A SEMI-AUTOMATIC PART-OF-SPEECH TAGGING SYSTEM FOR MIDDLE ENGLISH CORPORA: OVERCOMING THE CHALLENGES

Abstract

Historical corpus annotation is very much a manual, time-consuming task. The last few years have witnessed advances in the use of computational tools for the annotation of Middle English corpora. In 2007 an attempt at creating a semi-automatic system for part-of-speech (POS) tagging, based on the use of parallel texts, was developed at the University of Texas. Although this work still revealed manual annotation to be more accurate, it proved the potential of computational tools for the creation of tagging systems. We propose the development of a semi-intelligent and semi-automatic POS tagging program for ME corpora capable of tagging any given ME text with a high rate of success; no such computational system is currently available. This task entails challenges of a two-fold nature: a) linguistic difficulties; and b) computational limitations. This paper discusses these difficulties and provides possible solutions to them in order to create a tool that will facilitate POS tagging and help searching for linguistic information. **Keywords**: POS tagging, Middle English, historical corpora, computational linguistics.

Resumen

La anotación de corpus históricos es en gran medida una tarea manual y laboriosa. Los últimos años han sido testigos de muchos avances en el uso de herramientas computacionales para el etiquetado de corpus de inglés medio. En el 2007 la Universidad de Texas desarrolló un sistema semi-automático de etiquetado morfológico basado en el uso de textos paralelos y, aunque el estudio siguió revelando que el etiquetado manual era más preciso, demostró el potencial de las herramientas computacionales para la creación de sistemas de etiquetado. Proponemos el desarrollo de un etiquetador morfológico semi-inteligente y semi-automático para corpora de inglés medio capaz de etiquetar cualquier texto con mucha precisión; actualmente, no disponemos de tal sistema. Esta tarea supone desafíos tanto lingüísticos como computacionales. Este artículo analiza estos problemas y ofrece posibles soluciones al objeto de crear una herramienta que facilite el etiquetado morfológico y ayude en la búsqueda de información lingüística. **Palabras clave**: etiquetado morfológico, Inglés medio, corpora históricos, lingüística computacional.

1 Introduction

Electronic corpora are almost inexhaustible sources of linguistic knowledge. However, without the appropriate annotations most of this information would be as lost as a needle in a haystack. Part-of-speech (henceforth POS) annotation/tagging is undoubtedly the most common type of corpus annotation, simply because it stands as the basis of all corpus studies. Assigning POS-tags to raw corpora is essential for performing further analyses, such as syntactic parsing and semantic field annotation (McEnery and Wilson 1997), and furthermore to perform collocation studies and obtain word frequency lists, among others. All this is of great help in fields such as lexicography and language teaching and learning.

Many automatic POS taggers are available on-line nowadays that can tag large amounts of raw text in a matter of seconds. However, this task was entirely manual prior to 1971, when Greene and Rubin developed TAGGIT, the first POS tagging program. Although the TAGGIT system was very primitive and at first could guarantee a success rate of just 71%, many different systems have been developed over the years, each one providing new improvements, such as CLAWS (1983), developed by UCREL at Lancaster; the Brill Tagger (1993); or GENIA (2006), which also performs shallow parsing, and named-entity recognition for biomedical texts. Actually, the people behind CLAWS, which served to tag the famous BNC corpus, worked for a number of years on improving the system ever since it was developed in 1983, and by 1994 it could already claim a success rate of up to 97–98%. In light of this, most computational linguists today consider the automatic POS tagging process to be a close case, and although there is still much controversy as to what extent it is actually entirely solved (see Giesbrecht and Evert's 2009 discussion on the nature of five current German tagging systems). It is a fact that if, as Wolfgang Fischel claims, "human annotators agree in just 96% of the cases" (2009: 7) and this is the same percent of success that an automatic tagger can feasibly achieve on average, then the remaining percentage can be attributed to "the ambiguity in the language itself" (2009: 7) and not, therefore, on the tagging programme's limitations. But all in all, and bearing this in mind, we could easily consider the task of automatic POS tagging of English texts to be virtually resolved.

On the other hand, English historical corpora has lagged behind its modern counterpart: in fact, not until the last few decades has historical linguistics even become "strictly corpus-based". The common procedure was to take "a selective approach to empirical data" and simply to "look for evidence of a particular phenomena [...] making rough estimates at frequency" (McEnery and Wilson 1997). However, since 1984, when the *Helsinki Corpus of English Texts: Diachronic and Dialectal* (the most famous historical corpus of English) was compiled, many other historical corpora have been developed or are currently in the making: the *Innsbruck Computer Archive of Middle English Texts* (1994), the *Corpus of Early Middle English Tagged Texts and Maps* (1997), or the *Corpus of Early English Correspondence* (1998), to quote but a few.

Here in Spain we should mention the *Coruña Corpus*, developed at its namesake University, and *The Corpus of Late Middle English Scientific Prose,* currently being compiled with the collaboration of the Universities of Málaga, Oviedo, Murcia, Jaén and Glasgow. While similar in scope (the two of them deal with scientific English prose), there are also important differences between both projects. Most importantly, the former corpus is tagged and diachronic, while the latter is POS-annotated and synchronic. The *Malaga Corpus*, as we can call it for short, pursues the electronic editing of the Middle English material housed in the Hunterian Collection at Glasgow University Library. This corpus currently holds approximately 250,000 words, and the final target is to reach no less than half a million words.

In view of the late development of the creation of historical corpora, it stands to reason that the development of automatised POS tagging systems for such corpora is dilatory. The current state of art reveals only two attempts at creating an automatic system for the automatic POS tagging of English historical texts, the first regarding Old English corpora and the second dealing with the tagging of Middle English texts, which is in fact the object of our present study.

A part-of-speech tagger for OE was developed at Zurich University (Switzerland) by Beni Ruef; it consisted in a rule-based tagging system following transformational-based learning.[1] A manually tagged corpus of 108,000 words was employed for training the program into learning the rules of the language. The total rate of successful tagged words was of 88.5% (91.5% accuracy for known tokens and 56.5% for unknown tokens). As we can see, the main problem this system had is that it could not recognise items that had not previously been included during the training

[1] See also Miranda-García *et al.* 2000 and 2001 on the implementation of a POS tagger of OE, developed at the University of Málaga.

process. In turn, the ME POS tagging system, developed by the University of Texas at Austin in 2007, attempted to create a semi-automatised tagger for ME based on the alignment of already tagged parallel contemporary English texts. The parallel texts chosen were excerpts taken from the Bible. This new tagger was trained using the modern tagged texts as basis; through multiple alignment with the ME texts the appropriate tag was to be identified. Moreover, to ensure a higher rate of success and further automatisation a bigram tagger was trained on these alignments. Finally, the C&C (Curran and Clark) maximum entropy tagger, which was initially employed to tag the modern version of the Bible, was then bootstrapped onto the ME text which had been, in turn, tagged by the trained bigram tagger (Moon and Baldridge 2007: 393). They also attempted the use of unsupervised bootstrap methods to train the tagger without previously having to tag the texts manually. However results revealed that "a manually annotated training set of 400–800 sentences surpassed our best bootstrapped tagger". Overall, their methods managed to obtain "an accuracy of 84%" (Moon and Baldridge 2007: 391). Note further that this method presents a clear limitation as it relies on the existence of a text written in two languages and, consequently, proves unfeasible for the tagging of ME texts that do not have modernised equivalents, which is our case.

