

The Taming of the Shrew

Non-Standard Text Processing in the Digital Humanities

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität
Stuttgart zur Erlangung der Würde einer Doktorin der Philosophie (Dr. phil.) genehmigte
Abhandlung

Vorgelegt von

SARAH SCHULZ

aus

GÖPPINGEN

Hauptberichter: Prof. Dr. Jonas Kuhn
Mitberichter: Prof. Dr. Günther Görz

Tag der mündlichen Prüfung: 19.12.2017

Institut für Maschinelle Sprachverarbeitung der Universität Stuttgart

2018

AUTHOR'S DECLARATION

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen.

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: DATE:

ACKNOWLEDGEMENTS

My main motivation for pursuing a PhD was the curiosity to dive deeper into the world of natural language processing. After I had studied theater and media science and German in my bachelor's, I only spent an intense two and a half years studying computational linguistics in my master's. Naturally, I had the feeling that there was much more to be learned about this subject that seemed to fit my interests exceptionally well with its humanistic core and yet analytic methods. Now, after almost five years of research, I still have that feeling: learning never ends. However, I have learned a lot and not just about NLP but also about humanities and social sciences, about work methods, about people, about communication, about myself and about the world. This knowledge I owe to countless people I had the pleasure to work with throughout the last few years.

People who believe in you make all the difference. I want to thank my supervisor, Jonas Kuhn, for never doubting that NLP research is the right place for me and for all his support. I am grateful to Günther Görz who actually set the starting point for my career as a “computer person” many years ago and agreed to be the second examiner of my dissertation. I am also greatly indebted to Mechthild Habermann who gave me the necessary push to change my path.

While my name may be on the front cover of this dissertation, I am by no means its sole contributor. Thank you, Nils, you shared a lot of your experience with me. Also, your t-shirts always make my day. André, thanks for your help with all the web stuff. To all my colleagues at IMS: thank you for having an open door and an open ear. This is not self-evident and I appreciate it.

It is impossible to write a thesis about Digital Humanities without mentioning the countless colleagues from the humanities who discussed their research with me and who reminded me of why I started my academic career as a humanities scholar. Nora, Sandra, Max, Mareike and all CRETA members: thank you for being an inspiration and the best collaboration partners one can ask for.

And to all the people behind the scenes, especially Leo, my parents, Jürgen, Michi and Tarek: you inspire me, you motivate me and you kept me going.

TABLE OF CONTENTS

	Page
List of Abbreviations	ix
Abstract	xi
Deutsche Zusammenfassung	xiii
List of Tables	xv
List of Figures	xix
Preface	xxi
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	5
1.3 Publications Relevant for this Dissertation	6
2 Digital Humanities: Chance and Challenge	9
2.1 A Brief History of Digital Humanities	9
2.1.1 Origins	9
2.1.2 DH Conferences – The Witness of a Time Period	11
2.2 Digital Humanities: An Attempt at a Definition	13
2.3 Towards a Methodology	18
2.3.1 Interdisciplinarity and Collaboration	21
2.3.2 Reusability and Adaptability through Modularization	23
2.3.3 Specific Problem Solving	25
2.3.4 Evaluation	26
2.4 Research Contributions	27
3 Data in Digital Humanities: Wild and Sparse	29
3.1 Non-Standard Text	30
3.2 Why We Need Data: Machine Learning	35

TABLE OF CONTENTS

3.2.1	What is Machine Learning	35
3.2.2	(Big) Data and Digital Humanities	39
3.3	Annotation of Data	40
3.3.1	Corpus Annotation	40
3.3.2	Data Selection	43
3.4	Summary	46
4	Data Harvesting: Modularized and Adaptable Architectures for Digital Hu-	
	manities	47
4.1	Related Work	49
4.2	Evaluation Metrics	51
4.3	Data	52
4.3.1	The Werther Corpus	52
4.3.2	The Deutsches Textarchive (DTA) Corpus	52
4.3.3	Gutenberg Data for Language Modeling	53
4.4	Why OCR Post-Correction is Hard	54
4.5	Specialized Multi-Modular Post-Correction	54
4.5.1	Suggestion Modules	55
4.5.2	Decision Modules: the Ranking Mechanism	57
4.6	Experiments	57
4.6.1	Experimental Setup	57
4.6.2	Evaluation	58
4.7	Adaptability	60
4.8	Digitization Workflow	62
4.9	Research Contributions	64
5	Text Normalization	65
5.1	User-Generated Content - A Challenge for NLP	67
5.2	Text Normalization - Related Work	68
5.3	A Multi-Modular Approach Towards Normalization	71
5.3.1	Preprocessing Layer	72
5.3.2	Suggestion Layer	73
5.3.3	Decision Layer	77
5.4	Evaluation	77
5.4.1	Data Set	77
5.4.2	Modeling UGC Language	79
5.4.3	Evaluation Metrics	79
5.4.4	Experiments	80
5.4.5	The Bigger Picture - Extrinsic Evaluation and Portability	87

5.5	Research Contributions	88
6	Tool Adaptation for Non-Standard Text Processing	89
6.1	Training Data: The Influence of Quantity and Quality	93
6.1.1	Data Quantity and Quality	94
6.1.2	Manual Annotation – How much is enough?	95
6.1.3	Additional Data: Exploiting Existing Resources	98
6.1.4	Corpus Middle High German (ReM)	103
6.1.5	A General Model	107
6.1.6	Region-Specific Corpora	109
6.1.7	Summary	114
6.2	Finding the Right Method	116
6.2.1	Related Work	117
6.2.2	Learning from Within the Text	119
6.2.3	Stretching Out: Including Text-External Resources	121
6.2.4	Evaluation	123
6.2.5	Summary	125
6.3	Research Contributions	126
7	Exploiting Language Similarities: a Usecase	127
7.1	Code-Switching – Yet Another Deviation from the Norm	127
7.2	Related Work	129
7.3	Data	131
7.4	Automated Processing of Mixed Text	132
7.4.1	Language Identification	133
7.4.2	Part-of-Speech Tagging	134
7.5	Results	134
7.5.1	Language Identification	134
7.5.2	Part-of-Speech Tagging	135
7.6	Tools for Digital Humanities	138
7.7	Research Contributions	140
8	Problems Solved? – Future Directions for Digital Humanities	141
8.1	Towards Standard-Free Text Processing	141
8.2	Digital Humanities: Towards Key Concepts of a Methodology	143
8.3	DH and NLP: a Joint Future	144
	Bibliography	147

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
aaDH	Australasian Association for Digital Humanities
ACH	Association for Computers and the Humanities
ADHO	Alliance of Digital Humanities Organizations
ALLC	Association for Literary and Linguistic Computing
ASR	Automatic Speech Recognition
CA	Cluster Analysis
CER	Character Error Rate
CGN	Corpus Gesproken Nederlands
CoS	Code-Switching
CRF	Conditional Random Fields
CS	Computer Science
CSDH	Canadian Society for Digital Humanities
DH	Digital Humanities
DHQ	Digital Humanities Quarterly
DT	Decision Tree
DTA	Deutsches Textarchiv (German Text Archive)
EADH	European Association for Digital Humanities
ENHG	Early New High German
FEMA	Feature EMbeddings for domain Adaptation
HiTS	Historical TagSet
ICCH	International Conference on Computers and the Humanities
ISO	International Organization for Standardization
JADH	Japanese Association for Digital Humanities
LOB	Lancaster-Oslo-Bergen Corpus
LID	Language Identification
LSTM	Long Short-Term Memory
MHDBDB	Mittelhochdeutsche Begriffsdatenbank (Middle High German Conceptual Database)
MHG	Middle High German
ML	Machine Learning

LIST OF ABBREVIATIONS

MLP	Multi-Layer Perceptron
MNN	Multi-Layer Perceptron Neural Network
NE	Named Entity
NER	Named Entity Recognition
NHG	New High German
NLP	Natural Language Processing
NN	Neural Network
OCR	Optical Character Recognition
POS	Part-Of-Speech
ReM	Referenzkorpus Mittelhochdeutsch (Reference corpus Middle High German)
SCHN	Société canadienne des humanités numériques
SDI	Shannon Diversity Index
SGML	Standard Generalized Markup Language
SMS	Short Messages
SMT	Statistical Machine Translation
SNS	netlog message
STTS	Stuttgart-Tübingen-Tagset
TEI	Text Encoding Initiative
TTR	Type-Token Ratio
TWE	Twitter
UD	Universal Dependencies
UGC	User-Generated Content
UT	Universal Tagset
WER	Word Error Rate
W3C	World Wide Web Consortium
XML	Extensible Markup Language

ABSTRACT

Natural language processing (NLP) has focused on the automatic processing of newspaper texts for many years. With the growing importance of text analysis in various areas such as spoken language understanding, social media processing and the interpretation of text material from the humanities, techniques and methodologies have to be reviewed and redefined since so called *non-standard* texts pose challenges on the lexical and syntactic level especially for machine-learning-based approaches. Automatic processing tools developed on the basis of newspaper texts show a decreased performance for texts with divergent characteristics. Digital Humanities (DH) as a field that has risen to prominence in the last decades, holds a variety of examples for this kind of texts. Thus, the computational analysis of the relationships of Shakespeare’s dramatic characters requires the adjustment of processing tools to English texts from the 16th-century in dramatic form. Likewise, the investigation of narrative perspective in Goethe’s ballads calls for methods that can handle German verse from the 18th century.

In this dissertation, we put forward a methodology for NLP in a DH environment. We investigate how an interdisciplinary context in combination with specific goals within projects influences the general NLP approach. We suggest thoughtful collaboration and increased attention to the easy applicability of resulting tools as a solution for differences in the store of knowledge between project partners. Projects in DH are not only constituted by the automatic processing of texts but are usually framed by the investigation of a research question from the humanities. As a consequence, time limitations complicate the successful implementation of analysis techniques especially since the diversity of texts impairs the transferability and reusability of tools beyond a specific project. We answer to this with modular and thus easily adjustable project workflows and system architectures. Several instances serve as examples for our methodology on different levels. We discuss modular architectures that balance time-saving solutions and problem-specific implementations on the example of automatic postcorrection of the output text from an optical character recognition system. We address the problem of data diversity and low resource situations by investigating different approaches towards non-standard text processing. We examine two main techniques: text normalization and tool adjustment. Text normalization aims at the transformation of non-standard text in order to assimilate it to the standard whereas tool adjustment concentrates on the contrary direction of enabling tools to successfully handle a specific kind of text. We focus on the task of part-of-speech tagging to illustrate various approaches toward the processing of historical texts as an instance for non-standard texts. We discuss how the level of deviation from a standard form influences the performance of different methods. Our approaches shed light on the importance of data quality and quantity and emphasize the indispensability of annotations for effective machine learning. In addition, we highlight the advantages of problem-driven approaches where the purpose of a tool is clearly formulated through the research question.

Another significant finding to emerge from this work is a summary of the experiences and

ABSTRACT

increased knowledge through collaborative projects between computer scientists and humanists. We reflect on various aspects of the elaboration and formalization of research questions in the DH and assess the limitations and possibilities of the computational modeling of humanistic research questions. An emphasis is placed on the interplay of expert knowledge with respect to a subject of investigation and the implementation of tools for that purpose and the thereof resulting advantages such as the targeted improvement of digital methods through purposeful manual correction and error analysis. We show obstacles and chances and give prospects and directions for future development in this realm of interdisciplinary research.

DEUTSCHE ZUSAMMENFASSUNG

Die maschinelle Sprachverarbeitung (MS) hat sich viele Jahre lang hauptsächlich mit der automatischen Analyse von Zeitungstexten beschäftigt. Mit zunehmender Bedeutung automatischer Textanalyse in verschiedenen Bereichen wie Sprachverstehen, Verarbeitung sozialer Medien und der Interpretation von Texten aus den Geisteswissenschaften müssen sowohl Verarbeitungstechniken als auch Methoden überdacht und überarbeitet werden, da sogenannte *Nicht-Standard*texte eine Herausforderung auf lexikalischer und syntaktischer Ebene insbesondere für Ansätze des maschinellen Lernens darstellen. Automatische Verarbeitungswerkzeuge, die auf der Basis von Zeitungstexten entwickelt wurden, liefern schlechtere Ergebnisse für Texte mit abweichenden Merkmalen. Digital Humanities (DH) als Forschungsbereich, der in den letzten Jahren an Dominanz gewonnen hat, hält eine Anzahl unterschiedlicher Beispiele für solche Texte bereit. So erfordert die computergestützte Analyse der Beziehungen von Charakteren in Shakespeares Dramen die Anpassung von Verarbeitungswerkzeugen an das Englisch des 16. Jahrhunderts in dramatischen Texten. Die Untersuchung von Erzählperspektiven in Goethes Balladen wiederum bedarf Methoden, die deutsche Verse aus dem 18. Jahrhundert handhaben können.

In dieser Dissertation schlagen wir eine Methodik für MS in einer DH-Umgebung vor. Wir untersuchen, wie ein interdisziplinärer Kontext in Verbindung mit spezifischen Zielen innerhalb von Projekten den allgemeinen MS-Ansatz beeinflusst. Wir schlagen eine durchdachte Zusammenarbeit und erhöhte Aufmerksamkeit gegenüber der einfachen Anwendbarkeit resultierender Werkzeuge als Lösung für Unterschiede im Wissensschatz zwischen den Projektpartnern vor. Projekte in DH beschränken sich nicht auf die automatische Verarbeitung von Texten, sondern sind eingebettet in die Untersuchung einer bestimmten Forschungsfrage. Die dadurch entstehenden zeitlichen Einschränkungen erschweren die erfolgreiche Implementierung von Analysetechniken, zumal die Vielfalt von Texten innerhalb verschiedener Projekte die Übertragbarkeit und Wiederverwendbarkeit von Werkzeugen beeinträchtigt. Darauf antworten wir mit modularen und damit leicht anpassbaren Projektabläufen und Systemarchitekturen. Mehrere Projekte dienen als Beispiele für die vorgeschlagene Methodik auf verschiedenen Ebenen. Wir diskutieren eine modulare Architektur am Beispiel der automatischen Nachkorrektur der Ausgabertexte eines Optical Character Recognition Systems. Wir unterstreichen, wie diese Architektur eine zeitsparende Lösung mit einer problemspezifische Implementierungen verbindet. Wir befassen uns mit dem Problem der Datenvielfalt und geringen Ressourcenlage, indem wir verschiedene Ansätze zur Nicht-Standardtextverarbeitung untersuchen. Hierzu stellen wir zwei Ansätze vor: Textnormalisierung und Werkzeuganpassung. Die Textnormalisierung zielt darauf ab, Nicht-Standardtexte zu transformieren, um diese an den Standard anzupassen, wohingegen sich die Werkzeuganpassung auf die gegenteilige Richtung konzentriert, in der die Werkzeuge befähigt werden, eine bestimmte Art von Text erfolgreich zu handhaben. Wir konzentrieren uns auf die Aufgabe der Wortartenerkennung, anhand derer wir verschiedene Ansätze zur Verarbei-

tung historischer Texte als Instanz für Nicht-Standardtexte veranschaulichen. Wir diskutieren, wie das Niveau der Abweichung von einer Standardform die Ergebnisse verschiedener Methoden beeinflusst. Unsere Ansätze beleuchten die Bedeutung von Datenqualität und -quantität und betonen die Unverzichtbarkeit von Annotationen für effektives maschinelles Lernen. Darüber hinaus heben wir die Vorteile problemorientierter Ansätze hervor, bei denen der Zweck eines Werkzeugs durch die Fragestellung klar formuliert wird.

Ein weiterer wichtiger Befund, der sich aus dieser Arbeit ergibt, ist eine Zusammenfassung der Erfahrungen und des wachsenden Wissens durch gemeinsame Projekte zwischen Computer- und Geisteswissenschaftlern. Wir reflektieren verschiedene Aspekte der Ausarbeitung und Formalisierung von Forschungsfragen in DH und bewerten die Grenzen und Möglichkeiten, diese Fragen mit computergestützten Methoden zu beantworten. Ein Schwerpunkt liegt auf dem Zusammenspiel von Expertenwissen in Bezug auf einen Untersuchungsgegenstand und die Implementierung von Werkzeugen zu diesem Zweck. Dieser Vorteil wird verstärkt durch die Bereitschaft von Seiten der Geisteswissenschaften durch zielgerichtete manuelle Korrektur und Fehleranalyse zur gezielten Verbesserung digitaler Methoden beizutragen. Wir zeigen Hindernisse und Chancen auf und geben Perspektiven und Richtungen für die zukünftige Entwicklung in diesem Bereich der interdisziplinären Forschung.

LIST OF TABLES

TABLE	Page
2.1 Research activity taxonomy given in the TaDiRAH (Borek et al., 2016) initiative. . . .	19
2.2 Contextual levels that enclose NLP in a DH context and the issues, goals and strategies related thereto.	27
3.1 Tagging accuracy on POS tagging models for Latin dependent on the sampling method (random, type-token ratio based on word form (wf) and lemma (lem.) and Shannon-Diversity Index) and size of the training set.	45
4.1 Werther texts included in our corpus from different authors and times of origin. . . .	52
4.2 Werther-specific parallel corpus of OCR text and corrected text showing the number of tokens before and after post-correction along with WER and CER	57
4.3 DTA parallel corpus of OCR text and corrected text showing the number of tokens before and after post-correction along with WER and CER	58
4.4 WER and CER for both test sets before and after automatic post-correction for the system trained with the small training set (train) and the larger training set (train _{ext}). Baselines: the original text coming from the OCR system and the character-level SMT system trained on the Werther data.	58
4.5 Number of overcorrected, corrected and uniquely corrected words per module out of 17,367 tokens in test _{init} (2,726 erroneous words) and 13,304 tokens in test _{unk} (4,141 erroneous words)	59
4.6 Number of tokens in the English and French corpus provided by the competition on OCR-postcorrection.	61
4.7 Example of badly recognized text in the English part of the corpus.	61
4.8 The results reported in word error rate (WER) and character error rate (CER) for the English and French test set.	62
5.1 Number of tokens of the training, development and test sets listed by subgenre. . . .	77

5.2	Data statistics of the three genres of UGC: the number of messages and the number of tokens before and after normalization, together with the overall expansion rate (left-hand side); normalization effort expressed in the number of operations on character level (right-hand side).	78
5.3	Overview of corpora used for language modeling.	79
5.4	Evaluation results of the tokenization module.	80
5.5	Performance of the filtering methods.	81
5.6	WER, CER, precision and recall of the general modules with and without filtering of suggestions.	82
5.7	Number of problems each specialized module is responsible (RES) for, has solved correctly (COR) and has overcorrected (OVER).	83
5.8	Precision, recall and WER of the normalization in five different settings for each genre and on the entire test set.	85
5.9	Oracle recall values for the tuned, soft filtered genre-unbalanced system compared to the recall values achieved by the system in this setting without oracle.	86
5.10	Performance of different NLP tools before and after normalization with the all-data-in multi-modular system.	87
6.1	UD-Tagset. The tag SYM was not needed; we added combined tags for MHG as well as the tag SPUNCT to distinguish sentence-ending characters from other punctuation marks.	95
6.2	List of grammatical tags included in the MHDBDB along with examples for each category given by Mittelhochdeutsche Begriffsdatenbank (2017). This table is extracted from http://mhdbdb.sbg.ac.at/help/grammar-tags.de.html , possible mistakes are not corrected.	99
6.3	Features for the example word “næhest” (Engl. closest) used to disambiguate the grammatical information contained in the MHDBDB in context.	101
6.4	Comparison of tagging results achieved by using 20,000 manually annotated tokens (setting a), 10 million semi-automatically annotated tokens (90.7% annotation accuracy, setting b), combination of semi-automatically annotated and manually annotated (scaled up to 10 mio) data (setting a+b).	102
6.5	Direct mapping from HiTS to Universal Dependencies tagset.	104
6.6	Cross-evaluation results for experiments with the ReM and the disambiguated MHDBDB.105	
6.7	Cross-evaluation results after improvement of the mapping from HiTS to UD.	107
6.8	Overview of the subcorpora annotated along with the time of origin, number of tokens in the gold standards and number of tokens in the subcorpora which are used for training specific POS models.	108
6.9	Average number of sentences and tokens in train, development and test set of our gold standard.	118

6.10	Mapping between STTS and Universal Dependency POS tags.	122
7.1	Overview of POS-annotated CoS corpora. S&L'08:Solorio and Liu (2008b), V'14:Vyas et al. (2014), J'15:Jamatia et al. (2015), Ç&Ç'16:Çetinoğlu and Çöltekin (2016), S'16:Sharma et al. (2016), UT: Google Universal Tags (Petrov et al., 2012). UD: Universal Dependencies tag set (Nivre et al., 2016).	130
7.2	Labels annotated for LID along an explanation for each label and the occurrence in percent.	131
7.3	Labels annotated for POS tagging along with the explanation for each label and the occurrence in percent.	132
7.4	Performance of the CRF system for language identification compared to the baseline (BL). Precision, recall and F-score per class and macro-average of all classes.	134
7.5	Percentage of incorrectly labeled tokens per class along with the distribution of incorrect labels among the other labels.	135
7.6	Performance of the CRF systems for POS tagging compared to the majority baseline (BL1), the confidence baseline (BL2). CRF_{base} : system with the 13 basic features, $CRF_{predLID}$: system with predicted LID as an additional feature, $CRF_{goldLID}$: system with gold-standard LID as an additional feature. Precision (Pre), Recall (Rec) and F-score (F) per class and macro-average of all classes are given. The task-relevant results are emphasized in bold.	136
7.7	Percentage of incorrectly labeled tokens per class along with the distribution of incorrect labels among the other labels for the $CRF_{predLID}$ system.	137
7.8	Different portions of the training set along with precision, recall and F-score for LID and POS tagging.	138

LIST OF FIGURES

FIGURE	Page
1.1 Context levels of DH research that influence NLP approaches with their challenges and solution strategies.	4
2.1 Term frequency normalized to 1000 for concepts important to DH in the past 11 years in the Books of Abstracts from the Digital Humanities Conference.	12
2.2 Collaborative workflow of Digital Humanities projects.	17
2.3 Context levels of DH research that influence NLP approaches with their challenges and solution strategies.	20
2.4 A workflow describing two DH projects with different research objectives.	24
3.1 Constituent tree using Stanford’s CoreNLP constituency parser on the first stanza of “To a Lady”.	32
3.2 POS results a modern German tagger model (STTS tagset) on the first stanza of <i>The Songs of the Nibelungs</i>	34
3.3 Dependency analysis using Stanford’s CoreNLP dependency parser on verses from Shakespeare’s “Taming of the Shrew”.	34
3.4 The annotation workflow as described by Hovy and Lavid (2010)	41
3.5 Learning curves for different sampling methods for 2,000 to 20,000 tokens.	46
4.1 Three stages that a text has to go through from the scanned image of a book to the perfect transcription.	48
4.2 Multi-modular OCR post-correction system.	49
4.3 Scans of three different texts from our corpora. Emphasizes differences in quality of scan and differences in type setting, font and genre (e.g. drama).	53
4.4 Irregular type setting in German Gothic lettering. <i>sind</i> and <i>insgemein</i> are two separate words but yet written closely together.	55
4.5 Abstract workflow for the digitization from the scan to the digitized text.	62
4.6 Screenshots of two steps of the workflow implementation of OCR post-correction.	63

5.1	Multi-layer architecture of the UGC normalization system with the preprocessing layer on top, the context-based modules on the left-hand side, the token-based modules on the right-hand side and the decision module on the bottom.	72
6.1	Relative location of the data sets used in this chapter with respect to availability of resources and closeness to a standard form.	91
6.2	Learning curve for both classification algorithms trained on an increasing size of training data.	98
6.3	Results for the search word <i>schatz</i> for the Nibelungenlied at http://mhdbdb.sbg.ac.at where statistics for the word forms linked to this search are given together with direct links to the context in the Nibelungenlied.	100
6.4	Pipeline for incorporating a lexical resource into the development of a POS tagger model.	102
6.5	Example for the multilayer annotation in the ReM in CoraXML format for the word “mag”.	103
6.6	Accuracies achieved by the general POS tagger model on the genre-specific, author-specific and region-specific subcorpora.	110
6.7	Dimensions of non-standard text	119
6.8	Accuracies of all POS tagging approaches evaluated in a 10-fold Monte Carlo cross-validation setting along with the standard deviation of the accuracy values for the 10 samples are reported. Accuracy is given on the y-axis. The experiments are sorted by their increasing use of external resources and combination of classifiers: clustering (CA), conditional random fields classifier (CRF), MLP neural net (MLP NN), LSTM neural net (LSTM NN), MNN self-learning (MLP NN self) and CRF self-learning (CRF self) represent the experiments that use only text internal knowledge. On the right-hand side the results for experiments with external resources are listed in the following order: model transfer from New High German (NHG) and Middle High German (MHG), tritraining (TRI), majority voting (VOTE) and stacking (ST).	123
7.1	Simple web interface for the submission of mixed text for POS tagging.	139
7.2	Search interface of ICARUS returning results on a query for an English adjective followed by a Latin noun within the next 3 tokens.	139

PREFACE

This dissertation is about natural language processing (NLP) and more precisely about the processing of non-standard texts in the context of Digital Humanities (DH) projects. I¹ had the opportunity to learn more about DH-specific challenges in general throughout the last few years while working on NLP for DH. One important insight I gained is the fact that the methodology that computational linguists use in their research cannot deal with all aspects of the data and the context of DH. Thus, the main objective of this dissertation is to strengthen the peculiar characteristics of computer-aided aspects of DH projects. To do so, I will portray which implications the DH context can have for a methodology for NLP. I want to establish an interdisciplinary perspective on the topic, yet I am biased by the fact that I am a computational linguist. Interdisciplinary work is always a balancing act on a high wire and it is challenging to satisfy everybody involved. In order to paint a picture of these challenges and perks of automatic text processing for the humanities, I draw on my personal experience and introduce DH with respect to all aspects I deem relevant for the context of this thesis. In this specific case, this means that I mainly focus on digital literary studies as an interesting example for the ambitious application of digital methods to research questions from the humanities. I am aware, though, that DH is a heterogeneous field and others might perceive it in a way very different from the view I take in this dissertation.

As someone who has her roots in the humanities but grew to love the analytic and yet creative world of computational linguistics, the contributions I aim to make with this dissertation are manifold and aspire to bring the two fields closer together.

I want to facilitate a dialogue between humanists and computer scientist by sharing my experiences on the advantages and disadvantages of collaborative projects that I came across in years of interdisciplinary work. I remember to have followed countless heated discussions between humanists and computer scientists with a silent grin. Both parties meant to say the same thing but lacked the common vocabulary to communicate their thoughts; they unconsciously

¹Throughout this dissertation I will use both pronouns “I” and “we”. I will use “I” to indicate that the text speaks about my own decisions and choices, as well as to mark personal views. “We” is used whenever the text speaks about an insight or a result which was produced in a collaboration. Moreover, I use “we”, whenever the discourse is explanatory, such as an exposition of a proof. Therein, “we” stands for “me and the reader”.

talk past one another. I can, moreover, report on many unfulfilled expectations of collaboration partners since they did not have the basic understanding of their mutual fields to be realistic about what to expect. I hope to contribute to a basis for communication and mutual understanding for both fields by detailing points of view that might be self-evident for one, but not for the other person. I will stress the advantages that arise from collaborative work such as the mutual learning process, a heightened perception of which aspects are important for the other party and the facilitation that detailed knowledge of the subject of investigation can have especially for the development of modeling techniques.

Additionally, I hope to push DH forward. This field certainly suffers from ineffectiveness with respect to many aspects. Similar problems are tackled in the context of different projects repeatedly, even though there clearly is no need to reinvent the wheel over and over again. This is not due to a lack of commitment or intention of the research community. Often such ineffectiveness arises from lack of expertise. By pointing out which aspects of methodologies developed by the NLP community could be fruitful for DH and offering suggestions on how to adapt them to the DH context, I hope to facilitate research carried out in this field.

Eventually, I hope to draw the attention of the NLP community to the complex and interesting challenges that humanistic research objects offer for automatic processing. It is time to move on and turn towards more diverse manifestations of language and its context of use and the development of solutions that such scenarios require.

INTRODUCTION

1.1 Motivation

Humanities as the academic disciplines studying human society and culture can be traced back to ancient Greece. Over the course of the years, different subdisciplines developed a conglomerate of methods and analytic instruments to approach their research questions. A subfield that shows a special interest in a diversity of views on its objects of investigation are literary studies. Literature scholars have contrived different theories that can be applied to support text interpretation. For instance, structuralist criticism relates literary texts to universal structures such as narrative patterns or genre-specific structures. Vladimir Propp (1968) sets an impressive example of structured intertextual analysis of fairy tales by identifying prototypical functions. As another example, psychoanalytic literary criticism is influenced by *The Interpretation of Dreams* by Sigmund Freud (1899) which caused a massive surge of the use of methods borrowed from psychoanalysis to dive into the psychological motivations of the author or specific characters of the fictional world in the beginning of the 20th century. Bonaparte (1949) connects the fiction written by Edgar Allan Poe to his desire to be reunited with his dead mother. The results of one or several of such methods to approach literary texts flow together in a hermeneutic process of interpretation which eventually leads to an answer of a research question based upon the insights gained through applying these methods.

This collection of contributors to the hermeneutic process has recently been extended by a new methodology adding a digital component to research in the humanities. It has gained such prominence throughout different subdisciplines that its realizations are subsumed under their own name: Digital Humanities (DH). Being initially limited to frequency analysis of texts for the purpose of e.g. authorship attribution via stylometric analysis (Holmes and Forsyth, 1995), these digital methods are evolving to approach deeper and more complex concepts for text anal-

yses. It is no surprise that disciplines such as literary studies, which can be characterized by their variety of theories, demonstrate curiosity for a new method to potentially discover new vantage points for text interpretation. In his book “Distant Reading”, Moretti (2013) illustrates the conceptual development of digital methods for the analysis of literature and shows how to e.g. approximate plot analysis via ideas inspired by network theory. While individual research guided by the scholar’s intuitions still remains the prevalent form in the humanities, the introduction of formal digital methods and a collaborative context builds up an interesting tension. One important characteristic that all DH approaches have in common is their starting point; in the beginning of each analysis there is a research question or research interest. The goal is to support the answering process with the newly developed methodology. Thus, the digital method itself is embedded within a thematic context originating from the humanities. As the complexity of research questions that scholars approach with digital methods grows, the difficulty to formalize them to be fitting for a digital interpretation increases. This calls for an evolution of the methodology. To account for the challenging nature of these approaches, a new player entered the field in order to support such ambitious goals.

Natural language processing (NLP) as a field originating from the humanities has shown considerable interest in DH research. Collaborative work has e.g. been done by Kao and Jurafsky (2015) who report on the stylistic analysis of English poems using a variety of features motivated by traditional analytic techniques extracted with the help of methods from NLP. Milli and Bamman (2016) contribute to the deeper understanding of fanfiction by systematically comparing the characteristics found in such texts to the characteristics of their canonical work utilizing NLP techniques such as automatic character detection, gender identification and opinion prediction. This interest is not only motivated by the diversity and complexity of the research questions, which offers an ideal environment for the development of new methods and combined workflows, but also by the nature of texts found in the context of these research questions. Texts that serve as a basis for answering humanistic research questions are diverse with respect to their lexical and syntactic range. The object of investigation can be a play by Shakespeare, a sermon given in Latin mixed with Middle English or a collection of recipes from Medieval German times. In order to understand what makes these texts attractive, one has to understand which kinds of texts have been the focus of NLP for a long time. Plank (2016) calls it a “historical coincidence” that NLP has focused on the processing of newspaper language in its early days. This is due to its early availability in digital form which made newspaper texts to be what we consider the “standard”. This poses some issues. The standard form is the point of reference for definitions of basic concepts. This means that the characteristics of the standard define what a word is or give an idea of what constitutes a sentence. In a world in which NLP becomes increasingly important in the interaction between human and machine (Manaris, 1998), this fixation on one type of text that is far away from e.g. spontaneous speech is a clear disadvantage; other sorts of text do not necessarily corroborate our basic assumptions about characteristics of words, sentences but

also syntactic or lexical features that are defined by means of newspaper language. Thus, the results achieved with tools trained on such standard data are often disappointing when applied to non-standard texts (cf. Foster et al. (2011); McClosky (2010)). The goal to make NLP more applicable to other manifestations of a language is not new. Much work has been done in the field of domain adaptation (e.g. Blitzer et al. (2006); Daume III (2007); Ben-David et al. (2010)). However, a recurring issue is the lack of data available to develop and test new methods. DH as a source for diverse texts turned up at just the right time to meet the need for data in NLP. This data offers the opportunity to propose solutions for more flexible NLP. Following Plank (2016), three suggestions can be distinguished on how to approach the problem of non-standard text processing. She suggests the annotation of more data, the normalization of text towards the standard form and domain adaptation. In this dissertation, I will provide insights into all of these possible solutions and discuss their advantages and disadvantages in different contexts.

Even though joining forces between the humanities and NLP promises enhancements for both fields, it comes with a number of specific challenges. The tasks that this highly collaborative research field set out to solve require experts in more than one area which often leads to non-overlapping levels of expertise of the scientists involved. Interdisciplinary work requires a large degree of tolerance, awareness and trained communication skills. This concerns not only the collaboration between computer scientists and humanists but also among subdisciplines of both fields. Since collaboration must not refer to the mere combination of subparts of projects that are being processed separately by the respective experts, reflected inclusion of different viewpoints regarding all subparts is required. As an additional advantage, successful collaboration contributes to an increase in knowledge on both sides. Through DH collaborations, humanists learn a lot about abstraction from concrete instances whereas computer scientists can get immediate feedback from human experts about the strengths and limitations of their models. The context of a project with its specific research motivation necessitates NLP solutions that are time-saving as the application of digital methods is framed by the hermeneutic process. Ideally, methods and implementations should be reusable and adjustable to other research interests. Modularity of workflows and implementations could be a solution. This modularity also allows for concrete and applicable NLP solutions. With general submodules and a general basis, techniques are applicable and transferable between different data sets and can therefore help to account for specificities of texts at hand. Figure 1.1 illustrates how these contextual levels enclose NLP in the context of DH and underlines that the approach towards the incorporation of NLP into a digital methodology for the humanities has to account for these contexts.

The goal of this thesis is it to map out the challenges that come with this particular setting and suggest potential solutions and promising approaches and workflows for NLP in DH with an emphasis on solutions for non-standard text processing. I address the challenges in Chapter 2. In Chapter 3, I focus on the characteristics and peculiarities of the texts investigated

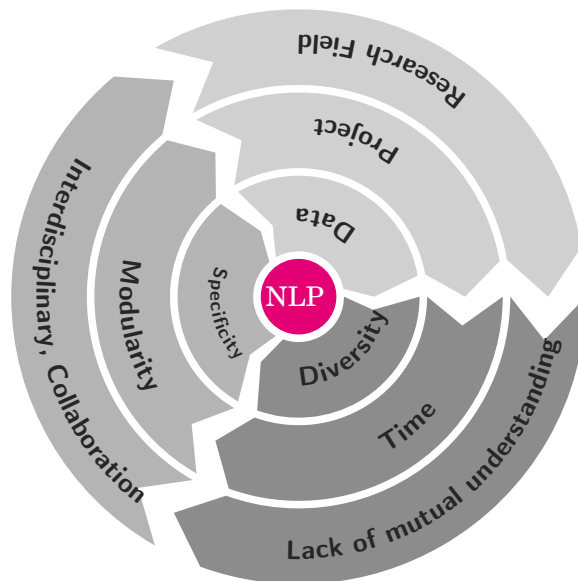


Figure 1.1: Context levels of DH research that influence NLP approaches with their challenges and solution strategies.

in this thesis since textual data is the linking element between NLP and DH. I motivate the need for text-specific NLP approaches by highlighting the implications of the characteristics of non-standard texts for a machine learning methodology. In Chapter 4, I suggest an adaptable pipeline for the digitization of texts from books as a solution to the reoccurring problem of low availability of texts that can serve as basis for DH research. Subsequently, I tackle two ways of computationally dealing with non-standard texts. Normalization of text aims at the assimilation of text characteristics to texts for which processing tools or data resources are available. In Chapter 5, I validate this solution with the example of Dutch user-generated contents. As an alternative, I investigate different methods of tool adjustment and look into the importance of data as well as advantages that algorithms yield for specific kinds of data. In Chapter 6, I connect my findings to different types of non-standard texts and show that there is no general approach towards non-standard text processing but that approaches are highly dependent on the characteristics of the text at hand. I especially highlight the importance of data quality and quantity and emphasize the indispensability of annotations for effective machine learning. In Chapter 7, I put the pieces back together. I show how I can include insights regarding methods investigated in Chapter 6 into a collaborative project. I underline how expertise from the humanities can support tool development and highlight the importance for application-oriented implementations.

1.2 Contributions

In this dissertation, I make a number of methodological contributions to the field of DH as well as to the area of non-standard text processing in NLP. DH, especially in its increasingly ambitious form, is still a young field of research. Even though involved researchers can profit from the interdisciplinary work, it suffers from a lack of primary “digital humanists”. I introduce a model of three contextual levels of DH that influence the nature of NLP for DH. On the first level, I discuss how the context of DH is normally accompanied by a lack of mutual expertise between the involved areas. The goal, however, is the support of humanities research with digital methods without the need to have humanists understand the details¹ of the implementation. This can be achieved by intense and reflected collaboration and communication.

The second level highlights the context of a project which is driven by a research question. Since the computer-aided part of DH projects are often only subparts of the entire workflow, there are serious time limitations with respect to the time available for method development. The goal, therefore, are transferable methods that can be reused and easily adapted among projects. As a solution, I suggest the modularization of workflows and system architectures, which allows for the introduction of problem-specific parts into established systems. These problem-specific solutions are conceptualized with the help of the expertise of the humanities scholar regarding the object of investigation.

The third contextual level is the data level. Central to the development of computational methods are the texts that have to be processed. However, the diversity of text characteristics within and across projects is immense and the lack of fitting processing tools for this kind of data is evident. The goal for this level are problem- and data-specific problem solutions that are conceptually transferable to other data sets. The concentration on specific aspects of a task facilitates the solution and simplifies successful implementations when combined with modularization.

I illustrate these aspects and suggested solutions with the help of diverse examples. I start with the implementation of a digitization workflow using a modular system architecture that allows for the introduction of problem-specific as well as general subsolutions to the problem. I account for a lower level of technical expertise of the user by integrating all modules by establishing an automatic workflow for the digitization process. Since I see the diversity of data as a central challenge for the transferability of methods between projects, I concentrate on solutions for the processing of texts with various characteristics. I detail the approach of text normalization on the example of Dutch user-generated content utilizing the architecture I introduced for the digitization process. This illustrates the flexibility of modular architectures not only regarding the specific data but also their adjustability to similar tasks. Eventually, I fathom the prospects of different approaches towards the processing of text in a low resource setting, investigating the influence of data quality and quantity as well as the significance of algorithms and

¹A basic understanding of the inner workings of the applied method, however, is recommended since this enables a reflected application of the method.

techniques for the performance of models. I investigate the task of part-of-speech (POS) tagging with the example of different historical languages.

I highlight applicability and usability of research results for the DH community as distinguishing aspect between NLP in the context of DH and general NLP. This is based on the contextual layer of the interdisciplinary research community where humanities scholars should be given easy access to the results of the automation process. I put this to practice by delivering actual implementations of solutions to support DH research through easily accessible web interfaces. Within the context of digitization, I implemented a pipeline reaching from optical character recognition via automatic post-correction of the resulting text and the provision of a format that can easily be accessed and processed further manually with an existing tool available in the community. The tools aided the compilation of a corpus of adaptations of Goethe's "The Sorrows of Young Werther". As it is a typical issue for historical languages that they lack a number of important preprocessing tools which influences the possibilities of deeper syntactic and semantic processing, I provide a POS tagger for Middle High German to improve this situation. This will accelerate the field of NLP for this stage of language since it opens a door for POS-dependent processing tools. Furthermore, I offer a combined pipeline for language identification and part-of-speech tagging for mixed texts using the example of mixed Latin-Middle English. I make sure that the output can be integrated with a query and visualization tool for the investigation of the results.

The following software is made available to the research community:

- optical character recognition post-correction system²
- POS tagger for Middle High German³
- system for language identification and POS tagging of Latin-Middle English text⁴

For all of these tools the source code and/or models have been published. This allows more advanced (DH-)users to retrain models and gain deeper insights into the implementation of the systems. However, the additional implementation of easy-to-use interfaces enables humanities scholars to easily access my tools and independently process their data without immediate support by their collaboration partner after the phase of development has been concluded.

1.3 Publications Relevant for this Dissertation

In the course of the past years, I worked with a lot of different researchers from different disciplines to answer research questions of humanistic nature and in computer science. Visiting different conferences, in the field of DH and NLP, allowed me to gain insights into both research

²<http://clarin05.ims.uni-stuttgart.de/ocr/>.

³<http://clarin05.ims.uni-stuttgart.de/mhdt/index.html>.

⁴<https://clarin09.ims.uni-stuttgart.de/normalisierung/mixed-pos.html>.

communities and connect with a number of research groups and people. This is reflected in the diversity of my publications. These publications are of varying focus but share the aspect of relevance for DH.



The chapters of this dissertation are based on the following publications

- Chapter 4

Publication *Multi-modular domain-tailored OCR post-correction* (Schulz and Kuhn, 2017).

Contribution I implemented the complete multi-modular system for this research. I offer an easy way to access the NLP pipeline and suggested an additional external tool for the inspection of the results. Project specific data has been provided by colleagues from literary studies.

- Chapter 5

Publication *Multimodular Text Normalization of Dutch User-Generated Content* (Schulz et al., 2016)

Contribution I implemented the multi-modular architecture and the majority of the modules for this research. The initial idea of the system architecture has been developed in collaboration with Bart Desmet, Orphée DeClercq, Véronique Hoste and Els Lefever. The preprocessing has been implemented by Bart and Orphée. The G2P2G and transliteration module has been suggested and implemented by Guy DePauw. Arda Tezcan contributed the idea to use a language model for the preclassification of tokens for normalization. Bart contributed the idea for the decision module. The article has been written collaboratively whereas the largest portion has been written by me.

- Section 6.1

Publication *From 0 to 10 Million Annotated Words – Part-of-Speech Tagging for Middle High German*. Manuscript (University of Stuttgart) under review for publication in “Language Technology for Digital Humanities” a special issue of “Language Resources and Evaluation” (Schulz and Ketschik, 2017)

Contribution The motivation of this work resulted from joint work between Nora Ketschik and myself. I implemented and trained the different systems compared in this article, the annotation guidelines were established in collaborative work. Nora annotated the data and shaped the process with insights from medieval German literature and linguistic points of views. Error analysis was done jointly at all times.

- Section 6.2

Publication *Learning from Within? Comparing PoS Tagging Approaches for Historical Text* (Schulz and Kuhn, 2016)

Contribution I am the only content contributor of this work.

- Chapter 7

Publication *Code-Switching Ubique Est - Language Identification and Part-of-Speech Tagging for Historical Mixed Text* (Schulz and Keller, 2016)

Contribution My contribution is the implementation and workflow conception of the entire NLP pipeline. The annotation guidelines were established in collaboration with Mareike Keller. There was a close feedback loop between Mareike and me which led to linguistically motivated improvement of the features used in the machine learning approach. Moreover, in agreement with Mareike, I offer an easy way to access the NLP pipeline and suggest an additional external tool for the inspection of the results. Mareike contributed the data and research question to the project. She annotated the data and gave feedback on the output of different stages of the systems. This facilitated the goal-oriented improvement of the system.

Publication *Challenges of Computational Processing of Code-Switching* (Çetinoglu et al., 2016)

Contribution This is an overview publication about the state-of-the-art of NLP approaches towards code switching. The largest portion of this work has been done by Özlem Çetinoglu with whom I also shared theoretical and practical insights into the processing of code switching research. In this dissertation only parts to which I contributed are used.

Publications relevant for the contents of a chapter are indicated in the introduction of the respective chapter.

DIGITAL HUMANITIES: CHANCE AND CHALLENGE

In this chapter, I introduce concepts that are relevant to understand natural language processing (NLP) in a Digital Humanities (DH) context. I draw a picture of the historical background of DH which has its beginnings with digital methods for linguistics. This is especially interesting since linguistics can be viewed as one of the most analytic humanities disciplines. This means that the subject of investigation, structure of language, is not per se something that is contrary to the formal and operationalized approach of computer science. This aspect is important in order to get a sense of the significance of a recent trend in DH which pursues a much harder task: the modeling of complex humanities research questions with digital methods. I give a short overview of different voices elaborating on DH in order to create an intuition concerning the diversity of approaches towards the inclusion of digital methods into the humanities. Based on this, I develop the definition of DH that underlies this dissertation. Subsequently, I zoom in to the consequences that the context of DH for how NLP has to be approached.

2.1 A Brief History of Digital Humanities

2.1.1 Origins

In the 1950s, Roberto Busa, theologian and linguist, decided to use computational power¹ in order to support his endeavor to lemmatize and digitize the massive corpus of Thomas Aquinas' works which comprises more than 10 million words. His efforts resulted in 56 printed volumes of all collocations included in this corpus, the *Index Thomisticus* (Busa, 1980). The *Index Thomisticus Treebank* project² started the syntactic annotation of this index in 2006. This corpus is a

¹Computational power provided by IBM.

²<http://itreebank.marginalia.it/>, 07/04/2017.

byword for the change that linguistics has undergone throughout the last 70 years. Fields that were characterized by the expertise of the individual, opened up to more quantifiable methods using corpus statistics as their supporting argument. However, in this example the application of digital methods merely extends an approach that already existed: frequency-based analysis of texts, especially in linguistics, was not invented with the introduction of digital methods. The involvement of computers allowed the expansion of analyses to larger amounts of texts and helped to shift the focus of the scholar back to the actual analysis and interpretation of the statistical findings. This focus on word-frequency-based analysis and therefore the restriction to research questions that can be answered with such methods has a prevalent influence on the orientation of the field. Apart from the mere analysis of linguistic structure such as e.g. the analysis of quantified noun phrases (Vannestål, 2004), throughout the last decades computational stylometry which is approximated via the frequency of function words representing author style has been a popular subject of investigation. The main reason for this limitation to analyses based on word counts was the lack of annotations. Annotations as the enrichment of texts with explicit linguistic, semantic or pragmatic information, sets the basis for analytic and yet complex approaches to understanding texts. Thus, with the rise of more structured and semantically rich annotations of often highly specialized nature, the complexity of questions that can potentially be answered with these methods increased. As an example, the annotation of named entities (Chinchor and Robinson, 1998) in texts can serve as a basis for the computer-aided analysis of character relationships in literary text (Chaturvedi et al., 2016).

This evolution from frequency-based computer-aided linguistics towards the support of humanistic research in e.g. literary studies was accompanied by the creation of national and international organizations which helped to establish a network of joint efforts in DH. The European Association for Digital Humanities (EADH) was founded in 1973 (back then bearing the name *Association for Literary and Linguistic Computing (ALLC)*) and to date liaises with three national DH organizations:

- Italian organization AIUCD - Associazione Informatica Umanistica e Cultura Digitale
- German language based DHd - Digital Humanities im deutschsprachigen Raum
- Nordic organization DHN - Digital humaniora i Norden.

Moreover, EADH is a founding chapter of the *Alliance of Digital Humanities Organizations (ADHO)* which was formed in 2005 and which is an international umbrella organization for regional DH organizations. Besides EADH, ADHO includes a number of associations:

- The Australasian Association for Digital Humanities (aaDH)
- The European Association for Digital Humanities, the Association for Computers and the Humanities (ACH)

- The Canadian Society for Digital Humanities / Société canadienne des humanités numériques (CSDH/SCHN)
- centerNet, Humanistica, L'association francophone des humanités numériques/digitales (Humanistica)
- The Japanese Association for Digital Humanities (JADH)

This notable tendency towards a joined and strongly connected international research community despite the inconceivable diversity of disciplines subsumed under the term *Digital Humanities* is remarkable.

Along with the tendency to organize and channel efforts in the fields of DH, early on journals dedicated to report on significant advances in research related to computational humanities got established. The first edition of *Computer and the Humanities* was already published in 1966. *The Journal Of Digital Scholarship In The Humanities* which publishes work related to digital literary studies and language research on behalf of the EADH and the ADHO exists since 1986. Since 2007, ADHO releases an open-access, peer-reviewed, digital journal which carries DH in its name. *Digital Humanities Quarterly* (DHQ) aims at providing a forum for everyone interested in DH and offers space for sharing theories, methods and technology. Even though text processing is emphasized in this thesis, DH is not merely restricted to text-based humanities. The first issue of *International Journal for Digital Art History*³ appeared in 2015. Yet, up until now the contributions to national and international DH conferences coming from e.g. musicology or art studies is vanishingly small. This might be related to the differences of digital methods that are applied. Text-based studies share the digital access provided by NLP, whereas artwork (or images thereof) would rather be approached with methods coming from image processing. Analysis of music could be supported by sound processing techniques. However, since the majority of humanities focuses on texts as their object of investigation, humanities disciplines with an emphasis on the analysis of other modalities are with only few peers.

2.1.2 DH Conferences – The Witness of a Time Period

The first ADHO conference was held 1989 at the University of Toronto as a joint event of ALLC and the International Conference on Computers and the Humanities (ICCH). This was already the 16th annual meeting of ALLC and the ninth annual meeting of the ACH-sponsored ICCH. Since 2006, this annual meeting is called *Digital Humanities* and includes additional organizers. The first edition in 2006 took place at the Sorbonne in Paris and the conference has since then paid a visit to three continents (Europe, North America and Australia) and twelve countries⁴.

Comparing the development of popular topics, it is most striking that the program offered and still offers an astonishing diversity in research topics from stylometry, over text mining,

³<http://journals.ub.uni-heidelberg.de/index.php/dah/>, 07/04/2017.

⁴For information about the history of conferences visit <https://adho.org/conference>, 07/04/2017.

semantic assessment of text and digital editions. As already mentioned, a strong focus on text-based research has been maintained over the years. This is partly due to the fact that humanities scholars have leaned on texts as their primary source of knowledge for many centuries. Moreover, text is a medium that is easily and intuitively accessible with a computer as opposed to e.g. images of art work. Nevertheless, there is still a large potential of DH barely explored and accessible by employing multi-modal material to shed light on one subject from different angles and through unusual combinations of perspectives. The surplus value of such approaches is demonstrated e.g. by a project including spatial data⁵ into the analysis of the dissemination of opinion via social media, whereby echo chambers as regions in which opinions are amplified via repetition can be identified (Hundt et al., 2017). Likewise in the context of user-generated online data, O’Halloran et al. (2014) combine text based and visual social media analysis to gain a social semiotic perspective on urban life in Singapore.

