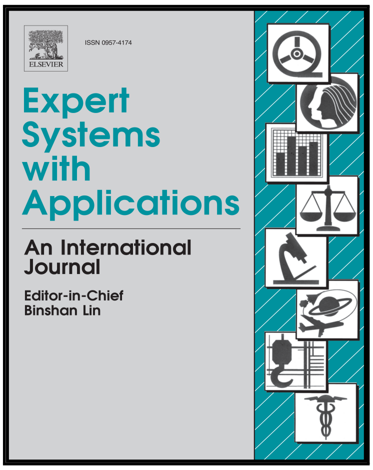
2017

# Using linguistic features to automatically extract web page title

Gali N



ISSN 0957-4174

Expert Systems with Applications

An International Journal

Editor-in-Chief
Binshan Lin

Using Linguistic Features to Automatically Extract Web page Title

Najlah Gali , Radu Mariescu Istodor , Pasi Fränti

PII: S0957-4174(17)30135-5
DOI: 10.1016/j.eswa.2017.02.045
Reference: ESWA 11153

To appear in: *Expert Systems With Applications*

Received date: 5 October 2016
Revised date: 27 February 2017
Accepted date: 28 February 2017

Please cite this article as: Najlah Gali , Radu Mariescu Istodor , Pasi Fränti , Using Linguistic Features to Automatically Extract Web page Title, *Expert Systems With Applications* (2017), doi: 10.1016/j.eswa.2017.02.045

**Highlights**

- Successful title extraction must analyze both the DOM nodes and the title tag
- Natural language processing improves the quality of the title.
- Visual and formatting features are less relevant for the task.
- Simpler classifier like k-NN perform as well as an advanced classifier like SVM.
- The proposed method significantly outperforms all existing ones by clear margin.

# Using Linguistic Features to Automatically Extract Web page Title

**Najlah Gali, Radu Mariescu Istodor, Pasi Fränti**

Machine Learning group, School of Computing, University of Eastern Finland, Joensuu FI-80101, Finland
{najlaa, radum, franti} @cs.uef.fi

**Abstract** Existing methods for extracting titles from HTML web page mostly rely on visual and structural features. However, this approach fails in the case of service-based web pages because advertisements are often given more visual emphasize than the main headlines. To improve the current state-of-the-art, we propose a novel method that combines statistical features, linguistic knowledge, and text segmentation. Using annotated English corpus, we learn the morphosyntactic characteristics of known titles and define a part-of-speech tag patterns that help to extract candidate phrases from the web page. To evaluate the proposed method, we compared two datasets Titler and Mopsi and evaluated the extracted features using four classifiers: Naïve Bayes, k-NN, SVM, and clustering. Experimental results show that the proposed method outperform the solution used by Google from 0.58 to 0.85 on Titler corpus and from 0.43 to 0.55 on Mopsi dataset, and offers a readily available solution for the title extraction problem.

**Keywords** Web content mining, Information extraction, Title extraction, Natural language processing, Machine learning

## 1 Introduction

Human is an impatient creature by nature when seeking information from the Internet; the user wants to obtain the right answer immediately and with minimal efforts (Marchlonini, 1992; Song, Xin, Shi, Wen, & Ma, 2006). In most applications, the title of the content is the first thing where the user pays attention. Even a single word or phrase can dramatically change the whole message of this content. To have a correct title is important and should therefore, be descriptive, concise, and grammatically correct (Hu et al., 2005).

In web pages, titles usually exist in two places for different purposes. First, the title is somewhere in the body text with high visual emphasis for the human reader. This is important for humans who browse the web page quickly on a computer display. Second, the title is placed in the title field (between <*title*> and </*title*> tag) for robots, crawlers, and programs that prepare a summary for the web page. However, the designers of the web page often ignore this or abuse the title tag by adding extra content like keywords, address or other less relevant text. Its content becomes then vague, incorrect, or it might be even missing completely (Xue et al., 2007). In mobile applications the problem exaggerates. The small screen of miniaturized devices are even more restrictive to the displayed content and requires the title to be also fitted spatially. The title is also needed for indexing in search engines like Google.

*Title extraction* aims at producing a compact title for a web page automatically. Due to the problems of the title tag, existing literature has mainly been focused on extracting the title from the body text. Methods have been developed for web pages of standard format such as news and pages of educational institutions. However, less attention has been given to service-based web pages such as *entertainment*, *sport*, and *restaurants*. Existing methods also make an assumption like the title is always located in the top region of the page and has visual prominence; they often fail to correctly extract the title of the service-based pages where the title is exchanged for a logo, or it is positioned elsewhere on the page (see Figure 1). For example, Hu et al. (2005) and Xue et al. (2007) explicitly state that the title must be in the top area of the page. Furthermore, Fan, Luo, and Joshi (2011) hypothesize that the title is located in the upper part of the main text. Changuel, Labroche, and Bouchon-Meunier (2009) implicitly assume that the title appears in the top portion of the page and as a result extract only the first 20 text nodes from the Document Object Model (DOM) tree.

Another assumption often made is that the title in a body is a separate line of text (i.e., it has its own text node in the DOM tree). However, the modern web page design allows the title to appear as a part of other phrases in the text node of the DOM tree. For example, <*h*1>Welcome to *Petter Pharmacy*, please select one of the five options below: </*h*1> will produce ill-fitting title unsuitable for mobile devices. According to our experiments, about 68% of the title nodes also contain additional information similar to the example and are therefore prone to errors in the title extraction.