As we can see, even though the last few years have witnessed this important advance in the development of a semi-automatic system for POS tagging for historical texts, much work still needs to be done before we can plead success. Nevertheless, these findings are encouraging as to the potential of computational tools for the creation of tagging systems and will undoubtedly set the ground-work for the development of a system of these characteristics.

We propose to devise a semi-intelligent and semi-automatic part-of-speech tagging program for ME corpora that is capable of tag any given ME text successfully with a very high rate of success, much more than any computational system of similar characteristics that we know of is currently able. However, this task confronts us with several challenges.

If contemporary English POS taggers pose problems when it comes to ambiguity and unknown words (among others), the range of difficulties encountered for the creation of an automatic POS tagger for ME words is considerably wider due to the nature of the language—mainly its orthographical variation. Consequently, we face challenges of a twofold nature: (*a*) linguistic difficulties; and (*b*) computational limitations. The present paper discusses the nature of each of these difficulties and provides solutions, whenever possible, to overcome them, in order to create a useful tool that will facilitate the POS tagging process and, therefore, help the linguist's search for linguistic information.[2]

The present paper is organised into 4 different sections. Section 2 deals with the challenges and is, accordingly, divided into two different subsections. The first (2.1) enumerates and discusses linguistic difficulties, and the second (2.2) accounts for computational limitations. Section 3, in turn, provides the possible solutions to overcome the difficulties enumerated in the previous chapter. And finally, section 4 provides the conclusions.

2 The challenges

Before we begin to discuss the difficulties posed by the design of a semi-automatic ME tagger, we consider it important to highlight and establish, if at a very basic level, the main steps involved for the creation of an algorithm for any automatic POS tagger, regardless of the language. We follow Wolfgang Fischl's summary for the task. He divides the process into three basic steps. First comes *tokenization*, wherein "the text is divided into tokens", including "end-of-sentence punctuation marks and word-like units". *Ambiguity look-up* then follows. Here each token that has been previously identified will be provided with a number of "possible part of speech tags". For example, ME *bath* would be tagged initially as both a noun and a verb. The final step is *disambiguation*: every word that has been assigned more than one tag in the previous phase will be given a single, correct tag. The program will have to choose the correct POS tag and assign it to the token in question. Homonyms and polysemic words

[2] The present research, entitled *Etiquetador Morfológico (POS-tagger) de Inglés Medio*, is funded by the Autonomous Government of Andalusia (project P09-HUM–4790). This grant is hereby gratefully acknowledged.

are particularly bound to undergo the latter process. This complicated task can be solved by using two different types of taggers: rule-based and stochastic ones (Fischl 2009: 2). Section 2.2, which deals with computational matters, expounds further on the nature of these two systems.

2.1 Linguistic difficulties

Spelling and word formation in the ME period was irregular due to a lack of standardisation in the language. Moreover, manuscripts were often compiled by different scribes or written by the same scribe but compiled from several different sources, many belonging to different dialects or even different languages, mainly Latin. The following section presents some of the linguistic problems that one encounters when dealing with ME texts.

2.1.1 Choosing the transcription

The first step to build a successful POS tagger for ME begins at the level of transcription. Depending on the type of transcription we are working with, the possibility of it being "taggable" will be more or less feasible, will be accomplished automatically or manually. We provide three different models of transcription below in order to ascertain, according to their specific features, whether they would be compatible with a semi-automatic POS tagging system.

Let us begin by considering a graphetic diplomatic transcription, maintaining the text as originally written by the scribe insofar as it not only preserves the original spelling, emendations and other scribal mistakes, but also reproduces the abbreviation symbols without expanding them. This model will normally also maintain the original punctuation as well. This type of transcription is completely incompatible will POS tagging, manual or automatic. First of all, it is graphetic and so reproduces every distinct letter type, resulting in a number of different graphs to represent the same letters. For example, in Fig. 1 below, we have an instance of two letter ⟨r⟩ shapes found within the same word. Due to the great number of symbols that the tagger would have to learn, it would be highly time-consuming and not at all practical. Furthermore, if no expansions are provided for the many abbreviations that appear across the witnesses, POS tagging is an impossibility even in manual tagging, as even if the tagger could be trained into understanding the different symbols, this would only be possible if there were a one-to-one correspondence between the symbols and the letters they represented during the period—which, of course, is not the case. The same symbols are frequently found to stand for different letters, not only intertextually, but also intratextually. See, for instance, the cases of ⟨pep*er*⟩ and ⟨p*ar*te⟩ (Fig. 2–3) where the groups ⟨ar⟩ and ⟨er⟩ have been abbreviated by means of the same symbol, a bar across the stem of letter ⟨p⟩. Last of all, if the punctuation remains the same as in the original MS, correct tokenization has also proved impossible.

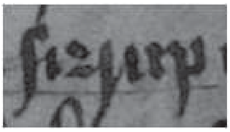


Fig. 1. f. 47v (Hunter 328)

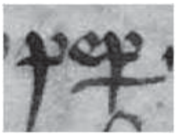


Fig. 2. f. 59r (Wellcome 397)

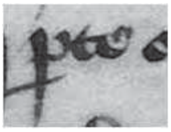


Fig. 3. f. 62v (Wellcome 397)

Our second model of transcription proposes, again, a semi-diplomatic transcription, using a graphemic—rather than a graphetic—approach, i.e. not distinguishing individual letter types (for example, *s longa* as opposed to diamond-shaped and sigma-shaped *s*) but presenting most phonemes in the text via one and the same graph (in the above case, ⟨s⟩ for all instances). Moreover, abbreviations would also be expanded. However, punctuation would still remain the same as in the original source, and, for this reason, our second model also has to be rejected for the achievement of a successful POS tagging process.

Our third and final model is also a graphemic semi-diplomatic transcription as the one above, only differing in that sentential punctuation is now soft-marked according to some rules. This feature is key to obtaining a model of transcription which feasibly allows for being processed by an automatic tagger, since having a more or less standard punctuation will allow a computer programme to identify sentence patterns with which to perform the automatic POS tagging process.

Note that all three models involved a (semi-)diplomatic transcription. Indeed a general edition, wherein punctuation, and sometimes even spellings, are regularised/standardised to a degree (take for instance the *Riverside Chaucer*), would solve many problems instantly. However, we wish to work with diplomatic transcriptions as we aim to provide the reader with versions of original sources as unbiased as possible, ones that are reliable for linguistic, codicological, palaeographical and historical research purposes.

2.1.2 Item/word recognition: tokenization

Across ME witnesses we find many instances of words that appear separated as if they were two different units when they are in fact just one, such as ⟨be fore⟩, ⟨with out⟩ and ⟨a boue⟩, below.