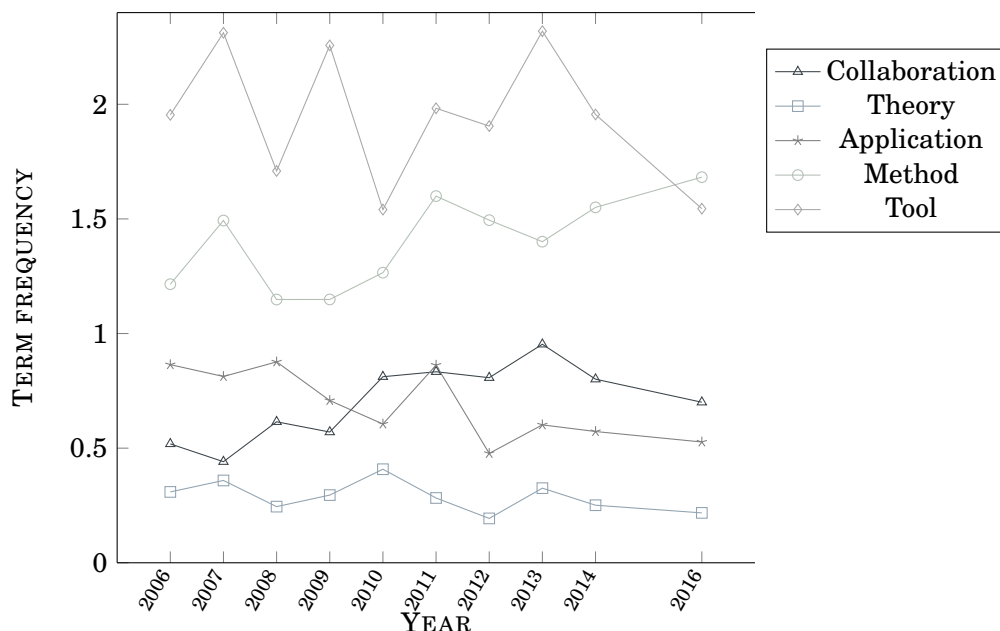


Figure 2.1: Term frequency normalized to 1000 for concepts important to DH in the past 11 years in the Books of Abstracts from the Digital Humanities Conference.

To gain an atmospheric picture on how DH relevant concepts changed over the years, I analyze word frequencies in the Books of Abstracts of the DH conferences from 2006 to 2016⁶. I use simple matching rules of word stems such as “collaborat*” to circumvent lemmatization. I normalize the word frequencies to 1000 words to account for different numbers of words in the Books of Abstracts of different years. The results are visualized in Figure 2.1.

Many of the concepts stay rather constant in their frequencies over the years. However, all of

⁵For an interdisciplinary introduction into space informed research in various fields cf. Warf and Arias (2008).

⁶2015 is missing since it has not been published as book of abstracts but only as html-based online proceedings.

them show at least one peak. *Theory* is the least mentioned concept of all five and reaches its highest point in 2010. In comparison to that, the concept of *application* shows a low point in 2010 and peaks one year later in 2011. In general, applications are mentioned much more frequently than theories. Along that line, *tools* are mentioned with the highest frequency. However, this concept shows the highest fluctuation throughout the years, reaching its peak in 2013. A rising tendency can be observed with *collaboration* with a peak at around 2013 but a rather constant growth in importance. The importance of *method* similarly gains importance rather slowly but constantly and reached its highest point recently in 2016.

2.2 Digital Humanities: An Attempt at a Definition

After discussing DH relevant concepts and the development of the field over the past decades one question arises: What does DH actually stand for? A glimpse at the literature shows that there is no agreed-upon definition on what constitutes DH, yet. In the following, different perspectives on DH are discussed and a narrower definition underlying this dissertation is introduced.

In their textbook, Jannidis et al. (2017, p.13) give a descriptive definition. They call DH the sum of all attempts to apply information technology to the subject of the humanities and illustrate the scope that research published under the term of DH can take. They mention texts as well as non-textual media as research objects alongside historical sources or the digital methods themselves as the object of investigation. Burdick et al. (2012, p. 24) state that “however heterogeneous, the Digital Humanities is unified by its emphasis on making, connecting, interpreting, and collaborating”, as an answer to voices claiming that a discipline called Digital Humanities cannot exist for the simple reason that there is no single discipline called humanities. This results in their definition expounding that DH “refers to new modes of scholarship and institutional units for collaborative, transdisciplinary, and computationally engaged research, teaching, and publication.” (Burdick et al., 2012, p. 122). The mere fact that there are textbooks such as Jannidis et al. (2017) and Burdick et al. (2012) that spend several hundred pages to illustrate the scale of DH shows its complexity and diversity. In the following, I attempt to give a rather shallow first idea of the main concepts and aspects of DH. The definitions quoted below, have been uttered on *Day of DH* between 2009 and 2014⁷ by members of the DH community.

There is the recurring opinion, that DH as such does not need a definition. This is reflected in the rejection (in often humorous ways) of definitions:

DH is me, Devin Higgins. Hi

(DEVIN HIGGINS, January 2014)

Who the hell knows....I certainly don't

(ETHAN WATRALL, January 2014)

⁷<https://twitter.com/DayofDH>, 07/04/2017, quotes have been collected by Jason Heppler: <https://github.com/hepplerj/whatisdigitalhumanities>.

A strange question. The only winning move is not to answer

(JEREMY BOGGS, January 2012)

Kirschenbaum (2014) closes his essay called *What is “Digital Humanities” and Why Are They Saying Such Terrible Things about it?* with the following conclusion:

I will be as plain as I can be: we will never know what digital humanities “is” because we don’t want to know nor is it useful for us to know.

(MATTHEW KIRSCHENBAUM, 2014)

Generally, the manifold definitions flying about in the community are dependent on the background of the individual. They put emphasis on different aspects of DH. A diversity in perspectives is characteristic for the field and in itself highlights a crucial aspect of DH. Nevertheless, there are basic concepts of DH that most people working in this field can agree on. One straight forward definition is given by David N. Wright.

The building and use of digital tools for studying the humanities.

(DAVID N. WRIGHT, January 2012)

This definition captures two aspects that lead to a number of implications for the nature of DH projects. One aspect is the use of digital tools for the purpose of answering questions in the fields of social sciences and the humanities. This requires a scholar who can actually interpret results suggested by these tools, thus someone equally specialized as the computer scientists but in a humanistic discipline. The other aspect that is mentioned is the building, thus the conception and implementation, of digital tools. Since humanities scholars are highly skilled in the reflected analysis and interpretation of humanistic subjects but not necessarily in the implementation of automatic routines, collaborative projects including experts in computational methods can support DH research without taking too much attention away from the humanistic core of the project. Computer scientists of any kind of specialization, typically from NLP, visualization, computer-human interfaces or similar areas are involved. These two aspects are by no means to be seen as two individual parts of DH. The building of the tool is initiated by the expertise of the humanities scholar and the concrete development of the methods should be a collaborative process. Likewise is it the task of the computer expert to facilitate the use of the resulting method for the humanities scholars which also includes the empowerment of the humanist to interpret the results through basic knowledge about the underlying technique.

The first challenge this setup reveals bare is the fact that these two expert groups are rarely overlapping. Ideally, this triggers what is described in a definition given by Dr. Craig Bellamy, an analyst within the DH based at the University of Melbourne in Australia and the founding secretary of the Australasian Association for Digital Humanities and co-chair of its recent inaugural conference.

The digital humanities is about creating people. Creating people who intersect with and apply the tools and methods of Computer Science using the principles, values, and techniques of the humanities. (CRAIG BELLAMY, January 2012)

This trend to educate people to create skill sets which enable well-founded DH research is also indicated by the emergence of study programs that teach DH as a discipline. These programs aim to create researchers who can take both perspectives, the point of view from the humanities side and the computer science perspective by simultaneously educating them in humanistic subject matters and computational methods and making them aware of new aspects that emerge in such contexts. Closely related to this is the aspect of mutual learning that is a crucial part of this rather early phase of DH. Learning is a key advantage of interdisciplinary research. This focus on learning also finds expression in the opportunities that DH can offer to support learning in general. Education studies as part of the humanities profit from advances in tools that can aid individual learning and serve different learning types with reasonable effort. Jana Remy, Associate Director of Digital Scholarship at Chapman University, emphasizes this supportive side of DH for different kinds of tasks:

It's using technology for humanities research, teaching, and publication.
(JANA REMY, January 2014)

In practice, as already mentioned there is often a lack of personnel that is experienced in both computer science and in humanities studies and thus Patrick Murray-John, Assistant Professor at George Mason University, is not too far from the truth when he defines DH as “smashing data into computers” indicating a lack of theory and methodology:

Short definition: Taking the Humanities, smashing it into computers, and seeing what happens. Long definition: Starting with ‘texts’, defined as broadly as the Humanities can sustain, seeing them as ‘data’ (as broadly as Computer Science can sustain), and using that view of the text/data to formulate and respond to new questions or issues in the Humanities. (PATRICK MURRAY-JOHN, January 2014)

This lack of structured and methodologically well-defined frameworks for DH is problematic. Yet, this seemingly unstructured approach in DH introduces new aspects to the humanities. The aspect of exploration and less theory-guided but data-driven research holds opportunities for new findings. At the same time, the immediate response to the output of a digital method forces a certain need for reflection upon computer scientists. This reflection ranges from the understanding of the suitability of features and algorithms to the limitations that digital methods have for deep semantic analysis. As automation needs to be understood in order to enable interpretation of the results they deliver, reflection is a part of DH that needs to be emphasized as it is done by Bobby Smiley in his definition.

The use of computational tools and techniques to explore questions in the humanities, and the concomitant reflection on the use of those tools and techniques in that exploration. (BOBBY SMILEY, January 2014)

Theoretical reflection is part of DH and is concerned with how the involvement of digital methods change the humanities and how data from the humanities as well as their questions change the way computer scientists work. The interpretation of results gathered by computational methods requires knowledge in both fields. As mentioned earlier, since “real Digital Humanists” are rare, the aspect of interdisciplinary work and communication is a crucial part of DH. Interdisciplinarity thus becomes the basic condition for the success of DH. Moreover, synergies are expected by the interaction of other research environments, research methodologies and points of view from a variety of humanities and social science disciplines.

The definition of DH underlying this thesis comes close to the one Laurie N. Taylor, Digital Scholarship Librarian at the University of Florida, gives:

[Digital Humanities are] [t]he humanities in and for a digital age
(LAURIE N. TAYLOR, January 2012)

I consider DH to be a new methodology in the collection of methods for (text) analysis. The fact that different fields within the humanities discover this new access to their objects of investigation at the same time and join forces makes it a movement that deserves its own label. Jockers (2013) calls DH a revolution for literary studies pointing out how especially the advances in annotated texts will change the possibilities for new research questions:

Though not “everything” has been digitized, we have reached a tipping point, an event horizon where enough text and literature have been encoded to both allow and, indeed, force us to ask an entirely new set of questions about literature and the literary record. (MATTHEW L. JOCKERS, 2013)

In fact, up to now DH has often just been the quantitative confirmation of knowledge that humanists have had for long already. This marks a phase in which computational methods are still – and rightly so – questioned in their abilities to capture the crucial points that are inherent to a humanist’s question. However, the goal has to be finding some middle ground on which these methods can maybe not fully account for the complexity of the research questions but still support a broader and deeper understanding of the texts. Through the advances in text annotations and the availability of automatic tools for the enrichment of texts with semantic information, confidence in the new methodology increases. These methods might cause a shift in how questions are asked, how views are shaped, how answers will sound. DH – and this is a subjective view – is not a discipline that can stand independently. It is rather an umbrella term for all those subdisciplines of traditional humanities which are willing and keen to add

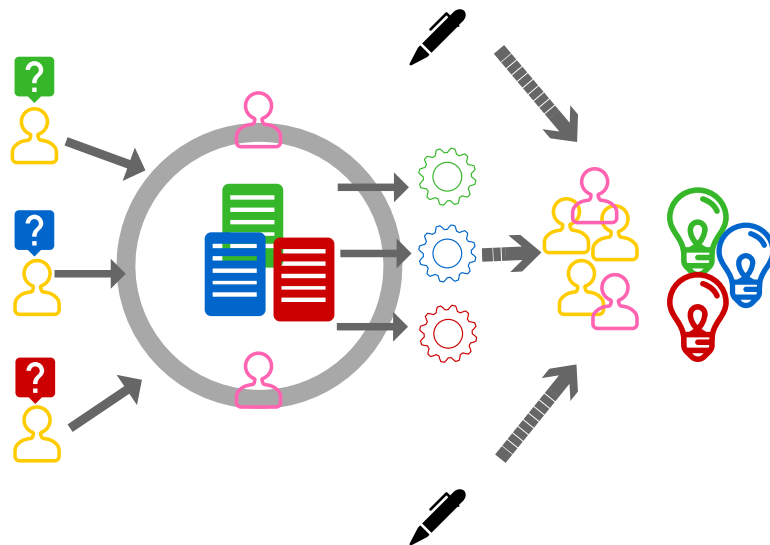


Figure 2.2: Collaborative workflow of Digital Humanities projects.

digital methods to their collection of traditional methods and use them as an extension and addition. What all of these branches of traditional humanities share are the challenges that this brave endeavor entails: the doubt of the validity that this automation involves, the blurriness and uncertainty that more curious and explorative approaches bring along and yet, the amazements that quantitative analysis can bring upon a scholar once he finds a well-known object of investigation in a new light.

Figure 2.2, illustrates the workflow advocated in this dissertation. Research questions coming from humanities scholars need to be formalized in a process involving close consultation with computational linguists. NLP techniques have to be adjusted to handle the specific characteristics of the text at hand. Since these texts are contributed from the humanities side, humanists can help to understand the specificities of the text. The computational methods return an output which corresponds to the operationalization. This output might highlight certain aspects of the data which in turn can lead to further insights into the subject under investigation. In order to streamline the results, humanists together with computer scientists possibly under consideration of further theoretical analytic methods interpret the results including knowledge about the digital method that has been applied, which eventually leads to a sound interpretation of the textual basis.

2.3 Towards a Methodology

I have established that DH per se is not a recent research methodology. Yet, just like it lacks a widely accepted definition, it does not yet have a full-fledged common set of methods, sources of evidence and infrastructures. This can mainly be attributed to the diversity of disciplines and thus the diversity of types of research questions and methodological approaches subsumed under the umbrella of DH. Each field brings its very own research tradition into the DH context which often comes with a specific way of approaching problems.

However, there are efforts to structure the field. With the goal to describe activities across disciplines Unsworth (2000) gives a coarse-grained list of so called “primitives”. These categories are abstract “functional primitives of humanities scholarship”. He names discovering, annotating, comparing, referring, sampling, illustrating and representing as basic building blocks of projects.

Similarly, the TaDiRAH (Borek et al., 2016) initiative ventures to compile a taxonomy of digital research activities in the humanities. It differentiates research activity, research object and research techniques. Whilst the techniques such as *Named Entity Recognition*, *Sentiment Analysis* or *Cluster Analysis* originate from computer science, many of the objects that are listed are clearly contributions from different humanities disciplines. Literature, manuscripts and sheet music are traditional research objects that have been investigated over the course of hundreds of years. However, the taxonomy also contains code, methods and infrastructures as new objects of investigation. Most interesting for the establishment of a methodology is the *activities* taxonomy. It lists eight main activities. They are displayed in Table 2.1. Some of the activities mentioned, such as creation and interpretation correspond to primitives such as annotating, sampling and discovering by Unsworth (2000). Yet, the activities are substantiated by the manifestations they can take in different project contexts.

This taxonomy is descriptive. It summarizes the variety of activities encountered in DH and focuses on the collection and creation of data sets as well as the analysis with the help of automatic processing tools and the aspect of storing data and disseminating results. The question arises whether there are certain activities that make up “good practice” of DH. Warwick et al. (2008) identify some of the activities as a determining factor for successful DH projects, namely documentation, users, management, sustainability and dissemination.

Borgman (2009) takes a rather normative perspective. She emphasizes that the “humanities need not emulate the sciences, but can learn useful lessons by studying the successes (and limitations) of cyberinfrastructure and eScience initiatives”. She identifies six factors inspired by the sciences which have implications for the future of digital scholarship in the humanities. She names the *print-only publication practice* which has to experience a shift to online publications, *data* as an essential future scholarly object, the need for *research methods* which are linked to the data under investigation, the necessity for an increase in *collaborative projects*, the requirement for *incentives to participate* in the form of open access with respect to sources

Activity	Specification
Capture	Conversion, Data Recognition, Discovering, Gathering, Imaging, Recording, Transcription
Creation	Designing, Programming, Translation, Web development, Writing
Enriching	Annotating, Cleanup, Editing
Analysis	Content Analysis, Network Analysis, Relational Analysis, Spatial Analysis, Structural Analysis, Stylistic Analysis, Visualization
Interpretation	Contextualization, Modeling, Theorizing
Storage	Archiving, Identifying, Organizing, Preservation
Dissemination	Collaboration, Commenting, Crowdsourcing, Communicating, Publishing, Sharing
Meta-Activities	Assessing, Community Building, Give Overview, Project Management, Teaching/Learning

Table 2.1: Research activity taxonomy given in the TaDiRAH (Borek et al., 2016) initiative.

and infrastructures and *cyberlearning* as a means to gain skills needed in this interdisciplinary field. Many of these aspects have flourished in the past few years. The number of collaborative projects has increased and infrastructures got established. However, infrastructures have to be general enough to be widely accepted in order to build the foundation of a methodology of the field.

In this dissertation, I abstain from the formulation of a methodology for DH but focus on a methodology for NLP in DH instead. Inspired by Borgman (2009), Unsworth (2000) and Borek et al. (2016), I focus on the implications that the multidisciplinary context has on the methodology of NLP. Similar to Borgman (2009), I concentrate on the comparison of sciences and humanities in order to give directions for structured text-based DH research. This comparative approach for the development of a useful methodology is grounded in the origins of DH. As introduced further above, linguistics can be considered to be one of the earliest disciplines among the humanities that decided to use computational methods. As a result, NLP constitutes its own discipline today. This discipline, however, has experienced a shift of emphasis. Even though its focus lies on the automatic processing of natural language, its aim is no longer exclusively the analysis of linguistic phenomena but rather offering general solutions in NLP for different levels of analysis of language. This tendency might relate back to the aim to abstract from concrete languages to describe general patterns in linguistics. Tasks in NLP are frequently solved **without any context of application** and tested on benchmark data sets to show the effectiveness of a technique. Due to this shift and the differences in the objective of the corresponding humanities disciplines, the DH and NLP communities have significant commonalities, while also differing in important ways. To get closer to a methodology of DH, I discuss aspects that DH and NLP have in common but emphasize the particular aspects that distinguish both fields. I believe that NLP holds a toolbox of methods and techniques that can successfully be adapted

to DH. Nevertheless, there are various aspects of DH that call for the introduction of specific concepts into this ready-made toolbox to fit the needs of DH research. These aspects originate from different **contextual levels that NLP is embedded in when applied within a DH research project**.

Figure 2.3 visualizes these contextual levels. I highlight the goals, current issues and my suggested strategies as key differences between NLP and DH. These aspects propagate through all levels. This means that the lowest level – the data level which directly influences the NLP approaches – incorporates the aspects related to the top-levels.

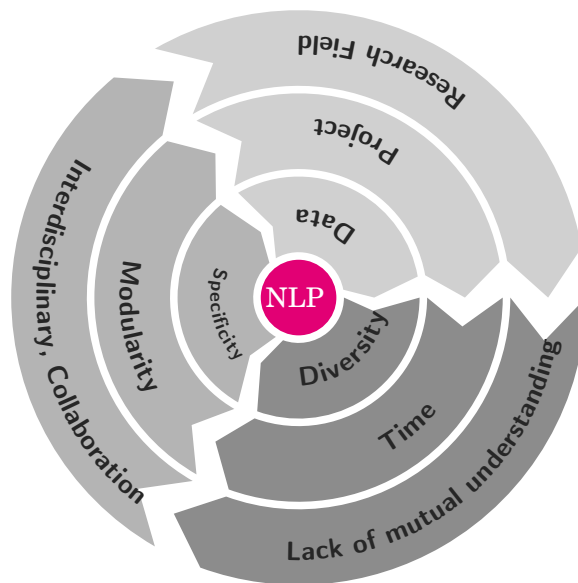


Figure 2.3: Context levels of DH research that influence NLP approaches with their challenges and solution strategies.

The first and broadest context layer is the layer of the **research field of DH**. Above, I have discussed the characteristics of this research area and have found **interdisciplinarity** to be one of the most prominent and both advantageous and difficult features. Interdisciplinarity is connected to potential weaknesses in mutual understanding for certain parts of collaborations. Thus, interdisciplinary research requires mechanisms which compensate for lack of shared knowledge. Certainly, interdisciplinarity can as well benefit the collaboration partners since it triggers the acquisition of new knowledge.

The next context layer is the **specific research project**. Research projects in the humanities usually have a humanistic research question at their core and therefore aim to answer this question with the support of digital methods. This requires project workflows that have a specific goal and follow specialized problem solving strategies. However, since time is of the essence, reusability and adaptability of already existing tools and workflows is a preferred characteristic. This can be tackled with the introduction of **modular system and modular project setups**

that allow for specific solutions via problem-relevant combinations of already existing partial solutions.

The last context level above the actual application of NLP techniques is the **data level**. Depending on the research question, the data that is used as a basis to answer the research question can vary a lot among projects. Texts from the humanities often deviate from the newspaper-based linguistic *standard* that NLP has primarily worked with for the last few decades and it is therefore necessary to come up with **data-specific solutions** for the specific kind of text a project is based on. Moreover, the availability of data is not guaranteed. Oftentimes data acquisition is the first step towards text-based (quantitative) research. These facts make the transferability of tools developed within one project to another difficult.

It is obvious that NLP as the lowest level has to deal with these influences coming from different contextual levels in order to guarantee successful implementation of digital methods for the humanities. In the following, I will discuss these aspects in more detail. I will touch upon categories introduced by Borgman (2009), Unsworth (2000) and Borek et al. (2016) since I believe that many of the points they make are essential for methodology of digital tool development for the humanities. I discuss the overall tendencies with respect to methodological decisions made in both general NLP and NLP for DH and link them back to the limitations, requirements and strategies that the different contextual levels induce. I advocate an NLP-based application-focused methodology for DH text processing.

2.3.1 Interdisciplinarity and Collaboration

As already mentioned in the beginning of this chapter, the DH community is strongly interested in interdisciplinary connections as indicated by the founding of organizations subsuming all fields interested in computer-aided humanities research. The motivation for collaborations is the assumption that collaborations will lead to new knowledge that goes beyond the mere sum of what participating individual parties of the collaboration could accomplish on their own.

However, collaborations come with certain challenges. Kanfer et al. (2000) discuss the tension between effort and knowledge growth in interdisciplinary collaborations. The most challenging – but at the same time most defining – of these interdisciplinary ties is the collaboration between humanists and computer scientists. At first glance, their collaboration seems imbalanced: The humanist delivers the research question and the computer scientist solely functions as a service provider to support the humanist with their methods. In many collaborations the goals are not defined in such a way that both parties benefit. Since DH can roughly be defined as the investigation of humanistic research questions with the help of computational methods, the clear beneficiary seems to be the humanist in this duo. However, any computer scientist should be clear about the variety of new computational challenges these collaborations offer. There are hardly more challenging tasks in NLP than working with texts from the humanities such as e.g. poems or narrative fiction. NLP has focused on the processing of newspaper text and texts

that are free of ungrammatical contents for long. Recently, the interest in so called non-standard data is growing with increased popularity of e.g. social media analysis. DH research yields the most diverse collection of this kind of data originating from all stages of language, reaching from poem to essays and beyond. Yet, there is still a certain degree of ignorance for the potential that DH holds for the development of NLP towards the processing of more diverse texts. This is one of the reasons that DH is still in a phase in which humanists do not fully trust automatically extracted knowledge. Likewise, computational linguists often still lack the confidence to tackle humanistic research questions with their methods.

The key to progress into the right direction is thoughtful collaboration. Siemens (2009) developed recommendations for successful interdisciplinary team work. They highlight mutual understanding of each other's goals, challenges and possibilities as important factors. Thus, one integral part of each collaborative project has to be the **establishment of a common vocabulary**. As trivial as this may sound it requires a lot of consciousness and sensitivity to detect situations in which communication fails without obvious signs. Raising awareness for the issue of differing terms and traditions of explaining, describing and tackling problems is the best basis for a slowly emerging common ground. Tolerance and mutual respect for what collaboration partners know and also might not know is equally crucial. Occasionally, disciplines tend to put their status above others. This might happen unconsciously and can result from a strong specialization of researchers who are often focused on just one way of thinking. Collaboration, however, should not be dominated by one discipline. Just if both (or all) parties contribute to an equal extent the collaboration can be fruitful for all partners, since otherwise a one-sided teacher-student relationship gets established. Such teacher-student relationships might indeed emerge throughout the process of a project. If they go both ways, however, all parties can profit from each other. This aspect of **learning** has been also emphasized by Borgman (2009).

Along these lines, I recommend a close **feedback loop between the experts from different backgrounds** within a project as a main component of a methodology of DH. Different ways to approach problems can be a productive source for progress. The computer scientist might request a formalization of the research agenda which forces the humanist to more concretely think about the goal and the necessary steps. In turn, detailed knowledge about a problem or specific data can support the development and improvement of automatic processing tools by pointing out the right direction. Immediate feedback on the strengths and limitations of computational models and the willingness to contribute manual corrections from the side of humanities scholars are a rare and luxurious situation in NLP which can be utilized for targeted refinement of modeling techniques.

Collaboration is successful when new knowledge arises from a vivid exchange for each of the collaboration partners. Thus, communication between collaboration partners, taking turns in teaching and learning and a close feedback loop between experts with different backgrounds should be taken into account when thinking about a methodological approach towards DH

projects.

An aspect that is often overlooked especially in the academic context of NLP research is the applicability of tools. Recently, the open source mentality amongst NLP researchers has increased considerably. However, tools are often badly maintained and it takes considerable knowledge of programming languages to make them work. This limits the usefulness of these tools for DH. Bulatovic et al. (2016) report on the importance of usability of tools and services based on usability studies. They highlight the interoperability aspect between several infrastructure components. Burghardt (2012) report on a gap between developers and scientific users. This gap becomes a severe problem in the context of DH. In DH collaborations, there is often a clearly defined project goal. Tools are developed for a specific purpose and will be applied. This forces humanists to utter their wishes with respect to the features a tool needs to have and requires the computer scientist to think beyond evaluation of a tool on a test set for proof of concept. Warwick et al. (2008) highlights the importance of the *user* as the addressee of research output in form of user-friendly tools.

An important contribution of this thesis is it therefore to raise awareness for **easily accessible tools** which provide the outcome of research to the potential user. However, this assumes good communication between D and H to become clear about the needs and limitations. This dissertation contains several examples of successful collaborations concluded with the publication of tools via webapplications. These have the advantage that the humanist can autonomously process data after the development phase is completed since they are easy to use and make the local installation of tools superfluous. The disadvantage of such solutions is the further “black-boxification” of the D-part. As mentioned earlier, teaching and learning is one of the key aspects of DH collaborations. Withholding the technical aspects of computer-aided components of a project indirectly keeps the humanities scholars from dealing with these components. However, in a well-structured and strongly intertwined work progress, the technical understanding which is also needed for a reflected use of resulting tools should ideally be covered by the development phase.

2.3.2 Reusability and Adaptability through Modularization

Projects in NLP often evolve around the improvement of a specific automatic processing task, such as parsing, or the understanding of algorithms for a specific task, such as the role of neural networks for prosody analysis. On the contrary, computer-aided projects with a humanistic research question at their core have to complete more steps to reach their goal. As just established, collaborations add phases of intense communication and mutual learning to the project workflow. Research questions have to be formulated, data as a basis for the analysis has to be collected, automatic processing tools have to be trained and applied and the results have to be evaluated in the context of the research question. Thus, a certain paradox emerges. I mentioned that the processing of texts in the context of DH is often challenging due to the nature of the

texts. However, the portion of time that can be spent on tool development is way smaller than in NLP projects. This establishes the need to find time-saving solutions. A key concept that can alleviate this time issue is the prioritization of **reusable and adaptable systems and project workflows**. Kuhn and Reiter (2015) advocate a modular architecture for DH workflows. They claim that **modularization as a key concept** in computer science is underexploited in DH. I agree that modularization on workflow level as well as on the level of implementation is the key to reusable results. DH projects often share a number of substeps that can be solved in similar ways. These substeps can be as abstract as the “primitives” introduced by Unsworth (2000) or the research activity taxonomy in TaDiRAH (Borek et al., 2016). I visualize a typical DH workflow and show how it generalizes over two different objectives in Figure 2.4. The abstract workflow described in the middle of Figure 2.4 is instantiated by the examples of network analysis on Middle High German texts and the analysis of mixed phrases in Latin-Middle English sermons. Even though the project goals and the techniques relevant to reach these goals (named entity recognition (NER) vs. part-of-speech (POS) tagging) are different, the steps that revolve around them are similar in both projects as summarized in Figure 2.4 (b). Thus, I advocate the idea of a **modularized, cross-project DH methodology**.

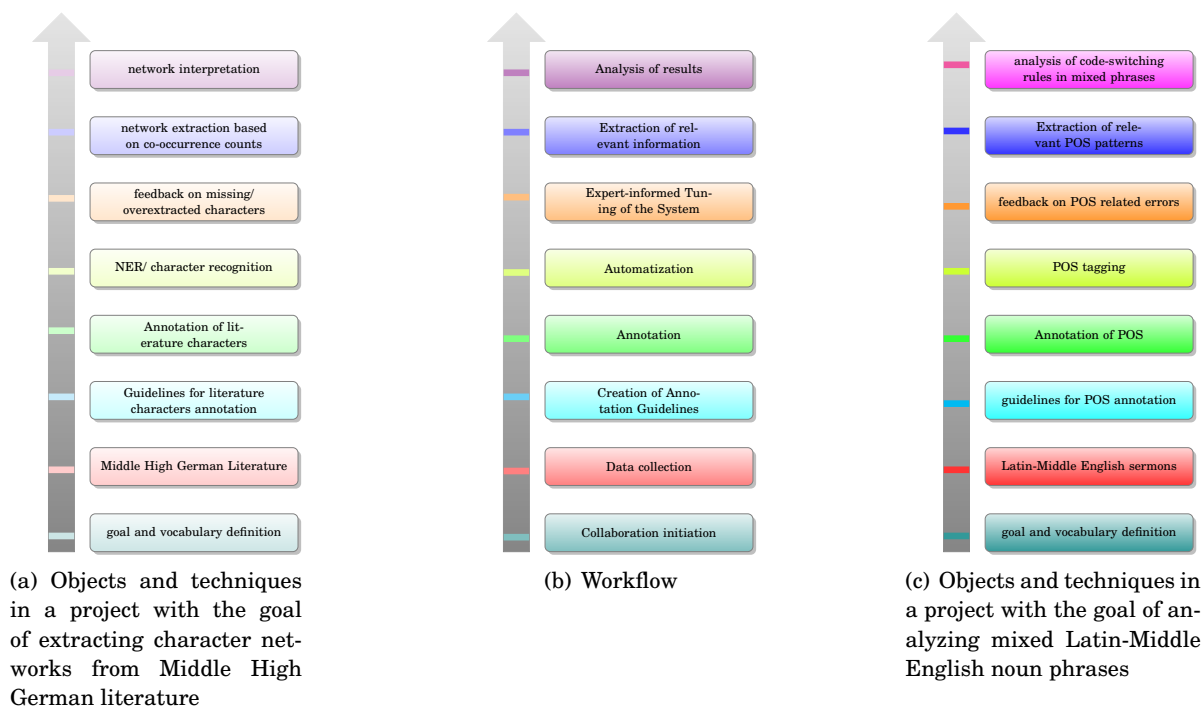


Figure 2.4: A workflow describing two DH projects with different research objectives.

The biggest challenge for the consolidation of such a modular methodology in DH, is the difference in problem solving strategies. A research question is traditionally examined within its context in humanities disciplines. Taking a problem apart into its individual subparts means

a temporary loss of context which on first sight seems counterintuitive. The challenge is it to bring the parts back together after a step-wise processing to consider context for the final analysis. Initiatives such as CLARIN⁸ and DARIAH⁹ focus on such modularized research infrastructures for the humanities. The underlying concept of these services comes from computer science. Through modularization and interfaces between modules, they promote a building block concept with the aim of serving as many different research objectives as possible. This principle has been known for long in NLP. Once such modules are established they can guarantee the reusability throughout different projects with little adjustments or simply through a different arrangement. Modules are shared across projects. POS tagging in our example in Figure 2.4 is given as the main technique to extract specific POS patterns from mixed texts. Moreover, POS tagging can function as a module in the development process for named entity recognizers by contributing informative features and thus become a module in network analysis projects. Due to the importance to the task of POS tagging for a series of subsequent task, we investigate this task in the context of non-standard data situations later in this dissertation.

2.3.3 Specific Problem Solving

The research goal of general linguistics is to find generalizable patterns. Similarly applied NLP is mostly interested in universal systems and findings. The central point is no longer the support of linguistic research questions with computational methods but rather the computational methods in themselves. NLP has evolved to a discipline in which general problem solving is highly emphasized. Tasks such as POS tagging get solved without a specific application purpose in mind. Data is often only used to test the algorithm at hand. Tools are considered to be useful if they can be applied to different texts. Recently, the trend has been shifting towards powerful models that can learn to solve multiple problems at once as e.g. suggested by Collobert and Weston (2008) or Kaiser et al. (2017). However, I mentioned that text processing in DH can be challenging due to the nature of texts. Thus, general problem solving often fails since the characteristics of the data used for training general models and the characteristics of DH data drift too far apart. Even though the complexity of the data and the objectives make DH challenging, the very specific foci of such projects come with certain advantages.

Specific research questions allow for **specific problem solving**. The knowledge about the context of application for a tool or analysis pipeline holds advantages. Specific (or goal-oriented) problem solving starts already with the **choice of specific data**. Consider a project including an optical character recognition post-correction system as a preprocessing step for the analysis of Goethe's works. This system does not have to perform well for texts from the 16th century since Goethe lived in the 18th and 19th century. Thus a training corpus for such a system can be compiled with respect to the needs of this system. In addition, a clear picture of what the data

⁸<https://www.clarin-d.de/de/>, 25/07/2017.

⁹<http://www.dariah.eu/>, 25/07/2017.

within a project looks like can help to adjust a tool **specifically to the data characteristics** which in all likelihood improves automatic processing results. As an example, tools in NLP often rely on a set of basic assumptions. One of these assumptions can be the availability of sentence delimiting punctuation marks. However, especially older texts often lack reliable punctuation. This knowledge is important when developing tools that work on sentence level. The definition of sentence in this context can e.g. be changed to those of verses in poems and verse novels or paragraphs or entire texts need to be chosen as minimal context. Moreover, automatic processing tools can be trained to **solve simpler tasks** than general tools that are not informed about the context of application. In a project that promotes the use of adjectives in *Beowulf*, there is no need to implement a full-blown automatic system for Old English POS annotation. It suffices to focus on the recognition of adjectives which should – with the help of an expert for Old English – be much simpler to extract. Feature sets can be designed for a specified task and also annotation can be done more easily, just covering certain aspects, not accounting for all possibly interesting concepts.

Even though specific problem solving and generic modular approaches seemingly build natural contrasts, easily adaptable architectures that can, with little effort, be customized to different text types and research questions, are the key to keeping a balance between time-saving modular solutions on the level of implementation and problem-specific content-related solutions.

2.3.4 Evaluation

Transparent and methodologically sound evaluation is one of the main criteria for judging the quality of research in NLP. This evaluation is normally applied to developed **tools** for a specific purpose and is based on a ground truth or gold standard (Sparck Jones and Galliers, 1996). These are usually manually annotated data sets – ideally by multiple annotators – that are supposed to contain the objectively correct answer. Partly due to this practice, NLP problems are formulated in such a way that there is a single objective correct answer. In case a subpart of a DH research project can be formalized in such a way, the evaluation methodology developed within the NLP community can suffice and should be applied to this subpart. This is the case for e.g. POS tagging or NER as part of network analysis. Bögel et al. (2015a) argue that the permission of ambiguity in annotations and thus in the evaluation data is crucial in the humanities. Due to the eventual (hermeneutic) interpretation for which the context is highly relevant, a phenomenon cannot be treated as something fixed and definite since the context will determine the actual meaning and interpretation. This makes the objective approach of NLP difficult for the application in a DH context.

While this problem describes the difficulties of evaluation on the level of automatic processing, the evaluation of an entire collaboration proves to be even more challenging. For the evaluation of the **overall success of an entire DH project** and the assessment whether a research question could be answered satisfactorily the NLP approach towards evaluation is not adequate.

Humanities research evaluation is based on recognition of the findings by the community. This is commonly indirectly measured via the number of citations a publication receives (Thelwall and Delgado, 2015). This measure can naturally also be applied to DH research. In addition, since DH projects often have output beyond publications such as e.g. tools, workflow documentations, digital edition etc., these aspects can be taken into consideration when evaluating the outcome of a project. One assessable aspect is the efficiency with which tools can be adapted to other data sets. This links back to the criterion of reusability and influences the outcome of a project as well as its usefulness to the community. Rockwell (2012) mentions several aspects that can be evaluated to assess the success of a DH project. Among other aspects, he mentions the accessibility of the study to the community, publication output, connectedness to other work in the field, archiving and long term accessibility. These factors for success are also described in Warwick et al. (2008).

2.4 Research Contributions

In this chapter, I have introduced the definition of DH that underlies this thesis. The main focus of this dissertation is the processing of non-standard text which I will approach under the consideration of the aspects discussed in Section 2.3. I formulate the contextual framework of this dissertation. By detailing the challenges and perks of DH research, I have motivated why I approach NLP in the context of such projects from a slightly different angle as it is done in general NLP. The contextual levels of NLP for DH are summarized in Table 2.2.

Contextual Level	Goal	Current Issue	Strategy
DH Field	applicable solution	① Lack of mutual understanding	① interdisciplinarity, collaboration, easy-to-use NLP
Project	adaptability, reusability	② time limitations, ① Diversity/sparsity of data,	② modularization of workflows and architectures, ①
Data	specific methods	lack of tools, ①, ②	specific problem solving, ①, ②

Table 2.2: Contextual levels that enclose NLP in a DH context and the issues, goals and strategies related thereto.

I have shown how the different context levels create certain issues for NLP, I have formulated the goals for successful implementations and suggest strategies to reach these goals.

In the following chapters, I will show how to emphasize reusability and adaptability on different levels of implementation, utilize the advantages that collaborative work offers, and demonstrate how specific problem solving facilitates the successful realization of computational support for humanistic research questions.

DATA IN DIGITAL HUMANITIES: WILD AND SPARSE

In this chapter, I discuss what constitutes non-standard texts in the context of this dissertation. The knowledge of the characteristics of these texts is essential for the understanding of potential problems encountered when using machine learning (ML) techniques for their automatic processing. Since ML-approaches are the most popular approaches towards automatic text processing, I investigate such techniques in this dissertation. Therefore, the basic understanding of ML and the relationship between data, features and transferability of such approaches is introduced in this chapter.

The term “data” has fueled a discussion in the Digital Humanities (DH) about whether or not one can speak of “data” (Marche, 2012) when talking about a research object of the humanities. Schöch (2013) concludes that “[d]ata in the humanities could be considered a digital, selectively constructed, machine-actionable abstraction representing some aspects of a given object of humanistic inquiry” and Schmidt (2012) calls it “digital surrogate”. I will use the word *data* to refer to digitized texts that serve as the basis for the computer-aided investigation of humanistic research questions.

I give examples to illustrate the wide spectrum of characteristics one can expect working with non-standard texts and detail how this poses problems for existing natural language processing (NLP) tools. Since low data availability is one of the primary characteristics of the data I work with, I moreover discuss this problem in the context of machine learning. Subsequently, I take a methodological perspective on this matter by emphasizing the importance of high-quality manual annotations as a basis for the automation of processing tasks. I show how existing workflows from NLP, such as the annotation workflow, can be adjusted to fit the needs of DH projects. I conclude this chapter with a small experiment in which I show how the selection of training data can influence the quality of resulting models.

3.1 Non-Standard Text

Non-standard text (occasionally also non-canonical text¹) is a term that refers to texts containing language which differs from some agreed-upon form. This definition shows that in order to define “non-standard” it is indispensable to previously define “standard”. It is important to note that a definition can only be given in a specific context and that a general definition – other than the observation that a standard is a convention – is neither achievable nor desirable. Therefore, in this dissertation the definition of the standard form and consequently the definition of a non-standard form will just be applicable within the context of NLP. In NLP this agreed-upon form, henceforth standard form, usually equals to the form of a language used in newspapers due to historical reasons. Newspaper texts were the first sort of texts widely available in digital form for the development of NLP tools. Therefore, this domain represents the standard domain for which these tools perform particularly well. Plank (2016) raises the question whether this standard would be different in a world in which NLP had its beginnings in a time when e.g. user-generated content (UGC) is widely available online. Presumably, the standard form would be different.

Stepping out of the standard domain and moving on to other sorts of texts, we can be faced with a drop in performance of automatic processing tools since the material the tools were trained with demonstrates properties different from those of non-standard text. This implies that any language for which sufficient digitized texts and trained resources are available, can be considered a standard form in the NLP sense and that there is more than one standard form, e.g. standard forms for different languages.

The deviation from the standard form can manifest on various levels and to different degrees. There are manifestations of a language that deviate on the **lexical level** from a standard form such as e.g. variants of the same language.

- (1) Het meisje draagt een jurk.
Het meisje draagt een kledingstuk.
‘The girl wears a dress.’

In Example (1), the two sentences stem from two varieties of Dutch, the first spoken in the Netherlands, the second spoken in Belgium (known as Flemish). The syntactic structure of the first and the second sentence is the same whereas two lexical items differ. Considering the availability of tools that rely on lexical features trained on the variety spoken in the Netherlands, this will influence performance on Flemish texts.

While these differences in the dictionary amount to a manageable number of words, as soon as pronunciation manifests in written text, as e.g. seen in UGC found on social media platforms online, they become increasingly problematic:

¹Personally, I experienced that the term non-canonical causes a lot of confusion especially in collaborations with literary scholars since they mistakenly interpret it as “not in the literary canon”, thus not part of a collection of major literary works. Therefore, I prefer the term non-standard over non-canonical.

- (2) *kebda ni gedaan*
ik heb dat niet gedaan
 I have that not done
 ‘I did not do that.’

In spoken Flemish, multiple words tend to melt together as exemplified in (2). The second line shows the normalized version² of line one. The token *kebda* resolves to three lexical items *ik heb dat*. This obviously has an influence on prediction tasks such as part-of-speech (POS) tagging where POS are assigned on token level. The difficulty is amplified when genre-specific characteristics crop up, such as flooding characters, abbreviations or emoticons in UGC:

- (3) @MisJeke hahaha you goooo girl;-) Laat je ma es volledig gaan
 @MisJeke hahaha you go girl;-) Laat je maar eens volledig gaan
 @MisJeke hahaha you go girl;-) Let you just one complete go
 ‘@MisJekeI you go girl;-) Let yourself completely go for once’

Example (3) illustrates a number of difficulties that UGC poses for NLP. First of all, we are regularly faced with code-switching. Code-switching describes the mixture of one or more languages within the same context, in this case one tweet. To appropriately process such data, the preliminary step of language identification has to be performed, which is difficult in the context of UGC since vocabulary and syntax differ from the standard form. The differences include the flooding of characters (*goooo*) or the assimilation of orthography to pronunciation such as *ma* instead of *maar* (Engl. but). Moreover, presumably simple tasks such as tokenization are impaired due to emoticons or so called *@-replies*. Besides this, there are numerous non-words (such as *hahaha* which symbolizes laughter) or abbreviations (such as *lol* for *laughing out loud*).

Beyond that, there are non-standard texts that differ from the standard **in terms of syntax** such as e.g. poetry:

- (4) O! had my Fate been join’d with thine,
 As once this pledge appear’d a token
 These follies had not, then, been mine,
 For, then, my peace had not been broken.
 (Lord Byron, To a Lady)

The parse tree for this first stanza of Lord Byron’s “For a Lady” (cf. Example (4)) is shown in Figure 3.1.

The parser struggles on one hand with the upper-case spelling at the beginning of a verse, as well as the interjection *O* followed by an exclamation mark. Even though the assignment of POS works well since the poem is not that different from standard English on the lexical level,

²Normalization is considered as the conversion to the standard form which in this case equals the Dutch spoken in the Netherlands.

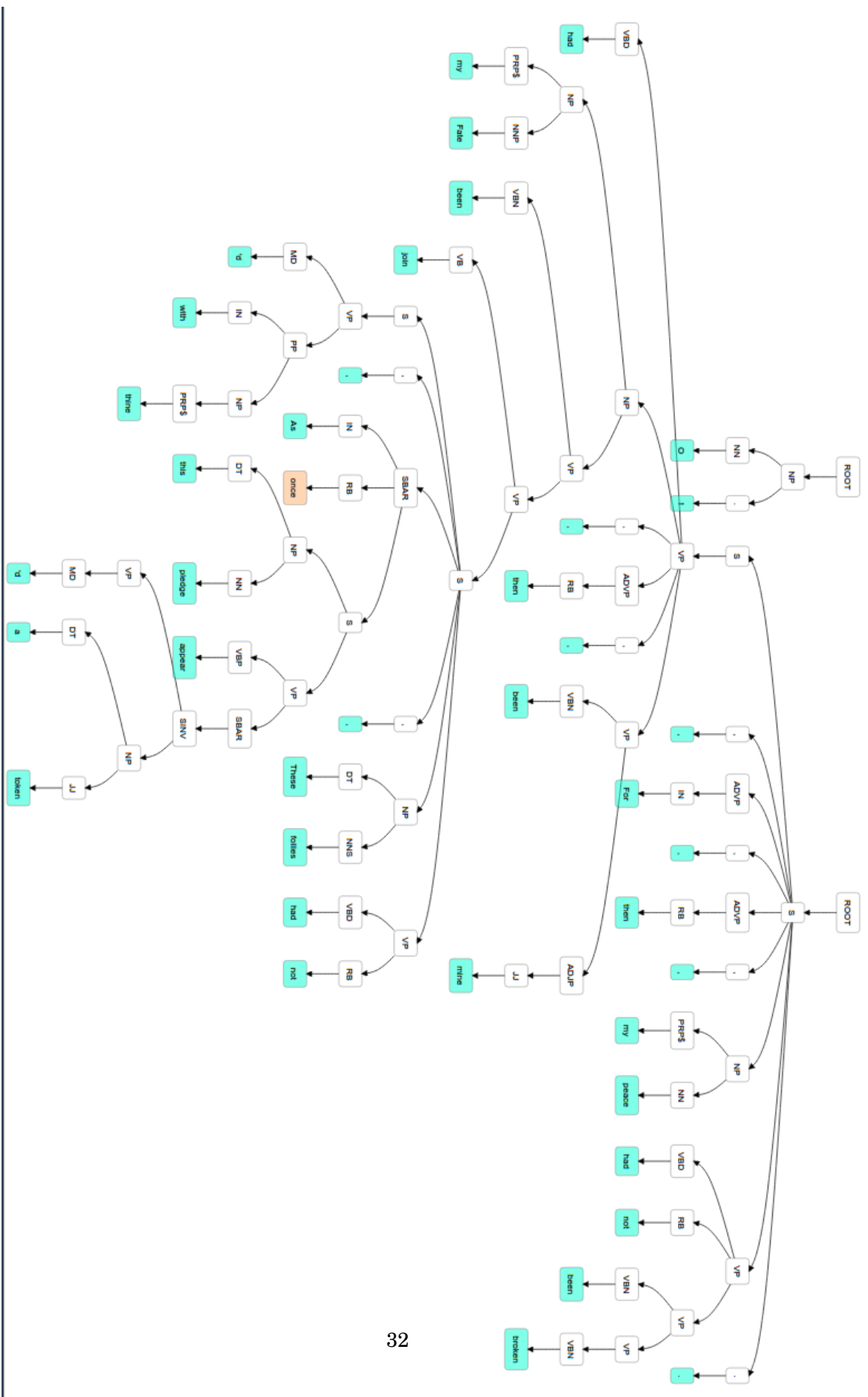


Figure 3.1: Constituent tree using Stanford's CoreNLP constituency parser on the first stanza of "To a Lady".

the parser struggles with the coordination of subclauses since there is an obvious deviation on the syntactical level. As a result, e.g. verse three (“These follies had not, then, been mine”) is torn apart.

There are texts that **combine both, lexical as well as syntactical deviations**, e.g. texts from former stages of a language. This can be illustrated by Middle High German (MHG), the historical stage of German spoken between 1050 and 1350, which differs considerably in syntax (cf. Ziegler and Braun (2010)) and lexicon from its modern stage.

- (5) a. Uns ist in alten mæren wonders vil geseit
 von helden lobebæren, von grôzer arebeit,
 von fröuden hôchgezîten, von weinen und von klagen,
 von küener recken strîten muget ir nu wunder hœren sagen.
- b. In alten Geschichten wird uns vieles Wunderbare berichtet:
 von ruhmreichen Helden, von hartem Streit, von glücklichen Tagen
 und Festen, von Schmerz und Klage, vom Kampf tapferer Recken:
 Davon könnt auch Ihr jetzt Wunderbares berichten hören.
- c. Wonderous things are told in ancient tales
 Of famous men and bold, of great travails,
 Of joy and festive life, of woe and tears,
 Of warriors met in strife – the wonder shall fill your ears!

Example (5) shows the first stanza of one of the most famous MHG works of Epic poetry, the “Song of the Nibelungs”. (5a) displays the MHG version, (5b) the modern German translation by Brackert (1971) and (5c) the translation into English by Ryder (1962). As opposed to modern German, adjectives can e.g. still appear postposed and the verb order differs (cf. *hœren sagen* (Engl. heard said) vs. *berichten hören* (Engl. said hear)). In addition, even though some words have kept their surface form, a few have changed their meaning, such as *hôchgezîten*) which means *festive life* and not weddings – as the modern German word *Hochzeiten* would lead one to believe. Other words were completely replaced by different words, such as *muget* (2. person plural of *mugen*) which translates to *könnt* (Engl. *can*) in modern German.

The results of a German POS tagger model (Schneider and Volk, 1998) on the first stanza of *The Songs of the Nibelungs* is shown in Figure 3.2.

Even though the result is not catastrophic – about half of the words are tagged correctly –, problems with special characters, word order and unknown words become apparent especially in verse 3 and 4.