In this work, we developed a novel method to overcome these problems in the title extraction for service-based content and mobile applications. Our key finding is that the title tag is still the best source. However, it needs to be segmented and further processed. Our preliminary version was presented in (Gali & Fränti, 2016). Here, we further enhance this approach by applying additional part-of-speech (POS) tagging. POS processes every word in the text and assigns them with a tag based on the relationship with adjacent and related words in the phrase, sentence, or paragraph. POS tagging has been successfully employed in other domains such as keyword extraction (Hulth, 2003). However, to our best knowledge, no language model has been applied for title extraction in the context of web pages. We are only aware the work of (Lopez, Prince, & Roche, 2010; Lopez, Prince, & Roche, 2014) giving POS model for mailing lists and news articles in the French language. We, therefore, investigate the contribution of POS tagging to this task.

We aim at identifying features that are independent of the format of the web page. Our method uses the following features: *syntactic structure*, *similarity with the link of the web page*, *appearance in the title tag*, *appearance in meta tags*, *popularity on the web page*, *appearance in heading tags*, *capitalization*, *capitalization frequency*, *independent appearance*, and *phrase length*. We consider four alternative classifiers: Naive Bayes, clustering-based, k-nearest neighbors (k-NN), and support vector machine (SVM), which to our knowledge have not been compared previously in the title extraction task.

We compare the proposed method against related ones with two datasets: Titler and Mopsi. Experiments show that our method gives a significant improvement, and achieves an accuracy of 0.85 with Titler dataset. The corresponding results of the baseline (title tag as such), Google, and the best content-based method are 0.52, 0.58 and 0.47 respectively.

The rest of the paper is organized as follows. In Section 2, we review existing methods for title extraction, and the new method is introduced in Section 3. Experiments are done in Section 4. Effect of POS pattern is studied in Section 4.5, feature extraction in 4.6, choice of the classifier in 4.7, and title selection methods in 4.8.

Comparisons to the existing methods are then performed in Section 4.9. We compare to all existing identification methods that are accessible. These include Styling (Changuel et al., 2009), TitleFinder (Mohammadzadeh, Gottron, Schweiggert, & Heyer, 2012), Title Tag Analyzer (Gali & Fränti, 2016), and the Baseline. We also compare to the titles provided by Google in the search results page. The results show that the proposed approach outperforms all the methods. The method with k-NN improves the Jaccard of the baseline from 0.50 to 0.84 on Titler corpus and from 0.44 to 0.59 on Mopsi dataset.

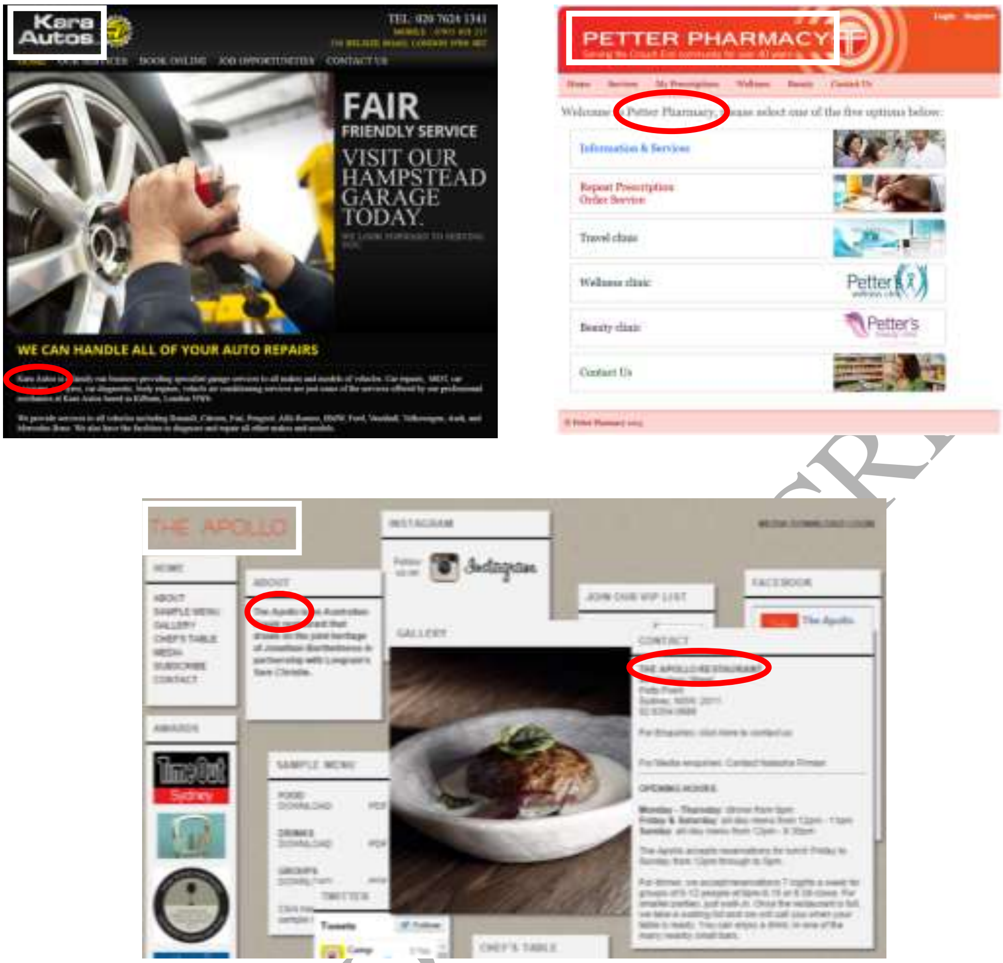


**Figure 1** Different layouts of web pages[1][2][3] (white squares refer to images of logos while red ovals refer to titles in the body of the web pages).