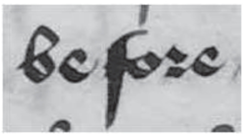


Fig. 4. f. 53v (Wellcome 397)

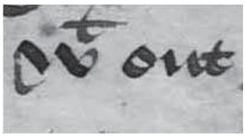


Fig. 5. f. 66v (Wellcome 397)

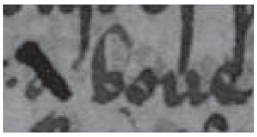


Fig. 6. f. 49r (Hunter 328)

We have just stated in 2.1.1 that we are to follow a semi-diplomatic transcription to reproduce the original source faithfully. Therefore, if the scribe wrote these words separately for any given reason, then we must respect this and reproduce it accordingly in the transcription. However, when feeding this transcription into a POS tagging program these words would be considered as two different items. For example, ⟨be fore⟩ would appear as ⟨be⟩ and ⟨fore⟩ and tagged as verb and preposition, respectively. We can also find the opposite situation: words that appear written continuously as a single token when they are in fact two separate words. Note cases as ⟨adragme⟩ and ⟨aman⟩ (Fig. 7–8), consisting of a determiner and a noun. These words would be understood as one item and as a result fail to be recognised by an automatic POS tagger.

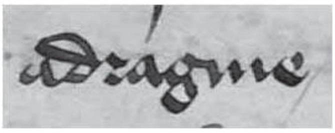


Fig. 7. f. 54r (Wellcome 397)

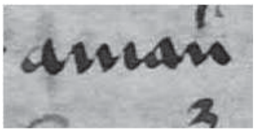


Fig. 8. f. 53v (Wellcome 397)

Another problem when dealing with ME texts is line-final word division. Sometimes we have a hyphen at the end of the line indicating that the word continues on the next line. Theoretically at least, we could train the tagger to recognise these hyphens, as they are accurately reproduced in the transcription. However, what happens when we have no hyphen indicating line-final word division, which is the unfortunate case more often than not?

The genitive morpheme is also trouble making. Apart from being a compound and, therefore, to be considered as one single unit (compounds will be discussed further below), how can we make the system understand that in a noun phrase like ⟨bores grece⟩ the first token is a noun in the genitive case and thus avoid the real danger that the system automatically interprets it as a noun in the plural? We use the apostrophe nowadays to tell the genitive singular ⟨'s⟩ from the plural ⟨(e)s⟩, and consequently most POS taggers for Present-Day English are trained to identify the ⟨'s⟩ morpheme as a separate unit. However, in the 15th-century the genitive ending *-es*, which survived from the OE declension for singular nouns, was still very much in use. The apostrophe did not appear until the ⟨e⟩ was finally dropped, since it fact it was used to indicate this contraction (Cavella and Kernodle 2003: 2).

Last of all, Middle English included letterforms which are no longer extant in our contemporary alphabet: these include, thorn ⟨þ⟩, yogh ⟨ȝ⟩ together with their respective capital counterparts ⟨Þ⟩, ⟨Ȝ⟩ and (since we work with semi-diplomatic transcription) we should probably include dotted ⟨ẏ⟩.[3] So, the tagger will have to be trained into recognising these letterforms, and moreover, into interpreting these letter forms as possible variants. Take for one the following spellings of the definite article: ⟨þe⟩, ⟨the⟩ and ⟨ye⟩. The tagger will have to realise that all these different letterforms are representing the same word and that they should accordingly acquire identical tags.

[3] Note that for early ME texts at least ⟨ð⟩, ⟨æ⟩, ⟨Ð⟩ and ⟨Æ⟩ should be added to the inventory.

2.1.3 Word identification

Once all the tokens have been suitably established, the system must recognise them in order to be able to add their corresponding morphological tag(s). However, this is not a straightforward process, as not all the items will be recognised by the tagger. Dialectal variants, scribal errors, roman numerals and terms belonging to other languages, mainly Latin, will be the main source of our problems.

A possible solution regarding dialectal variants and scribal errors would be to standardise and correct them, respectively. As we are dealing with semi-diplomatic transcription, this is naturally out of the question. The manual tagging process followed by the *Corpus of Late Middle English Scientific Prose* lemmatises the words according to the online version of the *Middle English Dictionary* (henceforth *e-MED*), but not only lemmas are provided, since their original spellings are also maintained. So, a semi-automatic tagger for ME would need to be trained to recognise these variants and, furthermore, to identify them as belonging to the same lemma. However, the real problem would arise when variants never seen before appear in a text, as a tagger trained on a specific set would not be able to recognise them. As for Roman numerals, they can easily be input into the system as indeed they already are in most Present day English taggers.

Concerning foreign terms (Latin, French, etc.), the basic problem is that we have a limited knowledge base. Our transcriptions, which have been manually annotated, have been lemmatised, as mentioned above, according to the entries recorded by the *e-MED*, but this source is of little use when it comes to such Latinate terms and other foreign words as were not considered borrowings by the editors of *MED*. Words not recorded in the *e-MED* have been tagged consulting other sources (see Moreno-Olalla & Miranda-García 2009: 137 for details). Furthermore, we can also rely sometimes on prior experiences, that is, if a particular word has already appeared in a previous text then we can tentatively assign it the same tag—but of course this practice is very limited. Moreover, if the tagger is solely trained on items that have already appeared then it is bound to encounter frequently new words which it is unable to identify. All in all, these problems are tied to the same main concern, the existence of “unknown” words that will not be identified by the system. This is, in fact, a difficulty that automatic POS taggers designed for present day languages still have to contend with.

2.1.4 Tagging criteria

Before attempting to devise a semi-automatic and semi-intelligent POS tagging system a criterion must be established for the task. What information do we want our tagger to provide? And, how do we want this information organised?