The same holds true for dramatic text. In “The Taming of the Shrew” by William Shakespeare the main character Katharina exclaims:

Uns	ist	in	alten	mæren	wunders	vil	geseit			
PPER	VVFIN	APPR	ADJA	ADJA	NN	NE	VVFIN			
von	helden	lobebæren	,	von	grôzer	arebeit	,			
APPR	NN	VVFIN	\$	APPR	ADJA	NN	\$			
von	frôuden	hôchgezîten	,	von	weinen	und	von	klagen	,	
APPR	NN	VVFIN	\$	NE	VVINF	KON	NE	VVINF		
von	küener	recken	strîten	muget	ir	nu	wunder	hœren	sagen	
APPR	ADJA	VVFIN	ADJA	ADJA	ADJ	ITJ	ADJA	NN	VVINF	

Figure 3.2: POS results a modern German tagger model (STTS tagset) on the first stanza of *The Songs of the Nibelungs*.

- (6) Nay, then,
Do what thou canst, I will not go to-day;
No, nor to-morrow, not till I please myself.

The wildness of Katharina, the shrew, can be found in her language. The verses are characterized by written orality in combination with upper-case writing in the beginning of verses and a vocabulary originating from the late 16th century. Automatic syntactic analysis of such dramatic text is difficult.

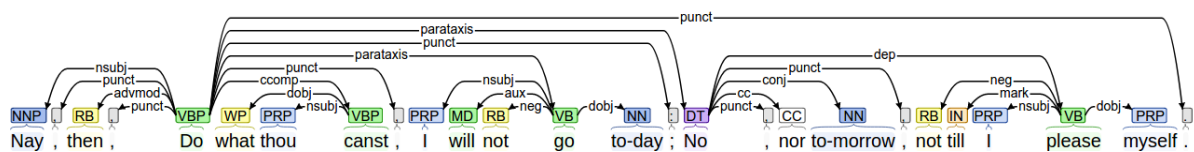


Figure 3.3: Dependency analysis using Stanford's CoreNLP dependency parser on verses from Shakespeare's "Taming of the Shrew".

Even though the POS tagging results are reasonably good despite the upper case writing of the beginnings of verses, the coordination of the sentences resulting from the imitated orality poses a problem for the parser (cf. Figure 3.3). For more successful processing the text first requires "taming".

Dependent on the level on which the data differs from the standard form and the degree to which it differs, various issues using standard text processing tools can be expected. Solutions to these issues depend on the type of data at hand. Examples (1) and (3) serve as instances that are still relatively close to the standard form. The deviations appear only on the lexical level and are systematic to a certain extend. In this case, **text normalization** can be a solution. Text normalization is the transformation of non-standard text to a standard form. Text normalization as an option to deal with the drop in performance of NLP tools observed for non-standard text is discussed in Chapter 5.

For historical texts or texts which follow another syntactical form (such as a poem), text normalization might not be promising. This has various reasons, one of which being the fact that

the structure and features of the texts would be altered to such an extent that a back-projection of annotations to the original text is difficult. However, normally one wants to work with the original form since, especially in DH, its specific features are often the focus of research. For these sorts of text, the **adjustment or development of customized tools** is more suitable. Different training techniques for non-standard text are discussed in Chapter 6.

3.2 Why We Need Data: Machine Learning

3.2.1 What is Machine Learning

ML is a research field which is concerned with teaching computers to learn through the use of statistics without being explicitly programmed. In the following, I will not give a complete introduction into ML but rather focus on the relevance of data for ML. Many of the problems in this dissertation are approached as ML problems and as such are influenced by the nature of the data. The statistical explanations are kept simple to make them accessible to the non-technical reader.

Shalev-Shwartz and Ben-David (2014) define ML as the automated detection of meaningful patterns in data. The goal of machine learning is the generalization of these patterns learned from known data to unseen data (Domingos, 2012). The capacity to learn is a basic characteristic of humans and animals. We learn from experience – our personal data. Humans have the advantage of having common sense that on one hand speeds up learning by specifically being able to confirm or reject assumptions by seeking for more input for uncertain cases and on the other hand helps to reject nonsensical conclusions drawn from incomplete or skewed data. Since computers lack this ability, the sensitive choice of data is crucial to the outcome of machine learning. Thus, the importance of data becomes immediately obvious. The patterns that are to be detected, however, can be very complex dependent on the task and the data. ML is especially helpful for tasks that cannot simply be described by a handful of handcrafted rules.

Supervised vs. Unsupervised Learning There are two basic types of learning. Viewing learning as a process of using experience to make predictions about similar situations, **supervised learning** describes a scenario in which we are provided with training examples (experiences or observations). Those training examples contain information about the so called *label* of an example. For instance, in school we learn that *beautiful*, *hopeful* and *careful* are adjectives. The label is thus the POS “adjective”. Our training set contains three data points. The pattern we might be able to extract from this data is the fact that all three words end in *-ful*. If we are faced with an unseen word such as *wonderful*, we can apply the knowledge we gained from our seen data, namely that words ending in *-ful* are adjectives. This allows us to label an unseen word with the correct label. More formally, supervised learning is having input variables (X)

and an output variable (Y) and using an algorithm for the mapping function from the input to the output.

Unsupervised learning in turn does not make use of labels (output variable (Y)). Assuming that you have the knowledge about what adjectives in English look like but not how to determine whether a word is an adverb or a noun, you might nevertheless be able to group them without knowing their label. Unsupervised learning relies on finding similarities of patterns in unlabeled data. For instance even without knowing the labels, a human can probably divide the words *clarity*, *actually*, *completely*, *ability*, *comfortably* and *solidity* into two groups. One of the groups contains the words ending on *-ty* the other group the words ending on *-ly*. If a computer is told to use the suffix as a criterion for these groups (or even a couple of other so called features), it can perform such a grouping task. In unsupervised learning this task is called clustering. Moreover, there are intermediate learning types called **weakly supervised learning** where the model is trained using examples that are only partially labeled.

In this dissertation mainly supervised methods are applied. Unsupervised and weakly supervised methods are implemented as points of comparison in some cases.

Formalization In the following, the framework of supervised learning is described. It is based on the formal model given by Shalev-Shwartz and Ben-David (2014) in their chapter about the statistical learning framework and by Manning and Schütze (1999, p. 575ff). The following ingredients are part of a supervised classifier:

- the learner’s input
 - **domain set or unlabeled set or test set:** a set X that we may wish to label. In our POS example this set will be the set of all English words, the domain points. Usually, these domain points are represented by a vector of feature values indicating characteristics of each word such as the suffixes, whether they start with an upper case letter or not, whether they contain a digit etc. I also refer to domain points as instances and to X as instance space.
 - **label set:** this set contains as many elements as you have labels to predict. Labeling words with the POS classes *noun*, *adjective*, *verb* and *rest*, one has four labels and the set therefore contains four elements. This can e.g. be represented as a set of numbers $Y = \{0, 1, 2, 3\}$ where 0 corresponds to noun, 1 to adjective, 2 to verb and 3 to rest.
 - **training data:** $S = ((x_1, y_1) \dots (x_m, y_m))$ is a finite sequence of pairs in $X \times Y$: that is, a sequence of labeled domain points. This is the input that the learner has access to (like a set of word that have been labeled with their part-of-speech and their suffixes, upper case letter information and whether they contain digits). Such labeled exam-

ples are often called training examples. I sometimes also refer to S as a training set³.

- the learner's output: The learner is requested to output a *prediction rule*, $h : X \rightarrow Y$. This function is also called *predictor*, a *hypothesis* or a *classifier*. This predictor can be used to predict labels of new domain points.

Algorithms Machine learning comes in many flavors. The different approaches can range from simple to tremendously complex. The common goal of these approaches is to estimate the functional relationship between the input features and the target variable. ML algorithms can be divided into parametric and nonparametric methods (Manning and Schütze, 1999, p. 49).

Parametric algorithms simplify the function used for prediction to a known form. These algorithms involve two steps. First a form for the function is selected and then the coefficients for the functions are learned from training data. Linear regression, logistic regression, perceptron, naive Bayes and simple neural networks are examples for such parametric methods. Even though these algorithms are simple and fast and can be trained with relatively small data sets, they have limitations. Due to the fixed form of the function they are highly constrained to this specific form. Normally, those methods are suited for simpler problems. In practice, they will very likely not match the underlying mapping function perfectly and can in many cases be a poor fit.

Nonparametric algorithms do not make such strong assumptions about the form of the mapping function. The form can therefore be learned directly from the data. The point is to balance the fit of the function to the training data and yet to maintain the ability to generalize to unseen data. Examples for such algorithms are k-nearest neighbors, decision trees or support vector machines. These models are more flexible and powerful but need more data to fit the form which also makes them slow. For successful modeling of a task with nonparametric algorithms the quality and quantity of training data is key.

Measuring success After fitting a machine-learning model, the next step is to assess the accuracy of that model. Having an impression of how well a classifier works before applying it to unseen data is essential in order to trust the results it returns and to comfortably carry out analyses on the results. Likewise, if the predictive performance is not satisfying for the task at hand, you can revisit your data and model to try to improve and optimize its accuracy. There are a variety of evaluation measures which usually make use of manually labeled gold standard (or ground truth) data (Sparck Jones and Galliers, 1996). These are evaluation sets for which the correct labels are known. Thus, it is possible to compare the predictions made by a classifier to the actual labels. In Section 2.3.4, I discussed the appropriateness of such numerical measures

³Despite the “set” notion, S is a sequence. In particular, the same example may appear twice in S and some algorithms can take into account the order of examples in S

to assess the quality or results of a project. For NLP subcomponents of such projects, numerical values might show improvements through changing e.g. a parameter of an algorithm which one can expect to reflect an improved performance for the whole system on an application task. Which evaluation measure to apply depends on the task and the distribution of labels in the data. Equally important as the choice of the right measure is the comparison of the results to an upper bound or lower bound (baseline) (Manning and Schütze, 1999, p. 233). The upper bound is usually the human performance on a task since it is assumed that a computer cannot outperform a human on complex tasks that require expert knowledge. A baseline gives an impression of how hard the task at hand really is. 60% accuracy can be quite a high accuracy value if the baseline achieves merely 20%. However, if a baseline already reaches 55% accuracy, 60% is not much of an improvement. Baselines can e.g. be results achieved by previously developed approaches, randomly assigned labels or the assignment of the majority label to all data points. Since the goal is to evaluate how well a classifier generally works on unseen data, cross-validation is a common method to ensure that the evaluation does not reflect oddities in the training as well as the test set (Manning and Schütze, 1999, section 6.2.4). By repeatedly splitting the labeled data into different training and test sets and averaging the results, a more realistic picture of the actual performance of a classifier can be given.

Non-standard data and machine learning The importance of data as a surrogate for real-world experiences in ML is significant. As mentioned above, this data is often presented to an algorithm in the form of vectors consisting of feature values. The set of informative features varies throughout tasks and data sets. Features can obviously also be extracted for non-standard data. The question arises where the problem of decreased performance of NLP tools on such data is rooted. The problem is illustrated by Example (7) and (8).

(7) They obviously will be there

(8) They obv will b there :)

The principle of features for learning relies on the fact that instances of the same class have similar feature vectors. In the example of distinguishing adjectives and adverbs, all adverbs have the same feature value for the feature *ends in -ly*. Example (8) is a version of (7) as it could be found in UGC. Due to character limitations in SMS or Tweets words often get abbreviated. Even though the abbreviated form *obv* for *obviously* still functions as an adverb, the telling feature of the *-ly* suffix would take another value than in the original sentence. Applying a POS tagger trained on data like Example (7) with only this feature would not label *obv* in Example (8) as an adverb. I make two observations. The first observations is that the characteristics of training data and the data of application should not differ considerably with respect to their features. This is the main problem when applying tools trained on standard data to data from non-standard domains. This holds for different tasks and levels of deviations from the norm. The

second observation is that the choice of features matters. How hard or easy learning is might be decided by how well your features correlate with your class (Domingos, 2012). In this case a feature has been chosen which is very sensitive to the surface realization of the specific word. If word order features would be included, they could make up for deviations in spelling, since the word order does not vary between Example (7) and (8). The fact that there are features that are more robust than others is utilized in feature-level domain adaptation (cf. Blitzer et al. (2006)). Another important factor is the size of the dataset. Especially for nonparametric algorithms a certain amount of data is needed since the predictor function is learned from the data without assumptions about its form. Data in DH, however, is often not available in sufficient quantity. How this is reflected in annotation practice, choice of algorithms and the general relationship between ML and DH is discussed in the following sections.

3.2.2 (Big) Data and Digital Humanities

In recent years, there has been a successful comeback of artificial neural networks (ANNs) in different fields of computer science and particularly successful in NLP. The advances in available memory and computational power in the last two decades solved some of the initial problems that neural networks (NNs) were faced within the early 90s. Today, artificial neural networks are often implemented using deep architectures of several hidden layers of artificial neurons (Bengio and Bengio, 2000; Ranzato et al., 2007). This method is referred to with the term *deep learning* which was first introduced in connection with ANNs by Aizenberg et al. (2000). The strong interest in this particular type of ML is partly grounded in the fact that the time-consuming feature engineering task is left to the learning algorithm itself⁴. This can be a big advantage for non-standard data. Given a complex task such as automatic syntax annotation, high quality features such as POS annotation or chunking are often a requirement for successful modeling with many traditional learning algorithms. However, this assumes the availability of preprocessing tools for this kind of data such as POS taggers or chunkers to extract these features. This assumption often is not met when working with non-standard texts. An avoidance of such feature input is desirable.

As NNs can be considered non-parametric algorithms as they do not assume any specific form of predictor function, they are in need of rather large amounts of data. Big data and ANNs therefore seem to be the perfect match (Chen and Lin, 2014). This builds up to a problem for non-standard text processing: even though the avoidance of explicit feature modeling in the framework of neural modeling would be an advantage, the lack of sufficient data makes these methods inappropriate. Kaplan (2015) argues that big data (Diebold, 2012) and DH are not contradictions by opposing Big Data Digital Humanities with Small Data Digital Humanities. Big Data Digital Humanities, he reasons, focus on massive cultural digital objects and include

⁴Even though this is only partly true since the problem is shifted to how to represent the input to a neural network and the choice of architecture.

large-scale corpora such as the millions of books scanned by Google and can therefore make use of recent developments in machine learning. This, however, covers just the few projects that are interested in these large-scale corpora. Schöch (2013) describes that big data in the humanities is not the same as big data in the natural sciences. He argues that there is neither a constant influx of new data as typical for big data such as e.g. the Internet nor are there amounts of data available that would seriously qualify as “big”. However, an important aspect of DH big data according to Schöch (2013) is the variety of formats, complexity and lack of structure which is what causes a shift from “close reading” to “distant reading” (Moretti, 2013) in the humanities. Borgman (2015) devotes an entire chapter to diversity of data due to diversity of research projects in data scholarship. This diversity is the reason of the claims made in Section 2.3. Methods have to be reusable and adjustable to account for this diversity and to serve as an orientation for different projects. At the same time due to this diversity specific problem solving is needed to account for specificities of data and research questions. In the following, I elaborate on thoughtful data annotation as basis for successful application of ML techniques in the context of humanities research.

3.3 Annotation of Data

Due to the lack of vast amounts of data, the data has to be “smart” (Schöch, 2013). This means that data has to be structured or semi-structured; it has to be explicit and enriched. This means that in addition to the raw text, it should contain markup, annotations and metadata. Labels are feedback for a classifier in order to group data points together. These labels are usually annotated manually and can contain information on different levels such as syntax, morphology, lexical information or semantics.

3.3.1 Corpus Annotation

Corpus linguistics as a field going back to the 1950s (Busa, 1980), involves the collection of texts in digital form for more than half a century already. In Chapter 2, I have described how the lack of annotations led to a concentration on research questions that could be inspected with the analysis of word frequencies. The tradition of enrichment of raw text corpora with manually annotated layers of information started in the 1970s with the Brown Corpus (Francis and Kucera, 1979) and the Lancaster-Oslo-Bergen Corpus (LOB) (Johansson, 1978). They contain POS annotations for the extraction of linguistically interesting patterns. These labeled data sets served as first training material for ML approaches in NLP. The success of these corpora triggered a predominance of ML in this field. Since ML entered into the world of DH, annotations became an indispensable element of its methodology. Within the last 40 years, the NLP community has worked out a feasible workflow for data annotation that serves to ensure high-quality annota-

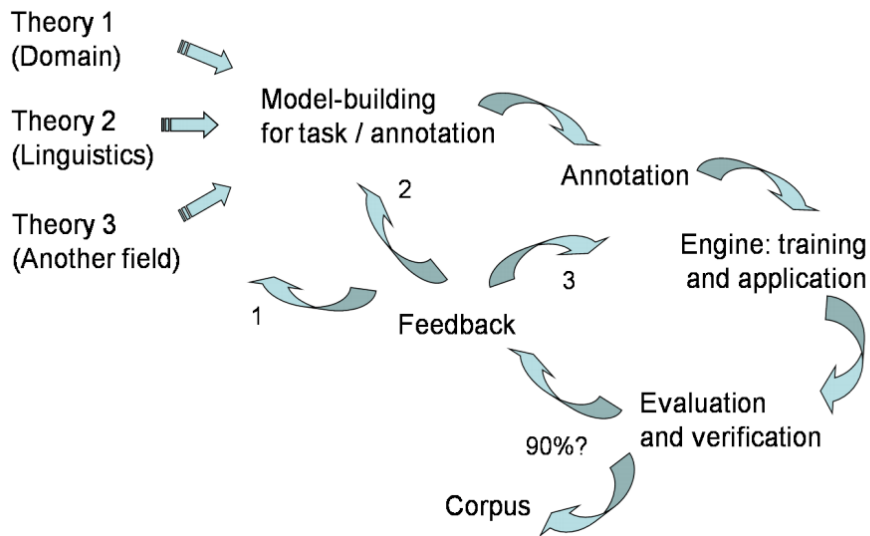


Figure 3.4: The annotation workflow as described by Hovy and Lavid (2010)

tions. I introduce the main concepts of this workflow based on the article by Hovy and Lavid (2010).

1. **selection of text:** Biber (1993) describes that corpus design is an iterative process that has to be initiated by theoretical research. In the context of general corpora, which serve as a basis for training generally applicable tools⁵, balancedness of genres and modality are important factors.
2. **annotation guidelines:** often also called codebook or manual, the set of tags used for annotation have to be determined based on a theory or linguistic concept and concrete decisions on how to annotate the data have to be made.
3. **pilot annotation:** some corpus fragments of the training corpus are annotated using the initial guidelines in order to determine the feasibility of the decision stated therein. These annotations are done by more than one annotator (parallel annotations).
4. **inter-annotator agreement:** determines the agreement between annotators and decides on a satisfactory level. If the agreement is too low, ML algorithms cannot be trained successfully. Refinements of the guidelines starting from step 2 are repeated until the agreement suffices.
5. **annotation of corpus:** with the final guidelines large portions of the corpus can be annotated.

⁵If such tools exists.

Figure 3.4 visualizes the annotation workflow described by Hovy and Lavid (2010). It shows how theories from the humanities initiate the compilation of annotation guidelines. These guidelines are then tested and refined in an iterative process until they reach a high level of inter-annotator agreement which shows that the right level of abstraction and concreteness has been reached.

Even though DH projects can draw on this well-developed NLP workflow for annotation, there are a few differences that have to be emphasized.

One difference is the representativeness of the corpus. Manning and Schütze (1999) define a corpus as “representative” of a phenomenon when what we find for the phenomenon in the sample corpus also holds for the general population or textual universe. In NLP, corpus development is rarely just for single use. Usually corpora are developed to serve as a training corpus for one of the basic NLP tasks in a language and should thus be representative for the entirety of a language. Corpora in DH often serve to answer a specific research question. This influences the selection of texts that are contained in this corpus as well as the categories that are annotated. In NLP, there are established definition for diverse categories such as POS and named entities which can be used for annotations with just slight adjustments for different languages. To be useful for the investigation of humanistic research questions, these categories often have to be extended or varied. An example for such a case is the CUTE project in which the basic definition of named entity is extended to make it more suitable for DH-specific research questions⁶. The guidelines of this project state that not only the mention of an actual name is considered an entity but any mention of an expression referring to an entity. Blessing et al. (2017) show how the creation of character networks in literary texts can profit from such an extended concept of entities since it leads to a more complete impression of character relations.

Another difficulty is the change of culture that comes along with rather operationalized annotation of concepts. Belanger (2010) describes the creation, use and organization of annotations in DH research and how the digital documents lead to a dual-medium representation. Moreover, the level of abstraction from actual data that is necessary to find a certain level of consensus is an unusual approach in the humanities. Annotations in DH can be seen as a close reading step in which the humanities scholars who annotate one or several aspects familiarize themselves with the textual material especially through the process of finding agreement between annotators. This leads to a deep understanding of the texts and a more objective view. Thus, annotation in DH is not merely a necessary step to create training data for ML approaches but rather a substantial part of understanding the problem and developing a theory. Bögel et al. (2015a) call this the *extended hermeneutic circle*.

The heureCLÉA project⁷ is an example of a DH project based on an NLP inspired annotation workflow. The project uncovers the methodological transformations that annotation brings upon

⁶<https://www.creta.uni-stuttgart.de/cute/datenmaterial/annotationsrichtlinien-1-1/>, 23/08/2017.

⁷<http://www.heureclea.de/>, 23/08/2017.

humanistic disciplines by investigating the interplay between humanities research informed annotations and ML. Bögel et al. (2015a) argue that Bögel et al. (2015b) argue that “epistemological reconceptualization[,] (...) propagation of an empiristic humanistic research practice[,] (...) renewed interest in the sociological dimensions of interpretative practices [and] (...) social dimensions of the humanities research practice itself” are influenced by this methodological shift. In contrast to NLP annotations which require unambiguous and agreeable annotation decisions, Bögel et al. (2015b) describe ambiguity as a condition for “an adequate conceptualization of the notion of ‘object’ common to the humanities”. They explain that a hermeneutic interpretation is not static but can only be successful in the historical context of an object. Therefore, there cannot be only one correct annotation. Zweig et al. (2017) implement the potential ambiguity or uncertainty with an extra label in their annotation of humor in Youtube⁸ comments. This explicitly models the fact that not all annotation decisions are straight forward and are often dependent on the context. Such information can be taken into account for the evaluation of a system.

These humanistic approaches towards textual phenomena lead to the desirability of flexible annotation schemes for DH. This goes hand in hand with the development of markup schemes that allow for such flexibility. Different markup schemes have been used for annotating structure of text and for text enrichment. The key requirements are reusability, interchange, system- and software-independence and portability to facilitate collaboration in the humanities. In 1986, the Standard Generalized Markup Language (SGML) was published as an ISO standard (ISO 8879:1986) (Goldfarb, 1990). In 1998, the World Wide Web Consortium (W3C) published their recommendation on Extensible Markup Language (XML). This laid the foundation for the probably most important DH markup language, the Text Encoding Initiative (TEI)⁹ which is an XML-based scheme, specifying encoding methods for machine-readable texts. It is chiefly used in the humanities, social sciences and linguistics. The reason for its success is the freedom of expression this scheme leaves to the user. It allows for their own theory of text by enabling the encoding of features they deem important in the text. At the same time, it ensures a certain degree of standardization and thus reusability and interoperability as pointed out by Renear (2004).

3.3.2 Data Selection

Apart from the creation of precise annotation guidelines, one has to take the composition of the data to be annotated into account. This composition might naturally be determined by the research question. In a project in which we focus on the analysis of Goethe’s *Werther*, we most likely want to annotate text parts coming from this work. This criterion of relevance is an important point and subsumes the condition that training data should be close to the target data,

⁸<https://www.youtube.com/>, 28/08/2017.

⁹<http://www.tei-c.org/index.xml>, 22/08/2017.

i.e. the data that will eventually be automatically annotated with the resulting tool. In NLP, researchers often tend to cover “as much ground” as possible in order to develop generally applicable tools. DH offers the advantage that the domain of application is known, which allows for a much more target-oriented collection of training data. Therefore, it is worth optimizing the data compilation to this end. First and foremost, it is important to represent all target categories in the annotations. Categories that do not appear in the manually annotated data, cannot be learned by a classifier and will therefore not be annotated automatically later. Furthermore, there is the recurrent question of quantity of annotations. The answer “the more the better” is neither concrete enough nor necessarily correct. The question arises, how many annotated examples are enough and whether there is an optimal distribution of categories to be annotated. It is important to point out that the answer to this question is dependent on the actual task and the number of labels.

I will investigate the example of POS annotation. To this end, I use the Index Thomisticus Treebank¹⁰, a Latin data set which is already annotated with 17 POS categories¹¹. This way, different sampling techniques along with different numbers of annotated instances can be tested in order to evaluate how these factors influence the performance of the resulting classifier.

The influence of different training set sampling methods is evaluated in terms of accuracy of the resulting tagger model.

Accuracy is defined as follows:

$$\text{(Accuracy)} \quad Acc = \frac{\text{true positives} + \text{true negatives}}{\text{true positives} + \text{true negatives} + \text{false positives} + \text{false negatives}}$$

The following four methods are compared:

1. random sampling
2. maximizing the type-token ratio (TTR) based on word forms
3. maximizing the type-token ratio (TTR) based on lemmas
4. maximizing the Shanon Diversity Index (SDI) of POS tags

The measures are defined as follows:

$$\text{(Type-Token Ratio)} \quad TTR = \frac{\# \text{ types in corpus}}{\# \text{ tokens in corpus}}$$

¹⁰<http://itreebank.marginalia.it/>, 21/6/2017.

¹¹For further information visit <http://universaldependencies.org/la/pos/index.html>.

(Shanon Diversity Index)

$$H = - \sum_{i=1}^S p_i \ln p_i$$

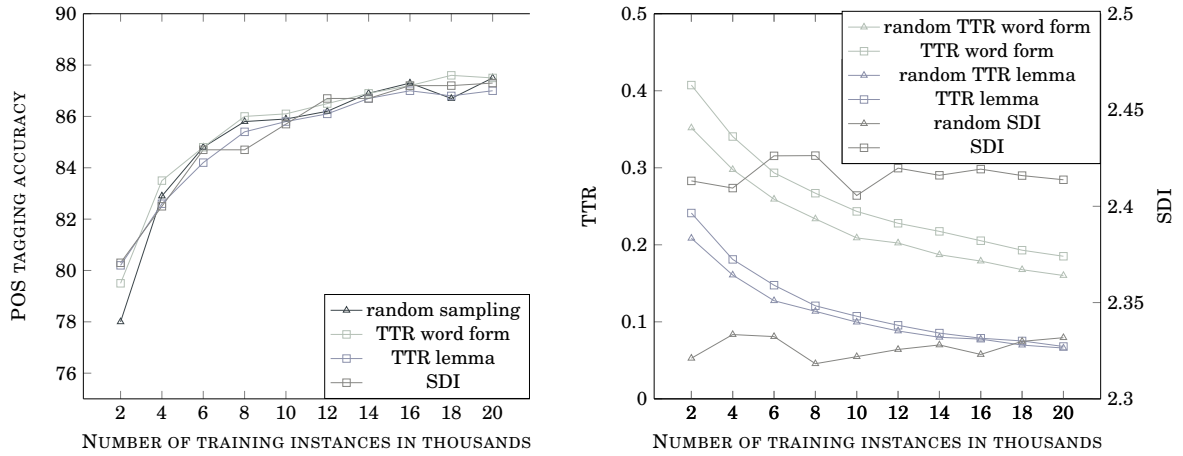
Type-Token Ratio is a measure commonly used in quantitative linguistics and gives insights into the diversity of vocabulary of a text dependent on the text length. The Shanon Diversity Index (Shannon, 1948) is commonly used in biology to characterize the diversity in a community. The proportion of species i relative to the total number of species (p_i) is calculated, and then multiplied by the natural logarithm of this proportion ($\ln p_i$). The resulting product is summed across species, and multiplied by -1. Replacing species by POS classes in our application guarantees us a high diversity of classes in our training set. This method can only be applied in scenarios in which one wants to sample from an already annotated data set and not for situations in which it still has to be decided which data portions ought to be annotated.

sampling	2,000	4,000	6,000	8,000	10,000	12,000	14,000	16,000	18,000	20,000
random	78.0	82.9	84.8	85.8	85.9	86.2	86.9	87.3	86.7	87.5
TTR wf	79.5	83.5	84.8	86.0	86.1	86.5	86.9	87.2	87.6	87.5
TTR lem.	80.2	82.6	84.2	85.4	85.8	86.1	86.7	87.0	86.8	87.0
SDI	80.3	82.5	84.7	84.7	85.7	86.7	86.7	87.2	87.2	87.3

Table 3.1: Tagging accuracy on POS tagging models for Latin dependent on the sampling method (random, type-token ratio based on word form (wf) and lemma (lem.) and Shannon-Diversity Index) and size of the training set.

The training sets are iteratively compiled for 2-4 by adding one sentence at a time from a set of 50 sentences, which increases the respective measure further until the target size of the set is reached. I do not choose the sentence that increases the measure the most to avoid a bias towards short sentences. Instead, TTR and SDI are calculated for all 50 sentences in the option pool and are then sorted in ascending order. The sentence that ranks at $\frac{2}{3}$ of the ordered list is added to the training set. This is repeated until the desired number of tokens per training set is reached. All four settings are evaluated for training set sizes between 2,000 and 20,000 tokens in incrementing steps of 2,000. The results are summarized in Table 3.1.

Notably, a crucial difference in accuracy between the sampling techniques can be observed for the smallest training set. The accuracy of the randomly sampled set, is consistently below the accuracy of those sampled with a more elaborate technique. Nevertheless, these differences disappear for larger training set sizes for all but the sets sampled based on higher TTR in which types are word forms. Even though the TTRs and SDIs are significantly different (cf. Figure 3.5(b)) from a rather small number of tokens on, these do not influence the accuracy of the resulting model. Thus, attention in sampling methods needs to be paid especially in projects where only a really small number of annotations are planned.



(a) Learning curve for all POS tagging models for Latin dependent on the sampling method and size of the training set. (b) TTR/SDI curves for all POS tagging models for Latin dependent on the sampling method and size of the training set.

Figure 3.5: Learning curves for different sampling methods for 2,000 to 20,000 tokens.

3.4 Summary

In this chapter, I introduce the definition for non-standard text along with two levels of deviation, namely lexical and syntactical deviation. I illustrate these deviations with different examples and I show how they influence typical NLP tasks. In order to describe the consequences that these deviations have for the application of standard tools on such data, I introduce basic concepts of ML-based techniques with the focus on supervised ML. I discuss the influence of the low resource situation on potential techniques and highlight the importance of targeted annotations and data selection for the successful modeling of humanistic research questions.

DATA HARVESTING: MODULARIZED AND ADAPTABLE ARCHITECTURES FOR DIGITAL HUMANITIES

In the previous chapter, I discussed the non-standard nature of texts in the Digital Humanities (DH) and the consequences for natural language processing (NLP) applications and machine learning techniques. In this chapter, I will take one step back and explore an option for data acquisition since the basic requirement for the successful implementation of such text-based projects often is a possible stumbling block. Large digital corpora comprising textual material of interest are rare. Archives and individual scholars are in the process of improving this situation by applying optical character recognition (OCR) to the physical resources. In the *Google Books*¹ project, books are being digitized on a large scale. But even though collections of literary texts like *Project Gutenberg*² exist, these collections often lack the texts of interest to a specific question.

As an example, we describe the compilation of a corpus of adaptations of Goethe's *The Sorrows of Young Werther*. This epistemic novel published in 1774 triggered an unprecedented flood of adaptations starting right after its publication and still persisting today (cf. Scherpe (1970), Piper and Algee-Hewitt (2014)). There are various aspects that a text has to exhibit in order to be considered an adaptation of Goethe's *Werther*, a so-called "Wertheriade". These aspects such as stylistic similarity, references to concepts invoked in *Werther*, typical character constellations and alike are discussed in secondary literature (cf. Martens (1985), Horré (1997)). Even though these aspects clearly constitute important factors for the identification of *Wertheriaden*, the definitions remain blurry because they are considered jointly and not as independent criteria. Creating a digital collection of adaptations allows for the systematic investigation of these

¹<https://books.google.de/>, 02/04/2017.

²<http://www.gutenberg.org>, 14/04/2017.

CHAPTER 4. DATA HARVESTING: MODULARIZED AND ADAPTABLE ARCHITECTURES FOR DIGITAL HUMANITIES

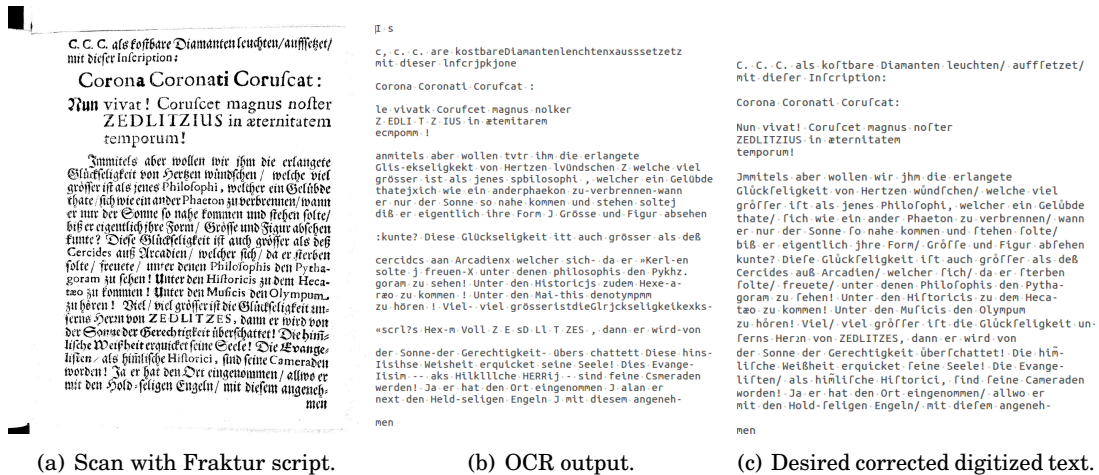


Figure 4.1: Three stages that a text has to go through from the scanned image of a book to the perfect transcription.

aspects, such as the analysis of character networks throughout the publishing history of this work (Murr and Barth, 2017).

The success of OCR is highly dependent on the quality of the printed source text. Recognition errors, however, impact the results of computer-aided research (Strange et al., 2014). Especially for older books which are set in hard-to-read fonts and with stained paper the output of OCR systems is not good enough to serve as a basis for DH research. Figure 4.1 shows how the text written in Fraktur in the scan shown in (a) contains a variety of recognition errors in (b). The desired output, thus the perfect transcription of the text in (a), is shown in (c). To reach this transcription, the recognized text needs to be post-corrected in a time-consuming and cost-intensive process.

We describe how we can support and facilitate the manual post-correction process with the help of informed automatic post-correction. We illustrate the importance of **reusability and adaptability** of NLP tools in DH which we discuss in Section 2.3.2. To account for the problem of relative data sparsity, we highlight how a generic but highly modularized architecture that is agnostic to a specific domain can be adjusted to text specificities such as genre and font characteristics by including only small amounts of domain-specific data. We emphasize the significance of **problem-specific solutions** for DH. Moreover, we show how the same architecture can be adjusted to different languages with only little expenditure of time. We suggest a system architecture (cf. Figure 4.2) with trainable modules which joins general and specific problem solving as required in many applications. We demonstrate that the concepts of **reusability and specific problem solving** are mutually compatible. We find that the combination of modules via a ranking algorithm, including a language model, yields results far above the performance

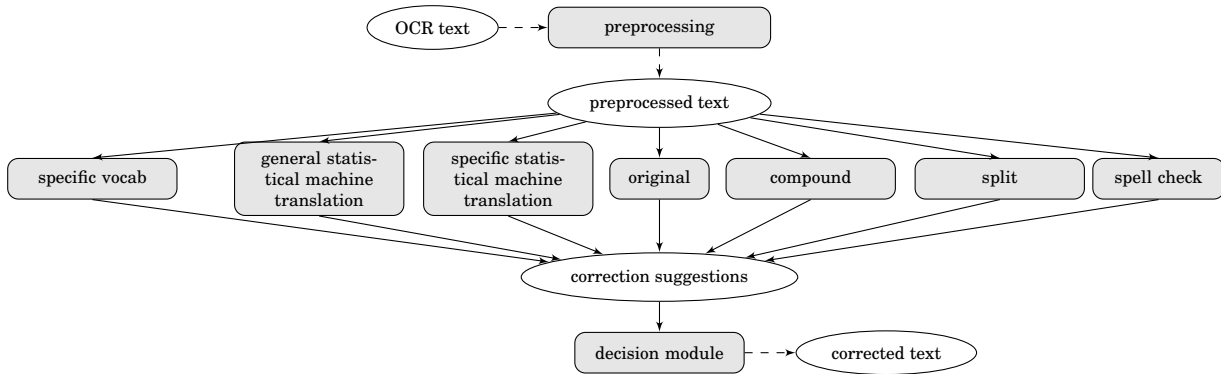


Figure 4.2: Multi-modular OCR post-correction system.

of single approaches.

In Section 4.1, we discuss the point of departure for our research and we introduce our evaluation metrics in Section 4.2. In Section 4.3, we present the data we base our system on. In Section 4.4, we illustrate the most common errors and describe our multi-modular, partly customized architecture. Section 4.5 gives an overview of techniques included in our system and the ranking algorithm. In Section 4.6, we discuss results, the limitations of automatic post-correction, and the influence the amount of training data has on the performance of such a system. A general architecture that is easy to adjust is an important part of DH projects, since time limits frequently do not allow for “from-scratch” development. We demonstrate that it is possible to easily adapt our architecture to other languages in Section 4.7. In Section 2.3.1, we address the need for real-world applications for DH. Section 4.8 describes a way to meet this need by efficiently integrating the results of our research into a digitization work-flow. We consider the **easy accessibility** of computational methods to be a central point in DH collaborations.



Publication

Parts of this chapter were published in Schulz and Kuhn (2017).

4.1 Related Work

There are two obvious ways to automatically improve quality of digitized text: optimization of OCR systems or automatic post-correction. Commonly, OCR utilizes only basic linguistic knowledge like the character set of a language or reading direction. The focus is on the image recognition aspect, which is currently often done with artificial neural networks (cf. Graves et al. (2009), Desai (2010)).

Post-correction is focused on the correction of errors in the linguistic context. It thus allows for the purposeful inclusion of knowledge of the text at hand, e.g. genre-specific vocabulary. Nevertheless, post-correction has predominantly been tackled in a way that is agnostic to the OCR

system as outlined below. As an advantage, post-correction can also be applied when no scan or physical resource is available.

There have been attempts towards shared data sets for evaluation. Mihov et al. (2005) released a corpus covering four different kinds of OCRed text comprising German and Bulgarian. However, in 2017 the corpus was untraceable for download and no recent research relating to the data could be found.

OCR post-correction is applied in a diversity of fields in order to compile high-quality data sets. This is not merely reflected in the homogeneity of techniques but in the metric of evaluation as well. While accuracy has been widely used as evaluation measure in OCR post-correction research, Reynaert (2008a) advocates the use of precision and recall in order to improve transparency in evaluations. Depending on the paradigm of the applied technique, even evaluation measures like BLEU score can be found (cf. Afli et al. (2016)).

Since shared tasks are a good opportunity to establish certain standards and facilitate the comparability of techniques, the Competition on Post-OCR Text Correction³ organized in the context of the 14th International Conference of Document Analysis and Recognition (ICDAR 2017) could be a milestone for more unified OCR post-correction research efforts.

Regarding techniques used for OCR post-correction, there are two main trends to be mentioned: statistical approaches utilizing error distributions inferred from training data and lexical approaches oriented towards the comparison of source words to a canonical form. Combinations of the two approaches are also available.

Techniques residing in this **statistical** domain have the advantage that they can model specific distributions of the target domain if training data is available. Tong and Evans (1996) approach post-correction as a statistical language modeling problem, taking context into account. Pérez-Cortes et al. (2000) employ a stochastic finite-state automaton along with a modified version of the Viterbi Algorithm to perform a stochastic error correcting parsing. Extending the simpler stochastic context-sensitive models, Kolak and Resnik (2002) apply the first noisy channel model, using edit distance from noisy to corrected text on character level. In order to train such a model, manually generated training data is required. Reynaert (2008b) suggests a corpus-based correction method, taking spelling variation (especially in historical text) into account. Abdulkader and Casey (2009) introduce a neural network for error estimation that learns to assess error probabilities from ground truth data which in turn is then suggested for manual correction. This decreases the time needed for manual post-correction since correct words do not have to be considered as candidates for correction by the human corrector. Llobet et al. (2010) combine information from the OCR system output, the error distribution and the language as weighted finite-state transducers. Reffle and Ringlstetter (2013) use global as well as local error information to be able to fine-tune post-correction systems to historical documents. In relation to the approach introduced by Pérez-Cortes et al. (2000), Afli et al. (2016) use statistical machine

³<https://sites.google.com/view/icdar2017-postcorrectionocr/home>, 03/07/2017.

translation for error correction using the Moses toolkit on a character level. Volk et al. (2010) merge the output of two OCR systems with the help of a language model to increase the quality of OCR text. The resulting corpus of yearbooks of the Swiss Alpine Club which has been manually corrected via crowdsourcing (cf. Clematide et al. (2016)) is available from their website.

Lexical approaches often use rather generic distance measures between an erroneous word and a potential canonical lexical item. Strohmaier et al. (2003) investigate the influence of the coverage of a lexicon on the post-correction task. Considering the fact that writing in historical documents is often not standardized, the success of such approaches is limited. Moreover, systems based on lexicons rely on the availability of such resources. Historical stages of a language – which constitute the majority of texts in need for OCR post-correction – often lack such resources or provide incomplete lexicons which would drastically decrease performance of spell-checking-based systems. Ringlstetter et al. (2007) address this problem by suggesting a way to dynamically collect specialized lexicons for this task. Takahashi et al. (1990) apply spelling correction with detection of the preceding candidate word. Bassil and Alwani (2012) use Google’s online spelling suggestions as they draw on a huge lexicon based on contents gathered from all over the web.

The **human component** as final authority has been mentioned in some of these projects. Visual support of the post-correction process has been emphasized by e.g. Vobl et al. (2014) who describe a system of iterative post-correction of OCRed historical text which is evaluated in an application-oriented way. They present the human corrector with an alignment of image and OCRed text and make batch correction of the same error in the entire document possible. They can show that the time needed by human correctors considerably decreases.

4.2 Evaluation Metrics

We describe and evaluate our data by means of word error rate (WER) and character error rate (CER). The error rates are a commonly used metric in speech recognition and machine translation evaluation and can also be referred to as length-normalized edit distance. They quantify the number of operations, namely the number of insertions, deletions and substitutions, that are needed to transform the suggested string into the manually corrected string and are computed as follows:

$$(WER) \quad WER = \frac{\text{word insertions} + \text{word substitutions} + \text{word deletions}}{\text{\# words in the reference}}$$

$$(CER) \quad CER = \frac{\text{char insertions} + \text{char substitutions} + \text{char deletions}}{\text{\# characters in the reference}}$$

1	Berichtigung der Geschichte des jungen Werthers	H. von Breitenbach	1775
2	Schwacher jedoch wohlgemeynter Tritt vor dem Riss, neben oder hinter Herren Pastor Goeze, gegen die Leiden des jungen Werthers und dessen ruchlose Anhänger	anonymous	1775
3	Lorenz Konau	David Iversen	1776
4	Werther der Jude	Ludwig Jacobowski	1910
5	Eine rührende Erzählung aus geheimen Nachrichten von Venedig und Cadir (first letter)	Joseph Codardo und Rosaura Bianki	1778
6	Afterwerther oder Folgen jugendlicher Eifersucht	A. Henselt	1784
7	Der neue Werther oder Gefühl und Liebe	Karl P. Bonafont	1804
8	Leiden des modernen Werther	Max Kaufmann	1901

Table 4.1: Werther texts included in our corpus from different authors and times of origin.

4.3 Data

As mentioned in the introduction, errors found in OCRed texts are specific to time of origin, quality of scan and even the characteristics of a specific text. Our multi-modular architecture paves the way for a solution taking this into account by including general as well as specific modules. Thus, we suggest to include domain-specific data as well as larger, more generic data sets in order to enhance coverage of vocabulary and possible error classes. The data described hereafter constitutes parallel corpora with OCR output and manually corrected text which we utilize for training statistical models.

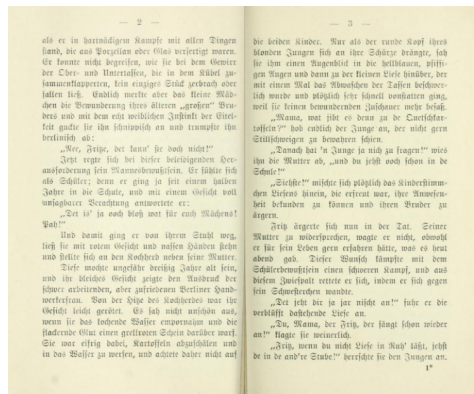
4.3.1 The Werther Corpus

Since our system is developed to help in the process of compiling a corpus comprising adaptations of Goethe’s *The Sorrows of Young Werther* throughout different text types and centuries, we collected texts from this target domain. To be able to train a specialized system, we manually corrected a small corpus of relevant texts (cf. Table 4.2). We use the output of Abbyy Fine Reader 7 for several Werther adaptations (Table 4.1), all based on scans of books with German Gothic lettering.

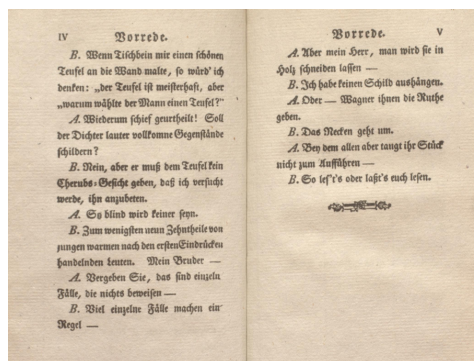
4.3.2 The Deutsches Textarchiv (DTA) Corpus

Even though manual OCR post-correction is a vital part of many projects, only very little detailed documentation of this process exists. *Das Deutsche Textarchiv* (The German Text Archive) (DTA) is one of the few projects providing detailed correction guidelines along with the scans and the text corrected within the project (Geyken et al., 2012). This allows the compilation of a comprehensive parallel corpus of OCR output and corrected text spanning a period of four centuries (17th to 20th) in German Gothic lettering. For OCR, we use the open source software *tesseract*⁴ (Smith and Inc, 2007) which comes with recognition models for Gothic font.

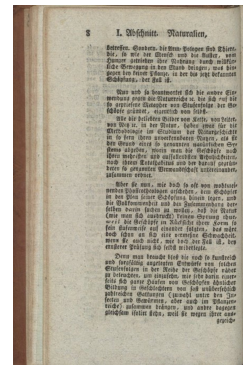
⁴Considering the open source aspect of our resulting system, we decided to use the open source OCR software *tesseract* and move away from Abbyy some time after our project started: <https://github.com/tesseract-ocr>.



(a) Werther der Jude (1910)



(b) Lorenz Konau (1776)



(c) DTA: Blumenbach (1791): Handbuch der Naturgeschichte

Figure 4.3: Scans of three different texts from our corpora. Emphasizes differences in quality of scan and differences in type setting, font and genre (e.g. drama).

4.3.3 Gutenberg Data for Language Modeling

Since the output of our system is supposed to consist of well-formed German sentences, we need a method to assess the quality of the output language. This task is generally tackled by language modeling. We compiled a collection of 500 randomly chosen texts from Project Gutenberg⁵ comprising 28,528,078 tokens. With its relative closeness to our target domain, it constitutes the best approximation of a target language. The language model is trained with the KenLM toolkit (Heafield, 2011) with an order of 5 on token level and an order of 10 on character level following De Clercq et al. (2013).

⁵Project Gutenberg. Retrieved January 21, 2017, from www.gutenberg.org.

4.4 Why OCR Post-Correction is Hard

In tasks like the normalization of historical text (Bollmann et al., 2012) or social media, one can take advantage of regularities in the deviations from the standard form that appear throughout an entire genre or in the case of social media e.g. dialect region (Eisenstein, 2013). Errors in OCR, however, depend on the font and quality of the scan as well as the time of origin, which makes each text unique in its composition of features and errors.

In order to exemplify this claim, we analyzed three different samples: *Lorenz Konau* (1776), *Werther der Jude* (1910) and a sample from the DTA data. Figure 4.3(a-c) illustrate the point at which the quality of scan is crucial for the OCR success. Figure 4.3(a) shows a text from the 20th century where the type setting is rather regular and the distances between the letters is uniform as opposed to Figure 4.3(b). Figure 4.3(c) shows how the writing from the back of the page shines through and makes the script less readable. Thus, we observe a divergence in the frequency of certain character operations between those texts: the percentage of substitutions ranges between 74% for *Lorenz Konau* and 60% for *Werther der Jude*, as well as 18% and 30% of insertions, respectively. The varying percentage of insertions might be due to the fact that some scans are more “washed out” than others. Successful insertion of missing characters, however, relies on the precondition that a system knows a lot of actual words and sentences in the respective language and cannot be resolved via e.g. character similarity like in the substitution from *l* to *t*.

Another factor that complicates the correction of a specific text is the number of errors per word. Words with an edit distance of one to the correct version are easier to correct than those with more than one necessary operation. With respect to errors per word our corpus shows significant differences in error distributions. Especially in our DTA corpus the number of words with two or more character-level errors per word is considerably higher than those with one error. For *Werther der Jude* (WER 10.0, CER 2.4) the number of errors in general is much lower than for *Konau* (WER: 34.7, CER: 10.9). These characteristics indicate that subcorpus-specific training of a system is promising.

4.5 Specialized Multi-Modular Post-Correction

In order to account for the nature of errors that can occur in OCR text, we apply a variety of modules for post-correction. Moreover, the modular implementation of the system ensures a certain flexibility with respect to the data and research objective.

The system proceeds in two stages. In the first stage, a set of specialized modules (Section 4.5.1) suggests corrected versions for the tokenized⁶ OCR text lines. Those modules can be context-independent (work on just one word at a time) or context-dependent (an entire text line

⁶Tokenizer of TreeTagger (Schmid, 1995).

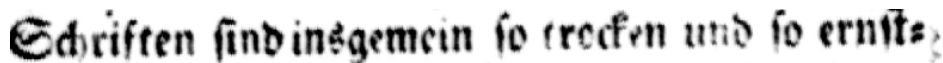


Figure 4.4: Irregular type setting in German Gothic lettering. *sind* and *insgemein* are two separate words but yet written closely together.

is processed at a time). The second stage is the decision phase. After the collection of various suggestions per input token, these have to be ranked to enable a decision for the most probable output token for that specific context. We achieve this by assigning weights to the different modules with the help of Minimal Error Rate Training (MERT) (Och and Ney, 2003).