## 2 Related work

Figure 2 shows typical steps for the title extraction (left) and possible approaches to each step (right); the modules that are covered in this work are highlighted in blue.

### 2.1. Content source for title

The title of a web page is usually found in one or more of three places: the title tag (i.e., between <*title*> and </*title*>), the text of the body, and the logo image. According to our experiments with 1002 websites, the occurrence of the title in these three places is as follows:

- Title tag (91%)
- Text of the body (97%)
- Logo (89%)

[1] http://karaautos.co.uk/
[2] http://www.petterpharmacy.co.uk/
[3] http://theapollo.com.au/

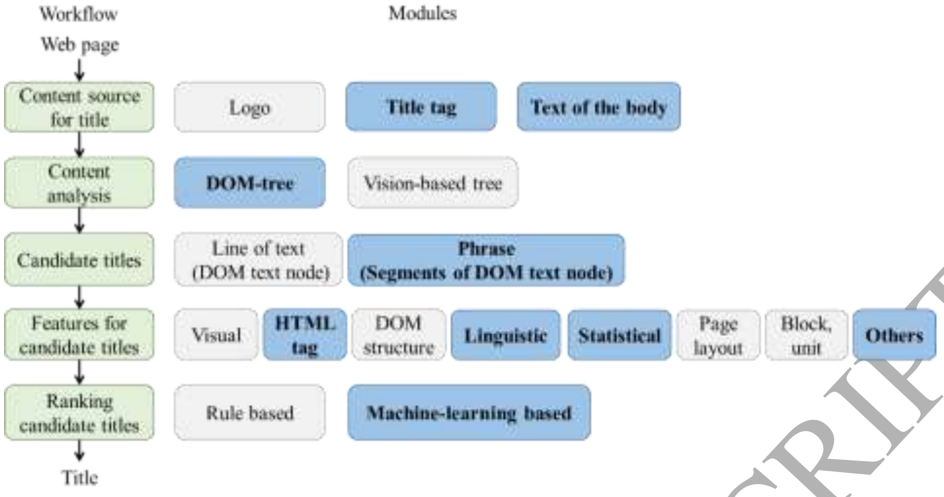


**Figure 2** Typical steps for title extraction.

The *title tag* is the obvious source, and the author of the page is expected to fill it with a proper title. However, people often do not complete this tag carefully as it does not have a visual impact on the page. A title tag often contains additional text, such as the name of the hosting website, information about the place offering services, a slogan, and contact details (see Table 1). The *body text* of a web page is a second source for a title. It has been given more focus by researchers given that a title in the body is visible to users and is thus expected to be written more carefully than the title tag (Changuel et al., 2009; Hu et al., 2005; Wang et al., 2009; Xue et al., 2007) However, extracting a title from the body of the web page is not an easy task, as roughly half of a page's content is irrelevant text (Gibson, Punera, & Tomkins, 2005). This irrelevant text (e.g., advertisements) is often given even more visual emphasis than the main headlines, which makes the task even more challenging. Furthermore, no standard location exists in relation to title placement. In this paper, we extract the candidate titles from both the body of the web page and the title tag.

**Table 1** The most typical problems related to title tag and the frequency they appear (according to our experiments).

| Type | Proportion (%) | Example | Annotated title |
|---|---|---|---|
| Long description | 62 | <title>Brook's Diner \| 24 Hampden Square, Southgate, London N14 5JR \| 020 8368 7201 \| eat@brooksdiner.com \| Like us on Facebook — Home</title> | Brook's Diner |
| Incorrect | 6 | <title>Hot Tubs, hot tub hire, swimming pools, Bristol, Gloucester</title> | Rio Pool |
| Vague | 2.4 | <title>home</title><br><title>index</title><br><title> \| </title> | Hellard Bros Ltd. |
| Short description | 0.5 | <title> Toby's Estate</title> | Toby's Estate coffee |
| Empty | 0.2 | <title> </title> | Zavino Hospitality Group |

The third source for a title is the *logo image*. However, extracting a title from this image would be very challenging. One reason is that the logo image must first be identified. Another reason is that the standard optical character recognition (OCR) approach would not generally work given that the content of the image is highly complex. We are not aware of any technique that attempts this approach. It should technically be possible, but as shown by the examples in Figure 3, such a technique would need to handle a wide variety of complex text fonts that involve shadowing effects, textures, and other artistic features.

### 2.2. Content analysis and candidate title extraction

Most title extraction methods use either DOM tree representation or combine the DOM structure with the visual cues of the page in a *vision-based tree*. The vision-based tree is built using the vision-based page segmentation algorithm (VIPS) introduced by Cai, Yu, Wen, and Ma (2003). The vision-based tree provides visual partitioning of the page where the blocks (i.e., DOM nodes) are grouped visually, while DOM tree describes the parent-child relationship between the tree nodes; therefore, it is not necessary that nodes in the vision-based tree correspond to the node in the DOM-tree.