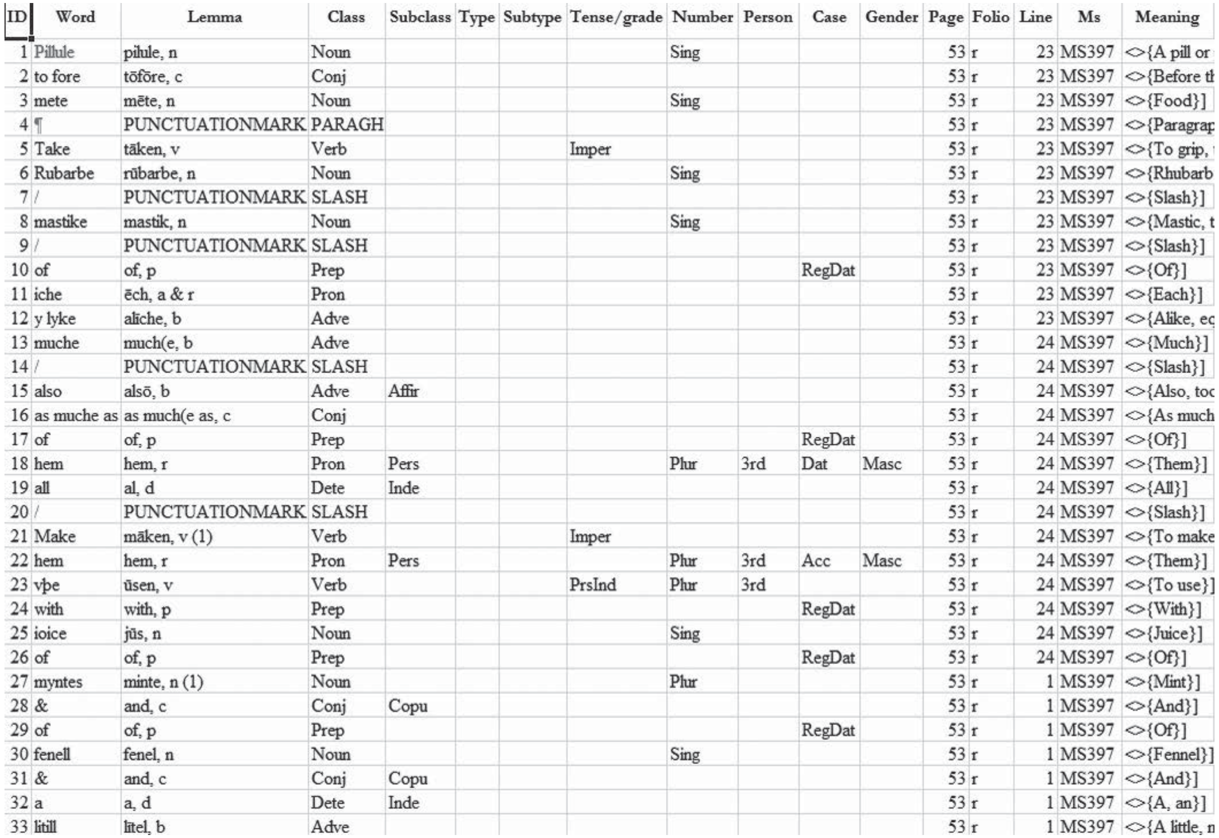
The manuscripts compiled in the *Corpus of Late Middle English Scientific Prose* have all been tagged manually according to the following criteria. First of all, the transcriptions are downloaded onto a Microsoft Excel spreadsheet, so that all the words appear vertically ordered in the first column. Then, each word is annotated with its corresponding lemma and morphological information in the remaining horizontal columns. Every word is tagged with the same information: lemma, word class, accidence, folio, line manuscript number, and meaning. Each lemma will moreover appear with its specific word class attached in order to procure disambiguation. The entries would look as presented on Fig. 9 below.

Our objective is to design a semi-automatic POS tagger that can provide the following information: (*a*) lemma (disambiguated according to its morphological category); (*b*) POS tag; and (*c*) accidence. In addition to that, we plan to offer some information on the dialectal provenance of each of the variants whenever this can be ascertained.

Moreover, our goal is not solely to provide tags at a simple word level. We also wish to take into consideration compound words, collocations and other phrases. So, our tagging system aims to perform POS tagging but also chunking to a certain extent, doubling up as a simple syntactic parser.

At word level, on virtually any text we will find both simple and compound nouns, such as ⟨enula campana⟩ and ⟨v levyd grase⟩, that should be tagged as one token only. However, this will cause difficulties for the system, as the terms are divided in the transcription and the tagger will assume that they are separate items. For example, ⟨v levyd grase⟩ would be tagged independently as a numeral determiner (v), an adjective (levyd) and a noun (grase), when our objective is in fact to tag the whole chunk as one noun.

Fig. 9. System of tags



ID	Word	Lemma	Class	Subclass	Type	Subtype	Tense/grade	Number	Person	Case	Gender	Page	Folio	Line	Ms	Meaning
1	Pillule	pilule, n	Noun					Sing				53	r	23	MS397	<>{A pill or
2	to fore	tōfōre, c	Conj									53	r	23	MS397	<>{Before tl
3	mete	mēte, n	Noun					Sing				53	r	23	MS397	<>{Food}]
4	¶	PUNCTUATIONMARK	PARAGH									53	r	23	MS397	<>{Paragrap
5	Take	tāken, v	Verb				Imper					53	r	23	MS397	<>{To grip,
6	Rubarbe	rūbarbe, n	Noun					Sing				53	r	23	MS397	<>{Rhubarb
7	/	PUNCTUATIONMARK	SLASH									53	r	23	MS397	<>{Slash}]
8	mastike	mastik, n	Noun					Sing				53	r	23	MS397	<>{Mastic, t
9	/	PUNCTUATIONMARK	SLASH									53	r	23	MS397	<>{Slash}]
10	of	of, p	Prep							RegDat		53	r	23	MS397	<>{Of}]
11	iche	ēch, a & r	Pron									53	r	23	MS397	<>{Each}]
12	y lyke	alīche, b	Adve									53	r	23	MS397	<>{Alike, eq
13	muche	much(e, b	Adve									53	r	24	MS397	<>{Much}]
14	/	PUNCTUATIONMARK	SLASH									53	r	24	MS397	<>{Slash}]
15	also	alsō, b	Adve	Affir								53	r	24	MS397	<>{Also, too
16	as muche as	as much(e as, c	Conj									53	r	24	MS397	<>{As much
17	of	of, p	Prep							RegDat		53	r	24	MS397	<>{Of}]
18	hem	hem, r	Pron	Pers				Plur	3rd	Dat	Masc	53	r	24	MS397	<>{Them}]
19	all	al, d	Dete	Inde								53	r	24	MS397	<>{All}]
20	/	PUNCTUATIONMARK	SLASH									53	r	24	MS397	<>{Slash}]
21	Make	māken, v (1)	Verb				Imper					53	r	24	MS397	<>{To make
22	hem	hem, r	Pron	Pers				Plur	3rd	Acc	Masc	53	r	24	MS397	<>{Them}]
23	vþe	ūsen, v	Verb				PrsInd	Plur	3rd			53	r	24	MS397	<>{To use}]
24	with	with, p	Prep							RegDat		53	r	24	MS397	<>{With}]
25	ioice	jūs, n	Noun					Sing				53	r	24	MS397	<>{Juice}]
26	of	of, p	Prep							RegDat		53	r	24	MS397	<>{Of}]
27	myntes	minte, n (1)	Noun					Plur				53	r	1	MS397	<>{Mint}]
28	&	and, c	Conj	Copu								53	r	1	MS397	<>{And}]
29	of	of, p	Prep							RegDat		53	r	1	MS397	<>{Of}]
30	fenell	fenel, n	Noun					Sing				53	r	1	MS397	<>{Fennel}]
31	&	and, c	Conj	Copu								53	r	1	MS397	<>{And}]
32	a	a, d	Dete	Inde								53	r	1	MS397	<>{A, an}]
33	litill	lītel, b	Adve									53	r	1	MS397	<>{A little, n

We also intend to identify common nouns and proper nouns, both simple, such as ⟨Galion⟩ (MS Hunter 497, f. 88v) and ⟨Eneas⟩ (Ms Hunter 497, f. 90r), and compound ones such as ⟨kyng*es* Rog*ere*⟩ (MS Wellcome 397, f. 54r) and ⟨*Christ*ofer Rochest*er*⟩ (MS Hunter 329, f. 30 v). How will the automatic system recognise proper names? Normally, they appear capitalised in ME witnesses. However, we must be aware that we can have instances of proper names which appear entirely in small case letters, such as ⟨uirgil⟩ (MS Hunter 497, 90r) or ⟨ypocras⟩ (Hunter 497, 48r), and also instances of common nouns that appear capitalised in the middle of a sentence, as in ⟨Myntt*is*⟩ (MS Hunter 328, 56v) and ⟨Coriand*er*⟩ (MS Hunter 328, 60v). Therefore, the system will not be able to rely on capitalisations in order to identify proper names.