4.5.1 Suggestion Modules

In the following, we give an outline of techniques included into our system.

Word Level Suggestion Modules

By combining token-based and context-based modules, we try to combine the best of different methods.

- **Original:** the majority of words do not contain any kind of error, thus we want to have the initial token available in our suggestion pool
- **Spell checker:** spelling correction suggestion for misspelled words with hunspell⁷
- **Compounder:** merges two tokens into one token if it is evaluated as an existing word by hunspell
- **Word splitter:** splits two tokens into two words using a compound-splitter module from the Moses toolkit (Koehn et al., 2007)
- **Text-Internal Vocabulary:** extracts highly frequent words from the input texts and suggests them as a correction of words with a small adjusted Levenshtein distance⁸

The compound and word split techniques react to the variance in manual typesetting, where the distances between letters vary. This means that the word boundary recognition becomes difficult (cf. Figure 4.4).

A problem related to the spell-checking approach is the limited coverage of the dictionary since it uses a modern German lexicon. Related to this is the difficulty of out-of-vocabulary words above average for literature text. Archaic words from e.g. the 17th century or named entities cannot be found in a dictionary and can therefore not be covered with any of the approaches mentioned above. However, especially named entities are crucial for the automatic or semi-automatic analysis of narratives e.g. with the help of network analysis. Our Text-Internal Vocabulary technique is designed to find frequent words in the input text, following the assumption that errors would

⁷<https://github.com/hunspell/hunspell>.

⁸OCR-adjusted Levenshtein distance taking frequent substitution, insertion and deletion patterns learned from training data into account.

not be regular enough to distort those frequencies. We compile a list from those high-frequency words. Subsequently, erroneous words can be corrected by calculating an OCR-adjusted Levenshtein distance. In this way misspelled words like *Loveuzo* could be resolved to *Lorenzo* if this name appears frequently. Since the ranking algorithm relies on a language model which will most probably not contain those suggestions, we insert the high-frequency words into the language modeling step.

Sentence Level Suggestion Modules

As has been suggested by Afli et al. (2016), we include Phrase-based **Statistical Machine Translation** (SMT) into our system. We treat the post-correction as a translation problem translating from erroneous to correct text. Like in standard SMT, we train our models on a parallel corpus, the source language being the OCRed text and the target language being manually corrected text. We train models on token level as well as on character-level (unigram). This way, we aim at correcting frequently mis-recognized words along with frequent character-level errors. We train four different systems:

- token level
 - domain-specific data (cf. Section 4.3.1)
 - general data (cf. Section 4.3.2)
- character level
 - domain-specific data (cf. Section 4.3.1)
 - general data (cf. Section 4.3.2)

The models are trained with the Moses toolkit (Koehn et al., 2007). Moreover, we use a subsequent approach by forwarding the output of the character-based SMT model to the token-based SMT.

Additional Feature

The information whether a word contains an error can help to avoid the incorrect alternation of an initially correct word (overcorrection). In order to deliver this information to the decision module without making a hard choice for each word, we include the information whether a word has been found either in combination with the word before or after in a corpus (cf. Section 4.3.3) into the decision process in the form of a feature that will be weighted along with the other modules. This naive language modeling approach allows for a context-relevant decision as to the correctness of a word.

set	# tokens (OCR)	# tokens (corr)	WER	CER
train	70,159	68,608	15.7	5.5
train _{ext}	133,457	131,901	12.9	4.0
dev _{SMT}	12,464	12,304	13.9	3.5
dev _{overall}	13,663	13,396	16.75	4.6
test _{init}	17,443	17,367	9.4	2.5
test _{unk}	13,286	13,304	31.2	9.2

Table 4.2: Werther-specific parallel corpus of OCR text and corrected text showing the number of tokens before and after post-correction along with WER and CER

4.5.2 Decision Modules: the Ranking Mechanism

Since the recognition errors appearing in a text are hard to pre-classify by nature, we run all modules on each sentence of the input, returning suggestions for each word. Since the output of some of our modules is entire sentences, input sentence and output sentence have to be word-aligned in order to be able to make suggestions on word level. The word alignment between input and output sentence is done with the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970), an algorithm originally developed in the context of bioinformatics.

It is the task of the decision module to choose the most probable combination of suggestions to build a well-formed sentence. To solve the combinatorial problem of deciding which suggestion is the most probable candidate for a word, the decision module makes use of the Moses decoder. As in general SMT, the decoder makes use of a language model (cf. Section 4.3.3) and a phrase table. The phrase table is compiled from all input words along with all possible correction suggestions. In order to assign weights to the single modules and the language model, we tune on the phrase tables collected from a run on our dev_{overall} set, following the assumption that suggestions of certain modules are more reliable than others and expect their feature weights to be higher after tuning.

4.6 Experiments

4.6.1 Experimental Setup

To guarantee diversity, we split each of the texts 1-4 (cf. Table 4.1) into three parts and combine the respective parts: 80% train (train), 10% development (dev_{SMT}) and 10% test (test_{init}).

Test setup We introduce two different test scenarios. Even though both test sets are naturally compiled from unseen data, the first test set consists of a self-contained Werther adaptation introducing new named entities, originating from a different source and thus showing a different

set	# tokens (OCR)	# tokens (corr)	WER	CER
train	3,452,922	3,718,712	41.6	13.2
dev	663,376	836,974	30.4	9.1

Table 4.3: DTA parallel corpus of OCR text and corrected text showing the number of tokens before and after post-correction along with WER and CER

training set	system	test _{init}		test _{unk}	
		WER	CER	WER	CER
	original text	23.5	15.1	36.7	30.0
train	baseline	22.0	13.2	26.6	26.3
	overall system	4.7	8.0	15.4	19.6
train _{ext}	baseline	21.1	11.7	24.0	20.4
	overall system	4.4	7.2	15.2	16.4

Table 4.4: WER and CER for both test sets before and after automatic post-correction for the system trained with the small training set (train) and the larger training set (train_{ext}). Baselines: the original text coming from the OCR system and the character-level SMT system trained on the Werther data.

error constitution. This constitutes an evaluation in which no initial manual correction as support for the automatic correction is included in the workflow. We henceforth call this unknown set *test_{unk}* (text 6).

In contrast, the second set contains parts of the same texts as the training, thus specific vocabulary might have been introduced already. The results for this test set give a first indication of the extent to which pre-informing the system with manually corrected parts of a text could assist the automatic correction process. Since this scenario can be described as a text-specific initiated post-correction, we henceforth refer to this test set as *test_{init}*.

We further on experiment with an extended training set train_{ext} (train with texts 7 and 8) to assess the influence of the size of the specific training set on the overall performance. The sizes of the data sets before and after correction along with WER and CER are summarized in Table 4.2. The sizes for the general data set before and after correction along with WER and CER are summarized in Table 4.3.

4.6.2 Evaluation

In the following we concentrate on the comparison of WER and CER before and after automatic post-correction. As a baseline for our system we chose the strongest single-handed module (SMT on character-level trained on Werther data).

Overall performance As indicated previously, our test sets differ with respect to their similarity to the training set. The results for both test scenarios for systems trained on our two training sets are summarized in Table 4.4. The results from *test_{init}* and *test_{unk}* show that our

module	# overcorr.	test _{init}		# overcorr.	test _{unk}	
		# corr.	# unique corr.		# corr.	# unique corr.
SMT Werther token	128	364	10	209	1,089	0
SMT Werther character	235	684	0	700	1,919	0
SMT Werther cascaded	273	697	2	728	1,933	4
SMT DTA token	2,179	229	8	1,627	893	19
SMT DTA character	4121	372	22	3,143	1,530	115
text-internal vocab	3,317	131	16	4,142	244	60
word split	594	3	0	720	45	2
spell check	1,329	219	15	2,819	731	40
compound	222	0	0	169	2	2
overall system	238	2171	-	675	2,642	-

Table 4.5: Number of overcorrected, corrected and uniquely corrected words per module out of 17,367 tokens in test_{init} (2,726 erroneous words) and 13,304 tokens in test_{unk} (4,141 erroneous words)

system performs considerably better than the baseline and can improve quality of the OCR output considerably.

For test_{unk}, the system improves the quality by almost 20 points of WER from 36.7 to 15.4 and over 10 points in CER from 30.0 to 19.6. For test_{init}, our system improves the quality of the text with a reduction of approximately 20 points of WER from 23.5 to 4.7 and 7 points in CER from 15.1 to 8.0. It is not surprising that the decrease in WER is stronger than the decrease in CER. This is due to the fact that many words contain more than one error and require more than one character level operation to get from the incorrect string to the correct string.

Only slight improvement can be shown by adding training material to the Werther-specific parts of the system (cf. train_{ext} row of Table 4.4). Merely the CER can be improved whereas the WER stays about the same. The improvement in test_{unk} is higher than for test_{init}.

Module-specific analysis Since a WER and CER evaluation is not expedient for all modules as they were designed to correct specific problems and not the entirety of them, we look into the specialized modules in terms of correct suggestions contributed to the suggestion pool and correct suggestions only suggested by one module (unique suggestions). As the system including the extended training set train_{ext} delivered slightly better results, in the following we will describe the contribution of the single modules to the overall performance of this system (cf. Table 4.5). For test_{unk} the number of corrected tokens along with the number of overcorrections is higher than for test_{init} throughout all modules. Clearly, for test_{init} the Werther-specific modules are strongest. The more general modules prove useful for test_{unk}. The number of corrected words increases for the SMT module trained on DTA data on character-level. The usefulness of the module extracting specific vocabulary (text-internal vocab) as well as the general SMT model and the spell checker becomes evident in terms of unique suggestions contributed by those modules.

The analysis of the output of the individual modules and their contribution to the overall sys-

tem uncovers an issue: those modules that produce a high number of incorrect suggestions, thus overcorrecting actually correct input tokens, are at the same time those modules that are the only ones producing correct suggestions for some of the incorrect input words. Consequently, those uniquely suggested corrections are not chosen in the decision modules due to an overall weak performance of this module. These suggestions are often crucial to the texts like the suggestions by the special vocabulary module which contain named entities or words specific to the time period. For our *test_{unk}* set, the text-internal vocabulary module yields around 60 unique suggestions, out of which 15 are names (Friedrich, Amalia) or words really specific to the text (*Auftrit* spelled with one t instead of two).

Challenges In the context of literature OCR post-correction is a challenging problem since the texts themselves can be considered *non-standard text*. The aim is not to bring the text at hand to an agreed upon standard form but to digitize exactly what was contained in the print version. This can be far from the standard form of a language. In one of our texts, we find a character speaking German with a strong dialect. Her speech contains a lot of words that are incorrect in standard German, however, the goal is to preserve these “errors” in the digital version. Thus, correction merely on the basis of the OCR text without consulting the printed version or an image-digitized facsimile can essentially never be perfect. It follows that the integration of automatic post-correction techniques into the character recognition process could lead to further improvements.

4.7 Adaptability

Reusability as a key concept in NLP for DH originates in the time limitations given in such projects. Since DH projects do not evolve around the development of tools but the analysis performed with the help of these tools in order to answer a specific question, the tools are expected to be delivered in an early phase of collaborative projects. From-scratch development easily exceeds these time limits. Therefore, tools need to be built in such a way that they can be adjusted to other sorts of texts, languages or even purposes (cf. Section 5) with minimal effort. In order to prove that our OCR post-correction system is modular enough to be adjusted to correct texts from other languages, we train two other versions of the system. We train systems for English and French with data released in the OCR post-correction competition organized in the context of the 14th International Conference of Document Analysis and Recognition (ICDAR 2017) (Chiron et al., 2017)⁹. The data is a subpart of the corpus collected in the context of the AmeliOCR project, led by the L3i laboratory (University of La Rochelle, France)¹⁰ and the Bibliothèque

⁹<https://sites.google.com/view/icdar2017-postcorrectionocr/home>, 3/07/2017.

¹⁰<http://www.bnf.fr/fr/acc/x.accueil.html>, 19/10/2017.

nationale de France (BnF)¹¹. For both languages, there are texts published in monographs and periodicals available. They originate from the last four centuries. The documents come from different collections (e.g. BnF, British Library) supported by various projects (e.g. Europeana Newspapers, IMPACT, Gutenberg, Perseus, Wikisource and Bank of wisdom) and therefore have been digitized using different OCR systems. The data is summarized in Table 4.6.

language	<i>train_{ocr}</i>	<i>train_{gold}</i>	<i>dev1_{ocr}</i>	<i>dev1_{gold}</i>	<i>dev2_{ocr}</i>	<i>dev2_{gold}</i>	<i>test_{ocr}</i>	<i>test_{gold}</i>
English	309,080	282,738	71,049	65,480	13,000	11,966	14,302	12,859
French	805,438	783,371	167,473	163,373	9,566	9,216	12,289	11,780

Table 4.6: Number of tokens in the English and French corpus provided by the competition on OCR-postcorrection.

The sizes of the training corpora are clearly much bigger than the size of our Werther-specific corpus but considerably smaller than the DTA corpus. We use two development sets, one for tuning the SMT models and the other for the tuning of the weights assigned to each of the modules in our overall system. We observe a much higher decrease in number of words from the OCRed text to the manually corrected text for English than for French. This might be due to the differences in quality of source material. Since the originals were not made available, we cannot verify this guess. Inspecting the data, however, shows examples where certain text areas seem to not be recognizable by the OCR system, which supports this claim (cf. Table 4.7)

OCRed	manually corrected
Certainly much superior thumb ot Liquid Blu ' I- 1 ' - o ' ' I ■ ● .ri ov T -The number of pers- ons killed in the rectnt ea-thquake shocks in Southern Italy is officially stated to bo eighty- six .	Certainly much superior thumb or Liquid of persons killed in the recent earthquake shocks in Southern Italy is officially stated to be eighty-six .

Table 4.7: Example of badly recognized text in the English part of the corpus.

We adjust our system to the language by retraining the SMT models and including spell-checkers for the respective languages. Due to the modular architecture these adjustments can be made easily and with a low expenditure of time. Since the data sets are a compilation of a variety of texts, we use all modules but the domain-specific SMT models. We solely include two SMT models per language, one on token-level and the other on character-level.

The test set does not comply with the official shared task set since the manually corrected data is not yet available for the test set. We test on a combination of periodicals and monographs.

The strongest unique modules for these two languages is the subsequent combination of the character-level SMT and the token-level SMT models (Cascaded). For English it performs just slightly worse on WER and even outperforms the overall system on the CER. For French, the

¹¹<http://www.bnf.fr/fr/acc/x.accueil.html>, 19/10/2017.

language	system	WER	CER
English	original	29.4	28.4
	SMT Cascaded	22.7	23.6
	overall system	22.1	24.5
French	original text	13.3	25.0
	SMT Cascaded	9.9	20.0
	overall system	8.7	21.5

Table 4.8: The results reported in word error rate (WER) and character error rate (CER) for the English and French test set.

overall system is clearly stronger than the Cascaded SMT system with more than 1 percent improvement of WER but also performs worse in terms of CER by 1.5 percent. Generally, the OCR post-correction system achieves about 25% reduction of WER for English and over 30% reduction in WER for French. The English data set generally poses a bigger problem for post-correction as also illustrated by the example in Table 4.7.

4.8 Digitization Workflow

We consider it an integral part of our research to make the resulting system available to the humanities. This goal requires that we develop a workflow that allows the humanities scholar to submit scans or already OCRed text files without any knowledge about the actual system architecture. The solution we implemented accepts various file formats such as pdf, jpeg and png and returns corrected texts. Since OCR can be a process that is very time-consuming, especially if done via a graphical user interface where only one scan at a time can be processed, we decided to facilitate the task by adding OCR to our pipeline. This results in a workflow with three steps. Firstly, we apply an OCR system. Subsequently, our post-correction system processes the output of the OCR system further. As an important final component of this workflow, we have to ensure that the output files which potentially contain remaining OCR errors are compatible with a system that allows for manual post-correction. This abstract workflow is illustrated in Figure 4.5.

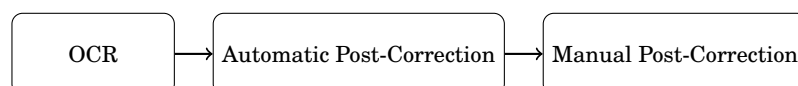


Figure 4.5: Abstract workflow for the digitization from the scan to the digitized text.

The implementation of an easy-to-handle workflow is an often underemphasized aspect of DH. It needs to be intuitive enough to not absorb the time that has been saved via automation. We have implemented a pipeline combining the automatic OCR recognition followed by automatic post-correction. For OCR, we utilize an open-source software called *tesseract* which

OCR und OCR Nachkorrektur

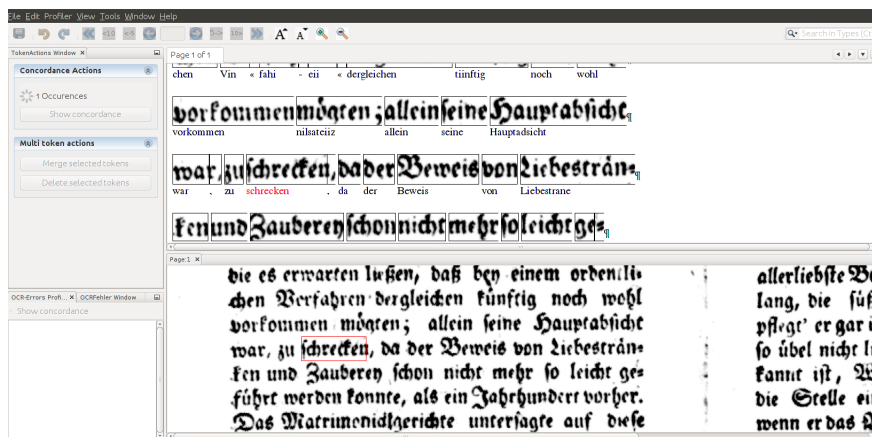
Email:

What's your email address?

Wählen Sie eine Textdatei (*.txt, *.pdf, *.tif, *.jpg, *.png usw.) von Ihrem Rechner aus.

Browse... No file selected. ... und ab geht die Post!

(a) Web interface for the submission of scans.



(b) PoCoTo showing the text from the scan aligned with the post-correction output.

Figure 4.6: Screenshots of two steps of the workflow implementation of OCR post-correction.

we have directly integrated with our automatic post-correction system. This means that the first and second step of the abstract workflow are combined into one step for our users. In our project, this has been realized as an easy web-based service (cf. Figure 4.6(a)) that allows the user to upload scans or images¹² online. The post-correction system returns an hocr file which is an OCR-specific XML-format. This format is readable by PoCoTo (Vobl et al., 2014), a tool for supporting manual post-correction of OCR'd text through alignment of image and digitized text (cf. Figure 4.6(b)). This system provides visual aid and batch correction support as an answer to the fact that automatic post-correction can hardly ever resolve all OCR errors. The tool has been developed at the University of Munich. By integrating already existing tools, we illustrate how the combination of tools specifically developed for our data set and already existing tools can build a complete pipeline in a time-saving manner. Moreover, we added our tool to the Clarin-D repository to ensure sustainability¹³.

¹²<http://clarin05.ims.uni-stuttgart.de/ocr/>, for access please contact the author.

¹³Clarin-D repository, metadata handle: <http://hdl.handle.net/11022/1007-0000-0007-C61A-D>.

4.9 Research Contributions

In this chapter, we focus on all three crucial aspects of the NLP for DH methodology suggested in Chapter 2: **specific problem solving, reusability of systems and application-oriented research**.

The Werther corpus is characterized by a specific vocabulary. The post-correction problem itself highly depends on the original sources of the text. Due to the complexity of OCR post-correction, there cannot be a general solution. Due to these facts, a system that is tuned to these texts is necessary for the highest possible success rate. We tackle this by including specialized modules into general architecture. This approach to **specific problem solving** exemplifies a difference in strategies between general and DH-focused NLP.

At the same time, **reusability** plays a crucial role in DH since there is rarely time to develop an entire processing pipeline from scratch for a specific type of text. We show that specific problem solving and reusability do not exclude each other. Reusability can be facilitated by highly modularized architectures and workflows. We present an example for a generic architecture which can easily be adjusted to specific texts. Thus, modularity and adaptability are key features that make systems valuable in such contexts. We can show that the enhancement of a general architecture by including small but specific data sets can improve results within a specific domain. Moreover, this combination of different techniques for OCR post-correction is significantly superior to single techniques. Especially the integration of SMT models on token level and character level contributes to the overall success of the system. Even though the ranking algorithm achieves large improvements, further potential lies in the inclusion of fine-tuned language models since the decision process highly depends upon it. The intrinsic characteristic of literature as being *non-standard* complicates the task. However, techniques that focus on these features such as our module that is specialized on extracting text-specific vocabulary show promising results for e.g. named entity correction.

In addition, we ensure the usefulness of the system by implementing an **easy-to-use digitization workflow**. By providing a web-based data submission interface, we guarantee that users using different operating systems can use our system. Moreover, we integrate a tool with visual post-correction support developed in the context of OCR post-correction which takes over where our system reaches its scope.

TEXT NORMALIZATION

Thus far, we have focused on text coming from traditional humanities disciplines such as literary studies. However, also texts from new media have come to the fore in recent years. With the advent of Web2.0, user participation on the Internet has become common practice. According to Murugesan (2007), Web2.0 is a conglomerate of technologies and strategies aimed at online user participation: it is highly dynamic and characterized by a productive user community. The online content these users produce is called user-generated content (UGC). Van Dijk (2009) discusses this new concept of *user* and their role and participation mechanisms in the virtual world. Phenomena known from face-to-face interaction are taken over into the virtual space and adjusted to it. This makes UGC an interesting research subject for the field of social science. Riegner (2007), for example, describes how the concept of word-of-mouth is adopted in cyberspace. From a commercial perspective, UGC has attracted the interest of research in a variety of text mining applications (Cortizo et al., 2012), including sentiment and opinion mining (Paltoglou and Thelwall, 2012), which is used in e.g. user-tailored advertising (Aven et al., 2009). Besides an interest for commercial applications, this kind of text holds potential for e.g. sociological applications. Similar methods can be used to automatically trace harmful content on social media (Peersman et al., 2011; Desmet and Hoste, 2014; Van Hee et al., 2015). This is especially important for the protection of teenagers and fits an urgent need. Royen et al. (2015) describe the harmfulness of cyberbullying on social network sites and state the need for prevention methods. It goes without saying that solving these tasks requires a deep linguistic processing of the text at hand.

The automatic analysis of UGC poses a problem for NLP, as discussed in Chapter 3, since the kind of language used in social media highly differs from standard language. Eisenstein (2013) even goes as far as to call it *bad language*. The noisy nature of UGC complicates the task of

automatically processing this valuable data source, because the performance of standard NLP tools significantly decreases on social media data (Melero et al., 2012; Eisenstein, 2013). This is because these tools have originally been developed for standard language and, as a consequence, cannot deal with many of the peculiarities encountered in UGC.

Two different computational approaches have been suggested to tackle this problem (Han et al., 2013; Plank, 2016): tool adaptation and text normalization. Tool adaptation aims at including UGC data into the training process. As such, tools are made robust with respect to the text type at hand. Work in this field has been performed by, amongst others, Ritter et al. (2011) for named-entity recognition (NER), Gimpel et al. (2011) for part-of-speech (POS) tagging and Foster et al. (2011) for parsing. A disadvantage of this approach is that it is non-transferable, which means that every single tool would have to be adapted individually. We will discuss this approach in Chapter 6. The other approach is text normalization, which envisages to first bring non-standard language closer to the “norm”, i.e. better conforming to the standard rules of spelling, grammar and punctuation of a particular language. In this way, standard NLP tools can be applied in a next step.

In this chapter, we follow the latter approach. We assess the **significance of text normalization for non-standard text processing**. Moreover, we explore the ways in which the concept of **reusability** can be extended to **adaptability across tasks**. To show that the system architecture introduced in Chapter 4 can not only be adapted to other languages but also another task, we adjust our multi-modular system to the task of text normalization.

The data used in our experiments has been collected in the context of a project with the goal to automatically monitor cyberspace applications. The ultimate goal is to prevent life and privacy threatening situations caused by harmful content online. The automatic tracing of such harmful content could help informed decisions by policy makers and law enforcement, online restorative and assistance services, moderators of social network sites, parents and – most importantly – by the young users themselves¹.

- (9) Pleeg gew zelfmoord, iedereen haat u.
Pleeg gewoon zelfmoord, iedereen haat je.
Commit just suicide, everyone hates you.

‘Just kill yourself, everyone hates you’

An example for such content is given in (9)². This example of so-called cyberbullying contains a direct prompt to commit suicide. Organizations such as suicide prevention centers monitor social media in order to find such utterances which gives them the possibility to intervene early enough. However, this post contains normalization issues such as the abbreviation of *gew* for *gewoon* (Engl. just) or the dialectal use of the formal pronoun *u* instead of *je* (Engl. you/yourself).

¹For more information on the project visit <http://www.amicaproject.be/>, 15/01/2018.

²Example given in Van Hee et al. (2015).

In order to reliably perform complex NLP tasks, such as the prediction of cyberbullying on social media posts, the data has to be normalized.

Several task-specific modules solve the different normalization problems that can be encountered in UGC (see below) similar to the OCR post-correction problem. More general modules are implemented to tackle all normalization issues in one step. To assess the suitability of the different modules, we evaluated the performance of each module separately. It has been shown in Chapter 4 that a multi-modular system covering a variety of approaches can outperform simple individual approaches. We thus combine the output of the different modules in several ways and analyze the overall performance of our system. We furthermore reveal the *impact of text normalization on different NLP tools* by comparing the output of NER, POS tagging and lemmatization on UGC before and after normalizing.

The remainder of this chapter is structured as follows. Section 5.1 discusses the characteristics of UGC, while Section 5.2 gives an overview of related work on text normalization. Section 5.3 presents how we use our modular approach for text normalization. In Section 5.4, we introduce the data sets that were used for the experiments, analyze the experimental results and illustrate the usefulness of text normalization on three NLP tasks.



Publication

Parts of this chapter were published in Schulz et al. (2016).

5.1 User-Generated Content - A Challenge for NLP

What started as online chatting on a PC and text messaging on cell phones has now evolved into a continuous stream of content that is being produced online using a variety of devices. This evolution has led to the creation of a language that often strongly deviates from standard language, characterized by abbreviations, omissions, spelling mistakes and grammatically incorrect utterances.

Eisenstein (2013) relates these phenomena to text input affordance, which might vary depending on the input method used (e.g. mobile phone keyboard vs. touch screen keyboard vs. computer keyboard). He also notes that the type of social media application (e.g. online chat, Internet forum, or social network status updates) influences the language used. In addition, social variables, such as age (Rosenthal and McKeown, 2011), ethnicity (Eisenstein et al., 2011) and location (Wing and Baldrige, 2011; Eisenstein et al., 2010) can influence wording and writing style. VandeKerckhove and Nobels (2010), for example, observe regional variation in UGC and discuss the example of different graphemic realizations of words ending in *-en* in Flemish (*-en*, *-n* or *-e*), and attribute these to different phonetic realizations depending on the regional dialect. They conclude that the large variety of dialects in Flanders leads to a strong variation in the graphemic realization of words in UGC.

VandeKerckhove and Nobels (2010) relate the language phenomena in UGC to two writing principles: *write as you speak* and *write as fast as possible*. Along the same line, De Clercq et al. (2013) divide the language deviations of UGC into three linguistically motivated categories, namely abbreviations, orthographic and phonetic variants. Very typical of UGC is the large number of *abbreviations*, which can be explained by different factors: space limitations (e.g. in Twitter posts or SMS) and time limitations. As VandeKerckhove and Nobels (2010) point out, the Internet is a medium in which communication is fast. Nevertheless, most abbreviations are easy to understand as they occur either frequently or are straightforward in a specific context. Social media users most commonly abbreviate *facebook* as *fb*, react to funny content with *lol* (laughing out loud) or talk about their *bf* (boyfriend) or *gf* (girlfriend). Quickly produced text also leads to *typos* and other orthographic issues. Uppercasing is often ignored, or unconventionally used to emphasize something or to convey a specific emotion. Letter transpositions can be observed due to fast typing and a lack of correction. Again, the frequency of these error types varies strongly depending on the social media application used.

The tendency to “write as you speak” can be observed across languages. It seems as if users mimic direct social interaction online by using phonetically motivated realizations of words. In English, this is largely realized by using homophonous graphemic variants of a word such as *r* for *are* or *dey* for *they*. In Dutch, words are often transformed or even fused on the basis of the regional pronunciation of the user. This leads to variants such as *zoiso* instead of *sowieso* (Engl. in any case) or *kheb* instead of *ik heb* (Engl. I have). Very typical of UGC is also that emotions are often orthographically expressed. This can be done in the form of flooding (i.e. the repetition of characters), capitalization and the productive use of emoticons.

Each of these characteristics contributes to the challenge of linguistically processing this type of text using standard NLP tools. In the next chapter, we give an overview of some research efforts that have attempted to automatically *normalize* such non-standard data.

5.2 Text Normalization - Related Work

Originally, *text normalization* referred to a preprocessing step for text-to-speech synthesis. It dealt with domain-specific problems that were often solved using hand-crafted rules. As such, the expected input was limited to a few patterns known a priori (Taylor et al., 1998) and the normalization problems were often restricted to words without context. In this form, the normalization problems to be addressed were often restricted to words without context and could therefore easily be solved at the token level using rules. Sproat et al. (2001) were the first to extend this technique by treating normalization as a language modeling problem and to propose a taxonomy of normalization types. This was done based on four rather distinct text types, newspaper articles, real estate ads, daily digests from a mailing list and recipes. Their work marked the beginning of more intricate text normalization research.

In more recent years, text normalization has been studied in the framework of UGC. As previously explained, this genre can be characterized by many issues which are not limited to the word level and very often context is needed to normalize correctly. Moreover, UGC is an umbrella term, which covers different text types such as SMS, tweets, chat logs and the like. As a consequence, the frequency and density of normalization problems also varies strongly depending on the social media application used. Han and Baldwin (2011) and Baldwin et al. (2013), for example, observe that English Twitter is more dissimilar compared to other forms of social media such as blogs and comments.

Previous research on UGC normalization has been performed on diverse languages using different techniques. Kobus et al. (2008a) introduced three metaphors to refer to these normalization approaches: the spell checking, translation and automatic speech recognition metaphor.

The **spell checking** metaphor leaves correct words untouched and only performs normalization on the incorrect words. Choudhury et al. (2007) use a Hidden Markov Model trained on SMS data to find the most probable mapping from an erroneous word to its standard equivalent, thus treating UGC as a noisy version of standard language. Closely related is the use of a dictionary containing both standard and OOV entries for the purpose of normalization. In this respect, Gouws et al. (2011) suggest a method for the extraction of frequent domain-specific lexical variants, which can serve as a basis for rule-based normalization systems. Such a system is described in Clark and Araki (2011). They normalize English tweets as a preprocessing step for machine translation from English to Japanese, based on a database of frequent erroneous words in Twitter posts and pattern matching rules. Since the coverage of these dictionaries often poses a problem, Han et al. (2012) introduces a method to automatically compile a large dictionary.

The **translation** metaphor treats social media language as the source language and standard language as the target language. As in general Statistical Machine Translation (SMT), a translation model is trained on parallel data. This model is then combined with a language model to transform a noisy input string into a string that is closer to the standard. The advantage of using SMT is that it directly makes use of contextual information during translation. This approach is described in Aw et al. (2006) who use phrase-based machine translation to normalize English SMS data and by Kaufmann and Kalita (2010) to normalize English tweets.

Pennell and Liu (2011) were the first to also perform machine translation at the character level, as well as Tiedemann (2012), who uses this technique to translate between closely related languages. Applied to abbreviation normalization, they find that character-based machine translation is more robust to new abbreviations. Li and Liu (2012) likewise describe a character-level machine translation approach to normalizing tweets and extend it to also work with character-blocks in order to improve on the automatic alignments. They also suggest a two-step MT approach that converts tokens into phonemes and phonemes into dictionary words, thereby incorporating the sensibility that people *write as they speak*. De Clercq et al. (2013) show an improvement using character-based models over token-based models for text normalization

when applying this technique to the entire range of normalization problems and not only to abbreviations. At the same time, Ling et al. (2013) introduce an approach based on paraphrasing by also building two translation models, one on the token-level and one on the character-level. Combining these two in a subsequent decoding step proved beneficial for normalizing English tweets.

Text found in social media also shares features with spoken language and the metaphor of **automatic speech recognition** utilizes this similarity. Here, text encountered in social media is treated as an alphabetic approximation of a phonetic string and is brought to a standardized written form using techniques from automatic speech recognition (ASR). Kobus et al. (2008b) propose an ASR-like system for text normalization based on this idea and, like Li and Liu (2012), combine it with SMT-like approaches to normalize French SMS messages. This metaphor has mostly been merged with other techniques to boost performance. Xue et al. (2011) show that combining phonetic with orthographic and contextual information together with acronym expansion works well for microtext normalization when combined in a multi-channel model. For their automatic dictionary construction, Baldwin et al. (2013) similarly rely on the morphophonemic similarity between standard tokens and ill-formed tokens, which leads them to use both edit distance and phonemic transcription to create word candidates, which are subsequently ranked by a trigram language model.

Some approaches fall beyond the scope of these metaphors, such as the character level sequence labeling technique described in Li and Liu (2012) and Li and Liu (2014), which uses a variety of phonetic, syllabic and orthographic features to construct likely abbreviations for words in a dictionary. This information is then used during testing as a reverse-lookup table to suggest expansions of observed OOV words. A similar approach is suggested in Liu et al. (2012) that learns character transformations on the basis of token-word pairs that were collected in an unsupervised fashion. Liu et al. (2012) also suggest a cognitive-sensitive visual priming technique that favors candidate words that are frequently used and bear an orthographic similarity to the token.

A log-linear model is proposed by Yang and Eisenstein (2013) that scores the conditional probability of a source and target sequence by means of language modeling of the latter and log-likelihood maximization of the former. They report state-of-the-art F-scores that improve on previous research efforts on the same data set (Han and Baldwin, 2011; Liu et al., 2012). Another log-linear approach, albeit over a series of replacement generators on the character level, is presented in Zhang et al. (2013), who evaluate the technique extrinsically, by comparing the performance of a dependency-parser on non-normalized, gold-standard and automatically normalized data.

With such a wide variety of techniques at our disposal, **system combination** seems very promising for text normalization. Yvon (2010) describes a normalization device based on finite state transducers using a phonetic representation as an intermediate step. He concludes that

the two systems perform better on different aspects of the task and that combining these two modules works best. A similar method is presented in Beaufort et al. (2010), who combine both spell checking and machine translation approaches on French data, which leads to good results. They conclude, however, that including phonetic information into the system is crucial.

Li and Liu (2012) demonstrate state-of-the-art performance using a rule-based combination of a variety of techniques. In later work by Li and Liu (2014) the rule-based approach is abandoned for a discriminative reranking technique that operates on the word level as well as on the sentence level. Similar to Liu et al. (2012), they also report good results when performing sentence level Viterbi decoding, through the incorporation of a language model. Finally, Wang and Ng (2013) report good results using a novel beam-search decoder that iteratively produces normalized sentence candidates according to several hypothesis producers and consequently evaluates these sentences on the basis of language model scores and a set of count feature functions.

For our approach, we assume that in order to find a way to automatically normalize highly diverse texts containing a wide variety of normalization issues, a multi-modular system is needed. Moreover, we utilize different techniques to interpret the metaphors (e.g. we include three techniques focusing on different spelling errors and implement different MT approaches both on the token level and character level). As such, we end up with a multi-modular system that should be able to tackle the full normalization task. Different to the approaches described above, we do not just combine two of the metaphors, but apply all three of them. Also in contrast to research efforts such as Yang and Eisenstein (2013) or Li and Liu (2014), we do not consider the non-standard tokens to be known in advance and consider their identification an integral and non-trivial part of the normalization task. We are the first to apply such an exhaustive approach on diverse genres of Dutch UGC.³

5.3 A Multi-Modular Approach Towards Normalization

Our multi-modular UGC normalization system relies on on the same architecture as the OCR post-correction system introduced in Chapter 4. Preprocessing and included modules vary from the OCR post-correction system due to the different nature of data encountered in UGC processing. The system consists of three main layers:

1. A preprocessing layer, in which the input text is split into tokens and flooding (word lengthening) is corrected.
2. A suggestion layer, in which each module generates suggestions; either for tokens (i.e. the token-based modules) or for a message as a whole (i.e. the context-based modules). Most

³Since our system works on Dutch text, we will illustrate various parts using Dutch examples, with an English translation.

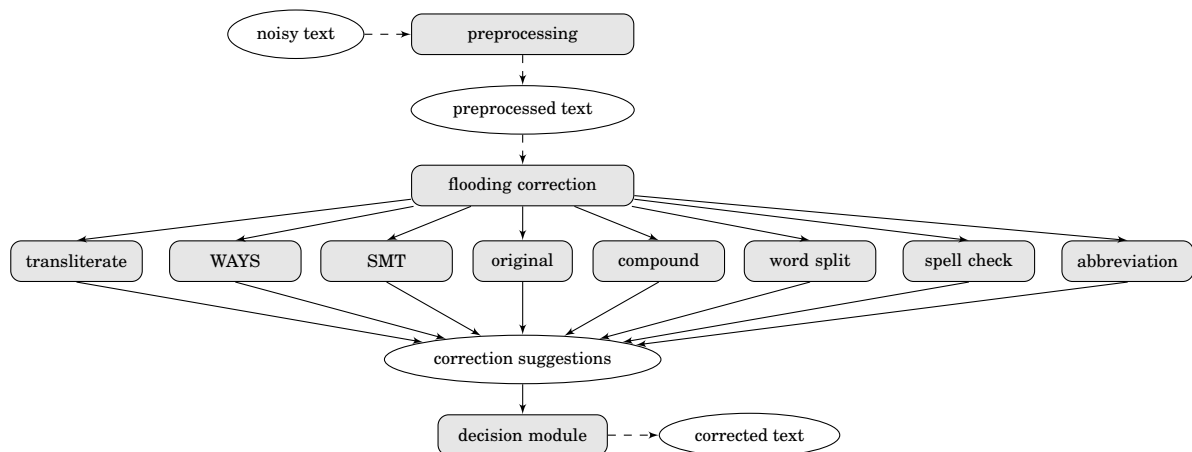


Figure 5.1: Multi-layer architecture of the UGC normalization system with the preprocessing layer on top, the context-based modules on the left-hand side, the token-based modules on the right-hand side and the decision module on the bottom.

of the token-based modules focus on well-understood normalization issues (such as abbreviations, compounds, split words). Context-based modules can operate on the word level, as well as the character level and differ from token-based modules, in that they can look beyond word boundaries to make normalization decisions.

3. A decision layer, in which the best combination of suggestions is chosen from the pool of suggestions.

The architecture of the multi-modular UGC normalization system is depicted in Figure 5.1.

5.3.1 Preprocessing Layer

Contrary to the OCR post-processing system where we simply take the tokenized lines recognized in the recognition step as an input, for UGC we add a specific preprocessing layer. This preprocessing layer consists of two modules. A first module splits the text into tokens, a task for which we adapted the rule-based tokenizer of Treetagger (Schmid, 1994) to cope with UGC-specific phenomena such as e-mail addresses, hyperlinks and emoticons. Whereas previous work focused on the tokenization of Twitter posts (O’Connor et al., 2010; Bontcheva et al., 2013), we investigate different genres of UGC, requiring us to build a more general tokenizer, covering a wider range of smileys, emoticons and other tokenization issues. The necessity to adjust the tokenization to the specificities of UGC, exemplifies that assumptions learned from the investigation of newspaper text with respect to words and punctuation are not universal to all kinds of text. Moreover, the definition of a sentence needs to be rethought:

- (10) Ik heb geprobeerd bellen . . . maar jij nam niet eens op :(
I have tried to call . . . but you picked not once up :(
'I have trid to call . . . but you did not even pick up :('

In (10), the subordinate clause 'maar jij nam niet eens op :(' is separated from the main clause using ' . . . ' indicating a break in the train of thought of the writer. Nevertheless, one would consider these two subclauses to make up one sentence. A standard sentence splitter, however, would split the sentence at the three dots. In addition, the emoticon in the end of the sentence apparently ends the sentence of the writer but since emoticons can appear at any place in a message, this is not a clear indicator. We therefore decide to work on the level of messages rather than sentences.

A second phenomenon dealt with in the preprocessing layer is character flooding, i.e. the repetition of the same character or character sequences, which is often used in UGC to express emotion, as illustrated in the example below. To reduce the number of out-of-vocabulary words in the subsequent modules, we limit the number of repetitions to a maximum of two for all characters except for the vowel "e", where a maximum of three is allowed. The flooding correction module makes use of the Hunspell spell checker⁴ to generate the most probable correction and to ensure that correct words are not overcorrected. The module corrects repeated characters and character combinations in the following way:

- (11) jij hebt egggggt zoooooooooooooooooooo onwijse mooie lipjes ...
jij hebt egg t zoo onwijse mooie lipjes ...
'you have really such incredibly beautiful little lips ...'

Note that version corrected for flooding still contains normalization problems. The flooding *o* is incorrectly substituted by *zoo* (Engl. zoo) and not by *zo* (Engl. such), as both words do exist in Dutch. The Dutch adverb *echt* (Engl. really) should be spelled with *ch* instead of *g*.

5.3.2 Suggestion Layer

The suggestion layer comprises a variety of modules which have been conceived to account for the different normalization issues encountered in UGC (cf. Section 5.1). Like in the OCR post-correction system, included modules can be divided into two main groups. The first group contains the **token-based modules**, which are responsible for a specific type of issues. The second group comprises **context-based modules**, which can correct a variety of normalization problems. Since the problem of text normalization shapes different from OCR post-correction, we include a slightly different collection of modules into our architecture.

⁴<http://hunspell.sourceforge.net/>, 19/08/2015.

The **token-based modules** are designed to solve specific normalization problems. They are not expected to return normalized messages but to find a solution to just one problem, more specifically to tackle abbreviations and various misspellings.

- ABBREVIATION module

Language used in UGC often shares certain abbreviations and uniform ways of reference such as hash tags in Twitter posts. Therefore, lookup approaches can cover a reasonable number of issues. The ABBREVIATION module relies on a dictionary of about 350 frequent abbreviations appearing in social media texts, such as *lol* (laughing out loud) and *aub* for *alstublieft* (thank you)⁵.

- SPELL CHECKING modules

The spell checking modules are also included in the OCR post-correction system even though the origin of errors is different. These modules account for normalization problems such as typos, for example the transposition in *spelne* which should be *spelen* (Engl. play), or orthographic mistakes such as the omission of diacritics, for example in *cafe*, which should be *café*. We include three modules in the suggestion layer which relate to the spell checking metaphor.

We use a plain SPELL CHECKER⁶, which uses Levenshtein distance to suggest the most probable correction. The SPELL CHECKER can correct minor misspellings in a word such as *gzien* to *gezien* (Engl. seen) or *zowiezo* to *sowieso* (Engl. in any case).

The second module that uses the spell checker is the COMPOUND module. It checks whether words that have been written as two separate words, should have been written together. It verifies all token bigrams and can solve cases such as split verbs, e.g. *langs komen* to *langskomen* (Engl. drop in), a phenomenon that frequently occurs in Dutch.

The WORD SPLIT module is the opposite of the COMPOUND module and splits words that have been erroneously written together. In UGC, words are often concatenated in order to save space. The WORD SPLIT module is based on the compound-splitter module of Moses (Koehn et al., 2007) and has been trained on the Corpus Gesproken Netherlands (CGN) (Oostdijk, 2000). Problems such as *misje* to *mis je* (Engl. miss you) or *perse* to *per se* (Engl. at any price) can be solved.

A problem related to the spell checking approach is the limited coverage of the word list that the spell checker is based upon. To improve the coverage, we extended the spell checker's dictionary with a word list containing about 2.3 million words compiled from a Dutch Wikipedia corpus. Considering the highly productive nature of UGC, this partly alleviates the problem of out-of-vocabulary words.

⁵This dictionary is available for download at http://www.lt3.ugent.be/amica/chat_abbreviations_dutch

⁶Hunspell: <http://hunspell.sourceforge.net/>, 19/08/2015.

The **context-based modules** have a wider range of responsibilities. Besides the SMT modules described in Chapter 4, we include two other context-based modules. They cover a variety of normalization issues and can solve phonologically motivated problems, as well as spelling mistakes and abbreviations. Their main strength is that they use contextual information during normalization.

- SMT modules

Following previous experiments described in De Clercq et al. (2013), the SMT models have been trained on the token and character level using Moses (Koehn et al., 2007). We include a token-unigram-based module, a character-unigram-based module, a character-bigram-based module and a combination of a token-based and a character-unigram-based module which is reported to perform best in De Clercq et al. (2013). The combination follows a cascaded approach, which means that we first process a message with the token-unigram-based module and subsequently forward the output of this module to the character-unigram-based module. The token model can solve problems of rather frequent shortenings, such as *ng* to *nog* (Engl. still) or *na* to *naar* (Engl. to). Character-based models on the other hand, tend to solve problems such as character transposition, but also problems across tokens such as fusions as in *kheb* and *ik heb* (Engl. I have). Additionally, they may offer better generalization, since they can learn productive alterations and correct them in words that do not occur in the training data.

- TRANSLITERATE module

This module approaches the normalization task as a transliteration problem to be solved using a discriminative sequence labeler. The normalization problem is defined on the level of the grapheme, not unlike the SMT-character-unigram module. It uses the manually annotated data of the training corpus (see Section 5.4) as an information source to build a supervised machine learning classifier in which each grapheme in the non-normalized input sequence is associated with a class in the output sequence. This class can be empty (deletion), the input grapheme itself or a sequence of graphemes, potentially also containing word boundaries (insertion). This is illustrated in the following example:

(12) *kebda ni gedaan*
ik heb dat niet gedaan
 ‘I did not do that’

In preprocessing, we first align these sequences using a dynamic programming script based on Wagner and Fisher (1974), so that they are of equal length:

```
+k +eb+da+ ni++ gedaann
ik heb dat niet gedaa+n
```

This data is consequently presented as training material to a memory-based learner (Daelemans and van den Bosch, 2005) that learns to associate the individual input graphemes with a contextually appropriate output class (input/output with “-” indicating a word boundary):

```
k/ik- e/he b/b- d/d a/at -/- n/n i/iet -/- g/g e/e d/d a/a a/a a/+ n/n
```

The classifier takes different types of context into account: the input characters on the left and the right of the current input character, but also the already transliterated output characters on the left.

- WAYS module

The WAYS module (*write as you speak*) attempts to model idiosyncrasies of UGC in which users write words as they speak, for example *kep* as the contracted representation of the expression *ik heb*, or *ma* instead of *maar*. The module is built as two machine learning classifiers: a grapheme-to-phoneme converter (G2P) and a consecutive phoneme-to-grapheme converter (P2G). We used the phonetic transcriptions of the CGN corpus (Oostdijk, 2000) to train our machine learning classifiers. CGN contains 136,000 transcribed sentences using graphemes and phonemes, as illustrated in the following example:

```
(13) die net daar in de zee ligt zeg maar
      di nEt tAr In d ze lIxt sEx mar
      ‘which is lying there in the sea say’
```

Similar to the TRANSLITERATE module, preprocessing involves aligning the sequences of graphemes, so that input and output sequence are of equal length.

```
die net daar in de zee ligt zeg maar
di+ nEt tA+r In d@ ze+ lIxt sEx ma+r
```

This is used as training material for the aforementioned memory-based learner, which now converts a sequence of graphemes into phonemes as follows:

```
d/d i/i e/+ -/- n/n e/E t/t -/- d/t a/A a/+ r/r -/- i/I n/n -/- d/d e/@ -/- z/z e/e
e/+ -/- l/l i/I g/X t/t -/- z/s e/E g/x -/- m/m a
```

Likewise, a memory-based learner was built that converts a sequence of phonemes back into graphemes.

Finally, as a high percentage of tokens do not contain normalization problems and should therefore not be changed, we also include the ORIGINAL input token in the word candidate list to ensure that we do not lose correct tokens in the input text. Therefore, the original module just adds the original token to the list of suggestions.

Subgenre	Train	Dev1	Dev2	Test	All
SMS balanced	6,665	1,137	1,138	2,150	11,090
SMS all	9,689	1,137	1,138	2,150	14,114
SNS balanced	5,706	929	829	1,701	9,165
SNS all	40,363	929	829	1,701	41,875
TWE	6,471	1,008	1,054	2,119	10,652
Total balanced	18,842	3,074	3,021	5,970	30,907
Total all	56,523	3,074	3,021	5,970	68,588

Table 5.1: Number of tokens of the training, development and test sets listed by subgenre.

5.3.3 Decision Layer

The decision layer is implemented in the same way as the decision layer of the OCR post-correction system. It is the task of the decision module to choose the most probable combination of suggestions to build a well-formed sentence. The language model has been built from a combination of four corpora using KenLM (Heafield, 2011) (see Section 5.4.2 for more details). The phrase table is a lookup table containing words and word sequences along with the normalization suggestions generated by the modules. The decoder can be tuned by allocating weights to the language model and phrase table, setting penalties for phrase reordering and sentence length. We also included features in the phrase table that indicated which module(s) generated a specific normalization suggestion. These features can be tuned as well. We assume that the normalization suggestions of certain modules are more reliable than others, and expect their feature weights to be higher after tuning. All tuning was performed on the development data (see Section 5.4 for a description of the data sets).

5.4 Evaluation

5.4.1 Data Set

The language encountered in UGC differs among different social media applications (Baldwin et al., 2013). To account for this variety, we include three different types of social media content in our corpus, namely texts from Twitter (TWE) accompanying a Flemish TV show⁷, texts from the social networking site Netlog⁸ (SNS) and short messages (SMS) from the Flemish part of SoNaR (Reynaert et al., 2010).

⁷The Voice of Flanders

⁸<http://nl.netlog.com/>; the SNS data is a combination of the Netlog data sets of De Clercq et al. (2013) and Kestemont et al. (2012)

Table 5.1 gives an overview of the size of our experimental corpus. In order to measure the cross-genre performance of our normalization system, we also compiled a genre-balanced data set, which includes an approximately equal number of tokens from each of the subgenres.

We split our corpus into a train set, development set and test set, setting aside about 60% for training, 20% for development and 20% for testing. We use half of the development set for tuning the individual modules (Dev1) and the other for tuning the overall system (Dev2).