The VIPS needs to refer to all styling information (including external sheets) to locate the proper place of the block in the tree. If the web page lacks rich visual properties, the hierarchies are incorrectly constructed. A wrong structure can also result from the algorithm not detecting separators represented by thin images. We, therefore, use DOM tree representation.

In both tree representations, existing methods use the entire text of the leaf nodes as candidate titles. In this paper, we extract only the relevant part of the text nodes by using POS tag patterns (see section 3.3).

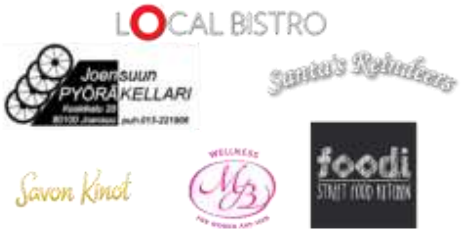


**Figure 3** Examples of web page logo.

### 2.3. Features for candidate titles

Researchers have extracted a wide range of features from either DOM or vision-based tree. Those found in the literature are listed below. The features used in this paper are underlined.

***Features from DOM tree:***

- *Visual*: font weight, font family, font color (Changuel et al., 2009; Hu et al., 2005; Xue et al., 2007); font style, background color (Hu et al., 2005; Xue et al., 2007); alignment (Fan et al., 2011; Hu et al., 2005; Xue et al., 2007); and font size (Changuel et al., 2009; Fan et al., 2011; Hu et al., 2005; Xue et al., 2007);
- *HTML tag*: bold, strong, emphasized text, paragraph, span, division (Changuel et al., 2009); image, horizontal ruler, line break, directory list (Hu et al., 2005; Xue et al., 2007); underline, list, anchor (Changuel et al., 2009; Hu et al., 2005; Xue et al., 2007); meta; title (Changuel et al., 2009; Fan et al., 2011; Mohammadzadeh et al., 2012); heading level (h1- h6) (Changuel et al., 2009; Fan et al., 2009; Gali & Fränti, 2016; Hu et al., 2005; Xue et al., 2007); and position in tags (Gali & Fränti, 2016);
- *DOM structure*: number of sibling nodes in the DOM tree (Hu et al., 2005; Xue et al., 2007); relation with the root, parent, sibling, next and previous nodes in term of visual format (Changuel et al., 2009; Hu et al., 2005; Xue et al., 2007);
- *Positional information*: position of the text unit from the beginning of the body of the page and width of the text unit with respect to the width of the page (Hu et al., 2005);
- *Linguistic*: length of text, negative words, positive words (Hu et al., 2005; Xue et al., 2007); position in text (Lopez, Prince, & Roche, 2011); syntactic structure, letter capitalization and phrase length; and
- *Statistical*: term frequency (Mohammadzadeh et al., 2012); term frequency-inverse document frequency (Lopez et al., 2011; Mohammadzadeh et al., 2012); capitalization frequency, and independent appearance.

***Features from vision-based tree:***

- *Page layout*: height, width, and position relative to the top left corner (Xue et al., 2007);
- *Block*: type, height, width, position (Wang et al., 2009; Xue et al., 2007); and front screen position (Wang et al., 2009);
- *Unit position*: from the top and left side of the page and from the top and left side of the block (Xue et al., 2007); and
- *Content*: number of words in a block (Wang et al., 2009).

***Other features:***

- Web page URL (Gali & Fränti, 2016).

The majority of these features are based on formatting, whereas the features we consider are independent of the design of the page.

### 2.4. Ranking candidate titles

Ranking techniques can be divided into two broad classes: rule based and machine learning based (Xue et al., 2007). *Rule-based techniques* use a set of predefined heuristic rules to score the candidate titles. These rules are derived from the content of the DOM tree (Fan et al., 2009; Gali & Fränti, 2016; Mohammadzadeh et al., 2012), the link structure between web pages (Jeong, Oh, Kim, Lyu, & Kim, 2014), and the text (Lopez et al., 2011). The key advantage of the rule-based technique is that it does not require training data. Moreover, the technique is easy for humans to interpret and improve, as the weighting procedure and scoring formulas are explicit. However, heuristic methods often require determining thresholds and weights for feature parameters, which are not always straightforward to calculate. For example, if the number of features is $n$ = 9 and each feature is assigned a value $m$ = 0 to 5, it takes O($m^n$) time to test all weight combinations. In this example, testing would take about four months if each attempt took 1 second.

In contrast, *Machine learning-based techniques* involve two steps: training and testing. In training, the goal is to learn a set of rules that maps the inputs to outputs, so that the rules generalize beyond the training data. In testing, the generated classifier receives unseen data as input and predicts the output values. Proper training of the model is the key to generalizing the classifier beyond the training data. Several machine learning algorithms have been considered by the existing methods. These include perceptron (Li, Zaragoza, Herbrich, Shawe-Taylor, & Kandola, 2002), decision tree (C4.5) (Quinlan, 1993), random forest (Breiman, 2001), support vector machine (SVM) (Vapnik, 1995), and conditional random fields (CRF) (Lafferty, McCallum, & Pereira, 2001). While SVM has shown to be an effective classifier for the title extraction task, it has not been compared against simpler algorithms such as Naïve Bayes (Domingos & Pazzani, 1997), k-nearest neighbor (k-NN) (Cover & Hart, 1967), and clustering (Fränti & Kivijärvi, 2000), all of which we investigate in this paper.