Collocations and other phrases are divided into two types: (*a*) those wherein all the units remain together in a consecutive sequence, such as ⟨because of⟩, ⟨take from⟩, ⟨to and fro⟩ (Hunter 503), ⟨yn as moche as⟩ (Hunter 513a) or ⟨in respect of⟩ (Hunter 513a), among many others; and (*b*) those that appear divided, such as ⟨not only ... but also⟩, ⟨whether ... or⟩, ⟨if ... than⟩ or ⟨neyther ... ne⟩, among others. Below, we have examples of these "divided" phrases found within context:

> **not only** puttyth oute sauerey hyr chylde whether yt be quyk or deed yf she ete sauerey. **but also** yf sauorey be under put to þe woman þat ys with chylde (MS Hunter 497, 28v)
>
> **whether** yt be quyk **or** deed (Hunter 497, 28v)
>
> **if** he parbrake malum significat . **than** serche þe wounde . & Chaffe þe brokyn bonys (Hunter 328, 64v)
>
> **Neyther** þe rose coloure **ne** þe lylye may ouerpasse þe uiolet (Hunter 497, 15v)

These phrases should be tagged as a single item. However, how can we make the system recognise them as such? As we can see, we have the same problem that we find with compound nouns. Furthermore, phrases that appear divided, such as ⟨not only ... but also⟩ pose even more of a challenge for the system since, as we have seen above, they can frequently be lines apart.

2.1.5 Homonyms

Homonyms can only be disambiguated within context, and, as a result, they are a source of problems for POS taggers, particularly since our target tagger also aims to provide the meaning of any given token. For example, PDE ⟨lap⟩ can be both a round of a race track and the part of the body when sitting down. In ME we also have homonymous words. The following sample pairs are entries taken from the *e-MED*:

lēchen, v (1) 'to cut, slice'	vs.	**lēchen**, v (2) 'to cure, treat'
whītel, n (1) 'a knife, dagger'	vs.	**whītel**, n (2), 'a blanket'

Note further that homonyms do not necessarily have to share the same POS: ⟨fair(e, n⟩ and ⟨fair(e, b⟩, both taken from the *e-MED*, are a noun meaning "an affair or business" and an adverb meaning "beautifully", respectively.

2.1.6 Word order

Word-order in Present Day English (henceforth PDE) is very fixed, which gives PDE automatic POS taggers a distinct advantage when it comes to item recognition. For instance, even if a word is unknown or ambiguous to the tagger, the system will frequently be able to tag the word successfully by means of analysing its immediate context, a task that is performed by many existing algorithms. On the other hand, in ME word-order was not so rigid. For example, a number of adjectives can appear before or after the noun they are modifying, as in ⟨pip*er* lon*g*⟩ (Hunter 328, f. 47v) or ⟨long*e* pep*er*⟩ (Wellcome 397, f. 55r), which is problematic for devising a POS tagger. Or let us take the following phrase, "mellic*ra*tu*m* cu*m* pip*er* lon*g* & Alys spe*ci*b*us*" (Hunter 328, f. 47v), and imagine that the word ⟨long⟩ is unknown to our system. If our tagger is trained according to the "adjective + noun" pattern, which is the fixed pattern in PDE, then a phrase like this is bound to be tagged erroneously. The context around the word ⟨long⟩ would be looked at, and as soon as the tagger realised that it had a noun to the right and the conjunction ⟨and⟩ to the left, the program would most likely assume (of course erroneously) that the unknown word is another noun, when it is in fact an adjective. Our ME tagger would then have to train the system into recognising different patterns for adjectives, and this would now cause ambiguity problems.

2.1.7 Punctuation

Tokenization relies heavily on sentential punctuation, as it allows a POS tagging system to recognise sentences and, as a result, to identify the items that compose it. However, ME punctuation poses a problem for this process due to its extreme haphazardness. The most common punctuation marks in ME include the period (.), the virgule (/) and the paragraph mark (¶), but, in any given witness, "the significance of a given mark varied almost as frequently as spelling did" (Petti 1977: 25). To make matters worse, "practice often differed from writer to writer" (Petti 1977: 25).[4] Therefore, we cannot train our tagging system to recognise ME punctuation if there are no standard rules. For example, the virgule was frequently employed between words as is our present-day comma, as in the following sequence: "Take tu*r*bite / clowes / armodactules / of eu*er*yche x dragynes [...]". But it could also function as a full stop, indicating the end of a paragraph. In a similar vein, the period (.) could also function as a comma or as full stop.

Another symbol found within our transcriptions is square brackets ([ ]). These have been added by the transcriber in order to re-introduce marginalia and other interlinear additions within the main body of the text, so they are not are not part of the original witness. However, as they are found within the transcription, and will therefore be fed into the system, they still have to be dealt with. We have two options: (*a*) to delete them before introducing the transcription into the system; (*b*) to train the tagging system to ignore these symbols and solely acknowledge their contents, which will appear in superscript, as in "put it into a fayre vessel $^{\text{[of glasse]}}$. & vse to drynk it often" (MS Hunter 328, f. 45v). Note incidentally that this example illustrates as well the use of the period (.) as a present day comma.

[4] See Calle-Martín 2004, Calle-Martín and Miranda-García 2005 and Marqués-Aguado 2009 for further information regarding the nature of ME punctuation.

2.2 Computational limitations

In order to devise an successful semi-automatic POS tagging system for ME it is necessary to look at the existing systems and algorithms that are currently available for the task, irrespective of their target language, in order to establish those methods that are best suited to fulfill our needs as to the information we wish our tagger to provide.

2.2.1 Classification

POS taggers can be classified into four basic types: (*a*) rule based; (*b*) stochastic or probabilistic; (*c*) hybrid; and (*d*) based on alignment and projection of parallel texts.

(*a*) *Rule based taggers*
Rule-based taggers, such as TAGGIT, employ specific rules to eliminate ambiguity. These rules can be fed into the system by introducing a pretagged corpus, a "tagger dictionary" (Van Guilder 1995: 1), wherefrom the program can be trained, and by introducing rules (manually or automatically) by searching for generalised patterns of word order, the most frequent clause patterns including noun phrases, verb phrases and prepositional phrases. On the one hand, manually established rules require a high level of linguistic knowledge of the language in question and, moreover, a great deal of human effort, which is precisely what we are trying to reduce. Furthermore, manually established rules limit the tagger to the language and/or domain which it was designed for in the first place, not allowing for accurate results when trying to tag texts of a different nature. On the other hand, automatic taggers learn the rules automatically, free of human intervention, through a training process on a previously tagged text.