All data have been manually normalized and annotated following the guidelines described in De Clercq et al. (2013). All operations that are necessary to transform the anomalous text into standard language have been added to the data. These operations are:

- insertions (INS): *stappe* → *stappen* (Engl. step)
- deletions (DEL): *schatjeeeee* → *schatje* (Engl. honey, darling)
- substitutions (SUB): *egt* → *echt* (Engl. really)
- transpositions (TR): *ftoo* → *foto* (Engl. photo)

This fine-grained annotation facilitates the analysis of normalization issues that are present in the data. Inter-annotator agreement was calculated between the two fully normalized versions for the SMS genre, which is the genre that includes the highest number of normalization problems. This was done by calculating the accuracy of taking one annotator as gold standard to score the annotations of the other. This results in an accuracy of 0.967 for both annotators. If we compare this to the non-normalized accuracy score, i.e. 0.839, we conclude that we have a nearly perfect inter-annotator agreement.

Genre	# Msg	Before	After	%	#INS	#DEL	#SUB	#TR
SMS	1,000	14,114	14,663	3.89	3,624	605	627	57
SNS	1,505	25,670	25,913	0.94	4,170	5,270	1,372	52
TWE	246	10,652	10,633	-0.18	1,104	394	270	9

Table 5.2: Data statistics of the three genres of UGC: the number of messages and the number of tokens before and after normalization, together with the overall expansion rate (left-hand side); normalization effort expressed in the number of operations on character level (right-hand side).

The normalization effort calculated on a part of our data can be seen in Table 5.2. The left-hand side of the table shows the number of messages and the number of tokens included in the corpus per subgenre before and after normalization, and the expansion rate. On the right-hand side the number of individual operations that have to be performed to reach the normalized version are shown. The large number of insertions hint at a high rate of abbreviations and phonologically realized words in our data, whereas deletions can be mainly attributed to flooding. Substitutions and transpositions roughly correspond to spelling problems. The slight decrease in tokens observed in Twitter data is due to words that are spread over multiple tokens in the original text which should actually be written as one word.

5.4.2 Modeling UGC Language

Apart from normalization problems, UGC language differs from standard language in terms of word choice, syntax and style as well. As the language model is a core element of the SMT modules and the decision module, we want to build a high-quality language model that fits the data that needs to be normalized as well as possible.

We have built language models from three corpora and combinations thereof. The corpora, listed in Table 5.3, were all chosen because of their relative closeness to the target domain, i.e. they all contain a high degree of spoken language features. In order to maximize this similarity, we also added all the training data of our UGC corpus. We used KenLM (Heafield, 2011) to evaluate the perplexity of different language models trained on different combinations with respect to our development corpus (Dev1). We varied the order of the models from 3-grams up to 6-grams, but could not observe any improvements above the order of 5. A 5-gram language model, built on the combination of all corpora, obtained the lowest perplexity of 7.4, and was used in the experiments.

Corpus	Sentences	Words
Corpus Gesproken Nederlands (CGN) (Oostdijk, 2000)	985,609	6,765,336
SoNaR (Oostdijk, 2008)	197,493	3,581,182
Open Subtitles Dutch (OSD)	11,788,416	90,147,315
Training set (TS)	3,721	56,523

Table 5.3: Overview of corpora used for language modeling.

5.4.3 Evaluation Metrics

We evaluated our results using standard evaluation measures for lexical normalization, i.e. word error rate (WER) and precision and recall calculated at the token level. Word error rate is a commonly used metric in speech recognition and machine translation evaluation. It takes into account the number of insertions, deletions and substitutions that are needed to transform the suggested string into the manually normalized string and is computed as follows:

$$(WER) \quad WER = \frac{\text{Insertions} + \text{Substitutions} + \text{Deletions}}{\# \text{ Tokens in the manual reference}}$$

Besides WER, we also calculate precision and recall which are widely used metrics in information retrieval. They give information about the degree of overgeneration and undergeneration in the suggested string. Precision and recall are computed as follows:

$$\text{(Pre)} \quad \textit{Precision} = \frac{\# \text{ Correct tokens}}{\# \text{ Tokens in the suggestion}}$$

$$\text{(Rec)} \quad \textit{Recall} = \frac{\# \text{ Correct tokens}}{\# \text{ Tokens in the manual reference}}$$

As the token-based evaluation metrics are rather strict and do not reward improvements that are not entirely correct (e.g. the suggestion *antworden* (correct form: *antwoorden* (Engl. answer)) for the anomalous form *antwrndn*), we also report Character Error Rate (CER). This is inherently the same formula as for WER, but instead of tokens it looks at characters. As we want to focus on the performance of the normalization modules, we take as input the manually tokenized and automatically flooding-corrected version of the data, and each time compare the output with the gold standard data set.

We evaluated the performance of the tokenizer and sentence splitting component in a separate experiment, in which we compared the automatically and manually tokenized strings. Tokenization in UGC is known to be a difficult task due to the productive use of emoticons, punctuation for emphasis and the appearance of concatenated words. The results in Table 5.4 show high precision scores, ranging between 0.98 and 0.99 for the three UGC genres. Recall scores are equally high, ranging between 0.97 and 0.99. Given that this preprocessing step comes before a whole range of normalization modules, high precision is important. We assume that some unsolved tokenization problems might find a solution during the normalization process. A notable problem for the tokenizer are cases in which words are strung together, such as *teveel* which should be tokenized into *te veel* (Engl. too much); this is also a problem for which a dedicated word split module was designed in the next layer.

Genre	SMS		SNS		TWE		ALL	
Metric	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Tokenization	0.98	0.97	0.98	0.98	0.99	0.99	0.98	0.98

Table 5.4: Evaluation results of the tokenization module.

5.4.4 Experiments

Our experiments are structured in two main parts. First, we investigate the performance of *each module separately* by presenting precision, recall, WER and CER scores. Because some modules were specifically designed to solve a certain type of normalization issue (cf. Section 5.1), we also performed a task-specific evaluation for them, i.e. the compound, abbreviation and word split module. This was done by evaluating each module with respect to its responsibility range,

which was manually annotated. In a next step, we evaluate the overall performance of our *multi-modular system* using the same evaluation metrics. In this setup, we apply different settings by weighting the individual modules differently for the decision making process, by adding more information about the necessity to normalize a token and by using different training sets.

Since only a portion of the tokens in the input sentence exhibit normalization issues, we experimented with filtering the module suggestions, to assess the impact on performance (both of individual modules and of the combined systems). The reasoning behind this is that we want to filter out unlikely suggestions to avoid overcorrection during the normalization process, viz. tokens which do not include any normalization problem should not be changed for the worse. We use two filters: a classifier trained on a bigram language model and a named entity recognizer. The classifier is trained on a simple bigram language model compiled from the data described in Section 5.4.2. We look up each token of the input sentence in the context of the preceding and subsequent token and only retain normalization suggestions for tokens for which we cannot find both bigrams in the language model.

The second filtering mechanism aims at detecting named entities (NEs). NEs typically consist of out-of-vocabulary words which should not be normalized. It is therefore important to recognize them as such in order to avoid overcorrection. Named entity recognition (NER) in tweets is a far from trivial problem (Liu et al., 2013): NEs in UGC often have different characteristics than in standard texts (NEs frequently lack capitalization or are introduced with specific characters such as @ or #), we developed a dedicated NER tool (Schulz, 2014). The NER tool is hybrid in the sense that it uses gazetteer lookup and classification. The gazetteers contain a variety of named entities. Moreover, it includes a simple pattern-matching rule to find words with a capitalized first letter that does not appear at the beginning of a sentence. Given the productive nature of NEs, we also added a dedicated conditional random field classifier trained on the training set of our corpus.

Module	SMS		SNS		TWE		ALL	
Metric	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
bigram LM	0.93	0.84	0.96	0.85	0.98	0.92	0.95	0.87
NER	0.65	0.69	0.38	0.39	0.93	0.39	0.76	0.58

Table 5.5: Performance of the filtering methods.

Table 5.5 shows the results of these two types of filtering for the three genres. For both techniques, we compared the output of the filtering with the gold standard. The precision obtained with bigram filtering is high, ranging between 0.93 and 0.98, whereas the recall scores range between 0.84 and 0.92. The precision of the NER module is high for Twitter data and reasonable for SMS. For SNS we observe a large number of tokens mistakenly classified as NE. This could be attributed to a non-standard usage of uppercase and lowercase letters.

Module	SMS			SNS			TWE			ALL			
	Pre	Rec	WER	Pre	Rec	WER	Pre	Rec	WER	Pre	Rec	WER	CER
baseline	81.6	78.2	21.8	79.7	76.0	24.2	96.3	96.2	4.3	86.6	84.1	16.1	7.7
Without filtering													
SMT Token	87.5	87.0	12.3	84.1	81.7	18.0	96.2	96.1	4.1	89.8	88.9	10.8	6.0
SMT Unigram	92.6	92.0	7.5	86.5	85.6	14.1	96.6	96.7	3.7	92.4	92.0	7.7	4.9
SMT Bigram	92.8	91.5	8.3	86.5	84.7	14.7	95.0	95.1	5.4	92.0	91.0	8.9	6.4
SMT Cascaded	89.9	90.3	9.6	86.0	85.4	14.0	96.2	96.4	3.8	91.1	91.2	8.2	5.1
WAYS	68.9	65.9	34.2	63.8	60.8	40.2	73.7	73.1	28.4	69.2	67.0	33.6	17.4
Transliterate	90.0	88.9	11.1	84.0	81.6	19.1	94.0	93.9	6.8	89.9	88.8	11.6	6.4
Spell checking	81.1	77.7	22.0	75.6	72.1	27.7	94.0	93.8	6.6	84.5	82.1	17.9	7.7
Abbreviation	82.3	79.1	20.9	79.7	76.6	23.7	96.3	96.2	4.1	86.8	84.6	15.5	7.4
Compound	81.9	74.7	29.6	80.2	73.0	31.0	96.9	91.1	15.4	87.0	80.2	18.5	9.3
Word Split	78.2	76.1	24.3	78.6	75.3	25.2	91.7	93.1	7.3	83.2	82.1	18.4	7.9
With filtering													
SMT Token	86.6	85.0	14.9	83.9	81.2	18.7	96.5	96.3	4.1	89.5	88.1	11.9	6.3
SMT Unigram	89.1	87.6	12.0	85.4	84.4	15.4	96.3	96.4	4.0	90.8	90.0	10.0	5.4
SMT Bigram	88.6	86.9	12.9	85.1	83.0	16.3	95.7	95.7	4.8	90.3	89.1	10.8	5.9
SMT Cascaded	88.2	87.0	12.7	85.4	84.3	15.2	96.3	96.4	4.0	90.6	89.7	10.2	6.4
WAYS	79.8	76.4	23.7	74.7	71.2	29.7	92.2	91.9	8.5	83.1	80.7	19.7	11.7
Transliterate	87.3	85.7	14.4	83.7	81.4	18.9	96.0	96.0	4.4	89.6	88.3	12.0	6.4
Spell checking	82.2	78.8	20.9	78.9	75.3	24.5	95.5	95.3	5.1	86.3	83.9	16.1	9.2
Abbreviation	82.4	79.2	20.9	80.0	76.7	23.5	96.3	96.2	4.1	87.0	84.7	15.8	6.9
Compound	81.7	78.1	22.1	80.0	75.0	25.0	96.4	95.9	4.8	86.7	83.9	16.5	7.8
Word Split	79.2	76.8	23.5	78.6	75.6	24.9	94.5	95.0	5.4	84.8	83.1	17.3	7.8

Table 5.6: WER, CER, precision and recall of the general modules with and without filtering of suggestions.

Module-Specific Evaluation

The evaluation scores for all modules are presented in Table 5.6. Baseline scores are calculated by comparing the manually tokenized original input with the gold standard normalized text. A first observation is that, in general, the performance varies significantly between the different UGC genres. The highest scores are obtained on the Twitter data, followed by SMS and SNS. This variation can be explained by the difference in density of normalization problems, which is in line with the data statistics that were presented in Table 5.2.

Table 5.6 also illustrates that the SMT and TRANSLITERATE modules reveal particularly high performance. The character-based SMT modules outperform all other modules, with and without filtering. It performs best with a WER reduction of almost 50% over the input text. CER shows a similar tendency. The strength of the character-based SMT modules lies in resolving concatenations such as *keb* to *ik heb*, whereas the token-based module is doing well in resolving frequent abbreviations. The TRANSLITERATE module also shows good normalization capabilities. Even though it does not contain any mechanism to prevent out-of-vocabulary words on the output side, it is able to resolve quite a few issues of compounding and cliticization.

We observe that without filtering, four modules are never able to beat the baseline (WAYS, PELL CHECKING, COMPOUND and WORD SPLIT). These are modules that typically overcorrect,

and as a result, we observe some moderate improvements after applying filtering.

While the WAYS module is able to model some aspects of write-as-you-speak effects, its usability on our data is rather limited. Correct words in the input sequence are very often converted erroneously through the processing chain. Furthermore, it is by definition not able to convert abbreviated forms, such as *ff* for *effe*, which are plentiful in our data. Finally, write-as-you-speak effects are very dependent on regional varieties of Dutch. As a result, a single pronunciation model capturing all such regional variants is just not tractable. Specialized region-specific WAYS modules may obtain better results.

Since the COMPOUND, ABBREVIATION and WORD SPLIT modules have been designed with a specific normalization issue in mind, these modules have a specific range of responsibilities (cf. Section 5.3.2). Table 5.7 gives an impression of the absolute number of problems each module is responsible for, based on a manual analysis and the actual performance with and without filtering. Besides the fact that the type of problems encountered in the three UGC genres differs considerably, we can also observe that some specialized problems are rather infrequent in our data, such as the small amount of compounding issues. We will now discuss the results of those three modules in closer detail.

Module	SMS			SNS			TWE		
	RES	COR	OVER	RES	COR	OVER	RES	COR	OVER
Without filtering									
Compound	2	1	96	7	4	60	8	4	136
Abbrev.	94	27	2	46	14	4	18	1	0
Word Split	10	0	57	26	2	15	0	0	80
With filtering									
Compound	2	0	3	7	1	6	8	1	8
Abbrev.	94	27	0	46	13	0	18	1	0
Word Split	10	0	38	26	2	10	0	0	31

Table 5.7: Number of problems each specialized module is responsible (RES) for, has solved correctly (COR) and has overcorrected (OVER).

Without filtering, the COMPOUND module can only solve about half of the problems of its responsibility range. In addition, we notice that it returns a lot of incorrect suggestions. As mentioned earlier, the number of problematic compounds is small in our test set. In total, there are just 17 problems that have to be solved, amongst which compounds such as *songkeuze* (Engl. song choice) and *dragqueen* (Engl. drag queen), which are very uncommon in Dutch. Introducing filtering leads to a drastic decrease in the number of overcorrections, but it also harms the ability of the module to solve problematic compounds correctly.

We can observe that the ABBREVIATION module is able to solve around 30% of the issues in the SMS and SNS genre, and 5% in the TWE genre. If we translate these numbers into precision and recall, we can see that it achieves a high precision and a rather low recall (averaged over all

genres, we reach a precision of 0.90 and a recall of 0.22). This high precision can be attributed to the lookup approach it is based on. The low recall points to a coverage issue of the dictionary. A manual analysis revealed for example that abbreviations such as *Hvj*, which stands for *hou van jou* (Engl. I love you), or *ipv* for *in plaats van* (Engl. instead of) remained uncorrected as they are not yet included in our dictionary, although they are highly frequent in Dutch. Nevertheless, high precision means that the module does not harm the overall performance of our system. Extending the dictionaries represented in this module could lead to a more valuable module contributing well to the normalization success. It is also worth mentioning that the filtering method works well for this module, because no overcorrections remain after filtering.

Finally, the WORD SPLIT module has the lowest performance of all. This can be attributed to the modules' inherent capacity to only split a word into two when those two words are actually existing and correct words. As a consequence, it cannot split words containing additional normalization problems. Typical examples are *kzit* which has to be split into *k* and *zit*. Since *k* has to be transformed into *ik* in order to build the correct bigram *ik zit* (Engl. I sit), the module cannot cope with those problems. The same problem occurs in fused words such as *loveyouuu*, where the second token is anomalous. Again we see that this module accounts for a large number of overcorrections. Introducing filtering leads to a decrease, but not as clearly as it was for the previous module.

To conclude, we can state that the actual responsibility range of the COMPOUND and WORD SPLIT modules looks rather limited. However, in order to evaluate the complementarity of the different modules, we also manually checked the number of unique suggestions each module (without filtering) proposes on the development data set. This revealed that even these three worst performing modules each return unique correct suggestions. We therefore decided to keep all modules in our multi-modular system and leave it up to the decision module to select the best suggestion. This is certainly not a trivial task. The two modules that suggest the highest number of unique correct suggestions (WAYS and SPELL CHECKING each offer 16) also generate the highest number of unique incorrect suggestions (1571 and 594, respectively).

Multi-Modular System

Having explored the performance of all modules separately, we also evaluated the interaction of all our modules in combination. As described in Section 5.3.3, we include features that provide information on which module(s) generated a normalization suggestion into the decoding process using the Moses decoder. Initially, these features were uniformly weighted (setting 1), but after further tuning on the development set (settings 2–5), we set the decoder to trust certain modules more than others. It is important to note that these are overall module weights, which do not take into account the particular normalization issue at hand.

Since we observed that filtering improved the output of some of the modules that tend to overcorrect, we also experiment with two different approaches to include this filtering in our

system. In one setting, which we label “hard filtering” (setting 3), we remove suggestions for tokens that according to the filters should not be normalized. In the second approach (setting 4), “soft filtering” is applied by adding this filtering information in the form of two additional features (NER and bigram LM) to the decoding process. The weights for these two additional features are tuned alongside other decoder parameters. In a last evaluation scenario, we built a system using all our training data using the best settings of the previous experiments (i.e. with tuning and soft filtering) and compare the results of the all-data-in setting to an all-data-in baseline. All in all, we have thus set up five evaluation scenarios:

- (1) genre-balanced system without tuning
- (2) genre-balanced system with tuning
- (3) genre-balanced system with tuning and hard filtering
- (4) genre-balanced system with tuning and soft filtering
- (5) all-data-in system with tuning and soft filtering

For the evaluation of the entire system, we decided to focus on minimizing WER. The first baseline is again calculated on the original, manually tokenized data. As a second baseline, we took the single best-performing module (MT UNIGRAM). A combined approach should in any case beat the second baseline in order to show that a combination of modules leads to an improvement over a single module approach.

System	SMS			SNS			TWE			ALL			
	Prec	Rec	WER	Prec	Rec	WER	Prec	Rec	WER	Prec	Rec	WER	CER
baseline	81.6	78.2	21.8	79.7	76.0	24.2	96.3	96.2	4.3	86.6	84.1	16.1	7.7
SMT Uni	92.6	92.0	7.5	86.5	85.6	14.1	96.6	96.7	3.7	92.4	92.0	7.7	4.9
1	89.6	87.3	12.8	84.9	81.8	18.4	96.7	96.5	3.9	91.0	89.2	11.0	6.1
2	92.2	92.2	7.5	87.5	87.5	12.4	97.0	97.2	3.2	92.8	92.8	7.2	4.9
3	88.7	87.6	12.1	86.1	85.8	14.0	96.2	96.4	3.9	90.8	90.4	9.7	5.5
4	91.3	92.7	7.0	87.6	87.4	12.6	96.9	97.1	3.2	93.1	92.9	7.1	4.8
SMT Uni all	92.9	92.2	7.4	88.1	87.8	12.0	95.8	96.2	4.1	93.4	92.5	7.4	4.6
5	93.5	93.0	6.7	89.1	88.2	11.5	95.9	96.3	4.0	93.2	92.9	6.9	4.7

Table 5.8: Precision, recall and WER of the normalization in five different settings for each genre and on the entire test set.

The results in Table 5.8 show that the genre-balanced system without tuning (setting 1) improves WER on the entire test set by about 30% over the first baseline and reaches high recall and precision scores. Model tuning (setting 2) improves results noticeably by lowering WER to 7.2%; a decrease of more than 50% over the baseline. This experimental set-up beats the best performing single module, which has a WER of 7.7%.

In order to gain some insight into the contribution of the different modules to the overall system, we inspected the feature weights of the modules. The weights do not entirely correlate

with the ranking of the performance of the modules in terms of WER, but do reveal the same tendency. The highest weight is allocated to the SMT modules. The abbreviation module, which shows reasonable performance, gets the third highest weight. As expected, modules that highly overgenerate receive a low weight.

Interestingly, we cannot show an overall improvement in WER over setting 2 by adding hard filtering (setting 3). It especially impairs results for the SMS test data which contain the highest number of normalization issues. This means that hard filtering removes too many correct suggestions for anomalous words. The CER values slightly improve by hard filtering, which can be explained by the limitation of overcorrection.

Soft filtering (setting 4) performs better in comparison to hard filtering on all genres. It appears that adding filtering information as decoding features to be tuned achieves slightly better results than when such filtering is absent (setting 2) for SMS and achieves the best scores for all data amongst the genre-balanced systems. This shows that flagging a token which contains a normalization problem by the bigram language model or a NE adds valuable information to the decoding process.

Adding more training data (setting 5) introduces a slight bias towards SNS data. The performance for TWE slightly suffers, whereas the performance for SNS and SMS noticeably improves, since we substantially extend the training set for SNS. The WER calculated on the entire test set is the lowest amongst all systems. We achieve significantly better results with our multi-modular system compared to the SMT Unigram module with all training data as a baseline. Significance has been calculated using the Monte Carlo algorithm (Efron and Tibshirani, 1986) with a resulting 95% confidence interval of 1.19 and 1.22 of difference in mean using 10,000 test suits.

Since we cannot presuppose that the decision module always picks the right suggestion even if it is provided by the modules, we also calculated the upper bound performance for system setting 5, which assumes a perfectly working decision module. These oracle values are shown in Table 5.9.

Genre	SMS	SNS	TWE	ALL
Oracle	96.2	93.7	98.2	96.3
5	93.0	88.2	96.3	92.9

Table 5.9: Oracle recall values for the tuned, soft filtered genre-unbalanced system compared to the recall values achieved by the system in this setting without oracle.

A first observation is that our system almost reaches the upper bound of 96.3 with an actual recall of 92.9, which means that the decision module performs really well. Nevertheless, the oracle values also show that not all normalization issues are handled by the modules of the suggestion layer. A manual inspection of the tokens for which no correct suggestions are provided, shows that those tokens often contain more than one normalization issue. An example is *tuurlyk*

for *natuurlijk* (Engl. of course) which is not only shortened but also has the homophones *ij* and *y* exchanged. Therefore, a spell checking approach or a machine translation approach will probably struggle to solve such issues since they deviate too strongly from the standard form. The problem of multiple corrections within one word could possibly be solved by a sieve technique in which modules are called consecutively instead of in parallel.

5.4.5 The Bigger Picture - Extrinsic Evaluation and Portability

Since the main motivation for text normalization is to counter the drop in performance of NLP tools on non-standard text, we also performed an extrinsic evaluation of our approach, similar to the work described in (Zhang et al., 2013). We evaluated the performance of a POS tagger (POS), a named-entity recognizer (NER) and lemmatizer (LEMMA) (van de Kauter et al., 2013) before and after normalization (NORM) on a test set from a subgenre which had not been included in training. Therefore, we additionally annotated 918 posts (7,610 tokens) from the social network ask.fm⁹ for these four tasks.

We used the best-working multi-modular system including all training data with soft filtering (setting 5) to normalize the posts. As can be seen in Table 5.10, for the normalization of this new subgenre, the system performs much better than the baseline (WER of 24.6).

To assess the impact of normalization on other NLP tasks, we include the results for our gold standard data to set the upper bound we can reach with perfect normalization and calculate accuracy and F-score. For all three tasks (POS tagging, lemmatization and named entity recognition), we observe a clear improvement after normalization, with an accuracy of 73.5% (after normalization) vs. 66.1% (before normalization) for POS tagging, and an accuracy of 80.7% (after normalization) vs. 71.5% (before normalization) for lemmatization.

Metric	WER		Accuracy		F-score
	NORM	POS	LEMMA	NER	
Gold standard	-	79.8	90.2	20.7	
Before normalization	24.6	66.1	71.5	18.5	
All-data-in system tuned module weights, soft filtering	14.9	73.5	80.7	20.4	

Table 5.10: Performance of different NLP tools before and after normalization with the all-data-in multi-modular system.

The performance improvement for NER, on the other hand, is very modest. The low scores of NER on the gold standard data set further illustrate that Named Entity Recognition is a very difficult task in UGC.

⁹<http://ask.fm/>

5.5 Research Contributions

Automatic normalization of UGC is a complex task with many challenges. In this chapter, we work with three different types of Dutch UGC, namely SMS, blog and forum posts and tweets. As can be seen in the expansion rate before and after normalization (Table 5.2) and the baseline WER scores (Table 5.8), the normalization effort for the different subgenres varies considerably, with tweets being easier to normalize than SMS and posts on social network sites.

To account for the diversity of normalization problems, we implemented eight different modules that make use of three well-known metaphors for normalization: spell checking, speech recognition and machine translation. The module-specific evaluation shows that especially the modules belonging to the machine translation metaphor (the SMT and TRANSLITERATE modules) perform well. However, as even the low-performing modules generated unique suggestions, we built a multi-modular system based on all modules.

The real challenge of the multi-modular system is the selection of the best (combination of) candidates from the pool of suggestions, which is the task of the decision module. We store all normalization suggestions in a phrase table and make use of the Moses decoder to tackle this problem. In contrast to previous research efforts that were limited to language model-based decoding, we use the phrase table infrastructure provided by Moses and add additional features to it that encode information about which module(s) generated a normalization suggestion. These features were tuned on the development set, thus permitting the decoder to learn to trust certain modules more than others. Furthermore, we experiment with two types of filtering (hard and soft filtering) to reduce overcorrection. The oracle values show that the decision module obtains a high performance, despite the large number of suggestions.

Since the main motivation for text normalization is the **improvement of the performance of state-of-the-art NLP tools** on UGC data, we also perform an extrinsic evaluation on data normalized by our system on yet another type of UGC, namely posts from ask.fm. We demonstrate that **automatic normalization indeed improves the performance of POS-tagging, lemmatization and NER**. However, the performance level of the standard NLP tools on UGC data (after normalization and even on the gold standard data) is still far below the performance level of those tools on standard language. This might be due to the high degree of syntactic anomalies and English words in Dutch UGC, which our system at this moment is not able to tackle.

We show that **modular software architectures do not only offer the opportunity to adjust to different data sets but moreover can be used for different objectives**. Text normalization shares characteristics with OCR post-correction as it attempts the automatic transformation of erroneous input text to some corrected form. By exchanging various modules with modules suited better to a specific problem, this generic architecture can easily be used to different ends.

TOOL ADAPTATION FOR NON-STANDARD TEXT PROCESSING

The application of automatic text processing methods has the purpose of enriching text with information on different levels such as syntax and semantics. As discussed in Chapter 3, in the context of Digital Humanities (DH) projects this commonly comes with a particular challenge: The texts under investigation usually deviate from some agreed upon form. In the previous chapter, we offer text normalization as a solution to this issue. In cases where the language under investigation differs considerably from its standard form, text normalization is a problem that is difficult to model. Errors made in the normalization propagate down to subsequent processing steps which influence the results of the final analysis. Moreover, text normalization requires a considerable amount of training data which DH projects are often lacking. Commonly, deviations from the standard form are exactly the focus of the investigation and thus have to be handled with care. If one wants to compare the different ways Mark Twain lets the characters speak in *Tom Sawyer*, the orality Twain gives to the different voices is the focus of investigation and must be preserved.

As an alternative to the normalization approach illustrated in Chapter 5, we propose a **data-oriented methodology for the development of dedicated tools for non-standard texts** in this chapter. Tools can be trained from scratch or adjusted from related languages. We focus on machine learning (ML) solutions. Training data, its quality and quantity, are the central points in natural language processing (NLP) with a focus on ML. As outlined in Section 3.2, those ML algorithms rely on the correctness of annotations. If one can provide the algorithm with enough data, statistics can be used to fit a model to solve problems that can be formalized in such a way that they can be modeled with a computer. In DH, the data sparsity issue caused by the before mentioned data characteristics makes ML solutions difficult. Often only small corpora are of interest to answer a specific research question. Even if annotated data is available for training,

the extraction of informative features as input to ML algorithms is difficult due to the lack of preprocessing tools for these specific kinds of data such as part-of-speech (POS) taggers, parsers and tools for automatic discourse analysis. This creates a vicious circle which traditionally is solved by expenditure of time and money for extensive manual annotation. This is worthwhile when developing tools and resources for popular languages which guarantee progress for a wider community, whereas for very specialized types of texts possibly only used within one project, this cannot be the solution. Thus, non-standard text processing poses challenges that have to be tackled in ways that are different from those for standard data.

POS tagging is the task of assigning a category from a given set to each input token of a text. It has a popular use as standalone application or is used as part of a preprocessing step for other tasks, e.g. parsing. It therefore needs to be done with high accuracy to ensure success in the subsequent task. Thus, it is a well understood field offering a variety of techniques suitable for different languages. We choose this problem for our experiments since it is considered to be “solved” for standard texts but not for non-standard texts with tagging accuracies that reach human performance. We focus on the POS tagging of historical texts. Historical stages of languages are one example of non-standard texts. These texts do not just differ from the modern stage of the language but commonly also show a large diversity within what is considered to be one stage of a language due to missing regulation of spelling and grammar and ununified vocabulary. Thus, features of texts coming from different regions, time periods, genres or even authors might vary enough to entail a noticeable difference in performance of automatic language processing tools. These features also make the normalization approach difficult, since they deviate significantly from the modern form. Even if texts were normalized before processing, important characteristics would be lost and could not be reflected in the processing results. Moreover, due to the diversity within one language stage, one normalization system would not suffice to normalize the different regional or time-dependent deviations.

The best POS taggers that are available for English reach an accuracy of up to 97.6% on the Wall Street Journal (Choi, 2016) and the best German models perform around 97% for newspaper texts (Brants, 2000; Schmid, 1994). However, the performance for texts from the web drops to 90–93% (Giesbrecht and Evert, 2009) and more significantly decreases for Middle High German to 45%¹.

In this chapter, we detail the difficulties inherent to non-standard text processing and **suggest techniques for successful automatic annotation**. Historical languages share important characteristics that can be utilized in its automatic processing: usually, earlier stages of a language share a fair number of features with their modern stage. We can exploit this relatedness to facilitate tagging by bootstrapping tools developed for the modern stage or by enhancing algorithms tailored to these older stages with information from these tools.

¹Accuracy achieved applying a German model trained on the German Universal Dependency Treebank using TreeTagger to our test set described in Section 6.1.

Appropriate approaches strongly depend on the characteristics of the data and the nature of the task. To be able to **compare techniques for different kinds of data**, we evaluate the task of POS tagging throughout different data sets with a varying degree of divergence from the standard form and differences in terms of availability of preprocessing tools. We experiment with three different degrees of non-standard data which allow for different approaches towards their processing. We investigate tagging of **Middle High German** (MHG) text, a former language stage of German, originating from between 1050 to 1350. Even though there are few automatic processing tools, there is a considerable amount of digitized data available. In addition, we experiment with Heinrich von Neustadt's *Apollonius von Tyrland*², a 20,645 verses long opus containing approximately 180,000 types and 800,000 tokens. This is a **unique text which mixes features from MHG as well as from New High German** and ranges therefore high on a scale of "non-standardness". There is no other text available that shares its really specific characteristics. As the last example of a text that deviates from a standard, we investigate the automatic processing of **mixed sermons written in Middle English and Latin** (Horner, 2006). These texts are constituted by a combination of two different standard forms and thereby develop characteristics not associated with either one of both standard forms. They are considerably close to both of these standard forms and available resources and tools can be exploited for their processing. We illustrate the processing of these sermons in the context of a research question coming from linguistics in Chapter 7.

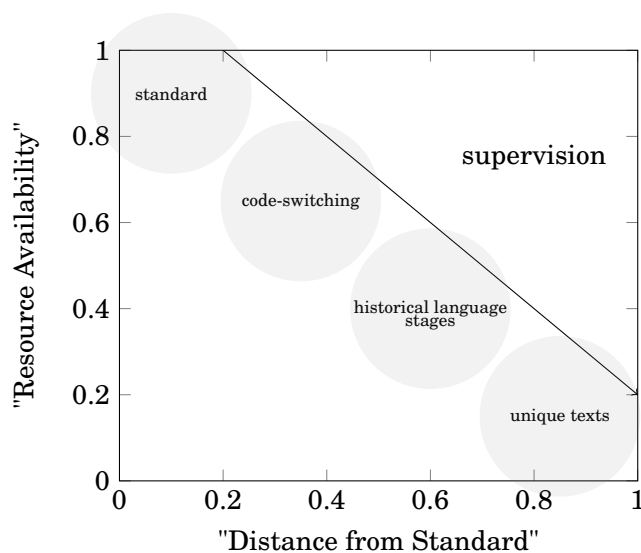


Figure 6.1: Relative location of the data sets used in this chapter with respect to availability of resources and closeness to a standard form.

The differences of the languages that manifest in these texts with respect to the standard

²Based on the Gotha manuscript edited by Samuel Singer, Berlin 1906. Digitalized version from <http://www.mhgta.uni-trier.de> (Gärtner, 2002).

can be expressed in two dimensions: the availability of data resources and automatic processing tools for a language and their dissimilarity to a language that is considered as standard. This is visualized in Figure 6.1. These two dimensions are by no means independent of each other. A language for which there are large amounts of annotated data available will not be considered a non-standard language in our context since it thereby establishes a standard itself. At the same time, a language that is relatively close to a standard form has a decent supply condition of processing tools coming from the standard domain, even though they might not work perfectly. With a falling availability of resources of whichever kind and simultaneously increasing distance of a text to a standard, the viability of supervised methods decreases. This means the further a text can be located in the lower right corner of this two-dimensional space, the more one needs to focus on weak supervision or even unsupervised learning techniques. We examine different aspects of non-standard text processing with respect to a text's localization in the spectrum of non-standard data.

At first, we investigate the **influence of the quality and quantity of training data** in Section 6.1. We introduce strategies for the adaptation of data resources developed for a different purpose to serve as training material for POS tagging. In Section 6.2, we subsequently aim to answer the question as to which **algorithms and processing techniques** are most promising in historical non-standard text processing.

In the following, we investigate three major questions:

1. In which way does the quantity and quality of training data influence the performance of a model?
2. How can existing POS models and resources be utilized in order to benefit the goal of tagging related text?
3. Which algorithms and techniques are suited best for the purpose of POS tagging in the domain of historical texts?

6.1 Training Data: The Influence of Quantity and Quality

Historical linguistics is a field in which scholars early on showed an affinity for computer-aided investigations of their research topics. One of the first digital corpora was a Latin corpus containing the works of Thomas of Aquin (Busa, 1980) constructed over the course of 30 years starting in the 1940s. It is thus even more astonishing that there is a lack of NLP tools for older stages of languages. This can be explained by the fact that even though digitized raw text is available for e.g. statistical analysis, there is hardly a tradition of enriching these corpora with manual annotations. This tendency to digitize historical text, however, could be exploited in order to change this situation. In the following, we want to investigate which approach is most promising when developing tools for historical stages of language. We exemplify our approaches by means of POS tagging of Middle High German (MHG) texts. POS taggers have been a formidable obstacle in the computer-aided analysis of Medieval German texts as they serve for the basic preprocessing before several more extensive steps in an automatic analysis of text. POS tagging has been tackled for different historical languages such as e.g. Ancient Greek (Celano et al., 2016), historical Dutch (Hupkes and Bod, 2016) and Coptic (Zeldes and Schroeder, 2015), which are trained on annotated historical corpora. Hardmeier (2016) trains taggers for Early Modern Swedish and German texts from between 1650 and 1800 using annotated corpora for only the modern stage of the languages but not for the historical stages. The historical languages described in this work, however, are close to the modern language due to their relatively late time of origin. Schulz et al. (2016) describe an approach to POS tagging of one specific MHG text. Barteld et al. (2015) train a POS tagger for Middle Low German. Dipper (2011) reports an accuracy of about 92% for tagging of specific dialects of Central German and Upper German trained on normalized lemmata. All of these models are either restricted to certain varieties of MHG or work on normalized text. To the best of our knowledge none of these models is publicly available.

The term Middle High German (MHG) denotes the stage of German spoken in the High Middle Ages (1050–1350), following the period of Old High German (750–1050) and preceding the period of Early New High German (1350–1650) (Hennings, 2003, p. 11–12). The notion High German refers to the distinction between the varieties of German spoken in the south of Germany (High German) and those spoken in the north (Low German), which were not affected by the second Germanic consonant shift. Beyond this large division into two language areas the MHG speaking area can be further subdivided into various dialects (e.g. Bavarian, Alemannic, East Franconian). The MHG literature shows a high diversity which arises from its different traditions. It has its beginnings in the 12th century and reaches its high point with the “classic” courtly literature between 1170 and 1230. Central “genres” in MHG literature are courtly romances, heroic epics, and lyrical poetry e.g. “Minnesang”. The genres differ in their form – strophes versus verses –, in their mediality – writing versus oral tradition – and in the subjects they discuss. Thus, the linguistic diversity of MHG is complemented by a heterogeneity of literary genres and traditions.

The development of tools for historical languages is clearly related to NLP for low resource languages where the biggest challenge is the lack of annotated data for the supervised training of classifiers. Some advantages arise, however, from working with historical data in the context of Digital Humanities projects. Firstly, in collaborations expert knowledge is accessible. This means that trained experts can provide annotations and give feedback on tagging results which can in turn be used for improvement. Moreover, even though there is a lack of annotated data for specific purposes such as POS tagging, early interest of historical linguists in digital methods results in a moderate availability of raw digitized data as well as digital lexical resources which can be exploited to make up for missing manual annotations.

We investigate various aspects of POS tagging in a non-standard data context generally following a supervised machine learning approach. We firstly evaluate the influence of data quality and quantity via different annotation schemes on tagging results and then show how to exploit other existing data resources. We show how we take advantage of the large size of the Mittelhochdeutsche Begriffsdatenbank (MHDBDB) by adjusting this database of MHG to our annotation scheme. As a result, we present a POS tagger which performs well on different genres, time periods and dialects of MHG.

We do not just report on the development of a tagger model but in addition aim to answer the following questions:

- How much annotated data is needed until the learning effect is decelerating?
- Is there a way to incorporate large resources that have been developed for a different purpose to improve tagging?
- Can large data quantity make up for low data quality?

**Publication**

Parts of this subchapter have been submitted for publication to a special issue of the journal “Language Resources and Evaluation”.

6.1.1 Data Quantity and Quality

We do not just report on the development of a tagger model but also aim to answer the following questions:

- How much annotated data is needed until the learning effect decelerates?
- In which way can large resources that have been developed for a different purpose be incorporated to improve tagging?
- Can large data quantity make up for low data quality?

Given that our collaboration includes scholars of medieval literature who are willing to invest some time into annotating POS tags, we investigate the quantity of POS annotated data needed to train a POS tagging model. It is often assumed that a large amount of annotated data is necessary to develop tools. We show that acceptable results can be achieved with a few thousand annotated words. Moreover, we demonstrate how resources that have been developed for other purposes can be harnessed when included in an intelligent manner. Data obtained in this manner is of lower quality regarding the task of POS tagging since the annotation is not manually overseen. Yet, the inclusion of such data into the training process can lead to an increase in the ultimate tagger quality due to the increase in data quantity.

6.1.2 Manual

Annotation – How much is enough?

Even though the comparison of results across languages and data sets is difficult due to differences in tagsets and annotation schemes, we have a look at sizes of training sets of previous approaches towards POS tagging to get an impression on how much data is recommendable. The state-of-the-art results for English POS tagging reported by Choi (2016) are achieved by training on a data set comprising more than 900,000 tokens. Brants (2000) show learning curves for POS tagging ranging from 1,000 to 320,000 tokens with accuracy values between 78.1% and 96.7%. On the other hand, Schmid (1994) reports state-of-the-art results of around 97.5% for a system trained on just 20,000 tokens along with a large list of word forms. Garrette and Baldrige (2013) describe how about 2,000 tokens of manual annotation can be exploited to train taggers for low resource languages. Even though they achieve impressive results, their work does not allow insights into how much results could be improved by just a little more annotation effort. This raises the question of how much training data is actually necessary until the learning curve decelerates, which we aim to answer in the following.

Data

We manually annotated a corpus consisting of 20,000 tokens with POS tags. This corpus is compiled by including parts of a variety of texts included in the Middle High German Conceptual

Tag	Explanation
ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Table 6.1: UD-Tagset. The tag SYM was not needed; we added combined tags for MHG as well as the tag SPUNCT to distinguish sentence-ending characters from other punctuation marks.

Database (MHDBDB) (Mittelhochdeutsche Begriffsdatenbank, 2017) which cover a period of four centuries. To make sure that we include different genres, dialects and language stages, we randomly select pieces from different texts. This accounts for the diversity we encounter in MHG. Initially, a part of the manual annotation (1,500 tokens) was done in parallel by two different annotators to compute the inter-annotator-agreement (Cohen’s kappa: 0,88 (Cohen, 1968)). Due to the rather low agreement, we investigated the disagreements and specify the guidelines with the help of examples accordingly. Disagreements became evident especially in cases such as the selectivity of participles and adjectives or determiners and adjectives. Our POS categories follow the tagset established by the Universal Dependency (UD) Project (Nivre et al., 2016), which provides a universal inventory of categories to facilitate a consistent annotation of similar constructions across languages and is thus also suitable for the annotation of historical languages. The POS tagset consists of 17 universal categories (cf. Table 6.1). The POS tags are strictly annotated in their syntactical context which avoids ambiguities. One surface form, e.g. “daz” (Engl. the, that) can represent different functions and POS classes in a sentence, as shown in Example 14:

- (14) a. Daz (article, DET) edel kint hât mir verjehen, daz ez in troume sî geschehen.
The (article,DET) noble child had me told, that it in dream was happened.
‘The (article, DET) noble child had told me that it had happened in a dream.’
- b. Wie staete ist ein dünnez eis daz (relative pronoun, PRON) ougestheize sunnen hât?
How stable is a thin ice that (relative pronoun, PRON) August hot sun has?
‘How stable is thin ice that (relative pronoun, PRON) gets hot in the August sun?’
- c. Daz (demonstrative pronoun, PRON) sage ich iu vür ungelogen.
That (demonstrative pronoun, PRON) say I you for truly.
‘This (demonstrative pronoun, PRON) I truly tell you’
- d. Der knappe wânde sunder spot, daz (subordinating conjunction, SCONJ) ieslicher waere ein got.
The squire believed without mockery, that (subordinating conjunction, SCONJ) each was a god.
‘The squire believed indeed that (subordinating conjunction, SCONJ) each was a god’

Word classes can be identified with the help of the substitution test according to which words can be substituted within a class to still yield syntactically valid sentences³. E.g. “schoene” (Engl. pretty) in 15a is an adjective and can therefore be replaced by another adjective (e.g. “minnicliche”, Engl. lovely), whereas in 15b “schoene” can only be substituted by another noun and is therefore annotated as a noun.

³Since MHG is a language for which a feeling for language is not a reliable criterion due to the lack of native speakers, we rely on the validity judgement of educated German medievalists.

- (15) a. daz schoene wîp (Engl. the pretty woman), or: daz minneclîche wîp (Engl. the lovely woman)
- b. die schoene saz bî ime (Engl. the beauty sits next to him), or: die maget saz bî ime (Engl. the maid sits next to him)

The distinction between determinant and adjective poses difficulties especially for indefinite words such as “manec”, “al” (Engl. many, all). Furthermore, the annotation of words which are in progress of being lexicalized or grammaticalized is difficult given the fact that the POS classes are often changed by this progress (e.g. the old form “sît daz” is annotated as adposition and pronoun since it is not yet lexicalized whereas the form “seitdem” (Engl. since then), the derived modern form from the combination of “sît” and “daz”, is a conjunction or an adverb. Both of them are equivalent in meaning). Since MHG allows the fusion of adjacent words, we extended the tagset by combining tags to annotate merged words such as “weistu” (“weist+du”, Engl. know+you) where a verb fuses with a pronoun. In such cases, the “+” represents the fusion of two or more words and allows it to be decomposed into its individual constituents which can be annotated (e.g. “weistu”: VERB+PRON).

Experiments

We use the manually annotated corpus to train POS models for MHG. To determine the point of deceleration when increasing the amount of data used for training, we subsequently enlarge our training data by 2,000 tokens at a time starting with a training size of 2,000 until we reach the full size of our available data set. In order to ensure that we are not simply capturing an oddity in the relation between the number of training instances and the training algorithm we use, we compare two different existing trainable taggers, which are based on two algorithms namely Decision Trees (DT) (TreeTagger, Schmid (1994)) and Conditional Random Fields (CRF) (Marmot, Müller et al. (2013)).⁴ For both taggers we use their default settings for training to avoid the influence of finding better hyperparameters on our data quantity experiment.

Since there is no lemmatizer for MHG, we solely base our POS taggers on the default features extracted by the implementations of these algorithms from the word surface. These features are character prefixes and suffixes of different length and the word itself. Context windows of width 5 are taken into account by the CRF algorithm and the 2 preceding words by the DT. The learning curves for both classifiers measured in accuracy are displayed in Fig. 6.2.

Results

As expected, the type-token ratio of the training set decreases with the increase in words. Most notably, the performance of the CRF model is consistently significantly⁵ above the performance

⁴We use 5-fold cross-validation for all settings. The test set splits are kept the same throughout the experiments.

⁵According to McNemar’s test using the “mid-p” variant (Fagerland et al., 2013).

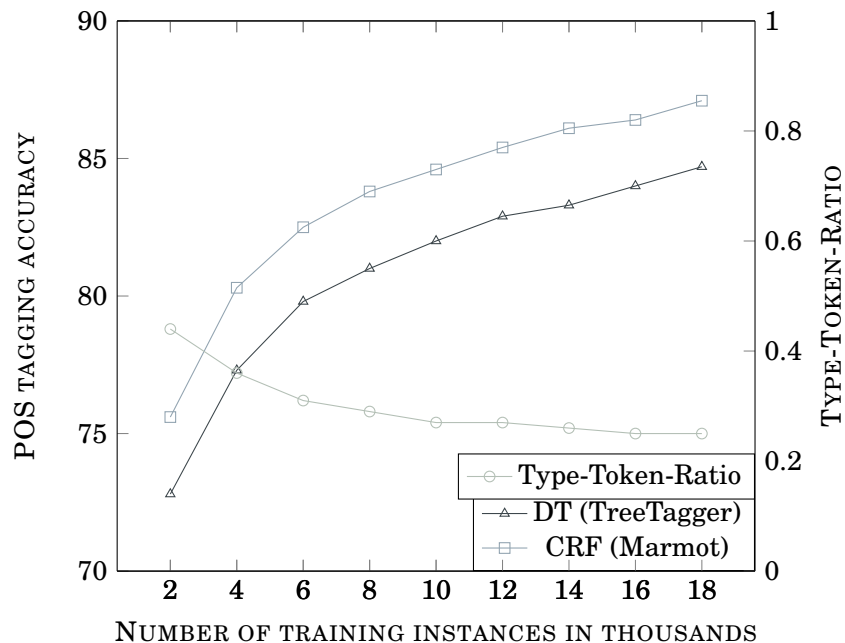


Figure 6.2: Learning curve for both classification algorithms trained on an increasing size of training data.

of the DT model. However, they show a similar learning pattern with respect to increase in training data. The effect of additional data is a steep increase for the first 6,000 tokens. From 12,000 tokens on, the increase of performance drops below 1%. Surprisingly, results of around 80% accuracy can already be reached with a training set size of just 6,000 tokens. This means that starting POS annotation from scratch leads to an expedient model relatively quick. A training set size of 18,000 tokens leads to 84.7% accuracy with TreeTagger and 87.1% with Marmot. Analyzing errors per POS classes, we find that increasing the data from 6,000 to 18,000 words significantly improves tagging accuracy for the smaller, less frequent classes such as particles, numbers and proper nouns but also for some high frequency classes such as nouns.

6.1.3 Additional Data: Exploiting Existing Resources

NLP for historical language has a lot in common with data processing for low resource languages: annotated data is often not, or insufficiently, available and also the amount of digitized raw data is limited. One distinctive feature, however, is the fact that there are often entire fields of research built around historical stages of a language. Even though the research tradition in these fields often focuses on other factors than automatic processing of the data and thus annotations are missing, there are commonly other resources available that can be utilized to substitute them. These resources can e.g. be dictionaries where the vocabulary of a language is listed along with grammatical characteristics. These type-based grammatical annotations can

then be projected onto tokens in context and thus can support POS tagging. Type-supervised POS tagging was first introduced by Merialdo (1994). Smith and Eisner (2005) depart from the assumption of having a complete tagging dictionary as given in Merialdo (1994) by deliberately removing knowledge to simulate a more realistic scenario. In a similar fashion, Goldberg et al. (2008) train a POS tagger for Hebrew. In addition to a lexicon, they assume the availability of a morphological analyzer, a tool typically not available for many historical languages. In Section 6.1.2, we show the influence of the size of high quality, manually labeled training data on the accuracy of the resulting POS tagger. In this subchapter, we investigate the influence of using a larger amount of training data with a reduced annotation quality on the results. We describe an approach departing from a large but not entirely reliable lexical resource as a basis for a POS tagger. We compare our results achieved using a large but qualitatively low resource to the POS tagger introduced in Section 6.1.2.

Data

Tag	Description	Example German	Example English
NOM	Noun	acker, zît	field, time
NAM	Name	Uolrîch, Wiene, Rhîn	Uolrîch, Vienna, Rhine
ADJ	Adjective	grôz, schoene	big, beautiful
ADV	Adverb	schone, schnelleclîche	already, fast
ART	Determiner	der, eine	the, a
DET	Demonstrative Pronoun	ditze, mîn, ieman	this, mine, someone
POS	Possessive Pronoun	mîn, dîn, unser	my, your, our
PRO	Pronoun	ich, es, wir	I, it, we
PRP	Preposition	ûf, zuo, under	on, to, under
NEG	Negation	nie, âne, niht	never, without, not
NUM	Numeral	ein, zwô, zweinzegest	one, two, twentieth
CNJ	Conjunction	als, und, abr	when, and, but
GRA	Graduation Particle	sêre, vil	very, much
IPA	Interrogative Particle	swer, war, wie	whom, where, how
VRB	Verb	liuhten, varn	shine, drive
VEX	Auxiliary Verb	haben, sîn	have, be
VEM	Modal Verb	müezen, suln	must, shall
INJ	Interjection	ahî, owê	ow, oh dear
CPA	Comparative Particle	als, wie	as, like
DIG	Digit	IX, XVII, II	IX, XVII, II

Table 6.2: List of grammatical tags included in the MHDDB along with examples for each category given by Mittelhochdeutsche Begriffsdatenbank (2017). This table is extracted from <http://mhdbdb.sbg.ac.at/help/grammar-tags.de.html>, possible mistakes are not corrected.