## 3 Title extraction

We consider the title extraction as a machine learning task, in which the computer is given a training data with assigned ground truth titles as the expected output. Three important issues are addressed: how to determine the candidate phrases, what features should be extracted, and which classifier to use. We add linguistic knowledge (syntactic POS) to the process to improve the extraction of the candidate phrases. The proposed method is based on four steps: extracting candidate phrases, feature extraction, phrase classification, and title selection (see Figure 4). A pre-processing step which involves corpus creation and learning POS patterns is applied before the training starts. The following subsections describe these steps in detail.

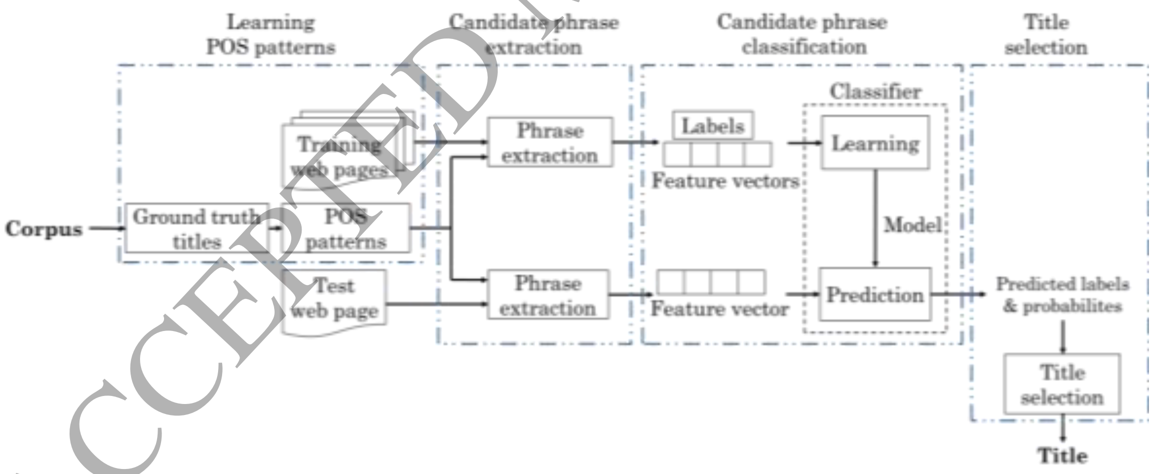


**Figure 4** Workflow for title extraction.

### 3.1. Corpus creation

Several corpuses have been created to evaluate title extraction methods. Changuel et al. (2009) have created two corpuses on education domain. The first corpus contains 624 websites in English and French languages, and they were collected by submitting queries to the search engine such as *chemistry + courses*. The second corpus contains 424 websites in the French language, and they were collected from an online educational portal *Eureka*[4]. Lopez et al. (2014) have created a corpus of 300 news articles from three French newspaper websites, and they cover *politics*, *sport*, *society*, and *science* domains.

[4] http://eureka.ntic.org/

However, they do not consider web pages of places that offer services such as *sports*, *hospitals*, *shops*, *banks*, and *restaurants*, or web pages that host information about these places such as *Wikipedia*, *Facebook*, *business directories*, and *information pages*. These types of web pages are more challenging because they do not follow a certain template or standard format. None of these corpuses are available in public.

We, therefore, built our corpus by collecting 1,002 unique websites from Google Maps[5] search results using queries such as *restaurant + Australia*, *hospital + Canada*, *pharmacy + London*, *fitness + Ireland*, and *auto repair + California*, to have reasonable geographical diversity and different layouts of websites. The main challenge of creating a corpus is that it is not enough to store the web link and the ground truth title. The entire content of the web page should be stored because the links will become obsolete quite fast. Storing the content takes lots of space especially when the web page contains maps and images. The websites were collected during 18-31 July 2014 and 19-23 April 2015, and they cover various domains— *food & drink*, *entertainment*, *auto & vehicles*, *beauty & fitness*, *health*, *sport*, and *hotels & accommodation*. The resulting corpus is publicly available[6].

Similarly to the previous studies (Lopez et al., 2014; Wang et al., 2009; Xue et al., 2007), we created the ground truth by manually extracting the titles from the pages. We define the *title* as the most obvious description of the web page (see Figure 5) following the specifications in (Xue et al., 2007) with a few modifications of our own:

- A web page can have more than one title, for example, we extracted *V-Café* and *Viet-Café* from *http://www.viet-cafe.com/*, *C&A Bennett Ltd* and *C&A Bennett Tiling Contractors* from *http://www.candabennett-tiling-bristol.co.uk/*;
- The title cannot be a part of a numbering or bullets;
- The title cannot be phrased like *last updated*, slogans like *aim high go low*, time, or address;
- The title should not be too long;
- The title must be concise and relevant to the page content;
- The title must be grammatically correct;
- The title must be understandable to humans.

We did not use specifications such as the title must be in the top region of the page, or that the title cannot be a link because the correct title can be located on any part of the page. Further, a title can be clickable, especially in the case of business directory pages, in which the title is usually linked to the home page of the service.