(*b*) *Stochastic or probabilistic taggers*
Stochastic or probabilistic taggers, as their name implies, rely on probabilistic methods for disambiguation. Usually trained on a previously tagged text, although not necessarily (see Eric Brill 1995 for more information regarding this matter), the system chooses those tags with the highest rate of frequency for the given word sequence. Simple stochastic taggers will assign tags relying solely on frequency, that is, an ambiguous

word will be tagged depending on the most frequent tag it has throughout the trainer text, which can cause words to be tagged erroneously. To surmount this problem many different types of stochastic taggers have been devised, all employing different systems based on probability. Let us look at seven commonly used stochastic systems: Hidden Markov Models, Maximum Entropy Taggers, decision trees, sliding windows, Support Vector Machines and memory based learning.

Hidden Markov Models (henceforth HMM), used in taggers such as HunPos or TATOO, combine "tag sequence probabilities and word frequency measurements" (Altunyurt, Orhan and Güngör 2007: 66), that is, they tag a word by making assumptions based not only on the frequency of the tag in itself, but also on the frequency of the tag appearing with the previous tags. This way the context of the word is taken into account and a tag will be applied or not depending on its preceding and succeeding words. For instance, note the position of the word *bathe* in the following phrases found in MS Hunter 497: "a bathe of hem" (f. 6v) and "wasshe or bathe well þe heed" (f. 26r). Out of context *bathe* could be either a noun or a verb, so to resolve ambiguity, a HMM would first of all calculate the probability of it being one or the other, depending on its rate of frequency found in the trainer text, and then it would look at the probability of these tags appearing with other tags, and thanks to the fact that in "a bathe of hem" *bathe* is preceded by a determiner the tagger will be able to correctly identify it as a noun, given that the structure *determiner* + *noun* is invariable across any text.

In turn, Maximum Entropy Taggers are statistical models based on mathematical formulas for automatic POS tagging. This model searches for the probability of distribution of the maximum entropy according to common restrictions and "combines diverse forms of contextual information in a principled manner, and does not impose any distributional assumptions on the training data" (Ratnaparkhi 1996: 133).

Decision Trees involve the creation of a decision tree generated from a previously tagged corpus that is used for the training process. The resulting tree will then be employed for the subsequent tagging of any text. Schmid claims that decisions trees would require a smaller training corpus than other methods, such as HMM, in order to obtain accurate results (Schmid 1994a: 48).

A Sliding Window is defined as "a system which assigns the part of speech of a word based on the information provided by a fixed window of words around it" (Sánchez-Villamil, Forcada and Carrasco 2004: 454), that is, that although the window, or frame, is mobile in itself, the number of words that is scanned at a time by that sliding window is always fixed. Sánchez-Villamil *et al.* present a tool that allows for the system to be trained from a raw corpus, i.e. unsupervised, not having to be tagged previously, furthermore implementing the tagger "exactly as a finite-state machine" (2004: 454).

Morphologic tagging can also be modeled and resolved with Artificial Neural Networks (ANN). These networks "consist of a large number of simple processing units" which "are highly interconnected by directed weighted links". Each unit will have its own *activation value*, this activation being "propagated to other units" through tile connections (all quotations from Schmid 1994b: 172). These networks can learn self-sufficiently by adapting the weight of their connections from a group of classified samples. This method has been claimed to "have shown performances comparable to that of Hidden Markov model systems or even better" (Lippmann 1989 cited in Schmid 1994b: 172).

The concept of Support Vector Machines (SVM) is defined by Pianta and Zanoli in the following manner:

> "Support Vector Machines are based on the Structural Risk Minimization strategy [7],[5] which aims at finding a hypothesis *H* for which we can guarantee the lowest true error, that is the probability that *H* will make an error on an unseen and randomly selected test example" (Pianta & Zanoli 2007: 8)

A SVM "performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels" (Electronic Statistics Textbook 2010). Murata, Ma and Isahara claim yet that SVM's can implement the POS tagging learning process efficiently only when large corpora are not being used as supervised data (2001: 24).

[5] From Vladimir N. Vapnik 1995: *The Nature of Statistical Learning Theory*. Springer.

Finally, the last stochastic method we will be dealing with is memory-based learning. This is a supervised inductive training method. When tagging a text both the tagged words of the training text and their context are stored in the system's memory and the words in our target text will be tagged according to their similarity with those words that have been stored in the memory. This system can provide assets such as the possibility of using a small tagged corpus for training, i.e. "incremental learning", and it apparently yields "good results on unknown words without morphological analysis", among others (Daelemans and Zavrel 1996: 25).

(*c*) *Hybrid taggers*

Hybrid taggers combine aspects of both stochastic and rule-based methods. Available hybrid taggers include, among others, systems based on transformation-based-learning (henceforth TBL) such as the Brill Tagger,[6] and those based on a combination of rules and HMM, such as CLAWS4.

TBL "is an error-driven approach to induce the retagging rules from a training corpus" (Algahtani, Black and McNaught 2009: 67). These taggers are trained from previously tagged texts. This process, however, is carried out in two different stages. Firstly, the words in the sample text are tagged according to their most frequent tag, and secondly (in the rule-learning stage), the tagger applies a series of possible disambiguation rules and evaluates them whilst gauging their adequacy, which is expected to reduce the error-rate. In this way the system can learn from its own errors and only learn those rules that are most effective. After the learning process has taken place, the texts are initially tagged without taking the rules into account; these are applied at a second stage to improve the first tagging.

(*d*) *Taggers based on alignment and projection of parallel texts*

This method, as seen above concerning the ME tagger developed at the University of Texas, relies on the existence of a text written in two different languages. One of the texts is tagged, either manually or by means of any

[6] Note that TBL can be classified as a rule-based method (see Brill 1992) and also as a stochastic method (Charniak 1997). However, given that it combines aspects of both methods, this study as considered it to be of a hybrid nature.

of the aforementioned automatic POS taggers, and then the other text is tagged by means of alignment and projection with the previously tagged text, therefore, automatically acquiring the same tags. Note that for our present objective this method is unfeasible, as no contemporary tagged versions of the ME texts we are dealing with are currently in existence.

2.2.2 Our beta tagger

Given the complexity of our objectives, the most adequate solution to fulfill our specific needs is to build a hybrid POS tagging system, wherein both rule-based and probabilistic methods are combined for the sake of a higher accuracy. At present we have a beta version of our target ME POS tagger that is fully operative and is continuously being improved by testing different methods. However, the tagger is able so far to tag items that have been previously trained into the system only, and does not deal as yet with unknown tokens.

We distinguished above the three stages involved in the POS tagging process: (*a*) "tokenization"; (*b*) "ambiguity look-up"; and (*c*) "disambiguation". Our current tagger is in the second phase of this process: it divides the text into tokens and then assigns all the possible tags that could possibly be applied to each of them. Take the token ⟨powder⟩ for instance. Our tagger is already able to assign it the POS tags of noun and verb, given that it can function as both, as seen from the following examples: "tyll the powder be consumed" (Hunter 503, p. 35) vs. "and powder it and medyl it wyth suger captyn" (Hunter 503, p. 124). The next step in line is hence to achieve a successful disambiguation, as ambiguity remains the main computational challenge.