The MHDDB is a long-term project with the goal of collecting as many complete MHG texts as possible and making them digitally available and searchable. After already 30 years of work

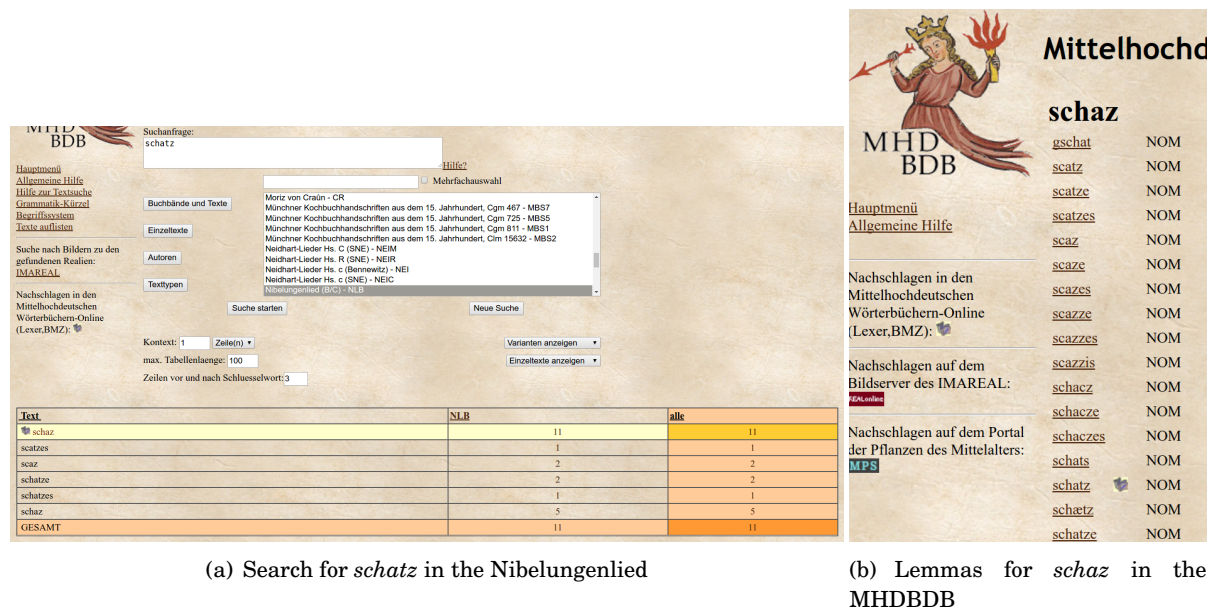


Figure 6.3: Results for the search word *schatz* for the Nibelungenlied at <http://mhdbdb.sbg.ac.at> where statistics for the word forms linked to this search are given together with direct links to the context in the Nibelungenlied.

it offers access to a large number of texts of MHG literature and provides search functions for linguistic or semantic queries. It resembles a comprehensive glossary of MHG with references to the source texts of the words. An example for a search placed at the interface online is displayed in Figure 6.3. The word forms are linked to a lemma entry in the database which contains all related word forms and their grammatical information.

The texts contained in the database cover four centuries (ca. 1100–1500), different dialects of MHG and Early New High German, and a wide range of literary genres (e.g. Arthurian romance, heroic epic, short epic, lyric) as well as non-literary texts (e.g. cookbooks or medical texts). The database encompasses 658 texts which amounts in total to nearly 10 million tokens. The data is tokenized, partly lemmatized and enriched with grammatical categories.

The grammatical categories (cf. Table 6.2) are similar to POS classes, but they cannot be equated for several reasons: Firstly, they are ambiguous and ignore the syntactical context of a token (cf. Example 16), secondly, they are incomplete because they do not cover all possible POS classes a word can be assigned to in different contexts (cf. Example 17), and in addition to this, they include morphological information which linguistically belongs to another level than POS tagging (cf. Example 18).

(16) “ist” (Engl. is): VRB|VEX; the disambiguation by context is missing according to which “ist” is either verb or auxiliary.

(17) “daz” (Engl. the, that): ART|CNJ; the categorization is incomplete since “daz” can also

be a pronoun (relative or demonstrative pronoun).

- (18) “drīvaltīgkeit” (Engl. trinity): NOM | NUM, “ungern” (Engl. unwillingly): NEG | ADV; the categorization includes morphological information such as NEG for the prefix "un".

The approach taken by the MHDBDB illustrated in Example 16 and Example 18 leads to the fact that approximately one half of the tokens (5,029,581 of 9,940,442 tokens) have multiple annotations, which means that they have more than one grammatical tag assigned to them. Furthermore, the data is only partially annotated: 2,823,327 of 9,940,442 tokens are neither lemmatized nor annotated with grammatical categories. Thus, the MHDBDB represents a large resource of MHG data but cannot be used for the task of POS tagging without adjustments since it has been developed for another purpose.

To incorporate the data into the development of a POS tagger for MHG, we take advantage of the manually annotated data described in Section 6.1.2. Since those texts were taken from the MHDBDB, we have a translation of the grammatical categories contained in the MHDBDB to the UD-tagset at our disposal. Some classes such as adverbs or conjunctions allow for a straight forward mapping to UD-tags (ADV >ADV, VEX >AUX, CNJ >CONJ) whereas other classes do not (for example, the tag ADJ can be ADJ or NOUN depending on its syntactical function in the sentence, cf. Example 15). Furthermore, multiple annotations (words have up to 5 grammatical tags) complicate the automatic disambiguation in context. This excludes the possibility to carry out a solely rule-based transfer from the MHDBDB annotations to in-context UD POS tags. The automatic translation technique is described in the following.

Experiments

To arrive at a POS tagger model, two steps are necessary: first, we disambiguate the grammatical categories in the MHDBDB with the help of a classifier trained on our manually annotated data. Subsequently, we can train a POS model on this newly compiled data source.

We start by training a CRF model (Marmot) and a DT model (TreeTagger) on the manually labeled training data introduced in Section 6.1.2 for the disambiguation of the grammatical tags available for the data described in Section 6.1.3. To this end, we extract features from the information contained in the database for all tokens where grammatical information is available. For

Feature	Example
surface form	næhest
2-gram word prefix	næ
3-gram word prefix	næh
2-gram word suffix	st
3-gram word suffix	est
lemma	nâch
2-gram lemma prefix	nâ
3-gram lemma prefix	nâc
2-gram lemma suffix	ch
3-gram lemma suffix	âch
is upper case	False
word length	6
MHDBDB tag1	GRAD
MHDBDB tag2	ADV
MHDBDB tag3	ADJ
MHDBDB2UD tag1	None
MHDBDB2UD tag2	ADV
MHDBDB2UD tag3	ADJ

Table 6.3: Features for the example word “næhest” (Engl. closest) used to disambiguate the grammatical information contained in the MHDBDB in context.

Data	Accuracy CRF	Accuracy DT
setting a	87.1	84.7
setting b	90.9	91.2
setting a+b	90.9	91.1

Table 6.4: Comparison of tagging results achieved by using 20,000 manually annotated tokens (setting a), 10 million semi-automatically annotated tokens (90.7% annotation accuracy, setting b), combination of semi-automatically annotated and manually annotated (scaled up to 10 mio) data (setting a+b).

tokens where annotations are missing, we introduce dummy features. A sequence labeling algorithm is trained with features listed in Table 6.3. The surface-related features can be extracted for every word. The MHDBDB annotation-based features are extracted for all words that have annotations. The number of MHDBDB tags along with their direct mapping to UD tags (regardless of the context) varies between one and five. The resulting disambiguation model is subsequently used to transform the database into a fully (but semi-automatically) annotated corpus. This corpus comprises almost 10 million tokens. However, due to the imperfect results of the disambiguation, this huge corpus has errors in about 9% of all annotations.

We compare the results achieved by the models trained on the manually annotated small training data to the results achieved by training a model on this huge but low quality training data. We use three settings for comparison, as visualized in Fig. 6.4:

- a) we only use the manually annotated corpus (18,000 tokens),
- b) we only use the automatically disambiguated corpus (10 million tokens),
- a+b) we combine the MHDBDB corpus with the manually annotated data scaled up to 10 million tokens (20 million tokens).

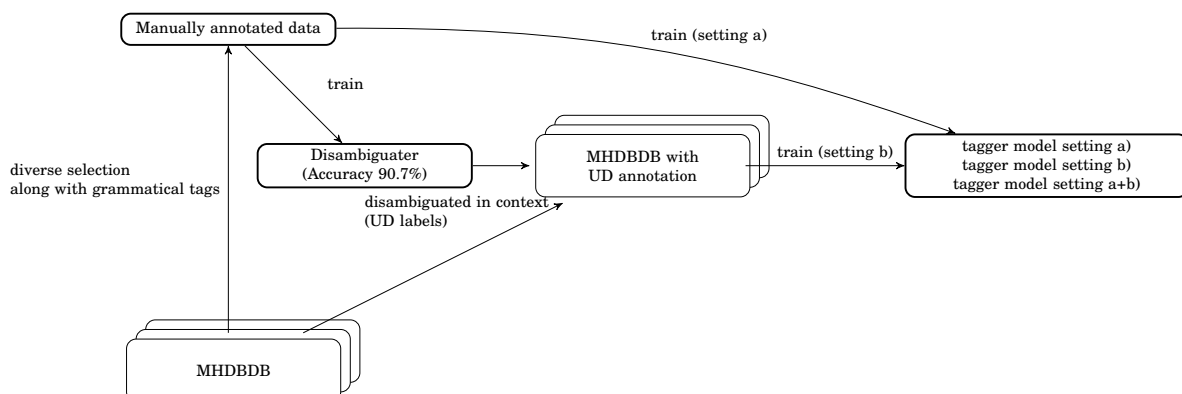


Figure 6.4: Pipeline for incorporating a lexical resource into the development of a POS tagger model

Results

The results are summarized in Table 6.4. These results show that task-external resources can be adapted to support training in a low resource situation. Even though the data quality with a disambiguation accuracy of 90.7% is not perfect, the massive increase in training size helps to improve the tagging performance significantly⁶ over the small but high-quality data set. A significant improvement of about 4% over the model that has been trained on the manually annotated data can be shown. The increase of data has an even more significant impact on the results achieved by training with TreeTagger. We improve the accuracy from 84.7 to 91.2. The differences between the two algorithms observed in our experiences with just little training data are evened out by the number of training examples. Thus, even though the data the models are based on contains annotation errors, we can still improve tagger performance. Especially the performance for nouns, numbers, proper nouns, verbs and adjectives can be improved due to a much wider lexical coverage. However, weighing the manually annotated data as equally strong as the automatically disambiguated data and including it into the training process does not yield better results. The differences between the two classifiers in setting b and setting a+b and within using the same classifier are not significant. In the following experiments, we rely on the DT model trained in setting a+b.

6.1.4 Corpus Middle High German (ReM)

Soon after we had finished our work on the automatic disambiguation of the MHDBDB, the Corpus Middle High German was released in December 2016. The ReM is a huge corpus of MHG texts subsuming four different corpora, comprising about 2.3 million words with several layers of annotations. Next to a *normal form*⁷, it contains lemma annotations as well as POS annotations.

⁶According to McNemar’s test using the “mid-p” variant (Fagerland et al., 2013).

⁷Normal form is a term used by Klein and Dipper (2016) and describes a word form to minimize the differences in spelling and use of diachritica, but do not standardize dialectal varieties (cf. Klein and Dipper (2016, pg. 7,14)).

```

-<token id="t29" trans="mag" type="token">
  <tok_dipl id="t29_d1" trans="mag" utf="mag"/>
  -<tok_anno ascii="mag" id="t29_m1" trans="mag" utf="mag">
    <norm tag="mac"/>
    <lemma tag="mügen"/>
    <lemma_gen tag="mügen"/>
    <lemma_idmwb tag="116097000"/>
    <pos tag="VMFIN"/>
    <pos_gen tag="VM"/>
    <infl tag="Ind.Pres.Sg.1"/>
    <inflClass tag="prpr"/>
    <inflClass_gen tag="prpr"/>
  </tok_anno>
</token>

```

Figure 6.5: Example for the multilayer annotation in the ReM in CoraXML format for the word “mag”.

An example for this multilayer annotation is given in Figure 6.5 in CoraXML format. The tagset used for the POS annotation is HiTS (Dipper et al., 2013). HiTS consists of 84 tags which can – with the exception of punctuation tags – deterministically be mapped to our UD tagset (cf. Table 6.5). The POS tag given in Figure 6.5 is mapped to AUX in the UD tagset.

HiTS	UD
ADJA, ADJD, ADJN	ADJ
ADJS	NOUN
APPO, APPR	ADP
AVD, AVG, AVNEG, AVW	ADV
CARDA, CARDD, CARDN	NUM
CARDS	NOUN
DDA, DDART, DDD, DDN	DET
DDS	PRON
DGA, DGD, DGN	DET
DGS	PRON
DIA, DIART, DID, DIN	DET
DIS	PRON
DNEGA, DNEGD, DNEGN	DET
DNEGS	PRON
DPOSA, DPOSD, DPOSGEN, DPOSN	DET
DPOSS	PRON
DRELS	PRON
DWA, DWD, DWN	DET
DWS	PRON
FM	X
ITJ	INTJ
KOKOM	PART
KON	CONJ
KOUS	SCONJ
NA	NOUN
NE	PROPN
PAVAP, PAVD, PAVG, PAVREL, PAVW	ADV
PG, PI, PNEG, PPER, PRF	PRON
PTKA, PTKANT, PTKINT, PTKNEG, PTKVZ	PART
PTKREL, PW	PRON
VAFIN, VAIMP, VAINF, VAPP, VAPS, VMFIN, VMIMP, VMINF, VMPP, VMPS	AUX
VVFIN, VVIMP, VVINF, VVPP	VERB
VVPS	ADJ
,(SPUNCT, PUNCT

Table 6.5: Direct mapping from HiTS to Universal Dependencies tagset.

Even though the annotation guidelines are mainly in accordance with our own annotation guidelines, we observe differences in annotation decision. In the phrase “die schoenesten unde die besten” (Engl. the most beautiful and the best) we treat the nominalized adjectives as nouns.

In the ReM they are annotated as adjectives.

Experiments

Having a fully annotated corpus with a size of over 2 million tokens, we are curious to compare our results to the results that can be achieved by training a model on the ReM corpus. We use the *normal form* since it is closest to the forms found in edited texts which often serve as a basis for DH research. We train the following models:

- a. train a model on the ReM data with HiTS annotation
- b. train a model on the ReM data mapped to UD annotation⁸

Even though the ReM contains POS annotation, the use of a different tagset makes a direct comparison of results difficult. We cross-evaluate the ReM models and our models on test sets coming from both corpora. We train a model on the original annotation with HiTS (setting a) and map the predictions made for our test set after tagging as well as a model for which we map the ReM data before training (setting b), and predicting and evaluating UD tags for both test sets. The results are summarized in Table 6.6.

model	test ReM	our test set
setting a	84.8	59.4
setting b	89.7	63.2
our model	74.3	91.2

Table 6.6: Cross-evaluation results for experiments with the ReM and the disambiguated MHDBDB.

Training and testing inside of the ReM delivers good results, even though the size of the corpus would suggest even higher accuracy. The results of training and testing on UD tags are higher since the number of tags is crucially smaller. Inspecting the errors found in the sets, we see that the punctuation annotation is problematic for the POS tagger. In the ReM, periods may serve as a sentence delimiter, a verse delimiter or as a comma. Accordingly, a comma can also take over the function of a sentence delimiter. This is an effect of edition practice since medieval manuscripts often lack punctuation and therefore the editor inserts punctuation at his or her own discretion. Thus, it is not particularly surprising that testing on our data the results achieve an accuracy significantly below the one our model achieves. Apart from the punctuation issue, this indicates that there are considerable differences in the annotation decisions even though they are not obvious from reading the annotation manual. In a way, we are faced with a situation as described in Section 6.1.3. This time the data has not been compiled with another purpose in

⁸Following the mapping given in Table 6.5.

mind but with a different understanding of several aspects of POS. In order to determine which these aspects are, we manually annotated 5,114 Tokens from the ReM corpus following our annotation approach outlined in Section 6.1.2. Comparing the automatically mapped subcorpus to our manually annotated corpus shows us the disagreements. The direct mapping from HiTS to UD tags leads to an accuracy of 91.4 when comparing with manually assigned gold labels following our own annotation approach.

An analysis of the disagreements leads to the following insights:

- sentence delimiting punctuations are hard to determine by the algorithm
- in the ReM, Latin is sometimes annotated and sometimes marked with FM for foreign material
- in the ReM, auxiliary verbs are also annotated as auxiliary when they are used as a main verb
- lexicalized adverbs consisting formerly of two adverbs are just tagged with one ADV in our corpus and not with ADV+ADV as in the ReM (“dâmite” (Engl. therefore) annotated as ADV in our corpus, ADV+ADV in the ReM)
- in the ReM, proper nouns are not always annotated as such
- in the ReM, adjectives in a nominal usage are often annotated as adjectives

Informed Tagset Mapping

In order to adjust these disagreements in POS tagging to our understanding of context-dependent annotation, we extend the direct mapping in Table 6.5 by rules to replace the tags in question. Similar to the disambiguation in Section 6.1.3, we thus adjust a resource to fit our needs as a resource for training. This time we resort to a heuristic approach. We include the following rules into the mapping mechanism:

- 1 replace all punctuation marks annotated as sentence delimiting with a period and annotate the tag SPUNCT
- 2 replace all punctuation marks not annotated as sentence delimiting with a comma and annotate the tag PUNCT
- 3 if there is no verb to be found close to an auxiliary⁹, tag an auxiliary with VERB
- 4 “niht” (Engl. not) is always tagged as PART
- 5 lexicalized adverbs consisting formerly of two adverbs are just tagged with one ADV not with a combitag ADV+ADV

⁹The definition of closeness being either a clause or in case of missing sentence delimiters the context window of 10.

- 6 “iè-man” and “niè-man” (Engl. somebody, nobody) are always tagged as PRON
- 7 words with the lemma “wante” (Engl. because), “ouch” (Engl. also) or “doh” (Engl. but) are tagged as CONJ if HiTS tag is KOUS or KO*
- 8 words with a lemma starting with upper case and NOUN tag are tagged as PROP
- 9 words tagged as ADJ that are preceded by a determiner and not preceded by a noun or adjective are tagged as NOUN (nominal adjective)

After these adjustments to the mapping mechanism we could only improve mapping accuracy by a mere 1%. This is mainly due to the fact that an accurate improvement of the agreement of the two annotation methods can only be achieved with the help of a broad understanding of context. With simple heuristics and missing sentence boundaries, in-context corrections are difficult. To test whether this small improvement affects the tagging results, we re-train a model for setting b on the adjusted ReM data following our extended mapping. Moreover, we combine the training sets of the ReM and the MHDBDB and train a combined model (setting c). The results are shown in Table 6.7.

Including heuristics to make the ReM annotation look more like our annotation does indeed boost the performance for almost all settings. The increase in improvement, however, is due to the consistency in punctuation. The combination of the ReM and the MHDBDB data leads to an improvement in accuracy of about 5% on the ReM test set, but does not yield any further improvement on our test set.

model	test ReM	our test set
setting b	92.4	72.8
setting c	89.6	91.2
our model	84.0	91.2

Table 6.7: Cross-evaluation results after improvement of the mapping from HiTS to UD.

We attempt the combination of corpora developed for similar texts, but following different annotation guidelines. Even though the models do not profit from a combination, the cross-evaluation can show annotation inconsistencies and lead to improvement inside of one model (cf. Table 6.6 and Table 6.7 for test ReM). For a successful combination of both resources, the combination of both taggers via stacking could be fruitful as shown by Schulz and Kuhn (2016).

6.1.5 A General Model

By developing a POS tagger for MHG, we close a gap in the preprocessing of Medieval German that will facilitate the further automatic processing of such texts. In the introduction of this article, we described the high degree of diversity of MHG texts, even though it is commonly

Genre or Author	Time of Origin	# Tokens GS	# Tokens Corpus
Arthurian Romance	ca. 1170–1470	2,057	2,559,402
Heroic Epic	ca. 1200–1400	1,984	967,458
Short verse narratives (“Mären”)	ca. 1220–1460	1,980	355,916
“Minnesang” (“Minnesangs Frühling”)	ca. 1150–1230	2,034	61,731
Hartmann von Aue	ca. 1180–1200	1,285	140,239
Wolfram von Eschenbach	ca. 1200–1220	1,237	247,309
Hessian	ca. 1165–1300	1,590	129,827
Middle Lower German	ca. 1170–1300	1,466	64,753

Table 6.8: Overview of the subcorpora annotated along with the time of origin, number of tokens in the gold standards and number of tokens in the subcorpora which are used for training specific POS models.

understood as one stage of the language. The question arises as to whether such a diverse language can be served well by just one model. In the following, we evaluate the performance of the best model described in Section 6.1.3 on subcorpora of different genres and authors. We show that the heterogeneity of the training corpus compiled from the MHDBDB leads to a generally applicable model.

Subcorpora

To cover important genres and significant authors of MHG, we compile corpora for a variety of genres following the classification of the MHDBDB. They differ in heterogeneity and quantity. Even though a thematic consistency is given, the Arthurian romance genre represents the most heterogeneous corpus, embracing texts from the early 13th century to the end of the 15th century written in rhyming couplets as well as in prose. The heroic epics (1200–1400) constitute the second largest genre-corpus and are more homogeneous regarding their form (stanzaic). The subcorpus of short verse narratives is smaller but comprises the highest number of single texts. Since the texts were mostly transmitted anonymously, deal with different topics and originate from different time periods, the corpus exhibits a relatively high diversity. In contrast, the “Minnesang” (including only the texts edited in Moser (1977)) is characterized by homogeneity given the fact that it is a lyric genre in strophic form with a restricted vocabulary, dealing with one and the same topic and comprising songs from a limited time period (1150–1230). In addition to the genre-specific corpora, we create two author-specific corpora consisting of the epic and lyric texts of Hartmann von Aue and of Wolfram von Eschenbach. The author-specific subcorpora do not contain dialectal and temporal linguistic varieties but do embrace different genres.

Another part of the diversity of MHG is based on linguistics such as dialectal varieties. The MHG speaking area is separated from the Middle Low German language area and furthermore subdivided into various dialects of MHG. To evaluate the performance of the tagger on linguistic varieties, we annotate two supplemental subcorpora: a corpus with texts from the Hessian

and Thuringian area containing texts in a dialect of MHG, and a corpus with texts from the Middle Lower German speaking area which are thus linguistically more distant from the MHG language. Those two corpora represent two of the subcorpora of the ReM (Reference Corpus of Middle High German¹⁰) (Klein and Dipper, 2016, p. 2). We will henceforth call them region-specific corpora¹¹. Thus, we do not only evaluate the performance of the POS tagger on different linguistic varieties and language areas but also test its applicability to “unknown” data. All subcorpora along with the number of tokens and the covered time periods are summarized in Table 6.8

Results

Evaluating the applicability of the POS tagger for MHG reveals that the model trained in setting a+b described in Section 6.1.3 using DT performs well throughout all genres and for the author-specific subcorpora covered by the MHDBDB, whereas the results for the region-specific subcorpora are much shorter of these for other subcorpora (cf. Fig. 6.6). This is on the one hand due to the fact that these corpora contain dialect forms that are not included in the MHDBDB (e.g. “niet” instead of “niht” (engl not); “sal” instead of “sol”, (Engl. should)). On the other hand, the region-specific corpora show a significantly higher number of fusions of words like “wandeer” (wande+er CONJ+PRON, Engl. because+he) which generally perform badly since they allow for numerous tag combinations. Besides this, we found minor reasons such as unknown characters such as verse delimiters that were not included in the MHDBDB which could not be labeled correctly. Moreover, the data lacks most of the punctuation marks, which leads to extremely long sentences and makes it difficult to identify syntactical units.

6.1.6 Region-Specific Corpora

The performance of the tagger on the region-specific corpora lags behind the performance on the other texts. We use these corpora as an example for specialized subcorpora, which share a considerable number of features with MHG but differ in other key characteristics due to their dialectal features. To give an example for how to improve tagging accuracy for text sorts related to – but not directly included in – the MHDBDB, we address domain adaptation as a solution and experiment with weakly-supervised learning.

Domain adaptation

Domain adaptation aims at extending a classifier’s abilities from a known source domain to a new, unseen target domain. The source domain usually provides reliably annotated data. A lot of work has been done on domain-adaptation in NLP e.g. Ben-David et al. (2010), Daume III

¹⁰<https://www.linguistics.rub.de/rem/>, 19/06/2017.

¹¹For more information on geographical deviation of the speaking areas compare Hennings (2003, p. 18–20)

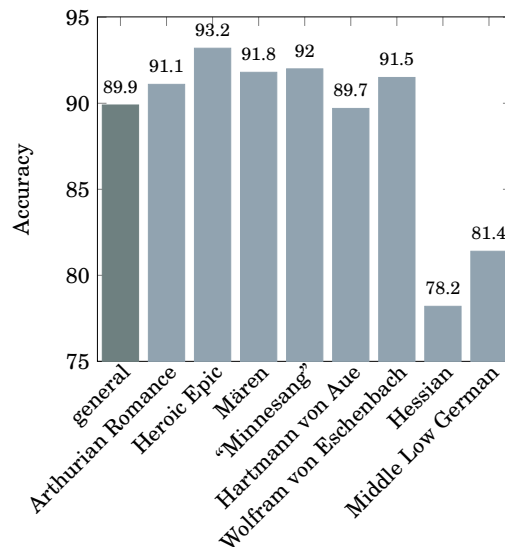


Figure 6.6: Accuracies achieved by the general POS tagger model on the genre-specific, author-specific and region-specific subcorpora.

(2007). The transfer to the new domain can be done with an emphasis on different levels, e.g. the feature space or the data set composition.

One strategy is the **transfer of feature knowledge** to the target domain. Blitzer et al. (2006) emphasize features equally important in source and target space in their structural correspondence learning approach. Another technique is described in Jiang and Zhai (2007) who take different distributions of instances and classification functions in the source and the target domains into account and exploit knowledge about the instances of the target domain to adapt to it. The most successful method reported for historical languages is *Feature Embeddings for domain Adaptation* (FEMA) suggested by Yang and Eisenstein (2015). They propose to learn domain-invariant properties of features from feature embeddings. These robust representations are learned from the combined source and target domain data using the skip-gram model introduced by Mikolov et al. (2013). Yang and Eisenstein (2016) report improvements for POS tagging of historical Portuguese and historical English.

Another factor is the **composition of the data set** used for training a domain-specific classifier. Commonly, the labeled data available in the source domain is extended by unlabeled data from the target domain. There are different techniques to perform this extension. Diverse self-training algorithms were introduced for different problems in NLP. One of the earliest is the Yarowsky algorithm (Yarowsky, 1995). Given a labeled data set in the source domain, it can be iteratively extended with unlabeled data from the target domain using different strategies to arrive at a data set which represents the target domain well enough to train a model on it. This way, the classifier bootstraps itself. Over the years, more elaborate techniques have been developed to integrate unlabeled data with labeled data, e.g. tritraining, first introduced by Zhou and

Li (2005) and successfully applied to POS tagging by e.g. Søgaard (2010).

Error-driven autocorrection

All of the above mentioned domain adaptation approaches rely on the availability of labeled source data. In a DH context this is a serious disadvantage since data that has been used for training a model is not always available due to copy right issues for example. Therefore, we introduce a technique in which we merely use the model trained on source data to tag unlabeled data. Subsequently, we learn to autocorrect the predicted labels from a very small sample of labeled data in the target domain and train a classifier on the automatically labeled and corrected target data. We iterate these steps until there is no improvement on the development set anymore. We call this approach **error-driven autocorrection**. This weakly supervised self-learning approach can be easily applied in a DH context constituting an automated active learning approach. The autocorrection is implemented as follows: We utilize the corpus-specific error distributions for each POS class learned from a small development set¹², thus a small annotated portion for each region-specific subcorpus. To use these distributions for improvement, we compile a confusion matrix containing all parts-of-speech along with the classes they are confused with frequently. Since POS tagging is context-dependent, we add statistical knowledge about highly frequent POS trigram patterns found in the development data. We label our unlabeled region-specific data with the general TreeTagger model and request the confidence values the model returns. We replace POS tags with a low confidence score (lower than 0.35) in the automatically tagged region-specific corpus. The POS tag with which the low confidence tag gets replaced is chosen based on two criteria: a) it is among one of the three most often confused POS tags for this tag in our development data, b) we choose the one tag out of those three tags that has the highest POS trigram count in our development if it is inserted at the position of the low confidence tag. Subsequently, we train a tagger model on the region-specific data. Note that there is no need for access to the labeled data from the source domain but merely the tagger model trained on labeled source data. This difference is crucial in comparison with the domain adaptation approach described further above, since it obviates the assumption that labeled data in a source domain is available. Example 19 illustrates the correction approach.

- (19) a. jâ wile ich dîner lêre vile gerne volgen
 yea want I your teaching much gladly follow
 ‘indeed I want to follow your teaching very gladly’
- b. PART_0.75 NOUN_1.0 PRON_0.99 DET_0.99 **NOUN_1.0** VERB_0.33 **ADV_0.98**
 VERB_0.99
- c. VERB is most often confused with AUX, ADV, ADJ

¹²Development set sizes for Hessian 1,072 tokens and for Middle Low German 1,007 tokens.

- d. **NOUN AUX ADV** and **NOUN ADJ ADV** is not found as a trigram sequence in the development data
- e. **NOUN ADV ADV** appears in the development data
- f. *jâ*.PART *wile*.NOUN *ich*.PRON *dîner*.DET *lêre*.NOUN ***vile***.ADV *gerne*.ADV *volgen*.VERB

The word *vile* is highly ambiguous in MHG and belongs to different POS classes depending on the context. Therefore, the confidence of the tagger is often rather low for this word. In this example, it is mistakenly labeled as verb as shown in 19b with a low confidence of 0.33. The label VERB is often incorrectly given to auxiliary verbs, adverbs and adjectives. We extract this information from the region-specific development set. To choose which one of those labels to select for correction, we access the POS trigram counts in the development set and pick the label for the word for which the trigram counts are highest, thus for which the context of the incorrectly labeled word supports the correction decision. In Example 19, the only trigram found in the development set is NOUN ADV ADV. We therefore replace the label VERB with the label ADV. The automatically corrected tagging is shown in 19f. Note that the word *wile* is incorrectly labeled as noun in this sentence as well. However, since the TreeTagger confidence lies above 0.35 for this word, our autocorrection approach will not change the label.

This approach is clearly a simple approximation of a correction in context and therefore makes wrong correction decisions as shown in Example 20. Even though it is correct in the assumption that the POS sequence ADV ADV VERB is valid, in this context it represents the less probable reading of the sentence, namely that the action of being born was performed in a noble manner. The second likely correction pattern ADV ADJ VERB is the correct labeling in this context.

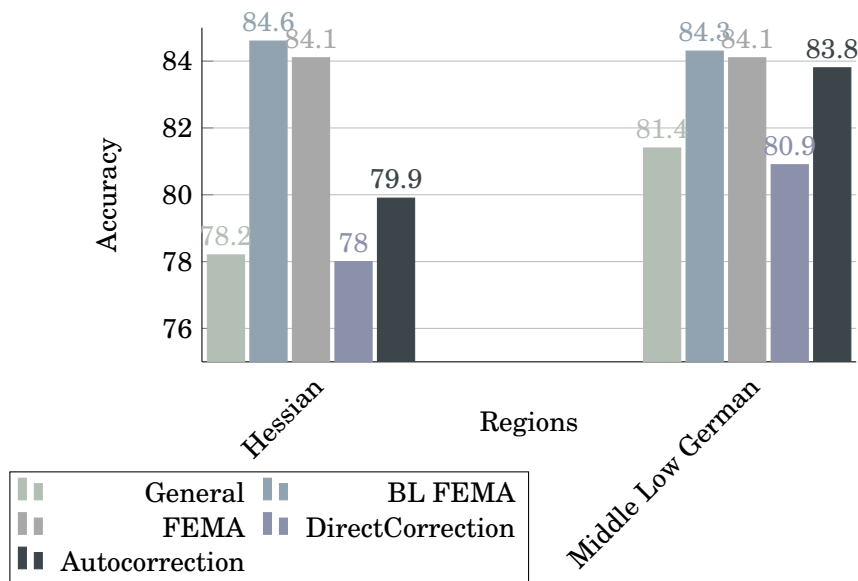
- (20) a. *newart nie keiser sô hêre geboren an der erde*
never was never emperor such noble born on the earth
‘there has never been born such a noble emperor on earth’
- b. VERB_0.92 ADV_0.99 NOUN_0.99 **ADV_1.0** VERB_0.32 **VERB_0.68** ADP_0.98 DET_1.0
NOUN_1.0
- c. VERB is most often confused with AUX, ADV, ADJ
- d. rank ADV AUX VERB rank 345 (improbable trigram)
- e. rank ADV ADV VERB rank 18 (most probable trigram)
- f. rank ADV ADJ VERB rank 84
- g. *never was*.VERB *never*.ADV *emperor*.NOUN *such*.ADV ***noble***.ADV *born*.VERB *on*.ADP
the.DET *earth*.NOUN

The main problem of domain-adaptation approaches relying on an extension of the training set is that they are based on sentence selection. However, the region-specific corpora lack considerably many sentence delimiters. We thus decide to experiment with a technique relying on

extrapolation of the feature space. We compare domain-adaptation in the flavor of FEMA (Yang and Eisenstein, 2015) and results of our error-driven autocorrection approach. Following Yang and Eisenstein (2015) and Ratnaparkhi (1996), a baseline is trained on three broad types of templates: five lexical feature templates, eight affix feature templates, and three orthographic feature templates using DT. Due to computational limitations, we use only half of our automatically disambiguated 10-million-token corpus described in Section 6.1.3 as labeled data and unlabeled data with corpus sizes of 129,827 for Hessian and 64,753 for Middle Low German.

Results

We evaluate FEMA in comparison to a baseline (BL FEMA) where the same features are used but without feature embeddings. This way, we get a realistic impression of how much the domain-adaptation technique contributes to the results. In addition, we compare the results achieved with the model described in Section 6.1.3 (General) and the results of our error-distribution-driven autocorrection (Autocorrection). For comparison, we apply this automatic error-correction directly to the test set (DirectCorrection) without iterative training of a model on auto-corrected unlabeled data to see whether the correction improves results.



The results for FEMA as well as for Autocorrection are significantly¹³ above the results achieved with the general model. We improve the tagging performance from 78.2% achieved by our general model on Hessian to 79.9% and from 81.4% to 83.8% on the Middle Low German sub-corpus applying error-driven autocorrection. Error-correction applied directly to the test set (DirectCorrection) impairs results. The reasons for improvement of FEMA to 84.1 for Hessian and 84.1 for Middle Low German, however, are not referable to the domain-adaptation via feature embeddings but rather the set of initial features used to train the DT suggested by Ratnaparkhi

¹³According to McNemar’s test using the “mid-p” variant (Fagerland et al., 2013).

(1996). This becomes evident comparing FEMA to its baseline. Thus, for our data adding feature embeddings does not show the effect described in Yang and Eisenstein (2016). This might be due to the fact that the variety is too close to the source language. Adding more features nevertheless improves results. Domain-adaptation requires the availability of the training data for the general model, whereas error-driven autocorrection relies on a very small development set to learn error distributions from the unlabeled target data and can thus be applied in scenarios in which source data is not accessible.

6.1.7 Summary

By illuminating the case of MHG, we demonstrate how to deal with a low resource, diverse and non-standard language in the domain of NLP. We investigate various aspects of POS tagging in a non-standard data context. Starting from manual annotation of data, we evaluate how much annotated data is needed to achieve acceptable results in the performance of a POS model. Surprisingly, we find that around 6,000 tokens are enough to reach an accuracy of 80 % and that the learning effect decelerates after around 12,000 tokens.

Another possibility to close the gap in data quantity in a low resource situation is the adaptation of existing data resources by adapting their annotation schemes. Departing from a large lexical resource, the MHDBDB, we show how to incorporate task-external annotated data via a disambiguation algorithm in the development process of a POS tagger. Moreover, it becomes clear that a large but qualitatively low resource achieves better results than a small, qualitatively high data set. Thus, we can conclude that data quantity – though not perfect – at a certain point outweighs the high quality of manually annotated data. Also, the choice of algorithm (CRF vs. DT), which had a relatively high influence on the small data set, loses its influence when working with a large amount of data. The example of the ReM shows how to incorporate another annotated resource even though it follows other annotation guidelines. By using an informed mapping from one tagset to the other we can make the two corpora more compatible.

Since the texts included in the MHDBDB cover different stages of the language (MHG and Early New High German), dialectal varieties as well as literary and non-literary genres, we can benefit from the heterogeneity of training data and develop a nearly “general” POS tagger for MHG. After having tested its applicability to different subcorpora, we found that the tagger performs well on the genre-specific and author-specific corpora taken from the MHDBDB, whereas the performance on the region-specific texts (Hessian and Middle Lower German) extracted from the ReM lags behind the others. We suggest a new weakly supervised technique based on error-driven autocorrection which can improve results on specific corpora. Experiments with FEMA domain-adaptation did not improve results but highlight the importance of feature selection.

Since the MHG has a lot in common with other non-standard and low resource languages, we plan on transferring our approaches to other historical languages which suffer from data-sparsity and thereby aim to contribute to solving the problem of lacking data in the context of

historical language data processing.

With the development of a POS tagger and the release of ReM, MHG moved one step further towards constituting a standard form by itself. In the next subchapter, we investigate a text for which MHG is considered one possible standard form. In order to guarantee sustainable retrievability, we submitted the best model for MHG POS tagging to the Clarin-D repository¹⁴. This ensures that the model as well as the metadata remain permanently findable.

¹⁴Clarin-D repository, metadata handle: <http://hdl.handle.net/11022/1007-0000-0001-877B-D>, landing page of TreeTagger where the model can be found: <http://hdl.handle.net/11022/1007-0000-0000-8E4D-B>.

6.2 Finding the Right Method

After we have investigated the influence of data quality and quantity on tagging results, we focus on training techniques in this subchapter. Various approaches have been applied to solve the task of automatic POS tagging and POS tagging in a low resource context. These approaches make assumptions such as the availability of resources for related languages or disposability of task-foreign resources as shown in the previous subchapter. These assumptions, however, are not always met.

To illustrate this, we investigate POS tagging of a unique late MHG text in the transition period between MHG and Early New High German (ENHG). This leads to a text with mixed features of two historical stages of German. *Apollonius von Tyrland* written by Heinrich von Neustadt (HvN) in the late 13th century is a translated text that shows an interesting relationship to its source text, a Latin original. HvN is suspected to have incorporated other sources into the translation of the text from Latin to German. An interesting question for medieval scholars is the verification of assumptions regarding a segmentation of this text into parts supposedly tracking back to different sources. In order to support this investigation with digital methods, the enrichment with linguistic features such as part-of-speech information for a detailed analysis of features related to content and to style seems promising.

We compare different approaches towards boosting performance of POS tagging of this text for which no suitable POS tagger is available and for which there is no or really limited annotated data. Departing from the assumption that we have no text-external resources at our disposal, we experiment with unsupervised and weakly supervised learning methods. Moreover, we follow experiments performed by Garrette and Baldrige (2013) who describe POS tagging research for low resource languages using really small amounts of annotated data. Unlike in Section 6.1.3 where we included task-foreign *lexical resources*, in this subchapter we include *tools* that have been developed for related languages into our experiments. The aim of this study is to evaluate the performance of POS tagging considering different supply conditions of data and related external resources utilized in different learning techniques. Moreover, we experiment with a variety of algorithms.

Our findings can serve as a reference point for DH projects dealing with non-standard data and offer a roadmap on how to approach text processing in similar contexts. We strive for a better idea of how one can gain performance. The approximation of the obtainable gain in proportion to the spent effort is an important consideration given that those texts often have very specific characteristics and developed resources might not be reused in another context.



Publication

Parts of this subchapter were published in Schulz and Kuhn (2016).

6.2.1 Related Work

Completely **unsupervised** POS tagging is still in its very early stages. Biemann (2006) relies on a graph clustering method. Unlike in current state-of-the-art approaches, the kind and number of different tags are generated by the method itself. Contrary to this, Haghghi and Klein (2006) use distributional prototypes in the learning process of their log-linear model. This way they inform the algorithm indirectly about the POS classes. These unsupervised or semi-supervised approaches make use of distributional semantics (Turian et al., 2010). In this context, the use of word embeddings has to be mentioned. Their ability to capture syntactic and semantic regularities (Mikolov et al., 2013) can be utilized to compensate for the high number of hapax legomena in sparse data by concentrating on the similarities of contexts in which they might appear. Word embeddings have been used by Lin et al. (2015) for unsupervised POS induction.

Weakly supervised techniques can involve supervision of different degrees and of different kinds. There are approaches using parallel data like Moon and Baldrige (2007) who use aligned text to compensate for the lack of annotated data in the language under investigation. Sánchez-Martínez et al. (2007) unsupervisedly train an HMM-based Occitan POS tagger used within an MT system using translation probabilities of tag assignments to inform the HMM. Agic et al. (2015) introduce an approach using the bible as a parallel corpus aggregating over the tags from annotated languages. This way, they train POS taggers for 100 languages such as Cakchiquel and Akawaio. Das and Petrov (2011) locate their approach on the unsupervised side, however, they use translated text in a resource-rich language for cross-lingual knowledge transfer. Several other approaches utilize lexicons providing the learning algorithm with possible valid POS for a part of the vocabulary (Ravi and Knight, 2009). Garrette and Baldrige (2013) show that there is no need for huge annotated corpora but that reasonable results can be achieved by generalizing from just a little amount of annotated data.

Moreover, POS taggers developed for closely-related languages can be applied as done in Zeman and Resnik (2008). This requires a proper mapping from one tag set to another.

In the field of low resource language processing, not just parallel data of closely-related languages is used, but the task is often tackled as domain-adaptation of tools developed for a related language. Blitzer et al. (2006) introduce structural correspondence learning for domain adaptation from newspaper text to the biomedical domain also for the setting when there is no labeled data from the target domain.

Being confronted with a diversity of methods to tackle POS tagging for underresourced languages, we investigate those being feasible regarding our data situation. Therefore, we focus on weak supervision following Garrette and Baldrige (2013), the unsupervised approach by Biemann (2006), model transfer similar to Zeman and Resnik (2008) and explore the opportunities that word embeddings (Mikolov et al., 2013) and combinations of methods hold.

set	# sentences	av. # tokens
train	100	1374
dev	100	1372
test	50	688

Table 6.9: Average number of sentences and tokens in train, development and test set of our gold standard.

Data

As introduced in Section 6.1, MHG texts are characterized by their high degree of diversity with respect to graphematic realization and choice of vocabulary (Dipper, 2010). Depending on the exact period and point of origin, the author and even the printer, a text may or may not be readable even for native speakers of modern German. In fact, even though MHG constitutes an early phase of nowadays German, it differs significantly with respect to different linguistic features. These characteristics make it impossible to directly use any off-the-shelf tool for automatic processing of this kind of text and moreover complicate the development of domain specific tools. We work on Heinrich von Neustadt’s *Apollonius von Tyrland*¹⁵, a 20,645 verses long opus containing approximately 180,000 types and 800,000 tokens. Heinrich von Neustadt lived in the 13th century and just two writings can be attributed to him, the other one being *Gottes Zukunft*. Considering these two texts as an independent text domain, this leaves us with a quite limited amount of data. Moreover, the language he uses can be located in an intermediate phase between MHG and ENHG. This is crucial to know since this means that neither tools developed for MHG (Dipper (2010); Bollmann (2013), the tagger introduced in the previous subchapter) nor tools for standard German will work reliably. However, its relative closeness to both can nevertheless be beneficial.

We annotated 250 sentences comprising 3625 tokens with Universal Dependency POS tags as described in Section 6.1.2. We use train and development sets of 100 sentences each since development will be used for training in some scenarios and a test set of 50 sentences (Table 6.9).

In the first phase of our experimentation we are evaluating different techniques using nothing but the text at hand. We call this learning from within. We use unsupervised methods as well as weakly supervised techniques. This scenario covers the lower right area in Figure 6.7. In the next sequence we will treat the text as any historical text for which we assume a closeness to other stages of the language. We utilize tools that have been developed for those stages in order to boost performance. Accordingly, we move up to the region covered by historical language stages in Figure 6.7 and the degree of supervision increases. We will refer to these techniques as text-external resource learning.

¹⁵Based on the Gotha manuscript edited by Samuel Singer, Berlin 1906. Digitalized version from <http://www.mhgta.uni-trier.de> (Gärtner, 2002).

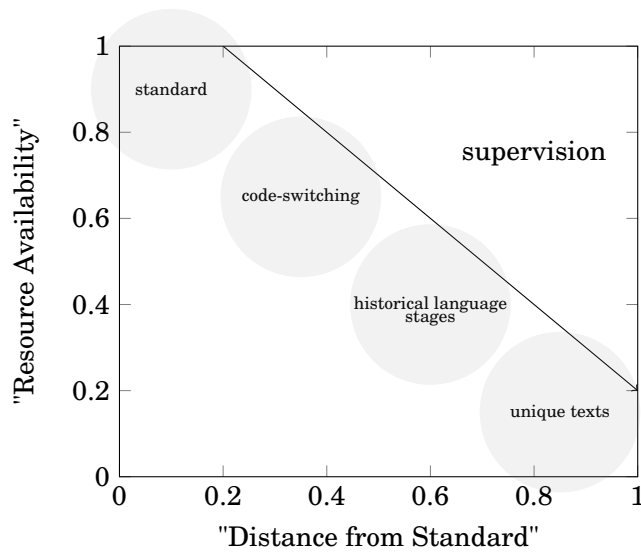


Figure 6.7: Dimensions of non-standard text

6.2.2 Learning from Within the Text

Training a tagger from scratch, we are confronted with the issue of extreme data sparsity. Different from a low resourced language, our text at hand provides us with just some thousand sentences in total (including a high number of hapax legomena) and considerably less annotated data. Thus, abstraction from the surface form is preferable. In the context of language modeling with the help of neural networks, it has been shown helpful to train what are known as word embeddings (e.g., Mikolov et al. (2013), Lebret and Lebret (2013)). These embeddings are high dimensional vectors representing features of words in a high feature space and are able to capture syntactic and semantic regularities (Mikolov et al., 2013). These characteristics make them a good departure point for a scenario in which one faces data sparsity.

We train 64-dimensional word embeddings using word2embeddings (Al-Rfou et al., 2013) and a window size of five tokens on our entire corpus. Although this oversteps the clear division between training and test data because those vectors summarize the context of the words in the entire corpus, we consider this a valid approach since we can assume the same treatment during application to the rest of the corpus. Moreover, we do not claim generalizability of our tagger to other data but are driven by the goal to tag in-domain text.

Word embeddings are used as a way to abstract from surface form in two of our approaches: in an unsupervised clustering approach and for training a multilayer perceptron neural net (MNN). We compare these approaches to a sequence labeling approach (CRF) (Lafferty et al., 2001) using only surface forms. To compare the performance of different neural net architectures, we additionally experiment with a long short-term memory (LSTM) neural net. Moreover, we investigate self-learning for the MNN and the CRF training aiming at further improvement.

K-Means Clustering

We experiment with **k-means clustering** informing the cluster analysis (CA) algorithm with the number of POS classes we have annotated in our gold standard. Moreover, we initialize our cluster centroids with prototypical words from the training data for each POS inspired by Haghighi and Klein (2006).¹⁶ This rather simple approach does not take the sequence in which words appear in the text into account but relies only on the context information encoded in the word embeddings that serve as features to locate each word in a multidimensional space. This means that each token can only be assigned to one POS.

Neural Networks

Neural networks are known for their success in many NLP applications. However, one characteristic emphasized is their ability to learn patterns from huge numbers of labeled instances. We have only a small number of training instances at our disposal, but nevertheless aim to evaluate the performance that can be reached with a neural approach. We train both a **multi-layer perceptron (MLP) neural net** using *nlpnet* (Fonseca et al., 2013) and an **(LSTM) neural net** using an integrated compositional character to word (C2W) model based on a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) using the *Java Neural Network (JNN) Toolkit* (Ling et al., 2015).

As an input to the **MLP**, we use the 64-dimensional word embeddings described further above. The architecture is relatively simple. We follow parameter settings that are reported as successful by Fonseca et al. (2013). We use 100 hidden neurons, a learning rate of 0.01 and hyperbolic tangent for activation.

The C2W technique of the used **LSTM** can be beneficial in case of a high number of OOV words, since character-level similarities between words can be utilized. Even though the JNN toolkit allows to extend the feature space by additional features, we do not add information on the words, suffixes and prefixes to the training process but rather rely on the C2W method. The bidirectional setting enables a notion of context. We use a learning rate of 0.01.

A comparison of the performance of these two architectures is interesting since LSTMs are known to capture long term dependencies and could therefore perform better in learning the structure of sentences.

Conditional Random Field

We train a **CRF** tagger (Lafferty et al., 2001) using a context window of 5 tokens and 6 features. We include the following features for each token:

- token is punctuation or not

¹⁶This also facilitates the evaluation because clusters can be mapped to POS more easily.

- word length
- character prefix of length 2
- character prefix of length 3
- character suffix of length 2
- character suffix of length 3

Self-Learning

Self-learning or self-training algorithms are bootstrapping methods with the goal to achieve improved performance of a supervised algorithm by employing different strategies to incorporate unlabeled data into an iterative training process (cf. e.g. (Mihalcea, 2004)). With the intention to overcome the sparsity of training data, we apply self-learning. We tag the unannotated part of our corpus with the CRF tagger and the neural net tagger, respectively. Subsequently, we sort the automatically tagged sentences by tagging confidence (Viterbi scores for the neural net and the conditional probability for the CRF) and add the best 200 sentences to our training data and retrain the tagger. We evaluate the performance before and after extension of the training data on the development set. In case the performance increases after extension, we keep the new classifier and start the next iteration by tagging the unannotated data anew. In case the performance decreases, we discard the new classifier and append the next 200 sentences of the automatically tagged data. This way we extend our training set by an average of 6 times¹⁷ for neural net training. Surprisingly, we cannot improve the CRF tagger. To make sure that the batch size of 200 sentences is not too big, we experiment with 100, 50 and 1. However, we consistently experience a decrease in performance even when just adding one automatically tagged sentence from our raw corpus to the training data.

6.2.3 Stretching Out: Including Text-External Resources

Following the assumption that closely-related languages have similar features, applying taggers trained for those languages is promising. We use the TreeTagger for German (Schmid, 1994) and the TreeTagger model described in Section 6.1.3. Both, New High German and Middle High German, share a considerable number of characteristics with the Apollonius text. We map the STTS (Schiller et al., 1995) to the UD tagset.