Two people were assigned to extract the ground truth titles independently on each other, and in the case of disagreement, a third person made a judgment between these two.

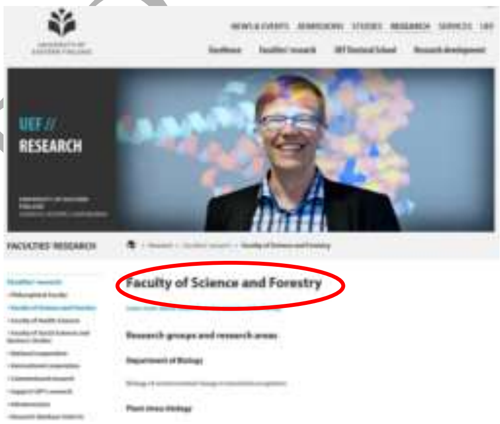

UEF // RESEARCH
FACULTIES' RESEARCH
Faculty of Science and Forestry
Research groups and research areas
Department of Biology

**Figure 5** Identifying the title of the web page[7].

## 3.2. Learning POS tag pattern

We define a set of specific POS tag patterns that correspond to the syntactic structure of the titles in the ground truth to remove the n-grams that have unwanted format. A POS tag is the part-of-speech label of a word in a text. For example, the

[5] http://maps.google.com

[6] http://cs.uef.fi/mopsi/TitlerCorpus/

[7] http://www.uef.fi/en/research/faculty-of-science-and-forestry

POS tag of *the university* is *the_DT university_NN*, where DT stands for determiner and NN stands for a noun. A POS tag pattern is a sequence of part-of-speech tags (e.g. <DT><JJ><NN>), where JJ stands for adjective. See Appendix A for a complete list of POS that we use in this paper.

We first extract all *n*-grams (*n*=1 to 6) as candidate phrases. We observed that the number of the candidates is excessively high (1,024,142), of which only 2,179 phrases are in the title class. To evaluate them all would slow down the process and it would also cause a high-class size unbalance. There are also many grammatically incorrect title candidates among the *n*-grams like *At Thai Food You*, *by Quay* and *Portishead Open Air Pool The*. Most of these can be eliminated by applying the POS patterns.

To generate the POS patterns, we have searched all POS tags that appeared among the ground truth titles in our corpus. We used the tagger developed by Stanford University[8] (Toutanova, Klein, Manning, & Singer, 2003). We observed that the following syntactic features are common for titles:

- Starts with a general noun, proper noun, personal pronoun, foreign word, adjective, determiner, adverb, cardinal number, preposition, or verb;
- Ends with a general noun, proper noun, personal pronoun, adjective, adverb, cardinal number, preposition, practical, verb or possessive ending;
- In case it contains more than two words, the middle words are allowed to be a general noun, proper noun, personal pronoun, adjective, determiner, cardinal number, preposition, foreign word, coordinating conjunction, adverb, or verb;
- Nouns appear much more than verbs see Table 2.

Based on these observations, we generate 151 patterns with the length varying from 1 to 6 (see Appendix B). In our study, we follow traditional English grammar, in which a noun preceded by determiners or premodifiers such as adjectives is considered a noun phrase. It is important to note that this set of syntactic patterns is language-dependent and applicable only to the English language. The same process could also be done for other languages if a set of ground truth titles and POS tagger exist.

**Table 2** Features of ground truth titles

| **POS tags** | **Presence in title (%)** |
|---|---|
| Nouns and proper noun | 83 |
| Determiner | 5 |
| Coordinating conjunction | 4 |
| Adjective | 3 |
| Possessive ending | 2 |
| Preposition or subordinating conjunction | 1 |
| Verb | 0.7 |
| Cardinal number | 0.7 |
| Foreign word | 0.2 |
| Pronoun | 0.2 |
| Adverb | 0.1 |
| Particle | 0.1 |

### 3.3. Candidate phrase extraction

We first construct the DOM tree of the web page and strip it off the <*script*> and styling tags because their text content is mainly used for functionality and styling. We used XPath[9], a query language for addressing parts of an XML document, to extract the text nodes as individual units from the tree. Then we add POS tags to each text node using Stanford Tagger. For example, a text node *Kingston General Hospital* becomes *Kingston_NNP General_NNP Hospital_NNP* after POS tagging. Later, we extract all phrases or (sequences of words) that match any of the patterns as potential candidates for titling. For example, we extract three nouns "*Kingston*", "*general*", "*hospital*" and three phrases "*Kingston general*", "*general hospital*", and "*Kingston General Hospital*" from *Kingston_NNP General_NNP Hospital_NNP* because they match the patterns <*NNP*>, <*NNP*>< *NNP*>, and <*NNP*><*NNP*><*NNP*>, which are all valid syntactic structures for titles.

### 3.4. Feature extraction

For each candidate phrase, we extract the following ten features to evaluate its importance.

[8] http://nlp.stanford.edu/software/tagger.shtml
[9] http://www.w3.org/TR/xpath20/

- Syntactic structure
- Similarity with the link of the web page
- Appearance in title tag
- Appearance in meta tag
- Popularity on the web page
- Appearance in heading (h1, h2…h6) tags
- Capitalization
- Capitalization frequency
- Independent appearance
- Phrase length

***Syntactic structure (POS tag patterns)***

The structure of the ground truth title (as detailed in subsection 3.2), provides useful information to determine the title phrases. Different representations can convert the text features into compact formats such as TF, TF-IDF and binary (Lan, Tan, Su, & Lu, 2009). We use binary representation because we aim at distinguishing the structure of the candidate phrases, but not to determine the importance of the individual tags.