The central trainer for our system is our tagger dictionary, compiled from the morphological information contained in the manually tagged transcriptions belonging to the *Corpus of Late Middle English Scientific Prose*. When a text is fed into the system, it is first of all tokenized: tokens are acknowledged and supralinguistic elements such as spaces and punctuation marks are skipped for the moment. Then the system searches for these tokens in the database and those that coincide with those words stored in the tagger dictionary are tagged automatically. Nothing really new here: this is how ordinary tokenizers work. However, in view of the aforementioned difficulties that the ME language poses for a successful tagging process, which cause so many words to remain unidentified, our system needs to be more complex. Therefore, the identification of the words that are unknown to the tagger becomes our second main computational challenge.

3 Solutions

3.1 Linguistic problems

Let us now move on to provide possible solutions to several of the aforementioned linguistic problems. Note that our linguistic problems are frequently solved by means of computational solutions. When a word is not identified automatically it will undergo several processes:

a) The first process surmounts our difficulties concerning line-final word division. Take, for instance, ⟨cle-pid⟩ or ⟨whi-ch⟩ when there is no hyphen indicating that these instances are just one word. The system performs the following procedure to identify them. First of all, it searches for the first part, in the case of ⟨cle-pid⟩ it will search for ⟨cle⟩, and if it cannot find it, the system will automatically unite it with the following token and perform the search again, this time searching for ⟨clepid⟩; if found, it will be added the corresponding tag(s).

Note that this solution only works for words which are divided into tokens that are unrecognisable to the system. For example, the word ⟨with out⟩ if divided would fail to undergo this process, as ⟨with⟩ would be identified and tagged as a preposition and ⟨out⟩ would also be found and tagged as an adverb. To avoid this problem, before any word is tagged by default the system will look at the word that follows it, to see if they exist in conjunction with another in the tagger dictionary. This way, ⟨with out⟩ will be successfully tagged as one token. Note that whenever a longer form is available in the tagger dictionary it will be favoured and chosen as the valid tag even if individual tags are found for the shorter tokens.

b) The second process solves the instances in which two words appear united. Take ⟨aman⟩, a determiner plus a noun, which was one of the aforementioned examples. The system first searches for the word as a whole, and since it will not found it begins to break down the word into two parts, going through all the possible combinations and searching for both parts respectively in the tagger dictionary. Our example ⟨aman⟩ can theoretically be divided in three ways: a-man, am-an and ama-n.

In this case, the first division is successful as the other two options are impossible. The system searches for ⟨a⟩ and then ⟨man⟩ and they acquire their respective tags, and what was once one token instantly becomes two.

c) For the system to recognise ME characters, such as ⟨þ⟩ and ⟨ȝ⟩, it is necessary to employ a character repertory that contains these characters. This corpus complies with version 5.0 of the Unicode standard, as it is becoming the most used one in the world.

d) As for ME irregular capitalisation, identification will pose no problems if the proper names appear in the training text, as they will automatically receive their corresponding tag, regardless of whether they happen to appear capitalised in our target text or not: the system will invariably check the word we wish to tag both in upper-case and lower-case letters. That is, if our text presents a proper noun in lower-case that appears capitalised in our database, the system will first look for it in lower-case exactly as it is found in the text, and then search for it with a capital initial. Likewise, for a common noun that appears capitalised the system will search for its capitalised form and, when not found, a lower case version will automatically be searched. However, this method can be problematic when we are dealing with ME characters, as the Unicode standard can sometimes cause problems when converting lowercase into uppercase and conversely. Moreover, our problem still remains for unknown proper nouns. How can they be identified?

e) Finally, to surmount the lack of standardisation in ME punctuation we initially developed a system of symbols that were to be introduced at transcription level and which would allow the tagger to recognise sentences. Three symbols were employed: ⟨**⟩, ⟨@@⟩ and ⟨%⟩, all of them respectively placed after the scribal punctuation. The two asterisks indicated that the system was to ignore the preceding punctuation mark. The two "at" symbols indicated that the mark we were dealing with should be acknowledged as valid also in PDE. Finally, the percent symbol indicated that the preceding mark of punctuation was added by the transcriber in order to provide a contemporary punctuation, irrespective of the value of the mark in the original MS. An example of this system looks as follows: ⟨& make .** a plast*er* & ley it to þe dyssese .@@⟩. Here we are telling the program to ignore the first period, and acknowledge the last one. Using this system we could teach the POS tagger to identify sentences without disturbing the scribe's original punctuation, as once the transcription has been fed into the system, the added symbols can be removed or made invisible.

A similar system of symbols was developed to lemmatise tokens that appeared divided, including compounds, collocations and other phrases. This was solved by placing ⟨ÑÑ⟩ in between the separated parts, this way telling the tagger that they were to be taken as one lemma. For example, ⟨withÑÑout⟩, ⟨enulaÑÑcampana⟩ and ⟨becauseÑÑof⟩. Furthermore, for those grammatical expressions that could appear lines apart (as in ⟨not only ... but also⟩) not only ⟨ÑÑ⟩ would be used to unite each section but arrows would also be added: > meaning that something follows; and < meaning that something precedes, as seen below.

"notÑÑonly> puttyth oute sauerey hyr chylde whether yt be quyk or deed yf she ete sauerey. <butÑÑalso yf sauorey be under put to þe woman þat ys with chylde" (MS Hunter 497, 28v).

However, although this system is feasible, it is clear that adding all these symbols is a tedious and time-consuming task for the transcriber as they have to be introduced manually, and all the more so since all the texts that were to be tagged by the system would have to undergo the same process. So, we aim to mark sentential punctuation instead by placing a bar or any similar symbol for the program to recognise full sentences.

3.2 Computational problems

Ambiguity will most likely be ultimately resolved by means of a HMM, grouped above within the stochastic systems. HMM are algorithms which tag words by taking into account (*a*) the frequency of the tag in itself across the trainer text; and (*b*) the frequency of the tag appearing with the previous tags, i.e its context. So, the tag will be applied or not depending on its preceding and succeeding words.

Again, unknown words remain as our main problem here. However, these may also be sorted by enlarging our tagger dictionary, by the assignment of the most frequent tag that is recorded within the tagger dictionary, by a simple analysis of the words context, and also by using the morphology of the word itself. In some cases, due to the unreliability of ME word order, inflectional morphology can be a very helpful resource for the tagging process.

4 Conclusions

This paper tried to fulfill the following purposes: (*a*) to present our objective: to design an accurate semi-intelligent and semi-automatic tagger for ME texts, as no such system is currently in existence; (*b*) to discuss the challenges that ME poses for our task, all these mainly due to a lack of standardisation in the language; (*c*) to present the different tagging systems that are available to suggest those that are most useful to fulfill our needs. The best system appears to be a hybrid one, which combines rule-based methods by employing a tagger dictionary from which the texts will acquire their corresponding tags, and stochastic and probabilistic methods in order to resolve ambiguity, mainly HMMs that can resolve ambiguity by looking at probabilities and the contextual information of the word in question. Finally, we aim (*d*) to provide the provisional solutions we have developed thus far to overcome the aforementioned problems.