Suspecting that different models have different strengths, we use the meta-learning method of stacking (Wolpert, 1992) to combine these advantages. We use the predictions of the weakly-supervised CRF classifier and the neural net classifier along with the predictions of the tree tagger models for MHG and NHG on the development for training a meta-learner. The meta-learner we use is a CRF classifier (Lafferty et al., 2001).

Moreover, we implement tritraining (Zhou and Li, 2005).

¹⁷In randomized sub-sampling setting.

STTS tags	UD tags
ADJA, ADJD	ADJ
ADV, PAV, PWAV	ADV
APPO, APPR, APPRART, APZR	ADP
ART, PDAT, PIAT, PIDAT, PPOSAT	DET
CARD	NUM
FM	X
ITJ	INTJ
KOKOM, KON	CONJ
KOUI, KOUS	SCONJ
NE	PROP
NN	NOUN
PDS, PIS, PPER, PPOSS, PRELAT, PRF, PWAT, PWS	PRON
PTK, PTKZU, PTKNEG, PTKVZ, PTKANT, PTKA	PART
TRUNC, XY	X
VVFIN, VVIMP, VVINF, VVIZU, VVPP	VERB
VAFIN, VAIMP, VAINF, VAPP, VMFIN, VMINF, VMPP	AUX
\$. \$., \$(PUNCT

Table 6.10: Mapping between STTS and Universal Dependency POS tags.

Cross-lingual model utilization

Working on a text with characteristics from both NHG and MHG, we use a tagger built for German and MHG respectively for our data. An issue arising from this otherwise simple approach of applying the model of a related language to another is the mapping of the tag sets. This process of mapping one tag set to the other is accompanied by a loss of information considering that even though languages might be related, they rarely cover exactly the same space of grammatical features. A solution is offered by language-independent tagsets such as the Universal Tagset (Petrov et al., 2012) or the Universal Dependency POS tagset (Nivre et al., 2015) which model POS on a level on which cross-lingual differences can be subsumed by a common tagset. Since the MHG tagger has been trained on UD tags already, the mapping issue only remains for the NHG model which has been trained on the STTS tagset. The mapping from a fine to a coarse tagset as in the case of NHG to UD tags is rather unproblematic as intended by these cross-linguistic annotation initiatives. The mapping from STTS to UD tags is described in Table 6.10.

Ensemble Learning

Ensemble learning is an approach in machine learning where the knowledge of multiple algorithms is employed to obtain better predictions. The basic idea is the combination of complementary strengths of different classifiers. We combine the knowledge of the MNN tagger, the CRF classifier introduced in Section 6.2.2 and the two taggers for closely-related languages. We implement three strategies: stacking (Wolpert, 1992) using a CRF meta-learner, voting and tritraining (Zhou and Li, 2005). The meta-learners base their decision upon the POS tags at-

tributed by each of the four taggers. The simplest technique is the unweighted majority voting approach (Boyer and Moore, 1991) in which we decide for the POS that has been voted dominantly by the classifiers in the ensemble. The stacking approach uses the surface form of the word and a context window of 5 over all the classifiers' predictions. We train the meta-classifier on the labeled development set. As another instance of self-learning and in this form an ensemble learning method this time using external classifiers, we use tritraining (Zhou and Li, 2005). We use two classifiers, our external taggers for MHG and the CRF tagger, to inform our third classifier about which sentence from the unlabeled data set to add to the training process. For this decision, we choose simple agreement of both classifiers on sentence level. We add all sentences labeled by our algorithm that have not more than one differently tagged word.

6.2.4 Evaluation

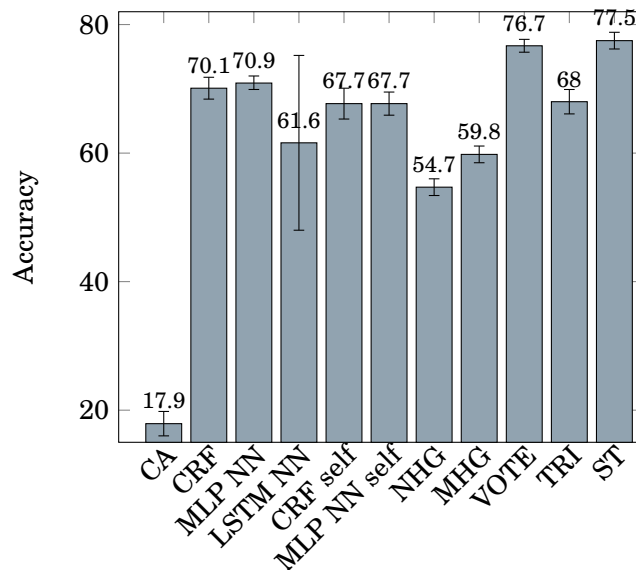


Figure 6.8: Accuracies of all POS tagging approaches evaluated in a 10-fold Monte Carlo cross-validation setting along with the standard deviation of the accuracy values for the 10 samples are reported. Accuracy is given on the y-axis. The experiments are sorted by their increasing use of external resources and combination of classifiers: clustering (CA), conditional random fields classifier (CRF), MLP neural net (MLP NN), LSTM neural net (LSTM NN), MNN self-learning (MLP NN self) and CRF self-learning (CRF self) represent the experiments that use only text internal knowledge. On the right-hand side the results for experiments with external resources are listed in the following order: model transfer from New High German (NHG) and Middle High German (MHG), tritraining (TRI), majority voting (VOTE) and stacking (ST).

Results

It is a challenge to evaluate the clustering performance and not a combination of clustering and mapping induction to the POS classes. Moreover, evaluation on a gold standard for POS tagging seems counter-intuitive given that the clustering is not informed about the task at hand. Vlachos (2011) advocates the evaluation as clustering-based word representation induction. Extrinsic evaluation is suggested as a solution to this problem. Having all these drawbacks in mind, we evaluate the overlap between the clustering results and the gold standard data without drawing strict conclusions about the usefulness of the clustering results for downstream tasks. To facilitate the mapping and weakly inform the clustering about the task at hand, we use a typical word for each POS as seed for each cluster inspired by prototype learning introduced by Haghighi and Klein (2006). This leaves us with four clusters in which none of the prototype words can be found and four clusters containing two of them. In favor of the clustering method, we assume a cluster containing two prototype words to cover both their POS classes. Those not containing any of the prototype words are analyzed with the help of the data in our gold standard. The POS most often found in the gold standard for the words in the cluster is attributed to it.

We evaluate our experiments in a 10-fold Monte Carlo cross-validation setting. Accuracy scores for all experiments averaged over all 10 samples are given in Figure 6.8 along with the standard deviation for the 10 samples. Statistical significance is calculated using McNemar's test (McNemar, 1947).

Cluster analysis performs significantly worse than all other approaches with an accuracy of 17.9. The fully supervised classifiers CRF and the multilayer perceptron neural net perform around 70% with no significant difference. The LSTM, however, shows a performance of only 61.6% accuracy and moreover a large standard deviation of over 13% over the cross-validation sets. Inspecting the separate cross-validation sets shows that there are two outliers with a performance of around 35% accuracy. Leaving these two outliers out of the evaluation of the rest of the sets leads to a performance of 67.2% of accuracy. Even though this is still lower than the accuracies achieved with the CRF or MLP NN classifier, this seems to be closer to the expected performance. We interpret the results as a demonstration of the shortcomings of neural net approaches, where wrong initialization and local minima in the error function can lead to poor results. At the same time it confirms the importance of cross-validation settings. The respective self-learning approaches lie significantly below the CRF and MLP NN classifier with performances of 67.7% accuracy. Also the tritraining does not exceed their level of performance. The tagger model for MHG introduced in Section 6.1.3 outperforms the modern German tagger model with a significant increase of over 5% but not the classifiers trained on the small annotated data set. This shows that applying models of related languages is only then a preferable solution if there are no annotations to obtain. However, their performances can outperform the best standalone classifiers (CRF and MLP NN) in an ensemble together with them. This indicates that Voting as well as stacking shows to be a valid technique to combine the strengths of

different classifiers in a meta-classification approach. With performances of 76.7 for voting and 77.5 for stacking, there is no significant difference for those combination strategies.

The results of this series of experiments confirm the observation that we made in Section 6.1.2. Already a small number of annotated instances can lead to reasonable results. 2500 training words can lead to a classifier with a tagging accuracy of around 70%. Really promising is the combination of classifiers in so-called ensembles. They outperform single classifiers by combining the strengths of different classifiers. This effect is reported to be amplified by bootstrap aggregating as done by e.g. Dietterich (2000). The model variance is promoted by training on random samples of the training data.

Discussion

Non-surprisingly, approaches using external resources perform generally better than approaches without external resources. However, also approaches only relying on a few annotated sentences achieve results that can serve as a basis for the investigation of many research questions in DH projects. Especially results achieved using a CRF classifier or a multilayer perceptron neural net are convincing. We want to emphasize that our weakly supervised methods make use of about 2000 tokens as opposed to e.g. 10 million tokens used for training of the MHG tagger. However, self-training approaches do not show any improvement but rather lower accuracy. Clustering, in turn, has to be evaluated in an extrinsic setting in order to make reliable statements about the usefulness.

6.2.5 Summary

In this subchapter, we give directions towards the tagging of languages or domains for which no labeled data is available. We can show that even for a very specific text we can successfully apply semi-supervised methods. Already using a small amount of annotated data can lead to reasonable results using neural nets. Adding resources developed for related languages boosts results even further. Thus, even though training data is sparse, algorithms designed for learning from huge amounts of data can perform reasonably well. Yet, the combination of resources in an ensemble method setup approach outperforms any single technique even though the involved individual approaches might be developed for text far from the text at hand.

6.3 Research Contributions

In this chapter, we set out to **explore techniques for the development of tools for non-standard texts**. The key strength of this chapter is the investigation of data with different degrees of deviation from their respective standard form which allows the examination of **different resource supply conditions** with respect to annotated data, tools for related languages and other available resources. The different conditions suggest different courses of action in order to develop dedicated tools for these texts.

We experiment with **different degrees of supervision** reaching from unsupervised methods to weak supervision to fully supervised techniques using little training data. With our experiments, we emphasize the importance of small sets of manual annotation as a basis for successful development. Already small sets enable the training of classifiers that perform reasonably well and moreover have the chance of utilizing them for the adaptation of task-external resources. These adapted resources can successfully be included in the development, even if they qualitatively lag behind manually annotated data. Attention should also be paid to the **choice of the training algorithm**. Results show that especially when working with small data sets different algorithms show different performances. This is, moreover, related to different feature extraction strategies. We find that the best strategy for the development of non-standard text processing tools is the **combination of different techniques** in an ensemble. By exploiting the strengths of different classifiers possibly trained on different data, we can improve over single-handed approaches.

Even though these findings have been made within the context of historical text processing which clearly profits from a tradition of digitizing text, we believe that these directions should prove to be particularly valuable for non-standard text processing in general. Other research shows that meta-classifiers are superior in many non-standard text contexts. Gamon (2010) reports a sustainable gain using ensemble methods in the context of error correction of language learners. Martínez-Cámara et al. (2014) show that stacking outperforms the individual classifiers for Spanish polarity classification and Kennedy and Inkpen (2006) present similar results on sentiment classification of English movie reviews.

EXPLOITING LANGUAGE SIMILARITIES: A USECASE

Throughout this thesis, we have investigated various aspects of non-standard text processing. In addition, we have strengthened the idea that the context of Digital Humanities (DH) adds some advantages and a shift in focus concerning general natural language processing (NLP) to this task. In this chapter, we aim at an illustration of how our findings support the successful realization of DH projects by means of an example. We apply techniques that proved useful in the context of text processing of historical languages. We give emphasis to three important aspects of DH collaborations. Firstly, we exemplify how **expert knowledge** can be exploited throughout the development process of methods for non-standard language processing. Secondly, we demonstrate how the **focus on a specific research question** can influence the objective and thereby facilitate the development of a classifier. Finally, we bring back our claim made in Section 2.3.1 that NLP in DH cannot stop with proof-of-concept systems. We argue that one key attribute of successful collaborations between computer scientists and humanists is the **easy availability of any resulting software** for humanities scholars. In the following, we outline a DH workflow motivated by a research question from the humanities and conclude with the implementation of a webapplication which answers to this question.

7.1 Code-Switching – Yet Another Deviation from the Norm

The analysis of mixed language is not a new field and has been extensively studied from several sociological and linguistic aspects (Poplack, 1980; Myers-Scotton, 1993; Muysken, 2000; Auer and Wei, 2007; Toribio and Bullock, 2012). This has also brought different perspectives on the definition and types of mixed language. Switching between sentences (*inter-sentential*) is distinguished from switching inside of one sentence (*intra-sentential*). Poplack (1980) defines *code-switching* as ‘the alternation of two languages within a single discourse, sentence or con-

stituent'. Muysken (2000) avoids this term arguing that it suggests alternation but not insertion, and prefers *code-mixing* for intra-sentential switching. Myers-Scotton (1993) employs the cover term *code-switching* for the use of two languages in the same conversation, sentence, or phrase. In this paper we follow her definition and use code-switching (CoS) for all types of mixing.

However, CoS is not just a recent phenomenon but can already be observed in medieval writing. As has been pointed out in several studies (Wenzel, 1994; Schendl and Wright, 2012; Jefferson et al., 2013), historical mixed text is an interesting, yet still widely unexplored, source of information concerning language use in multilingual societies of Medieval Europe. Even though some studies use text corpora in order to qualitatively describe the phenomenon (cf. Nurmi and Pahta (2013)), a deeper analysis of the underlying structures has not been carried out due to the lack of adequate resources.

Computational approaches in the analysis of CoS data are quite recent as compared to linguistic studies. The first theoretical framework to parse code-switched sentences dates back to the early 1980s (Joshi, 1982), yet few studies are done in the 2000s (Goyal et al., 2003; Sinha and Thakur, 2005; Solorio and Liu, 2008a,b). With the beginning of the last decade, this picture has changed due to increasingly multi-cultural societies and the rise of social media. Supported by the introduction of annotated data sets on several language pairs, different tasks are applied to CoS data.

The characteristics of mixed data affect tasks in different ways, sometimes changing the definition (e.g. in language identification, the shift from document-level to word-level), sometimes by creating new lexical and syntactic structures (e.g. mixed words that consist of morphemes from two different languages). Thus, there is no doubt that mixed data calls for dedicated tools tailored to the specific problems and contexts encountered. In order to take these specialties into account, these different cases have to be understood. This way, differences in techniques for monolingual and mixed language processing can be unfolded to yield good results.

In order to pave the way for an in-depth corpus-based analysis, we promote the systematic annotation of resources and concentrate on developing and implementing automatic processing tools. To this end, combining forces from humanities and computer science (CS) seems promising for both sides. As an additional challenge, joint work in this context and with a specific purpose in mind does not just require developing proof-of-concept tools. We need to tackle the issue of how to make tools available to Humanities scholars. Consequently, we do not just focus on developing techniques for automatic processing but also take into consideration how to share tools and make them useful for interpreting and analyzing data.

For the project presented in this study, we annotate Macaronic sermons (Horner, 2006)¹ with language information and POS, and use this resource to develop tools for automatic language identification (LID) on the word level and POS tagging of mixed Latin-Middle English text. The resulting tools allow for the automatic annotation of larger quantities of text and thus for

¹We are greatly indebted to the Pontifical Institute of Mediaeval Studies (PIMS), Toronto, for their support and kind permission to use a searchable PDF version of the sermon transcripts.

the investigation of CoS constraints within specific syntactic constructions on a larger scale. In particular, we aim at an analysis of CoS rules within nominal phrases.

In the following example, the determiner and modifier (*þe briȝt / the bright*) are written in Middle English whereas the head of the noun phrase (*sol / sun*) is written in Latin. Keller (2017) provides an analysis of adjectival modifiers in the framework of the Matrix Language Frame model introduced by Myers-Scotton (1993 and following).

þe	briȝt	sol	sapiencie	subtrahit	lumen	suum
the	bright	sun	wisdom	withdraws	light	its
eng.	eng.	lat.	lat.	lat.	lat.	lat.

The goal is the extraction of such phrases with the help of POS patterns along with the language information for all words of each phrase.

The body of this chapter is organized as follows. Section 7.2 gives an overview of work that has been done in the context of CoS. In Section 7.3, we describe the data set that serves as a basis for the experiments described in Sections 7.4 and 7.5. Section 7.6 details how our tools are made available for wider use by the academic community.



Publication

Parts of this chapter were published in Schulz and Keller (2016) and Çetinoğlu et al. (2016).

7.2 Related Work

Previous work on automatic processing of mixed text can be divided into two main areas: research on LID and work on POS tagging.

LID for written as well as for spoken CoS has been tackled for a wide range of language pairs and with different methods. Lyu and Lyu (2008) investigate Mandarin-Taiwanese utterances from a corpus of spoken language. They propose a word-based lexical model for LID integrating acoustic, phonetic and lexical cues. Solorio and Liu (2008a) predict potential CoS points in Spanish-English mixed data. Different learning algorithms are applied to transcriptions of code-switched discourse. Jain and Bhat (2014) present a system on using conditional posterior probabilities for the individual words along with other linguistically motivated language-specific as well as generic features. They experiment with a variety of language pairs, e.g. Nepali-English, Mandarin-English or Spanish-English. Yeong and Tan (2011) use morphological structure and sequence of syllables in Malay-English sentences to identify language. Barman et al. (2014) investigate mixed text including three languages: Bengali, English and Hindi. They experiment with word-level LID, applying a simple unsupervised dictionary-based approach, supervised word-level classification with and without contextual clues, and sequence labeling using CRFs.

POS tagging is the second most popular task after language identification in the current state of CoS research. Unlike LID, CoS does not change the definition of the task. Nevertheless the task gets harder compared to tagging monolingual text. While state-of-the-art models reach over 97% accuracy on canonical data², in work on CoS data scores mostly around 70% are reported.

One problem, as expected, is the lack of large annotated data. Table 7.1 shows all the POS-annotated CoS corpora to our knowledge and their sizes. CoS POS tagging requires more annotated data compared to monolingual tagging, as CoS increases the possible context of tokens.

Corpus	Language	Tokens	Tag set
S&L08	En-Es	8k	PTB ³ + 75 Es
V'14	En-Hi	4k	12 UT + 3 NE
J'15	En-Hi	27k	34 Hi + 5 Twitter
ICON'15 ⁴	En-Hi	27k	34 Hi + 5 Twitter
	En-Bn	38k	34 Hi + 5 Twitter
	En-Ta	7k	17 UD
Ç&Ç'16	De-Tr	17k	17 UD
S'16	En-Hi	11k	12 UT

Table 7.1: Overview of POS-annotated CoS corpora. S&L08:Solorio and Liu (2008b), V'14:Vyas et al. (2014), J'15:Jamatia et al. (2015), Ç&Ç'16:Çetinoğlu and Çöltekin (2016), S'16:Sharma et al. (2016), UT: Google Universal Tags (Petrov et al., 2012). UD: Universal Dependencies tag set (Nivre et al., 2016).

The last column of Table 7.1 shows the tag sets used in annotating POS. Only one corpus uses language-specific tags (Solorio and Liu, 2008b), which predates universal tag sets. With the introduction of Google Universal Tags (UT) (Petrov et al., 2012) and later its extended version Universal Dependencies (UD) tag set (Nivre et al., 2016), preference has moved to using a common tag set for all tokens. Vyas et al. (2014) employ 3 additional tags for named entities. Jamatia et al. (2015) and ICON 2015 Shared Task use a Hindi tag set that is mappable to UT. They also adopt 5 Twitter-specific tags.

Solorio and Liu (2008b) show that high accuracy English and Spanish taggers achieve only 54% and 26% accuracy respectively on their data, indicating that off-the-shelf monolingual taggers are not suitable for CoS text. Common methods applied to overcome this problem in several experiments (Solorio and Liu, 2008b; Vyas et al., 2014; Jamatia et al., 2015; Sharma et al., 2016) are to choose between monolingual tagger outputs based on probabilities, utilizing monolingual dictionaries and language models and applying machine learning to the annotated CoS data. One feature that deviates from standard POS tagging is language IDs, which are shown

²[https://aclweb.org/aclwiki/index.php?title=POS_Tagging_\(State_of_the_art\)](https://aclweb.org/aclwiki/index.php?title=POS_Tagging_(State_of_the_art)), 20/06/2017.

³Solorio and Liu (2008b) report the tagset is a slightly modified version of PTB but do not give the exact number of tags.

⁴Data from the ICON 2015 Shared Task on *Pos Tagging For Code-mixed Indian Social Media Text*. It is available at <http://amitavadas.com/Code-Mixing.html>

label	explanation	%
l	Latin	60.5
e	Middle English	24.6
a	word in both languages	1.8
n	Named Entity	1.0
p	punctuation	12.1

Table 7.2: Labels annotated for LID along an explanation for each label and the occurrence in percent.

to be quite useful in previous work. Thus another challenge that comes with CoS is predicting language IDs as a prior step to POS tagging.

Solorio and Liu (2008b) achieve a high score of 93.48% with an SVM classifier, but this could be partly due to monolingual English sentences that constitute 62.5% of the corpus. In corpora with a higher level of mixing, e.g. (Vyas et al., 2014; Jamatia et al., 2015; Sharma et al., 2016) best scores drop to 65.39%, 72%, and 68.25% respectively. At the ICON 2015 Shared Task, the best system has an average of 76.79% accuracy. These scores show POS tagging on CoS data has room for improvement.

Considering the rather limited number of automatic processing tools for our languages at hand, we focus on those methods suggesting the application of shallow features for written language. Thus, we renounce morphological processing as described in Yeong and Tan (2011) and prosodic features since we are working with written text.

7.3 Data

The texts addressed in the following are so-called Macaronic sermons (Horner, 2006), a text genre containing diverse CoS structures of Middle English and Latin which is thus highly informative both for historical multilingualism research and for computational linguistics. Our aim is to investigate phrase-internal CoS. This requires language information on the token level on one hand and a basic understanding of the syntax of a sentence on the other. We aim at POS tagging as a basis for a pattern-extraction-based approach. In particular, we are interested in extracting mixed-language nominal phrases with a focus on determiners, attributive adjectives and adjective phrases as adnominals.

Since we are often dealing with a critically low data situation in DH focusing on historical topics, we experiment with a data set which can realistically be acquired with just a few hours of annotation effort. This implies that our approach is easily applicable to language pairs for which there is only a limited amount of annotated data. Our annotated corpus comprises about 3000 tokens.

In a first step, we annotate the tokens for the following language information, mostly Latin

label	explanation	%
ADJ	adjective	8.0
ADP	adposition (pre- and post)	7.9
ADV	adverb	6.0
CONJ	conjunction	7.9
DET	determiner	6.8
NOUN	noun (common and proper)	29.1
NUM	cardinal number	0.03
PRON	pronoun	4.3
PRT	particle or other function word	3.2
VERB	verb (all tenses and modes)	14.4
X	foreign word, typo, abbrev.	0.06
.	punctuation	12.3

Table 7.3: Labels annotated for POS tagging along with the explanation for each label and the occurrence in percent.

and Middle English. The two languages share a small part of their vocabulary. Those words can e.g. be simple function words such as *in*. For these items the attribution to one or the other language is not possible. We label these words with a separate tag to preserve the information that no decision on language could be made. Moreover, we mark named entities since they are often not part of the vocabulary of a language, as well as punctuation. Just about 25% of the tokens are Middle English, compared to more than 60% of Latin words (cf. Table 7.2). Our data set comprises 159 sentences with an average length of 19.4 tokens. Overall we observe 316 switch points, which means an average number of two CoS points per sentence.

In a second step, we annotate coarse-grained POS using the Universal Tagset (UT) suggested by Petrov et al. (2012). This choice facilitates a consistent annotation across languages since language specificities are conflated into more comprehensive categories. Nouns constitute by far the most frequent POS (cf. Table 7.3), which makes our data set a promising source for the investigation of nominal phrases.

7.4 Automated Processing of Mixed Text

We model LID and POS tagging as both two subsequent tasks in which POS tagging builds upon the results of the LID and two independent tasks where POS tagging and LID do not inform each other. LID can be understood as a step to facilitate POS tagging and any further processing of mixed text. In order to be used as a feature for POS tagging, it needs to be solved with a high accuracy to avoid error percolation through the entire processing pipeline.

7.4.1 Language Identification

We use an approach similar to the one described by Solorio and Liu (2008a). Since there is no available lemmatizer for Middle English, in contrast to Solorio and Liu (2008b) we cannot add lemma information to our training. To compensate for the lack of lemmas, we include POS informed word lists for both languages extracted from manually annotated corpora. Following the POS introduced by the universal dependency initiative (Nivre et al., 2016), we extract lists for the following POS: adjectives, adverbs, prepositions, proper nouns, nouns, determiners, interjections, pronouns, verbs, auxiliary verbs and conjunctions. For Middle English, we extract these lists from the Penn Parsed Corpora of Historical English (Kroch and Taylor, 2000). For Latin, we revert to the Latin corpora included in the Universal Dependency treebank namely Latin Dependency Treebank 2.0 (LDT) (Bamman and Crane, 2011), Latin-PROIEL UD treebank (Haug and Jøhndal, 2008) and the Latin-ITTB UD treebank (McGillivray et al., 2009). In case a word is found in one of the lists, we add its POS as a feature.

CRF classifiers are known to be successful for sequence labeling tasks. Based on features extracted from the results given by monolingual taggers for our data, we train a CRF classifier (Lafferty et al., 2001) combining those features with several other features. The features we implement are the following:

- 1 surface form
- 2 POS tag TreeTagger Latin
- 3 TreeTagger confidence Latin
- 4 POS tag TreeTagger Middle English
- 5 TreeTagger confidence Middle English
- 6 POS from Middle English word list
- 7 POS from Latin word list
- 8 character-unigrams prefix
- 9 character-bigrams prefix
- 10 character-trigrams prefix
- 11 character-unigram suffix
- 12 character-bigram suffix
- 13 character-trigram suffix

Features 2-5 are generated by the Latin and Middle English TreeTagger (Schmid, 1995), respectively. This means that this method is only an option for languages for which a TreeTagger model is available or can be trained⁵. We include character-n-gram affixes from length 1-3 to

⁵We want to thank Achim Stein, University of Stuttgart, for providing the parameter file for Middle English.

	label	l	e	a	n	p	all
Pre	BL	68.9	0.0	0.0	0.0	100	33.8
	CRF	93.1	93.9	45.5	0.0	98.7	66.0
Rec	BL	100	0.0	0.0	0.0	99.4	40.0
	CRF	97.6	92.1	7.1	0.0	98.9	59.2
F	BL	81.6	0.0	0.0	0.0	100	36.3
	CRF	95.3	93.0	14.9	0.0	99.3	59.9

Table 7.4: Performance of the CRF system for language identification compared to the baseline (BL). Precision, recall and F-score per class and macro-average of all classes.

account for the fact that Latin is characterized by a relatively restricted suffix assignment. In addition, we use a context window of 5 tokens on all features.

7.4.2 Part-of-Speech Tagging

For POS tagging, we use the same features as described in Section 7.4.1 (CRF_{base}). In order to investigate the influence of LID as a feature on POS Tagging, we also train the CRF classifier ($CRF_{predLID}$) using information generated by the LID system (feature 14.a). Since we cannot assume perfect LID, we evaluate the performance of a CRF classifier ($CRF_{goldLID}$) having the gold standard LID (feature 14.b) at its disposal. In this way, we can investigate to which degree differences in the quality of LID influence the POS tagging quality.

14.a LID label predicted by the system described in Section 7.4.1

14.b gold LID label manually annotated for our corpus

7.5 Results

We evaluate our systems in a 10-fold cross-validation setting using 80% for training, and 10% each for development and testing. We tune the hyper-parameter settings of our learning algorithm on our development set by testing different manually chosen parameter settings. The CRF classifier is trained with the CRF++ toolkit (Lafferty et al., 2001) using L2-regularization and a c-value of 1000. We report average results over all sets.

7.5.1 Language Identification

Since the sermons are primarily written in Latin featuring Middle English insertions, we use a combination of Latin and perfect punctuation labeling as a majority baseline (BL) for our LID system. We report per-class precision, recall and F-score along with macro-averages for the overall system. We do not report accuracy since the number of instances per class highly varies.

As was to be expected, our system reliably finds the right label for Latin text and just a little less so for English. We attribute the poor performance for named entities and words appearing in both languages to the low number of training instances in our corpus.

In order to investigate the primary sources of errors, we inspect the incorrectly labeled tokens per class. Table 7.5 shows that all but 2.4% of the Latin tokens are labeled correctly. The erroneous labels can be attributed to about 84% to English, 7% to the class that can appear in both languages. The remaining 9% contain wrong labels for punctuation. The performance for English tokens is slightly lower with an error rate of 7.9% incorrect labels which are almost all tagged as Latin. This can be due to the fact that our data contains more Latin tokens overall. The same effect is observable for the labels *a* (word in both languages) and *n* (named entities). Since the corpus contains just a few instances with those labels, they get incorrectly assigned to Latin. The small error in classifying punctuation appears in one of our cross-validation sets where colons are not part of the training but the test set.

7.5.2 Part-of-Speech Tagging

For the evaluation of our POS tagger, we use two baselines. We compare the output of our systems to the output of the monolingual Latin tagger after mapping the Latin tagset to the UT. Moreover, we add a strong baseline, drawing on the confidence feature of the monolingual TreeTagger models. We choose the POS label of the monolingual tagger with a higher level of confidence. In case the label indicates that a word is a foreign word, we choose the label from the other language (in our case Middle English). We map all POS tags to the UT. Per-class results along with macro-F-score are shown in Table 7.6.

All our systems beat the baseline systems for almost all classes (except for BL2 adverb and verb) (cf. Table 7.6). With overall F-scores between 67.4 and 67.7 our systems achieve better F-scores than the baseline systems with an F-score of 46.7 and 55.5, respectively. For our further analysis we leave the results for NUM and X aside cause they appear just once and three times in the entire corpus, respectively. Even though the average scores for all classes combined range just between about 60 and 90, we achieve good results for classes with a high number of tokens in our corpus (e.g. nouns and verbs), and also for adpositions and conjunctions. Since macro-F-score

label	% err	% l	% e	% a	% n	% p
l	2.4	-	84.1	6.8	0.0	9.1
e	7.9	95.0	-	3.3	0.0	1.7
a	92.9	90.4	9.6	-	0.0	0.0
n	100	90	10.	0.0	-	0.0
p	0.5	100	0.0	0.0	0.0	-

Table 7.5: Percentage of incorrectly labeled tokens per class along with the distribution of incorrect labels among the other labels.

	label	ADJ	ADP	ADV	CONJ	DET	NOUN	NUM	PRON	PRT	VERB	X	.	all
Pre	BL1	43.3	92.0	72.9	85.1	25.0	71.1	0.0	30.5	0.0	55.8	5.1	100	48.4
	BL2	55.7	83.1	68.6	87.2	37.5	82.5	0.0	34.5	23.2	78.2	7.1	100	54.8
	CRF _{base}	68.1	92.0	81.2	88.8	79.3	85.2	0.0	82.2	71.4	85.9	0.0	98.2	69.4
	CRF _{predLID}	69.2	92.8	79.5	89.7	78.9	85.3	0.0	82.2	72.5	86.2	0.0	98.2	69.5
	CRF _{goldLID}	69.4	92.4	80.0	90.4	77.8	85.6	0.0	82.2	72.5	86.4	0.0	98.4	69.6
Rec	BL1	51.0	80.6	56.8	63.1	3.3	79.4	0.0	45.1	0.0	76.5	1.0	98.4	46.3
	BL2	51.8	89.7	68.6	81.1	8.6	90.6	0.0	53.4	23.2	84.4	100	98.4	65.8
	CRF _{base}	60.0	86.0	67.6	88.1	82.3	95.3	0.0	66.2	60.6	86.9	0.0	98.7	66.0
	CRF _{predLID}	60.4	85.5	69.2	88.9	82.3	95.4	0.0	66.2	58.6	87.6	0.0	98.4	66.0
	CRF _{goldLID}	65.1	89.1	74.2	89.4	80.0	90.3	0.0	73.3	64.8	87.0	0.0	98.7	66.2
F	BL1	46.9	85.9	63.8	72.5	5.9	75.0	0.0	36.4	0.0	64.5	9.8	99.2	46.7
	BL2	53.7	86.3	68.8	84.1	14.0	86.4	0.0	41.9	36.5	81.2	13.3	99.2	55.5
	CRF _{base}	63.8	88.9	73.7	88.5	80.8	90.0	0.0	73.3	65.6	86.4	0.0	98.4	67.4
	CRF _{predLID}	64.5	89.0	74.0	89.3	80.6	90.1	0.0	73.3	64.8	86.9	0.0	98.3	67.6
	CRF _{goldLID}	65.1	89.1	74.2	89.4	80.0	90.3	0.0	73.3	64.8	87.0	0.0	98.7	67.7

Table 7.6: Performance of the CRF systems for POS tagging compared to the majority baseline (BL1), the confidence baseline (BL2). CRF_{base}: system with the 13 basic features, CRF_{predLID}: system with predicted LID as an additional feature, CRF_{goldLID}: system with gold-standard LID as an additional feature. Precision (Pre), Recall (Rec) and F-score (F) per class and macro-average of all classes are given. The task-relevant results are emphasized in bold.

gives equal weight to all classes the numbers might be misleading, depending on the purpose of the system. Given that we built the POS tagger with a specific task in mind, namely the extraction of nominal phrases, we calculate the F-score for the POS classes relevant to this task (determiners, adjectives and nouns). This gives a task-specific macro F-score of 78.2 (CRF_{base}), 78.4 (CRF_{predLID}) and 74.5 (CRF_{goldLID}), respectively. Those F-scores are noticeably above the average F-scores for the overall systems and also beat the task-specific F-scores of BL1 (42.6) and BL2 (51.4). The relatively high average recall of almost 80 for these three labels combined for all three systems is important for the task whereas precision has lower priority, since the extracted phrases are manually inspected afterwards. Since our LID system performs well, the system with automatically predicted labels shows a slight increase in performance compared to the system without LID information. The system with manually annotated LID information yields the best performance. However, according to McNemar’s test the differences are not statistically significant.

The analysis of the incorrectly labeled tokens shows which POS tags are difficult to distinguish (cf. Table 7.7). Since we are especially interested in adjectives, an error rate of 40% is rather high. Out of these, about 63% have been incorrectly labeled as nouns, which has considerable negative effect on our objective, especially since most of the incorrectly labeled nouns are labeled as adjectives. Almost 70% of the adjectives that are incorrectly labeled as nouns are Latin. This can be explained by the morphology of adjectives in Latin. As Latin adjectives and

label	% err	ADJ	ADP	ADV	CONJ	DET	NOUN	PRON	PRT	VERB	.
ADJ	39.6	-	2.1	3.1	0.0	9.3	62.9	0.0	1.0	20.6	1.0
ADP	14.6	11.4	-	8.6	6.5	5.7	11.4	0.0	37.1	14.3	2.9
ADV	30.8	19.3	5.3	-	10.5	5.3	33.3	7.0	1.8	14.0	0.0
CONJ	11.1	0.0	0.0	37.0	-	11.1	7.4	22.2	11.1	7.4	3.7
DET	17.7	16.2	10.8	10.8	2.7	-	32.4	10.8	8.1	8.1	0.0
NOUN	4.6	56.1	0.0	9.8	0.0	0.0	-	2.4	0.0	26.8	4.9
PRON	33.8	8.8	0.0	2.2	15.5	31.1	20.0	-	2.2	17.8	2.2
PRT	41.4	4.9	12.2	14.6	17.1	22.0	14.6	2.4	-	12.2	0.0
VERB	12.4	25.5	3.6	1.8	0.0	7.3	54.5	5.5	0.0	-	1.8
.	1.6	33.3	0.0	0.0	16.7	0.0	50.0	0.0	0.0	0.0	-

Table 7.7: Percentage of incorrectly labeled tokens per class along with the distribution of incorrect labels among the other labels for the CRF_{predLID} system.

nouns often have similar, if not the same suffixes of case marking, the two classes cannot be distinguished using the suffix as a defining feature. These difficulties are also observed by von der Brück and Mehler (2016) who present a morphological tagger for Latin.

	þis	made	hom	to	lede
	this	made	them	to	lead
lang.	eng.	eng.	eng.	eng.	eng.
gold	PRON	VERB	PRON	PRT	VERB
pred	PRON	VERB	PRON	PRT	VERB
	super	terram	celestem	conuersacionem	
	on	earth	heavenly	regime	
lang.	lat.	lat.	lat.	lat.	
gold	ADP	NOUN	ADJ	NOUN	
pred	ADP	DET	NOUN	NOUN	

The first half of the sentence⁶ is written in Middle English. The assigned POS tags are correct and also the first Latin word after the CoS point is labeled correctly. The subsentence *terram clestem conuersacionem* is tagged in the pattern of a noun phrase with a determiner and a compound noun instead of a prepositional phrase *super terram* (Engl.: on earth) and a noun phrase *clestem conuersacionem* (Engl.: heavenly behavior) consisting of an adjective and a noun. The similar syntactic function of pronouns (in case of possessive pronouns and demonstrative pronouns) and determiners leads to an additional source of error.⁷ The following example displays a tagging error in which the demonstrative pronoun *isso* (Engl.: this) is used as a pronoun. Since it can be used as a determiner in other sentences, the tagger mislabels it as a determiner here.

⁶Translation by Horner (2006): *this made them lead on earth a heavenly regime.*

⁷Translation by Horner (2006): *in it there is no confidence.*

size	LID			POS		
	Pre	Rec	F-score	Pre	Rec	F-score
800	56.3	56.8	56.5	60.8.1	54.6	56.8
1600	56.6.0	57.8	57.2	66.7	63.0	64.6
2400	66.0	59.2	59.9.3	69.5	66.0	67.6

Table 7.8: Different portions of the training set along with precision, recall and F-score for LID and POS tagging.

	In	isto	non	est	fiducia
	In	this	not	is	confidence
lang.	lat.	lat.	lat	lat.	lat.
gold	ADP	PRON	PRT	VERB	NOUN
pred	ADP	DET	PRT	VERB	NOUN

On closer inspection, we find that many of the incorrectly tagged words appear in POS sequences which are either rarely or not at all contained in the training data. We predict that adding more training data will significantly decrease errors of this kind. Since data sparsity in general is an issue dealing with historical text, we investigate how different sizes of the training set influence the results. We compare results for 800 tokens, 1600 tokens, and for the complete training set (around 2400 tokens).

With an increase of training instances, the results improve for both tasks (cf. Table 7.8). The increase from 800 to 1600 is higher than from 1600 to 2400. This suggests that the F-score might grow logarithmically with increasing training size.

7.6 Tools for Digital Humanities

Since the aim of our project is not only to build a proof-of-concept system but to enable Humanities scholars to automatically process their data with the help of our tools, we implement a simple web service in Java to offer an easily accessible interface to our tool (cf. Figure 7.1)⁸. Moreover, we added our tool to the Clarin-D repository to ensure sustainability⁹.

The data is returned in a format compatible with ICARUS, a search and visualization tool which primarily targets dependency trees (Gärtner et al., 2013). Despite the present lack of a dependency-parsed syntax layer, ICARUS offers the opportunity to inspect the data and pose complex search requests, combining the three layers of token, language information and POS tag. Figure 7.2 shows a query that extracts all sequences of a determiner in either of both languages followed by a Middle English adjective followed by a Latin noun (cf. Figure 7.2). ICARUS shows the results within the sentence of origin. ICARUS also allows searches including gaps.

⁸The web service is hosted at <https://clarin09.ims.uni-stuttgart.de/normalisierung/mixed-pos.html>. For access, please contact the author.

⁹Clarin-D repository, metadata handle: <http://hdl.handle.net/11022/1007-0000-0007-C61B-C>.

Part-of-speech tagging: mixed text Home Impressum

Bitte laden Sie eine utf-8 kodierte txt-Datei hoch

Textdatei

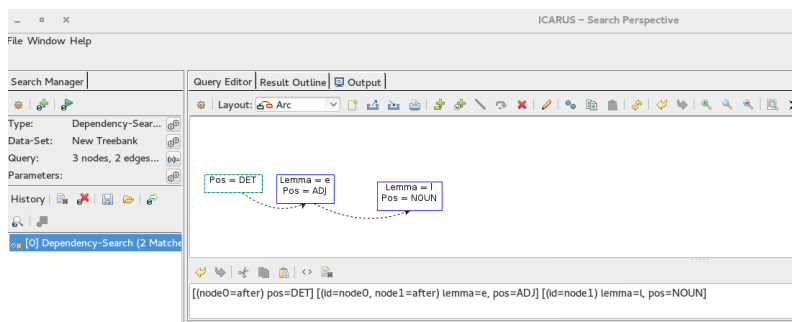
Choose File No file chosen

Das Ergebnis erhalten Sie per Email. Bitte geben Sie Ihre Email-Adresse an:

Email

Submit

Figure 7.1: Simple web interface for the submission of mixed text for POS tagging.



(a) Formulation of a search query in ICARUS.

1: pe e DET	2: brigt e ADJ	3: sol NOUN	4: sapienciel NOUN	5: subtrahit VERB	6: lumen NOUN	7: suum PRON
-------------------	----------------------	---------------------	----------------------------	---------------------------	-----------------------	----------------------

1: pe brigt sol sapienciel subtrahit lumen suum
2: lift vp tuum oculum ad istam blessic idem

(b) Results shown by ICARUS

Figure 7.2: Search interface of ICARUS returning results on a query for an English adjective followed by a Latin noun within the next 3 tokens.

This is helpful since nominal phrases vary according to the number of adjectives and as to whether or not they contain an overt determiner. Thus, flexibility in formulating the search query facilitates an in-depth search of all possible constructions.

Our method can easily be adapted to other languages by inserting the fitting monolingual taggers (TreeTagger) and POS related word lists (if available). For this purpose, the code is publicly available on Github¹⁰.

¹⁰<https://github.com/sarschu/CodeSwitching>.

7.7 Research Contributions

We show the implementation and application of two systems, one for language identification and one for POS tagging, developed for a specific purpose capitalizing on the **insights we gained throughout our experimentation** in Section 6.1 and Section 6.2. We achieve reasonable results given the very low number of annotated training instances. Considering the detailed error analysis for our system, we can purposefully extend our training data in order to correct the sources of error in the future by for example adding monolingual data from the Penn-Helsinki Parsed Corpus of Middle English (Kroch and Taylor, 2000).

We believe that not just the development of tools but also the support with respect to applying them constitutes an important component of successful collaboration between humanities and CS. In return, a **task-oriented tool development** along with immediate **feedback on the performance and analysis of error from the humanities side** facilitate the implementation of systems that do not only serve the proof of a concept but are applied to real-world data. We believe that this kind of collaboration is the way to give CS the chance to support other fields in their research and find new and interesting challenges throughout this work.

PROBLEMS SOLVED? – FUTURE DIRECTIONS FOR DIGITAL HUMANITIES

In this dissertation, I set out to introduce various aspects of Digital Humanities (DH) research and relate them to natural language processing (NLP). These aspects especially emphasize the differences between general NLP and application-oriented text processing in an interdisciplinary context. The main focus of this work is on the automatic processing of non-standard texts. Throughout my dissertation, I visit a whole string of examples illuminating challenges and potential solutions. I report on a computer-aided digitization project and discuss two main approaches towards non-standard text processing – text normalization and tool adaptation – through the example of different kinds of text. These DH model projects highlight the aspects of application-oriented, reusable, problem-specific and collaborative implementations from various angles and on different levels. Since it is the main focus of this thesis, I will start to summarize the findings on non-standard text processing. Subsequently, I will broaden the scope of the discussion to more general aspects of DH and NLP, further challenges and opportunities.

8.1 Towards Standard-Free Text Processing

Non-standard text processing is a difficult task. This is generally not due to the individual characteristics of each single non-standard text genre, but rather due to a fixation of NLP on one specific sort of text over the past 30 years. With various examples from different kinds of text, I show that some of the basic assumptions that are taken for granted, such as a clear definition of sentence boundaries or the definition of a token, can cause problems in approaches towards non-standard text processing. I demonstrate how the use of emoticons and other unconventional usages of punctuation marks in user-generated content (UGC) requires an adjusted definition

of these concepts. We e.g. work on the level of messages for UGC rather than on sentence level. Moreover, one has to be aware that the lack of punctuation in historical texts impairs approaches such as sentence-based self-learning.

Processing of non-standard texts is not hard per se. Neither is the vocabulary in Middle High German bigger than in modern newspaper texts nor is the syntax more complex. A standard language is defined by a sufficient availability of annotated data and a certain knowledge of the characteristics of the language found in this sort of text. Thus, any sort of text can be a standard form given a sufficient availability of data. An implication of this is that there is not a single standard but various standards. An ambitious aspiration is to move away from an NLP situation in which there is one very strong standard form that determines the limitations of our processing tools towards a broader understanding of language in NLP. All of these issues root in a machine learning (ML) tradition for solving NLP tasks. The fact that learning takes place in a relatively static feature space makes a transfer to data that varies in its characteristics difficult. Starting from ground zero is not a solution for every newly encountered genre or language. Thus, there are two main approaches for tackling the processing of such data: the assimilation of the non-standard texts to the feature space of the training data (text normalization) or the desensitization of processing methods to only one data source (tool adjustment). Tools need to be less dependent on specific characteristics of one sort of text. In addition, text annotation efforts have to be extended to all kinds of text.

I illustrate the approach of **text normalization** by means of normalization of Flemish UGC. I demonstrate how normalization can successfully improve results of standard NLP tools on such data by adjusting non-standard texts to the training domain. Text normalization itself, however, is again based upon the availability of training data. For extremely small genres or texts which are far away from the standard with respect to their characteristics, this approach is not promising. In anticipation of a world in which the NLP community aims to cover as many languages as possible, this is likely to be a rather inefficient solution.

As an alternative, I show various methods of **tool adaptation** and training techniques that can be applied to non-standard texts. I focus on ideally utilizing small training sets, exploiting task-related resources and tools and taking advantage of resources from similar languages. The main insight is the fact that there is no single solution and no “best” approach to non-standard text processing. Solutions are dependent on the data situation and the proximity to better-resourced languages. Nevertheless, the experiments and insights of this dissertation can serve as a guidebook and orientation for which techniques are promising.

The Universal Dependency Project¹ provides a good data basis for language processing across language boundaries. Largely unified annotations of language phenomena help to transfer knowledge from one to another language. The resources collected within this project thus far, however, often also just extend to the “standard manifestation” of a language. A similar scenario

¹<http://universaldependencies.org/>, 01/09/2017.

for texts coming from DH is envisioned by Bamman (2017). He argues for a shared repository of linguistic annotations for all kinds of data which enables the leveraging of complementary sources. It is an open question how to handle the diversity of resource questions and project-specific annotations in such a context while still enabling the interoperability of schemes. Yet, shared diverse data indeed is the foundation of a standard-free NLP.

8.2 Digital Humanities: Towards Key Concepts of a Methodology

One aspect that can make DH inefficient is that there are only few standardized methodologies. This results in an overload through the repetition of basic tasks throughout different projects. As an attempt at a solution, I introduce key concepts for the successful development of NLP tools in the context of DH but also for workflows in DH projects in general.

I show how flexible and modular system architectures can be used to optimize different objectives and to fit different data sets. The concepts of **reusability and adaptability** as time-saving factors determine the way that NLP has to take in order to be successful in a DH context. However, not only the implementation of NLP tools can profit from the concept of reusability. Many of the steps in a project workflow such as data collection, annotation and analysis of results follow a pattern. These patterns can be optimized and transferred between projects just like workflows for the development of tools.

Furthermore, I show how using **expert knowledge** as input during the development process can improve performance. Especially in the process of data annotation and feature design, experts from the humanities can contribute valuable insights to ensure successful ML solutions. The deep knowledge of the data can compensate for the often only small number of training examples. Immediate feedback on the performance of a modeling technique and the expertise in annotating data for targeted improvement of a system, as e.g. done in active learning environments, are advantages that arise from the collaborative context.

In line with this, I illustrate that **problem-specific solutions** are vital for successful implementations in DH. Putting the focus on small subproblems without having to solve a task in a general manner, can make up for the lack of larger amounts of data. I present this by means of pattern extraction from code-switching texts. The focus on the extraction of adjectives, determiners and nouns facilitates the task compared to the task of general POS tagging.

In addition, I emphasize the importance of the **easy accessibility of tools**. General solutions and proof-of-concept systems are not enough in the context of DH. **Applications** need to be delivered to the project partner tailored to the demands and abilities of the user.

8.3 DH and NLP: a Joint Future

The added benefit of DH for NLP became obvious through this dissertation. NLP has focused on a limited variety of texts. DH offers plenty of different kinds of text and interesting problems for computational models along with the aspect of application which will force NLP researchers to take a step forward and develop real NLP for real people. Other non-standard text domains can profit from methods developed in the context of NLP for DH.

In this dissertation, I focus on a methodology from the computational point of view. Workflows for steps apart from the automatic processing, however, have to be established as well. I review one of these steps in Section 3.3 of this thesis, in which I detail how the annotation method that has proven useful in NLP can be adapted to fit the need for more flexible annotation in DH. The differences in the ways in which problems are approached in the humanities and in computer science are more challenging: often it is hard to define research problems from the humanities in such a structured way that formalization is easily possible. In addition, the hermeneutic interpretation requires the embedded interpretation of a research object in its context. Contrastively, computer science often approaches problems in a strongly modularized way. Problems are taken apart into subproblems which leads to a temporary decontextualization. The real challenge is it, therefore, to harmonize these two strategies. A narrow feedback loop between all collaboration partners helps to understand subparts and supports the maintenance of the “bigger picture”.

In Chapter 2, I argue that the “real digital humanist” – a person who has deep knowledge in computer science as well as in a humanities discipline – is still rare. This is a crucial shortcoming that makes DH vulnerable to criticism. There is **DH scepticism** eloquently formulated by e.g. Marche (2012). He reduces the significance of DH to some minor niche fields inside the humanities. He mainly criticizes the inadequacy of treating literature like data. However, his definition of data as a complete collection is not used as such by researchers in the DH. He seems to assume an unreflected and meaningless approach toward the analysis of literary texts which is neither the goal nor common practice in recent DH. Kirsch (2014) rightfully discusses the limitations of DH and warns against following a rhetoric style that preaches the redemption of the humanities. Clearly, this is a reminder for a reflected usage of digital methods for the investigation of humanistic research questions and has its justification in a debate in which a few seem to forget that computers will likely never outperform humans in tasks such as the interpretation of literary texts. Reflection of automatically achieved results is a key point in a competent analysis for which a basic understanding of the inner workings of the applied algorithm is required. In turn, the mere competence to interpret why an algorithm suggests certain results is not sufficient for the correct interpretation of these results in the humanistic context. Proper reflection on automatically extracted data can be decisive for determining the validity of results of an entire DH project. Kirsch (2014) neglects that such meaningful implementations of thoughtful and successful DH projects are widely achieved, as pointed out by Worthey (2014),

as an answer to Kirsch's article. Thus, his generalizations seem hard to maintain.