In the analysis, we observed that the maximum word counts for the titles are 6. Therefore, we compute a vector of size (21×6=126) to represent the syntactic structure feature. This transformation is similar to token-vector approach, in which the order of the tokens is preserved.

- Each title is represented by its POS tags. For example, *The Path Café* is represented by *DT*|*NN*|*NNP*|, and *Leon Restaurants* is represented by *NNP*|*NNPS*|.
- The tags are then binarized by giving each 21 possible slot, and activating the slot corresponding to a specific tag. We recognize 20 POS tags in the titles, and we use slot 21 for other tags. For example:

| | | | | | |
|---|---|---|---|---|---|
| The | Path | Cafe | ‘ ’ | ‘ ’ | ‘ ’ |
| DT | NN | NNP | | | |
| [00**1**000000000000000000] | [00000000**1**000000000000] | [0000000000**1**0000000000] | [0] | [0] | [0] |
| | | | | | |
| Leon | Restaurants | ‘ ’ | ‘ ’ | ‘ ’ | ‘ ’ |
| NNP | NNPS | | | | |
| [0000000000**1**0000000000] | [00000000000**1**000000000] | [0] | [0] | [0] | [0] |

We later denote this feature as POS feature.

***Similarity with the link of the web page***

Words in the link of the web page are usually precise and relevant to the content of the page; therefore, we hypothesize that a candidate phrase that has high similarity to these words is more likely to be the title of the page. We compute the similarity between the phrase and the words in the web link using the Dice coefficient (Brew & McKelvie, 1996). It counts the number of shared character bigrams divided by the total number of bigrams in both strings:

$$similarity\ (p, s) = \frac{2 \times |bigrams(p) \cap bigrams(s)|}{|bigrams(p)| + |bigrams(s)|} \tag{1}$$

where bigrams is the number of adjacent character pairs in the candidate phrase ($p$) and the words in the link ($s$) respectively. The reason for choosing this measure is that it is robust to the change of the order of the words, and it treats strings with small differences as similar. These kinds of variations are expected between the web page link and the extracted phrases; an exact match would not be as useful. A measure like edit distance would recognize the reverse order of two strings as a mismatch. For example, the edit distance between the two strings *nba mcgrady* and *macgrady nba* is very low, although they refer to the same title (Wang, Li, & Feng, 2014). We normalize the similarity scores to the scale [0, 1] as follows:

$$S_s(p) = \frac{similarity\ (p, s)}{\max(similarity(p_r, s))} \tag{2}$$

where *r* is the number of candidate phrases from the web page.

***Appearance in title and meta tags***

The content of the title tag is the second most important source that yields useful hints for the correct title. The meta tag with *name=title* or *property="og: title"* usually contains the same information as the title tag when it exists. Therefore, a candidate phrase that appears in these tags is valuable. The scores resulting from the comparison of the candidate phrase, the title and the meta tags are binary: 1 if a match is found in any of the tags; 0 otherwise.

***Popularity on the web page***

We use term frequency (TF) to count the number of times a candidate phrase appears on the web page in total. More popular phrases have better chances of being a title. We normalize the score by the frequency of most popular phrase ($p_r$) to the scale [0, 1] as follows:

$$S_f(p) = \frac{TF(p)}{max(TF(p_r))} \tag{3}$$

***Appearance in heading tags***

Heading (h1, h2…h6) tags emphasize the headlines and important text on the web page. We consider a phrase that appears in any of the heading tags more important than other candidates. We navigate through the entire page and count the number of times a candidate phrase appears in each heading. A phrase within <*h*x></*h*x> tag is given a score between [0, 1] as follows:

$$H(p) = \sum_{i=1}^{6} w_i f_i$$

$$S_h(p) = \frac{H(p)}{\max(H(p_m))} \tag{4}$$

where $f_i$ is the frequency of a candidate phrase (*p*) in heading $h_{i,}$ and $w_i$ is the weight of heading $h_i$. Similarly to Fan et al. (2011), the weights of the headings are fixed to (6, 5, 4, 3, 2, 1) respectively, with $h1$ having the biggest weight and $h6$ the lowest weight, *m* is the total number of candidate phrases that appear in all heading tags.

***Capitalization and capitalization frequency***

Capitalization feature takes value 1 if the candidate phrase starts with a capital letter; otherwise, it will be 0. The hypothesis is that majority of the titles starts with a name. However, a phrase might also start with a capital letter depending on its position in the text such as sentence-initially; therefore, we also consider capitalization frequency. It counts the number of times a candidate phrase starts with a capital letter on the web page. We use Eq. 3 to compute the normalized score for capitalization frequency, (but counting only when the phrase starts with a capital letter).

***Independent appearance***

Important phrases such as a title and a headline can appear separately in the nodes of the DOM tree. Therefore, we consider a candidate phrase that appears independently more important than a phrase that appears as a part of other phrases in the text nodes. We calculate the normalized score similarly as in Eq. 3, (but counting only when the phrase appears individually on the web page).