As we have seen, our project is still very much work in progress. Even though some problems have been solved, we are still looking for suitable solutions for the successful identification of the genitive case and, more importantly, for the tagging of unknown tokens. These remain the main stumbling-blocks that hinder the development of a tool that may facilitate the part-of-speech tagging process and thus aid the linguist's search for useful information.

Melania Sánchez Reed & Antonio Miranda García
University of Málaga

References

AlGahtani, S., W. Black and J. McNaught 2009: Arabic Part-of-Speech Tagging Using Transformation-Based Learning. In K. Choukri and B. Maegaard, eds., *Proceedings of the Second International Conference on Arabic Language Resources and Tools*. (Cairo, Egypt): 66–70.

Altunyurt L., Z. Orhan and T. Güngör 2007: Towards Combining Rule-based and Statistical Part of Speech Tagging in Agglutinative Languages. *Computer engineering* 1.1: 66–69.

Brill, E. 1995: Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. *Proceedings of the 3rd Workshop on Very Large Corpora*. Massachusetts Institute of Technology (Cambridge, Massachusetts). 1–13.

Brill, E. 1992: A Simple Rule-Based Part of Speech Tagger. In M. Kauffman, ed., *Proceedings of the DARPA. Speech and Natural Language Workshop*. (San Mateo, California): 112–116.

Calle-Martín, J. 2004: Punctuation Practice in a 15th-Century Arithmetical Treatise (MS Bodley 790). *Neuphilologische Mitteilungen* 4. 407–422.

Calle-Martín, J. and A. Miranda-García 2005: Editing Middle English Punctuation. The Case of MS Egerton 2622 (ff. 136–165). *International Journal of English Studies* 5. 27–44.

Cavella, C. and R. A. Kernodle 2003 [2010/08]: How the Past Affects the Future: The Story of the Apostrophe'. *AU TESOL Working Papers 2*. American University: Washington, DC. http://www1.american.edu/tesol/Working%20Papers/wpkernodlecavella.pdf

Charniak, E. 1997: Statistical Techniques for Natural Language Parsing. *AI Magazine*, 18.4. American Association for Artificial Intelligence. 33–43.

Daelemans W. and J. Zavrel 1996: MBT: A Memory-Based Part of Speech Tagger-Generator. *Proceedings of the Fourth Workshop on Very Large Corpora (ACL SIGDAT)*. Copenhagen. 14–27.

Fischl, W. 2009: Part of Speech Tagging – A Solved Problem? Unpublished work from seminar/lecture. Technical University of Vienna: Austria. 1–8.

Giesbrecht, E. and S. Evert 2009: Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus. *Proceedings of the 5th Web as Corpus Workshop* (WAC5). Donostia. 27–35.

Kurath, H., S. Kuhn, J. Reidy and R. E. Lewis eds. 1956–2001 [2010/08]: *Middle English Dictionary*. Ann Arbour: University of Michigan Press. http://quod.lib.umich.edu/m/med/lookup.html

Lippmann, R. P. 1989: Review of Neural Networks for Speech Recognition. *Neural Computation, Vol. i*. MIT Press. 1–38. Cited in Schmid, H. 1994. Part-of-speech tagging with neural networks. *Proceedings of the 15th Conference on Computational linguistics*. Kyoto: Japan. 172–176.

Marqués-Aguado, T. 2009: Punctuation Practice in the Antidotary in G.U.L. MS Hunter 513 (ff. 37v–96v). *Miscelánea* 39. 55–72.

McEnery, T. and A. Wilson 1997 [2010/08]: *Corpus Linguistics*. http://www.lancs.ac.uk/fss/courses/ling/corpus/contents.htm.

Miranda-García, A., J. Calle-Martín, D. Moreno-Olalla and J. L. Triviño-Rodríguez 2001: CALLOE: A Pedagogical Tool for the Learning of Old English. *Old English Newsletter* 34.3. 12–20.

Miranda-García, A., J. L. Triviño-Rodríguez and J. Calle-Martín 2000: A Morphological Analyzer of Old English Texts (MAOET). In A. M. Hornero and M. P. Navarro, eds. *Proceedings of the 10th International Conference of SELIM*. Zaragoza, Institución Fernando el Católico: 127–145.

Moon, T. and J. Baldridge 2007: Part-of-Speech Tagging for Middle English through Alignment and Projection of Parallel Diachronic Texts. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Prague, Czech Republic): 390–399.

Moreno-Olalla, David and Miranda-García, Antonio 2009: *An Annotated Corpus of Middle English Scientific Prose*: aims & features. In Díaz Vera, Javier Enrique & Caballero Rodríguez, Rosario eds., *Textual Healing: Studies in Medieval English Medical, Scientific and Technical Texts*. Bern-New York, Peter Lang: 123–140.

Murata, M., Q. Ma and H. Isahara 2001: Part of Speech Tagging in Thai Language Using Support Vector Machine. *Proceedings of Natural Language Processing and Neural Networks (NLPNN)*. Tokyo: Japan. 24–30.

Petti, A. G. 1977: *English Literary Hands from Chaucer to Dryden*. Harvard University Press: Cambridge Massachusetts.

Pianta, E. and R. Zanoli 2007: Tagpro: a System for Italian POS Tagging Based on SVM. *Contributi Scientifici*, 4.2. 8–9.

Ratnaparkhi, A. 1996: A Maximum Entropy Model for Part-of-Speech Tagging. *Proceedings of the Empirical Methods in Natural Language Processing*. Philadelphia, PA. 133–142.

Ruef, B. [n. d.] [2010/08]: Transformation-Based Learning and Part-of-Speech Tagging of Old English. Zurich. https://files.ifi.uzh.ch/cl/gschneid/KorpusSeminar/Beni_TBL_Slides.pdf

Sánchez-Villamil, E., M. Forcada and R. Carrasco 2004: Unsupervised Training of a Finite-State Sliding-Window Part-of-Speech Tagger. In Vicedo, J.

L. *et al*. eds., *Proceedings of the 4th International Conference España for Natural Language Processing (EsTAL)* (Alicante, Spain): 454–463.

Schmid, H. 1994a: Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of International Conference on New Methods in Language Processing*. (Manchester, UK): 44–49.

Schmid, H. 1994b: Part-of-Speech Tagging with Neural Networks. *Proceedings of the 15th Conference on Computational linguistics* (Kyoto, Japan): 172–176.

StatSoft, Inc. 2010 [2010/08]: *Electronic Statistics Textbook*. Tulsa: StatSoft. http://www.statsoft.com/textbook/

Van Guilder, L. 1995 [2010/08]: Automated Part of Speech Tagging: A Brief Overview. Handout for LING361. Georgetown University. http://ccl.pku.edu.cn/doubtfire/NLP/Lexical_Analysis/Word_Segmentation_Tagging/POS_Tagging_Overview/POS%20Tagging%20Overview.htm.

Received 28 Oct 2010; *revision received* 17 Nov 2010; *accepted* 22 Nov 2010