Allington et al. (2016) reduce DH to “the promotion of project-based learning and lab-based research over reading and writing, the rebranding of insecure campus employment as an empowering ‘alt-ac’² career choice, and the redefinition of technical expertise as a form (indeed, the superior form) of humanist knowledge” and deprive them of their defined goal of using “digital or quantitative methodologies to answer research questions in the humanities”. This opinion has to be considered within a bigger debate introduced by Harpham (2005). He describes a crisis of the humanities in the early 21st century:

Sometimes the crisis – whose dimensions can be measured by declining numbers of enrollments, majors, courses offered, and salaries – is described as a separate, and largely self-inflicted, catastrophe confined to a few disciplines; sometimes it is linked to a general disarray in liberal education, and sometimes to the moral collapse and intellectual impoverishment of the entire culture. But one point emerges with considerable regularity and emphasis: humanistic scholars, fragmented and confused about their mission, suffer from an inability to convey to those on the outside and even to some on the inside the specific value they offer to public culture; they suffer, that is, from what the scholar and critic Louis Menand calls a “crisis of rationale”.

(Harpham (2005, p.21-22))

The point of departure for the fast rise of popularity of digital methods in the humanities was the search for a way out of this crisis that according to Harpham (2005) has lasted for already half a century. This illustrates the high hope that the mere existence of DH as such fuels. I see the real endeavor in creating something new beyond the traditional humanities. This means that DH is not merely seen as the upscaling of traditional methods to more textual evidence without questioning the implications this might have for the results. The opposite approach undeniably exists. Especially the problem of financial shortcomings in humanities research might be a motivation for one or the other project proposal referring to the application of digital methods. This, however, does neither solve the crisis of the humanities nor are these the projects in which the employment of digital techniques unravels new insights.

A key point for new paradigms in DH is a shared way of evaluation. Zuccala (2013) claims that “the products of Humanities research are not ‘empirical’ enough for objective forms of evaluation”. Instead, evaluation is mainly based on metrics derived from academic citations (Thelwall and Delgado, 2015). The Dutch Royal Academy of Arts and Sciences has published a report on assessment within the humanities³. Therein they request a peer-reviewed, twofold assessment

²Bethany Nowvskie has called “#alt-ac” positions doing the digital humanities- “alternative academic careers” – including postdocs, jobs in libraries, and administrative and staff positions at newly founded or expanding digital humanities centers.

³<https://www.knaw.nl/en/news/publications/quality-indicators-for-research-in-the-humanities>, 25/08/2017.

strategy based on scholarly output and benefit for the society. The scholarly aspect can be indicated by publications, reviews and citations, prizes and personal grants whereas the societal aspect can be assessed via specialist publications, contract research, projects in collaboration with civil-society actors and societal prizes. Since the focus of DH differs from the humanities in the sense that it aims for more objective research, this objectivity can and must be reflected in its evaluation. Steiner et al. (2014) suggest a user-centered evaluation methodology for humanities research environments and they emphasize that there are different groups of users that expect different functionalities. This relates back to the aspect of interdisciplinarity where easy applicability cushions the differences in expertise with computational methods. Besides, reproducibility is an aspect that is supported by a formalized approach. LeBlanc (2017) names two important aspects of reproducible research: verification and inspiration. Introducing the criterion of reproducibility into an evaluation methodology could enforce a methodological approach throughout a project that ensures validity and makes the approach more transparent. As a side effect, this would lead to a reduction of developmental overhead since components are easier to identify and to reuse. Along these lines, there is the criterion of sustainability. Since a major surplus of this kind of research is the archiving and the increase of accessibility from everywhere and from any time, sustainable support and future maintenance can be an important point for evaluation. However, since this aspect often shows at a later date, it cannot be used immediately as feedback for assessing the quality of research at the time of publication.

Whilst many humanists fear the diminution of the genius of the individual, I strongly believe that the genius of the individual does still prevail in DH. Computational methods do not replace the ambitious step of making sense of whatever the data shows. It is clear that humanistic research questions are far too complex to be operationalized to their full extent. However, a formalized investigation can highlight outliers and thus lead to refinements of existing theories. This interplay between the underlying theory and the challenges but also the surprises that computational modeling holds is one of the most valuable aspects of DH. The limitations of computer-aided analysis become clear. Yet, this is only a disappointment when the expectations towards such methods are unrealistic. Their proper application and engagement, however, are the real goal which requires trained humanists. In addition, a more formal approach brings higher transparency to the line of argumentation. It merely points towards the places where the individual might find the relevant information in order to put it all together into a traceable, yet inspired line of argumentation.

The future of the humanities will be a digital one. The humanities are generally defined as the disciplines that examine human culture, or in other words the products of the human mind. Since human life and human creativity increasingly take place in a digital environment, the science of understanding these products has its future in digital methods.

BIBLIOGRAPHY

- Abdulkader, A. and Casey, M. R. (2009). Low Cost Correction of OCR Errors Using Learning in a Multi-Engine Environment. In *Proceedings of the 10th International Conference on Document Analysis and Recognition, (ICDAR 2009)*, pages 576–580.
- Afli, H., Qiu, Z., Way, A., and Sheridan, P. (2016). Using SMT for OCR Error Correction of Historical Texts. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 962–966, Paris, France. European Language Resources Association (ELRA).
- Agic, Z., Hovy, D., and Søgaard, A. (2015). If all you have is a bit of the Bible: Learning POS taggers for truly low-resource languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, (ACL 2015)*, pages 268–272, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Aizenberg, I. N., Aizenberg, N. N., Butakoff, C., and Farberov, E. (2000). Image Recognition on the Neural Network Based on Multi-Valued Neurons. In *15th International Conference on Pattern Recognition (ICPR 2000)*, pages 2989–2992.
- Al-Rfou, R., Perozzi, B., and Skiena, S. (2013). Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL 2013)*, pages 183–192, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Allington, D., Brouillette, S., and Golumbia, D. (2016). Neoliberal Tools (and Archives): A Political History of Digital Humanities. <https://lareviewofbooks.org/article/neoliberal-tools-archives-political-history-digital-humanities/#!>
- Auer, P. and Wei, L. (2007). *Handbook of Multilingualism and Multilingual Communication*, volume 5. Walter de Gruyter.
- Aven, B. L., Burgess, D. A., Haynes, J. F., Merino, J. R., and Moore, P. C. (2009). Using Product and Social Network Data to Improve Online Advertising. US Patent App. 11/965,509.

- Aw, A., Zhang, M., Xiao, J., and Su, J. (2006). A Phrase-based Statistical Model for SMS Text Normalization. In *Proceedings of the International Conference on Computational Linguistics (COLING 2006) / Annual Meeting of the Association for Computational Linguistics (ACL 2006)*, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Baldwin, T., Cook, P., Lui, M., MacKinlay, A., and Wang, L. (2013). How Noisy Social Media Text, How Different Social Media Sources? In *Sixth International Joint Conference on Natural Language Processing (IJCNLP 2013)*, pages 356–364. Asian Federation of Natural Language Processing.
- Bamman, D. (2017). Natural Language Processing for the Long Tail. In *Book of Abstracts of Digital Humanities 2017*.
- Bamman, D. and Crane, G. (2011). The Ancient Greek and Latin Dependency Treebanks. In Sporleder, C., Bosch, A., and Zervanou, K., editors, *Proceedings of the Workshop on Language Technology for Cultural Heritage, Theory and Applications of Natural Language Processing*, pages 79–98. Springer Berlin Heidelberg.
- Barman, U., Das, A., Wagner, J., and Foster, J. (2014). Code-mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code-Switching*.
- Barteld, F., Schröder, I., and Zinsmeister, H. (2015). Unsupervised regularization of historical texts for POS tagging. In *Proceedings of the 4th Workshop on Corpus-based Research in the Humanities (CRH)*. Polish Academy of Sciences.
- Bassil, Y. and Alwani, M. (2012). OCR Post-Processing Error Correction Algorithm Using Google’s Online Spelling Suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3(1):90–99.
- Beaufort, R., Roekhaut, S., Cougnon, L.-A., and Fairon, C. (2010). A Hybrid Rule/Model-based Finite-state Framework for Normalizing SMS Messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 770–779, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Belanger, M.-E. (2010). Annotations and the Digital Humanities Research Cycle: Implications for Personal Information Management. Conference Poster.
- Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman Vaughan, J. (2010). A Theory of Learning from Different Domains. *Mach. Learn.*, 79(1-2):151–175.
- Bengio, Y. and Bengio, S. (2000). Modeling High-Dimensional Discrete Data with Multi-Layer Neural Networks. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, pages 400–406.

- Biber, D. (1993). Representativeness in Corpus Design. *Literary and Linguistic Computing*, 8(4):243–257.
- Biemann, C. (2006). Unsupervised Part-of-speech Tagging Employing Efficient Graph Clustering. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING 2006) and 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006): Student Research Workshop*, pages 7–12, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Blessing, A., Echelmeyer, N., John, M., and Reiter, N. (2017). An End-to-end Environment for Research Question-Driven Entity Extraction and Network Analysis. In *Proceedings of the Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 57–67, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 120–128, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bögel, T., Gertz, M., Gius, E., Jacke, J., Meister, J. C., Petris, M., and Strötgen, J. (2015a). Collaborative Text Annotation Meets Machine Learning: heureCLÉA, a Digital Heuristic of Narrative. *DH Commons*.
- Bögel, T., Gertz, M., Gius, E., Jacke, J., Meister, J. C., Petris, M., and Strötgen, J. (2015b). Gleiche Textdaten, unterschiedliche Erkenntnisziele? Zum Potential vermeintlich widersprüchlicher Zugänge zu Textanalyse. In *Book of Abstracts of DHd 2015*.
- Bollmann, M. (2013). POS Tagging for Historical Texts with Sparse Training Data. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability in Discourse (LAW7-ID 2013)*, pages 11–18.
- Bollmann, M., Krasselt, J., and Petran, F. (2012). Manual and semi-automatic normalization of historical spelling – Case studies from Early New High German. In *Proceedings of KONVENS 2012 (LThist 2012 workshop)*, pages 342–350.
- Bonaparte, M. (1949). *The Life and Works of Edgar Allan Poe. A Psychoanalytic Interpretation*. Imago Pub. Co.
- Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M., Maynard, D., and Aswani, N. (2013). TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pages 83–90. INCOMA.

BIBLIOGRAPHY

- Borek, L., Dombrowski, Q., Perkins, J., and Schöch, C. (2016). TaDiRAH: a Case Study in Pragmatic Classification. *Digital Humanities Quarterly*, 10(1):online.
- Borgman, C. L. (2009). The Digital Future is Now: A Call to Action for the Humanities. *Digital Humanities Quarterly*, 3(2):online.
- Borgman, C. L. (2015). *Big Data, Little Data, No Data: Scholarship in the Networked World*. MIT Press.
- Boyer, R. S. and Moore, J. S. (1991). *MJRTY – A Fast Majority Vote Algorithm*, pages 105–117. Springer Netherlands, Dordrecht.
- Brackert, H. (1971). *Das Nibelungen: Mittelhochdeutscher Text und Übertragung*. Fischer Taschenbücher. Fischer Bücher.
- Brants, T. (2000). TnT: A Statistical Part-of-speech Tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLC 2000)*, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bulatovic, N., Gnadt, T., Romanello, M., Stiller, J., and Thoden, K. (2016). Usability in Digital Humanities - Evaluating User Interfaces, Infrastructural Components and the Use of Mobile Devices During Research Process. In Fuhr, N., Kovács, L., Risse, T., and Nejdil, W., editors, *Proceedings of Research and Advanced Technology for Digital Libraries: 20th International Conference on Theory and Practice of Digital Libraries (TPDL 2016)*, pages 335–346. Springer International Publishing.
- Burdick, A., Drucker, J., Lunenfeld, P., Presner, T., and Schnapp, J. (2012). *Digital Humanities*. The MIT Press.
- Burghardt, M. (2012). Annotationsergonomie: Design-Empfehlungen für linguistische Annotationswerkzeuge. *Inf. Wiss. & Praxis*, 63(5):300–304.
- Busa, R. (1980). The annals of humanities computing: The index Thomisticus. *Computers and the Humanities*, 14(2):83–90.
- Çetinoğlu, Ö. and Çöltekin, Ç. (2016). Part of Speech Tagging of a Turkish-German Code-Switching Corpus. In *Proceedings of LAW-X*.
- Çetinoğlu, Ö., Schulz, S., and Vu, N. T. (2016). Challenges of Computational Processing of Code-Switching. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 2016), Workshop on Computational Approaches to Linguistic Code Switching (CALCS 2016)*, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Celano, G., Crane, G., and Majidi, S. (2016). Part of Speech Tagging for Ancient Greek. *Open Linguistics*, 2(1):393–399.

- Chaturvedi, S., Srivastava, S., III, H. D., and Dyer, C. (2016). Modeling Evolving Relationships Between Characters in Literary Novels. In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Meeting of the Association for the Advancement of Artificial Intelligence (AAAI 2016)*, pages 2704–2710. AAAI Press.
- Chen, X.-w. and Lin, X. (2014). Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*, 2:514–525.
- Chinchor, N. and Robinson, P. (1998). MUC-7 named entity task definition (version 3.5). In *Proceedings of the 7th Message Understanding Conference*.
- Chiron, G., Doucet, A., Coustaty, M., and Moreux, J.-P. (2017). Icdar2017 competition on post-ocr text correction. In *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 1423–1428.
- Choi, J. D. (2016). Dynamic Feature Induction: The Last Gist to the State-of-the-Art. In *Proceedings of the Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2016): Human Language Technologies (HLT)*, pages 271–281, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., and Basu, A. (2007). Investigating and modeling the structure of texting language. *International Journal on Document Analysis and Recognition*, 10(3):157–174.
- Clark, E. and Araki, K. (2011). Text normalization in social media: progress, problems and applications for a pre-processing system of casual English. *Procedia - Social and Behavioral Sciences*, 27:2–11.
- Clematide, S., Furrer, L., and Volk, M. (2016). Crowdsourcing an OCR Gold Standard for a German and French Heritage Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70:213–220.
- Collobert, R. and Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 160–167, New York, NY, USA. ACM.
- Cortizo, J. C., Carrero, F., Cantador, I., Troyano, J. A., and Rosso, P. (2012). Introduction to the Special Section on Search and Mining User-Generated Content. *ACM Transactions on Intelligent Systems and Technology*, 3(4):65:1–65:3.

- Daelemans, W. and van den Bosch, A. (2005). *Memory-Based Language Processing*. Cambridge University Press, Cambridge, UK.
- Das, D. and Petrov, S. (2011). Unsupervised Part-of-speech Tagging with Bilingual Graph-based Projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011): Human Language Technologies (HLT) - Volume 1*, pages 600–609, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Daume III, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 256–263, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- De Clercq, O., Desmet, B., Schulz, S., Lefever, E., and Hoste, V. (2013). Normalization of Dutch user-generated content. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013)*, pages 179–188. INCOMA.
- Desai, A. A. (2010). Gujarati Handwritten Numeral Optical Character Reorganization Through Neural Network. *Pattern Recognition*, 43(7):2582–2589.
- Desmet, B. and Hoste, V. (2014). Recognising suicidal messages in Dutch social media. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 830–835, Paris, France. European Language Resources Association (ELRA).
- Diebold, F. X. (2012). A Personal Perspective on the Origin(s) and Development of Big Data: The Phenomenon, the Term, and the Discipline, Second Version. PIER Working Paper Archive 13-003, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.
- Dietterich, T. G. (2000). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2):139–157.
- Dipper, S. (2010). POS-Tagging of Historical Language Data: First Experiments. In Pinkal, M., Rehbein, I., Schulte im Walde, S., and Storrer, A., editors, *Semantic Approaches in Natural Language Processing: Proceedings of the Conference on Natural Language Processing (KONVENS 2010)*, pages 117–121, Saarbrücken, Germany. Universaar.
- Dipper, S. (2011). Morphological and Part-of-Speech Tagging of Historical Language Data: A Comparison. *JLCL*, 26(2):25–37.
- Dipper, S., Donhauser, K., Klein, T., Linde, S., Müller, S., and Wegera, K. (2013). HiTS: ein Tagset für historische Sprachstufen des Deutschen. *JLCL*, 28(1):85–137.

- Domingos, P. (2012). A Few Useful Things to Know About Machine Learning. *Commun. ACM*, 55(10):78–87.
- Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54–75.
- Eisenstein, J. (2013). What to do about bad language on the internet. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013): Human Language Technologies (HLT)*, pages 359–369, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Eisenstein, J., O’Connor, B., Smith, N. A., and Xing, E. P. (2010). A Latent Variable Model for Geographic Lexical Variation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 1277–1287, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Eisenstein, J., Smith, N. A., and Xing, E. P. (2011). Discovering Sociolinguistic Associations with Structured Sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011): Human Language Technologies (HLT) - Volume 1*, pages 1365–1374, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Fagerland, M. W., Lydersen, S., and Laake, P. (2013). The mcnemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC Medical Research Methodology*, 13(1):91.
- Fonseca, E. R., Luís, J., and Rosa, G. (2013). Mac-Morpho Revisited: Towards Robust Part-of-Speech Tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 98–107.
- Foster, J., Çetinoglu, Ö., Wagner, J., Le Roux, J., Nivre, J., Hogan, D., and van Genabith, J. (2011). From News to Comment: Resources and Benchmarks for Parsing the Language of Web 2.0. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 893–901. Asian Federation of Natural Language Processing.
- Francis, W. N. and Kucera, H. (1979). Brown Corpus Manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.
- Freud, S. (1899). *Die Traumdeutung*. Franz Deuticke.
- Gamon, M. (2010). Using Mostly Native Data to Correct Errors in Learners’ Writing: A Meta-classifier Approach. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2010): Human Language Technologies (HLT)*, pages 163–171, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).

- Garrette, D. and Baldridge, J. (2013). Learning a Part-of-Speech Tagger from Two Hours of Annotation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2013): Human Language Technologies (HLT)*, pages 138–147, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Gärtner, K. (2002). Comprehensive Digital Text Archives: A Digital Middle High German Text Archive and its Perspectives. In *First EU/NSF Digital Libraries All Projects Meeting*.
- Gärtner, M., Thiele, G., Seeker, W., Björkelund, A., and Kuhn, J. (2013). ICARUS – An Extensible Graphical Search Tool for Dependency Treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013): System Demonstrations*, pages 55–60, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Geyken, A., Haaf, S., Jurish, B., Schulz, M., Thomas, C., and Wiegand, F. (2012). TEI und Textkorpora: Fehlerklassifikation und Qualitätskontrolle vor, während und nach der Texterfassung im Deutschen Textarchiv. In *Jahrbuch für Computerphilologie*.
- Giesbrecht, E. and Evert, S. (2009). Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus. In Alegria, I., Leturia, I., and Sharoff, S., editors, *Proceedings of the 5th Web as Corpus Workshop (WAC5)*.
- Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011): Human Language Technologies (HLT): Short Papers - Volume 2*, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Goldberg, Y., Adler, M., and Elhadad, M. (2008). EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start). In McKeown, K., Moore, J. D., Teufel, S., Allan, J., and Furu, S., editors, *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2008)*, pages 746–754, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Goldfarb, C. F. (1990). *The SGML Handbook*. Oxford University Press, Inc., New York, NY, USA.
- Gouws, S., Hovy, D., and Metzler, D. (2011). Unsupervised Mining of Lexical Variants from Noisy Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011), First Workshop on Unsupervised Learning in NLP*, pages 82–90, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Goyal, P., Mital, M. R., Mukerjee, A., Raina, A. M., Sharma, D., Shukla, P., and Vikram, K. (2003). A bilingual parser for Hindi, English and code-switching structures. In *Proceed-*

- ings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, page 15, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868.
- Haghighi, A. and Klein, D. (2006). Prototype-driven Learning for Sequence Models. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL 2006)*, HLT-NAACL 2006, pages 320–327, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Han, B. and Baldwin, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011): Human Language Technologies (HLT) - Volume 1*, HLT 2011, pages 368–378, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Han, B., Cook, P., and Baldwin, T. (2012). Automatically Constructing a Normalisation Dictionary for Microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP 2012) and Computational Natural Language Learning (CoNLL 2012)*, pages 421–432, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Han, B., Cook, P., and Baldwin, T. (2013). Lexical Normalization for Social Media Text. *ACM Transactions on Intelligent Systems and Technology*, 4(1):5:1–5:27.
- Hardmeier, C. (2016). A Neural Model for Part-of-Speech Tagging in Historical Texts. In *26th International Conference on Computational Linguistics (COLING 2016), Proceedings of the Conference: Technical Papers*, pages 922–931.
- Harpham, G. G. (2005). Beneath and Beyond the “Crisis in the Humanities”. *New Literary History*, 36(1):online.
- Haug, D. T. T. and Jøhndal, M. L. (2008). Creating a Parallel Treebank of the Old Indo-European Bible Translations. In Sporleder, C. and Ribarov, K., editors, *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34.
- Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT 2011)*, pages 187–197, Stroudsburg, PA, USA. Association for Computational Linguistics.

BIBLIOGRAPHY

- Hennings, T. (2003). *Einführung in das Mittelhochdeutsche*. De Gruyter Studienbuch. De Gruyter.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computing*, 9(8):1735–1780.
- Holmes, D. I. and Forsyth, R. S. (1995). The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing*, 10(2):111–127.
- Horner, P. J. (2006). *A Macaronic Sermon Collection from Late Medieval England: Oxford, MS Bodley 649*. Pontifical Institute of Mediaeval Studies Toronto: Studies and texts. Pontifical Institute of Mediaeval Studies.
- Horré, T. (1997). *Werther-Roman und Werther-Figur in der deutschen Prosa des Wilhelminischen Zeitalters*. PhD thesis, St. Ingbert.
- Hovy, E. and Lavid, J. (2010). Towards a ‘Science’ of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. *International Journal of Translation Studies*, 22(1):13–36.
- Hundt, M., Schneider, B., El-Assady, M., Keim, D. A., and Diehl, A. (2017). Visual Analysis of Geolocated Echo Chambers in Social Media. In Puig, A. P. and Isenberg, T., editors, *Proceedings of EuroVis 2017 - Posters*. The Eurographics Association.
- Hupkes, D. and Bod, R. (2016). POS-tagging of Historical Dutch. In Nicoletta Calzolari (Conference Chair) and Khalid Choukri and Thierry Declerck and Sara Goggi and Marko Grobelnik and Bente Maegaard and Joseph Mariani and Helene Mazo and Asuncion Moreno and Jan Odijk and Stelios Piperidis, editor, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Jain, N. and Bhat, R. A. (2014). Language Identification in Code-Switching Scenario. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 87–93.
- Jamatia, A., Gambäck, B., and Das, A. (2015). POS Tagging for Code-Mixed English-Hindi Twitter and Facebook Chat Messages. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2015)*, pages 239–248. INCOMA.
- Jannidis, F., Kohle, H., and Rehbein, M., editors (2017). *Digital Humanities: Eine Einführung*. Metzler, Weimar, 1 edition.
- Jefferson, J. A., Putter, A., and Hopkins, A. (2013). *Multilingualism in Medieval Britain (c. 1066-1520): Sources and Analysis*. Medieval texts and cultures of Northern Europe. Brepols.

- Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in NLP. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL 2007)*, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Jockers, M. L. (2013). *Macroanalysis: Digital Methods and Literary History*. Topics in the Digital Humanities. University of Illinois Press.
- Johansson, S. (1978). *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers*. ICAME collection of English language corpora. Univ., Department of English.
- Joshi, A. K. (1982). Processing of sentences with intra-sentential Code-Switching. In *Proceedings of International Conference on Computational Linguistics (COLING 1982)*, pages 145–150.
- Kaiser, L., Gomez, A. N., Shazeer, N., Vaswani, A., Parmar, N., Jones, L., and Uszkoreit, J. (2017). One Model To Learn Them All. cite arxiv:1706.05137.
- Kanfer, A. G., Haythornthwaite, C., Bruce, B. C., Bowker, G. C., Burbules, N. C., Porac, J. F., and Wade, J. (2000). Modeling Distributed Knowledge Processes in Next Generation Multidisciplinary Alliances. *Information Systems Frontiers*, 2(3):317–331.
- Kao, J. T. and Jurafsky, D. (2015). A computational analysis of poetic style: Imagism and its influence on modern professional and amateur poetry. *Linguistic Issues in Language Technology*, 12(3):1–31.
- Kaplan, F. (2015). A Map for Big Data Research in Digital Humanities. *Frontiers in Digital Humanities*, 2:1.
- Kaufmann, M. and Kalita, J. (2010). Syntactic normalization of Twitter messages. In *International Conference on Natural Language Processing (COLING 2010)*.
- Keller, M. (2017). Code-switched adjectives and adverbs in Macaronic sermons. *Studies in Language Variation and Change 2: Shifts and Turns in the History of English*, page 17.
- Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125.
- Kestemont, M., Peersman, C., Decker, B. D., Pauw, G. D., Luyckx, K., Morante, R., Vaassen, F., van de Loo, J., and Daelemans, W. (2012). The Netlog Corpus. A Resource for the Study of Flemish Dutch Internet Language. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1569–1572, Paris, France. European Language Resources Association (ELRA).

BIBLIOGRAPHY

- Kirsch, A. (2014). Technology Is Taking Over English Departments: The false promise of the digital humanities. *New Republic*. <https://newrepublic.com/article/117428/limits-digital-humanities-adam-kirsch>.
- Kirschenbaum, M. (2014). What Is “Digital Humanities,” and Why Are They Saying Such Terrible Things about It? *Differenes*, 25(1):46–63.
- Klein, T. and Dipper, S. (2016). Handbuch zum Referenzkorpus Mittelhochdeutsch. Technical report, University of Bochum. <https://www.linguistics.rub.de/rem/documentation/index.html>.
- Kobus, C., François, Y., and Géraldine, D. (2008a). Transcrire les SMS comme on reconnaît la parole. In *Actes de la Conférence sur le Traitement Automatique des Langues (TALN2008)*, pages 128–138.
- Kobus, C., Yvon, F., and Damnati, G. (2008b). Normalizing SMS: Are Two Metaphors Better Than One? In *Proceedings of the International Conference on Computational Linguistics (COLING 2008)*, pages 441–448.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL 2007) on Interactive Poster and Demonstration Sessions*, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Kolak, O. and Resnik, P. (2002). OCR Error Correction Using a Noisy Channel Model. In *Proceedings of the Second International Conference on Human Language Technology Research, HLT 2002*, pages 257–262, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Kroch, A. and Taylor, A. (2000). The Penn-Helsinki Parsed Corpus of Middle English (PPCME2). Department of Linguistics, University of Pennsylvania. CD-ROM, second edition, release 4 (<http://www.ling.upenn.edu/ppche-release-2016/PPCME2-RELEASE-4>).
- Kuhn, J. and Reiter, N. (2015). A Plea for a Method-Driven Agenda in the Digital Humanities. In *Book of Abstracts of Digital Humanities 2015*.
- Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- LeBlanc, M. D. (2017). Toward Reproducibility in DH Experiments: A Case Study in Search of Edgar Allan Poe’s First Published Work. In *Book of Abstracts of Digital Humanities 2017*.

- Lebret, R. and Lebret, R. (2013). Word emdeddings through hellinger PCA. *Computing Research Repository*, abs/1312.5542:482–490.
- Li, C. and Liu, Y. (2012). Improving Text Normalization using Character-Blocks Based Models and System Combination. In *Proceedings of the International Conference on Computational Linguistics (COLING 2012)*, pages 1587–1602, Mumbai, India. The COLING 2012 Organizing Committee.
- Li, C. and Liu, Y. (2014). Improving Text Normalization via Unsupervised Model and Discriminative Reranking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2014), Student Research Workshop*, pages 86–93, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Lin, C., Ammar, W., Dyer, C., and Levin, L. S. (2015). Unsupervised POS Induction with Word Embeddings. *Computing Research Repository*, abs/1503.06760:1311–1316.
- Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2013). Paraphrasing 4 Microblog Normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 73–84. Association for Computational Linguistics (ACL).
- Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fernandez, R., Amir, S., Marujo, L., and Luís, T. (2015). Finding function in form: Compositional character models for open vocabulary word representation. In Márquez, L., Callison-Burch, C., Su, J., Pighin, D., and Marton, Y., editors, *Proceedings of the Conference on Empirical Methods in Computational Linguistics (EMNLP 2015)*, pages 1520–1530, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Liu, F., Weng, F., and Jiang, X. (2012). A Broad-Coverage Normalization System for Social Media Language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1035–1044, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Liu, X., Wei, F., Zhang, S., and Zhou, M. (2013). Named Entity Recognition for Tweets. *ACM Transactions on Intelligent Systems and Technology*, 4(1):3:1–3:15.
- Llobet, R., Cerdan-Navarro, J.-R., Perez-Cortes, J.-C., and Arlandis, J. (2010). OCR Post-processing Using Weighted Finite-State Transducers. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR 2010)*, pages 2021–2024, Washington, DC, USA. IEEE Computer Society.
- Lyu, D. and Lyu, R. (2008). Language identification on code-switching utterances using multiple cues. In *9th Annual Conference of the International Speech Communication Association (INTERSPEECH 2008)*, pages 711–714.

BIBLIOGRAPHY

- Manaris, B. (1998). Natural Language Processing: A Human-Computer Interaction Perspective. *Advances in Computers*, 47:1 – 66.
- Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Massachusetts Institute of Technology.
- Marche, S. (2012). Literature Is Not Data: Against Digital Humanities. <https://lareviewofbooks.org/essay/literature-is-not-data-against-digital-humanities/>.
- Martens, L. (1985). *The diary novel*. Cambridge University Press Cambridge.
- Martínez-Cámara, E., Martín-Valdivia, M. T., Molina-González, M. D., and Perea-Ortega, J. M. (2014). Integrating Spanish lexical resources by meta-classifiers for polarity classification. *Journal of Information Science*, 40(4):538–554.
- McClosky, D. (2010). *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. PhD thesis, Brown University, Providence, RI, USA. AAI3430199.
- McGillivray, B., Passarotti, M., and Ruffolo, P. (2009). The Index Thomisticus Treebank Project: Annotation, Parsing and Valency Lexicon. *TAL*, 50:103–127.
- McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.
- Melero, M., Costa-Jussà, M. R., Domingo, J., Marquina, M., and Quixal, M. (2012). Holaaa!! writin like u talk is kewl but kinda hard 4 NLP. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3794–3800, Paris, France. European Language Resources Association (ELRA).
- Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.
- Mihalcea, R. (2004). Co-training and Self-training for Word Sense Disambiguation. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2004)*, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Mihov, S., Schulz, K. U., Ringlstetter, C., Dojchinova, V., and Nakova, V. (2005). A Corpus for Comparative Evaluation of OCR Software and Postcorrection Techniques. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR 2005)*, pages 162–166.

- Mikolov, T., Yih, W., and Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013): Human Language Technologies (HLT)*, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Milli, S. and Bamman, D. (2016). Beyond Canonical Texts: A Computational Analysis of Fanfiction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2048–2053, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Mittelhochdeutsche Begriffsdatenbank (1992-2017). Mittelhochdeutsche Begriffsdatenbank (MHDBDB). <http://www.mhdbdb.sbg.ac.at/>.
- Moon, T. and Baldridge, J. (2007). Part-of-Speech Tagging for Middle English through Alignment and Projection of Parallel Diachronic Texts. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP 2007) and Computational Natural Language Learning (CoNLL 2007)*, pages 390–399, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Moretti, F. (2013). *Distant Reading*. Verso Books.
- Moser, H., editor (1977). *Des Minnesangs Frühling: Unter Benutzung der Ausgaben von Karl Lachmann und Moriz Haupt, Friedrich Vogt und Carl von Kraus. Bearbeitet von Hugo Moser und Helmut Tervooren*. S. Hirzel, 36 edition.
- Müller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Murr, S. and Barth, F. (2017). Digital Analysis Of The Literary Reception Of J.W. V. Goethe's Die Leiden Des Jungen Werthers. In *Book of Abstracts of Digital Humanities 2017*.
- Murugesan, S. (2007). Understanding Web 2.0. *IT Professional*, 9(4):34–41.
- Muysken, P. (2000). *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press.
- Myers-Scotton, C. (1993). *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press.
- Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

BIBLIOGRAPHY

- Nivre, J., Agić, Ž., Aranzabe, M. J., Asahara, M., Atutxa, A., Ballesteros, M., Bauer, J., Bengoetxea, K., Bhat, R. A., Bosco, C., et al. (2015). Universal Dependencies 1.2. <http://universaldependencies.org/>.
- Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A Multilingual Treebank Collection. In Chair), N. C. C., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Nurmi, A. and Pahta, P. (2013). Multilingual practices in the language of the law: Evidence from the Lampeter corpus. In Jukka Tyrkkö, O. T. and Salenius, M., editors, *Ex Philologia Lux: Essays in Honour of Leena Kahlas-Tarkka (Mémoires de la Société Néophilologique de Helsinki XC)*, pages 187–205. Société Néophilologique.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.
- O’Connor, B., Krieger, M., and Ahn, D. (2010). TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *Proceedings of the Fourth International Conference on Artificial Intelligence (AIII) Conference on Weblogs and Social Media*, Washington, DC, USA. The AAAI Press.
- O’Halloran, K., Chua, A., and Podlasov, A. (2014). The Role of Images in Social Media Analytics: A Multimodal Digital Humanities Approach. *Visual Communication*, pages 565–588.
- Oostdijk, N. (2000). The Spoken Dutch Corpus. Overview and First Evaluation. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)*, pages 887–893, Paris, France. European Language Resources Association (ELRA).
- Oostdijk, N. (2008). SoNaR: STEVIN Nederlandstalig Referentiecorpus.
- Paltoglou, G. and Thelwall, M. (2012). Twitter, MySpace, Digg: Unsupervised Sentiment Analysis in Social Media. *ACM Transactions on Intelligent Systems and Technology*, 3(4):66:1–66:19.
- Peersman, C., Daelemans, W., and Van Vaerenbergh, L. (2011). Predicting Age and Gender in Online Social Networks. In *Proceedings of the 3rd International Workshop on Search and Mining User-generated Contents, SMUC 2011*, pages 37–44, New York, NY, USA. ACM.
- Pennell, D. L. and Liu, Y. (2011). A character-level machine translation approach for normalization of SMS abbreviations. In *Proceedings of 5th International Joint Conference on Natural*

- Language Processing*, pages 974–982, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Pérez-Cortes, J. C., Amengual, J., Arlandis, J., and Llobet, R. (2000). Stochastic Error-Correcting Parsing for OCR Post-Processing. In *15th International Conference on Pattern Recognition (ICPR 2000)*, pages 4405–4408.
- Petrov, S., Das, D., and McDonald, R. (2012). A Universal Part-of-Speech Tagset. In Chair), N. C. C., Choukri, K., Declerck, T., Doan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, Paris, France. European Language Resources Association (ELRA).
- Piper, A. and Algee-Hewitt, M. (2014). The Werther Effect I: Goethe Topologically. *Distant Readings: Topologies of German Culture in the Long Nineteenth Century*, pages 155–184.
- Plank, B. (2016). What to do about non-standard (or non-canonical) language in NLP. In Dipper, S., Neubarth, F., and Zinsmeister, H., editors, *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, volume 16 of *BLA: Bochumer Linguistische Arbeitsberichte*, pages 13–20, Bochum, Germany.
- Poplack, S. (1980). Sometimes I'll start a sentence in Spanish y termino en Espanol: toward a typology of code-switching. *Linguistics*, 18(7-8):581–618.
- Propp, V. (1968). *Morphology of the folktale*. Publication of the Indiana University Research Center in Anthropology, Folklore, and Linguistics. University of Texas Press.
- Ranzato, M. A., Boureau, Y.-L., and LeCun, Y. (2007). Sparse Feature Learning for Deep Belief Networks. In *Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS2007*, pages 1185–1192, USA. Curran Associates Inc.
- Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)*, pages 133–142, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Ravi, S. and Knight, K. (2009). Minimized Models for Unsupervised Part-of-speech Tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL 2009) and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, pages 504–512, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Reffle, U. and Ringlstetter, C. (2013). Unsupervised Profiling of OCRed Historical Documents. *Pattern Recognition*, 46(5):1346–1357.

- Renear, A. (2004). Text Encoding. In Schreibman, S., Siemens, R., and Unsworth, J., editors, *A Companion to Digital Humanities*. Oxford: Blackwell.
- Reynaert, M. (2008a). All, and only, the Errors: more Complete and Consistent Spelling and OCR-Error Correction Evaluation. In Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008)*, Paris, France. European Language Resources Association (ELRA).
- Reynaert, M. (2008b). Non-interactive OCR Post-correction for Giga-scale Digitization Projects. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2008)*, pages 617–630, Berlin, Heidelberg. Springer-Verlag.
- Reynaert, M., Oostdijk, N., Clercq, O. D., van den Heuvel, H., and de Jong, F. (2010). Balancing SoNaR: IPR versus Processing Issues in a 500-Million-Word Written Dutch Reference Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2693–2698, Paris, France. European Language Resources Association (ELRA).
- Riegner, C. (2007). Word of mouth on the web: the impact of Web 2.0 on consumer purchase decisions. *Journal of advertising research.*, 47(4):436–437.
- Ringstetter, C., Schulz, K. U., and Mihov, S. (2007). Adaptive Text Correction with Web-crawled Domain-dependent Dictionaries. *ACM Transactions on Speech Language Processing*, 4(4):9–9:36.
- Ritter, A., Clark, S., Mausam, and Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1524–1534, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Rockwell, G. (2012). Short Guide To Evaluation Of Digital Work. *Journal of Digital Humanities*, 1(4):online.
- Rosenthal, S. and McKeown, K. (2011). Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-social Media Generations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT 2011*, pages 763–772, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Royen, K. V., Poels, K., Daelemans, W., and Vandebosch, H. (2015). Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability. *Telematics and Informatics*, 32(1):89 – 97.

- Ryder, F. (1962). *The Song of the Nibelungs*. Wayne State University Press.
- Sánchez-Martínez, F., Armentano-Oller, C., Pérez-Ortiz, J. A., and Forcada, M. L. (2007). Training Part-of-Speech Taggers to build Machine Translation Systems for Less-Resourced Language Pairs. *Procesamiento del Lenguaje Natural*, pages 257–264.
- Schendl, H. and Wright, L. (2012). *Code-Switching in Early English*. Topics in English Linguistics (TiEL). De Gruyter.
- Scherpe, K. (1970). *Werther und Wertherwirkung: zum Syndrom bürgerlicher Gesellschaftsordnung im 18. Jahrhundert*. *Werther und Wertherwirkung: Zum Syndrom bürgerlicher Gesellschaftsordnung im 18. Jahrhundert*. Gehlen.
- Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1995). Vorläufige Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, and Seminar für Sprachwissenschaft, Universität Tübingen.
- Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, pages 44–49.
- Schmid, H. (1995). Improvements In Part-of-Speech Tagging With an Application To German. In *In Proceedings of the Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 1995), SIGDAT-Workshop*, pages 47–50, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Schmidt, D. (2012). The Role of Markup in the Digital Humanities. *Historical Social Research / Historische Sozialforschung*, 37(3 (141)):125–146.
- Schneider, G. and Volk, M. (1998). Adding manual constraints and lexical look-up to a Brill-tagger for German. In *Proceedings of the ESSLLI-98 Workshop on Recent Advances in Corpus Annotation*.
- Schöch, C. (2013). Big? Smart? Clean? Messy? Data in the Humanities. *Journal of the Digital Humanities*, 2(3):2–13.
- Schulz, S. (2014). Named Entity Recognition for User-Generated Content. Proceedings of the Student Session of ESSLLI 2014, unpublished, online: <http://www.kr.tuwien.ac.at/drm/dehaan/stus2014/proceedings.pdf>.
- Schulz, S. and Keller, M. (2016). Code-Switching Ubiquitous - Language Identification and Part-of-Speech Tagging for Historical Mixed Text. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2016)*, pages 43–51, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).

- Schulz, S. and Ketschik, N. (2017). From 0 to 10 Million Annotated Words – Part-of-Speech Tagging for Middle High German. Manuscript University of Stuttgart. Submitted for review.
- Schulz, S. and Kuhn, J. (2016). Learning from Within? Comparing PoS Tagging Approaches for Historical Text. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Schulz, S. and Kuhn, J. (2017). Multi-modular domain-tailored OCR post-correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2706–2716, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Schulz, S., Pauw, G. D., Clercq, O. D., Desmet, B., Hoste, V., Daelemans, W., and Macken, L. (2016). Multimodular Text Normalization of Dutch User-Generated Content. *ACM Transactions on Intelligent Systems and Technology*, 7(4):61:1–61:22.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA.
- Shannon, C. (1948). A mathematical theory of communication. *Bell system technical journal*, 27:379–423.
- Sharma, A., Gupta, S., Motlani, R., Bansal, P., Shrivastava, M., Mamidi, R., and Sharma, D. M. (2016). Shallow Parsing Pipeline - Hindi-English Code-Mixed Social Media Text. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pages 1340–1345, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Siemens, L. (2009). ‘It’s a team if you use “reply all”’: An exploration of research teams in digital humanities environments. *Literary and Linguistic Computing*, 24(2):225–233.
- Sinha, R. M. K. and Thakur, A. (2005). Machine Translation of Bi-lingual Hindi-English (Hinglish) Text. In *Proceedings of the MT Summit*.
- Smith, N. A. and Eisner, J. (2005). Contrastive Estimation: Training Log-linear Models on Unlabeled Data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*, pages 354–362, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Smith, R. and Inc, G. (2007). An overview of the Tesseract OCR Engine. In *Proceedings of the 9th IEEE International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 629–633.

- Søgaard, A. (2010). Simple Semi-supervised Training of Part-of-speech Taggers. In *Proceedings of the Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2010) Conference Short Papers*, pages 205–208, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Solorio, T. and Liu, Y. (2008a). Learning to Predict Code-switching Points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 973–981, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Solorio, T. and Liu, Y. (2008b). Part-of-speech Tagging for English-Spanish Code-switched Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 1051–1060, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Sparck Jones, K. and Galliers, J. R. (1996). *Evaluating Natural Language Processing Systems: An Analysis and Review*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Sproat, R., Black, A. W., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. (2001). Normalization of non-standard words. *Computer, Speech and Language*, 15(3):287–333.
- Steiner, C. M., Agosti, M., Sweetnam, M. S., Hillemann, E.-C., Orio, N., Ponchia, C., Hampson, C., Munnely, G., Nussbaumer, A., Albert, D., et al. (2014). Evaluating a digital humanities research environment: the CULTURA approach. *International Journal on Digital Libraries*, 15(1):53–70.
- Strange, C., McNamara, D., Wodak, J., and Wood, I. (2014). Mining for the Meanings of a Murder: The Impact of OCR Quality on the Use of Digitized Historical Newspapers. *Digital Humanities Quarterly*, 8(1).
- Strohmaier, C. M., Ringlstetter, C., Schulz, K. U., and Mihov, S. (2003). Lexical Postcorrection of OCR-Results: The Web as a Dynamic Secondary Dictionary? In *Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2-Volume Set, 3-6 August 2003, Edinburgh, Scotland, UK*, pages 1133–1137.
- Takahashi, H., Itoh, N., Amano, T., and Yamashita, A. (1990). A spelling correction method and its application to an OCR system. *Pattern Recognition*, 23(3-4):363–377.
- Taylor, P., Black, A., and Caley, R. (1998). The Architecture of the Festival Speech Synthesis System. In *Proceedings Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 147–151, Blue Mountains, Australia. International Speech Communication Association.
- Thelwall, M. and Delgado, M. M. (2015). Arts and humanities research evaluation: no metrics please, just data. *Journal of Documentation*, 71(4):817–833.

- Tiedemann, J. (2012). Character-based Pivot Translation for Under-resourced Languages and Domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2012*, pages 141–151, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Tong, X. and Evans, D. A. (1996). A Statistical Approach to Automatic OCR Error Correction in Context. In *Proceedings of the Fourth Workshop on Very Large Corpora*, pages 88–100.
- Toribio, A. J. and Bullock, B. E. (2012). *The Cambridge handbook of linguistic code-switching*. Cambridge University Press.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, ACL 2010, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Unsworth, J. (2000). Scholarly Primitives: What Methods Do Humanities Researchers Have in Common, and How Might Our Tools Reflect This? Humanities Computing: Formal Methods, Experimental Practice, King’s College symposium, London.
- van de Kauter, M., Coorman, G., Lefever, E., Desmet, B., Macken, L., and Hoste, V. (2013). LeTs Preprocess: The multilingual LT3 linguistic preprocessing toolkit. *Computational Linguistics in the Netherlands Journal*, 3:103–120.
- Van Dijk, J. (2009). Users like you? Theorizing agency in user generated content. *Media, Culture & Society*, 31(1):41–58.
- Van Hee, C., Lefever, E., Verhoeven, B., Mennes, J., Desmet, B., De Pauw, G., Daelemans, W., and Hoste, V. (2015). Detection and fine-grained classification of cyberbullying events. In *Proceedings of the 10th Recent Advances in Natural Language Processing (RANLP 2015)*, Hissar, Bulgaria.
- VandeKerckhove, R. and Nobels, J. (2010). Code eclecticism: Linguistic variation and code alternation in the chat language of Flemish teenagers. *Journal of Sociolinguistics*, 14:657–677.
- Vannestål, M. (2004). *Syntactic Variation in English Quantified Noun Phrases with All, Whole, Both and Half*. Acta Wexionensia. Växjö University Press.
- Vlachos, A. (2011). Evaluating unsupervised learning for natural language processing tasks. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 2011), Workshop on Unsupervised Learning in NLP*, pages 35–42, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).

- Vobl, T., Gotscharek, A., Reffle, U., Ringlstetter, C., and Schulz, K. U. (2014). PoCoTo - an Open Source System for Efficient Interactive Postcorrection of OCRed Historical Texts. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage (DATECH 2014)*, pages 57–61, New York, NY, USA. ACM.
- Volk, M., Marek, T., and Sennrich, R. (2010). Reducing OCR errors by combining two OCR systems. In Sporleder, C. and Zervanou, K., editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2010)*, pages 61–65.
- vor der Brück, T. and Mehler, A. (2016). TLT-CRF: A lexicon-supported morphological tagger for Latin based on conditional random fields. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, LREC 2016, pages 1514–1519, Paris, France. European Language Resources Association (ELRA).
- Vyas, Y., Gella, S., Sharma, J., Bali, K., and Choudhury, M. (2014). POS Tagging of English-Hindi Code-Mixed Social Media Content. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP 2014)*, pages 974–979, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Wagner, R. A. and Fisher, M. J. (1974). The string-to-string correction problem. *Journal of ACM*, 21(1):168–173.
- Wang, P. and Ng, H. T. (2013). A Beam-Search Decoder for Normalization of Social Media Text with Application to Machine Translation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013): Human Language Technologies (HLT)*, pages 471–481, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Warf, B. and Arias, S. (2008). *The Spatial Turn: Interdisciplinary Perspectives*. Routledge Studies in Human Geography. Taylor & Francis.
- Warwick, C., Galina, I., Terras, M., Huntington, P., and Pappa, N. (2008). The master builders: LAIRAH research on good practice in the construction of digital humanities projects. *Literary and Linguistic Computing*, 23(3):383 – 396.
- Wenzel, S. (1994). *Macaronic sermons: bilingualism and preaching in late-medieval England*. Recentiores : Later Latin Texts and Contexts. University of Michigan Press.
- Wing, B. P. and Baldrige, J. (2011). Simple Supervised Document Geolocation with Geodesic Grids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ALC 2011): Human Language Technologies (HLT) - Volume 1*, HLT 2011, pages 955–964, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).

BIBLIOGRAPHY

- Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, 5:241–259.
- Worthey, G. (2014). Why are such terrible things written about DH? Kirsch v. Kirschenbaum. *Blog entry*. <https://digitalhumanities.stanford.edu/why-are-such-terrible-things-written-about-dh-kirsch-v-kirschenbaum>.
- Xue, Z., Yin, D., and Davison, B. D. (2011). Normalizing Microtext. In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AIII-11), Workshop on Analyzing Microtext*, volume WS-11-05, pages 74–79, San Francisco, USA. AAAI.
- Yang, Y. and Eisenstein, J. (2013). A Log-Linear Model for Unsupervised Text Normalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 61–72, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Yang, Y. and Eisenstein, J. (2015). Unsupervised Multi-Domain Adaptation with Feature Embeddings. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2015): Human Language Technologies (HLT)*, pages 672–682, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Yang, Y. and Eisenstein, J. (2016). Part-of-Speech Tagging for Historical English. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016): Human Language Technologies (HLT)*, pages 1318–1328, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL 1995)*, ACL 1995, pages 189–196, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Yeong, Y. and Tan, T. (2011). Applying Grapheme, Word, and Syllable Information for Language Identification in Code Switching Sentences. In *International Conference on Asian Language Processing (IALP 2011)*, pages 111–114.
- Yvon, F. (2010). Rewriting the orthography of SMS messages. *Natural Language Engineering*, 16(2):133–159.
- Zeldes, A. and Schroeder, C. T. (2015). Computational Methods for Coptic: Developing and Using Part-of-Speech Tagging for Digital Scholarship in the Humanities. *Digital Scholarship in the Humanities*, 30(suppl_1):i164.
- Zeman, D. and Resnik, P. (2008). Cross-language parser adaptation between related languages. *NLP for Less Privileged Languages*, pages 35–42.

- Zhang, C., Baldwin, T., Ho, H., Kimelfeld, B., and Li, Y. (2013). Adaptive Parser-Centric Text Normalization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013), Volume 1: Long Papers*, pages 1159–1168, Stroudsburg, PA, USA. Association for Computational Linguistics (ACL).
- Zhou, Z.-H. and Li, M. (2005). Tri-Training: Exploiting Unlabeled Data Using Three Classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541.
- Ziegler, A. and Braun, C. (2010). *Historische Textgrammatik und Historische Syntax des Deutschen. 2 Bände: Traditionen, Innovationen, Perspektiven / Diachronie, Althochdeutsch, Mittelhochdeutsch; Frühneuhochdeutsch, Neuhochdeutsch*. De Gruyter, Incorporated.
- Zuccala, A. (2013). Evaluating the Humanities: Vitalizing ‘the forgotten sciences’. *Research Trends*, pages 3–6.
- Zweig, L., Liu, C., Hiraga, M., Reed, A., Czerniakowski, M., Dickinson, M., and Kübler, S. (2017). FunTube: Annotating Funniness in YouTube Comments. In *Proceedings of the International Workshops on Treebanks and Linguistic Theories (TLT 2017), Workshop Corpora in the Digital Humanities (CDH)*.