***Phrase length***

The last feature is the length of the candidate phrase. Figure 6 shows the distribution of the title and non-title phrases in our corpus. Most extracted phrases are less than eight characters long; however, they are not as likely to be titles as phrases which are longer. Phrases longer than 8 characters are more likely to be title phrases, and phrases shorter than this are less likely to be the title. We compute the score of the length feature by counting the number of phrases that have the same length of the candidate phrase divided by the most frequent length. For example, if the length is 16 characters and the frequency of this length is 69, then the length score is 69/124= 0.56, where 124 is the frequency of the most frequent length (9 character-long titles).

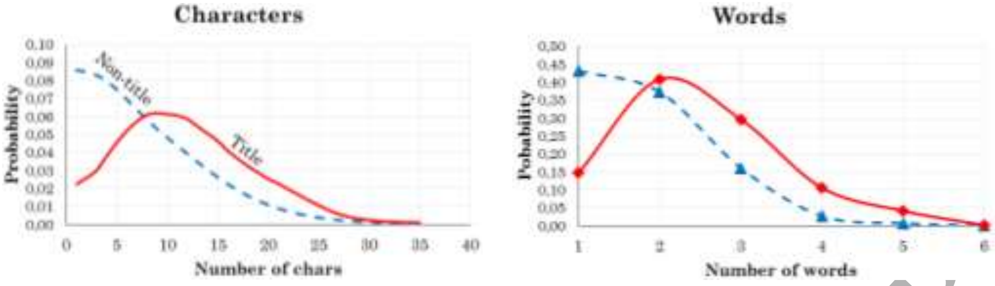


**Figure 6** Probability of length for titles and candidate phrases: characters (left) and words (right).

## 3.5. Classification of candidate phrase

After the features for all candidate phrases have been computed, we generate the feature vectors and manually label them either as *title* or *non-title*, based on the similarity of the candidate phrase with the ground truth titles. In the case of a perfect match, we label the feature vector as a *title*; otherwise, we label it as *non-title*. We then proceed by training different models to extract the titles. We consider four alternative classifiers: Naive Bayes, clustering, k-NN, and SVM.

The four classifiers consist of two phases: learning and testing. In the learning phase, the sequences of feature vectors {($\boldsymbol{X}_1$, $Y_1$), ($\boldsymbol{X}_2$, $Y_2$)… ($\boldsymbol{X}_n$, $Y_n$)} from the training set are the input to the classifier. Here $\boldsymbol{X}_i$ denotes a feature vector of a candidate phrase; $Y_i$ denotes the class label of the vector $\boldsymbol{X}_i$, and *n* is the number of the candidate phrases in the training set. In the testing phase, a sequence of feature vectors {$\boldsymbol{X}_1$, $\boldsymbol{X}_2$…$\boldsymbol{X}_r$} with unknown class labels is input to the classifier, and the output is a sequence of predicted labels {$Y_1$, $Y_2$…$Y_r$}, where *r* denotes the number of candidate phrases on a web page.

### 3.5.1. *Naive Bayes*

Naive Bayes (Domingos & Pazzani, 1997) is a probabilistic model, in which conditional probabilities are calculated from the training data. Classification is done by taking the one with the highest probability given by the features. It is based on Bayes' theorem (Duda, Hart, & Stork, 2001). Naive Bayes assumes that the probability of a feature is independent of the other features. Given a feature vector $X$=($x_1$, $x2$,…, $x_D$) and a class label ($y$), the goal is to find the most likely class for $X$.

In the learning phase, we estimate the prior probability $P(y)$ for each class $y\in$ {title, non-title} by counting their relative frequency in the training data. We also use the training data to estimate the conditional probability $P(x_i|y)$ for each feature $x_i$ occurring in the class $y$. We use multinomial distribution (Manning, Prabhakar, & Hinrich, 2008) to estimate the probability $P(x_i|y)$ because it performs better than Gaussian and Bernoulli models for the text classification task according to (McCallum & Nigam, 1998; Manning et al., 2008).

In the testing phase, we calculate the posterior probability ($\hat{y}$) for each class given the feature vector, and then assign the candidate phrase to the most probable class. We select the phrase with the highest title probability.

$$\hat{y} = arg \max_{y\in Y} P(y) \prod_{i=1}^{D} P(x_i|y) \tag{5}$$

### 3.5.2. *Clustering-based*

The overall process of the clustering-based model is shown in Figure 7. In the learning phase, we cluster the feature vectors from the training dataset, using Randomized Swap algorithm (RS) (Fränti & Kivijärvi, 2000). Random Swap alternates between performing k-means and randomly relocating centroids to allow k-means to escape local optimum. It has a time complexity of $O(MNDI)$ where $M$ is the number of clusters, $N$ the number of feature vectors, $D$ is the number of dimensions and $I$ the number of swaps. A higher number of iterations typically yield better results in terms of total squared error.

After we obtain the clustering solution, we determine the dominant label of each cluster. For example, if a cluster contains 20 titles and 16 non-titles, it is labeled as a title cluster with 56 % probability. Likewise, a cluster that contains 6 titles and 130 non-titles is labeled as a non-title cluster with 96 % probability.

In the testing phase, we compute the feature vector for all candidate phrases and map them to the nearest cluster centroid using Euclidean distance in the feature space. This takes $O(MD)$ time per candidate title. The candidate phrase that is classified to a cluster with the highest title probability is chosen.

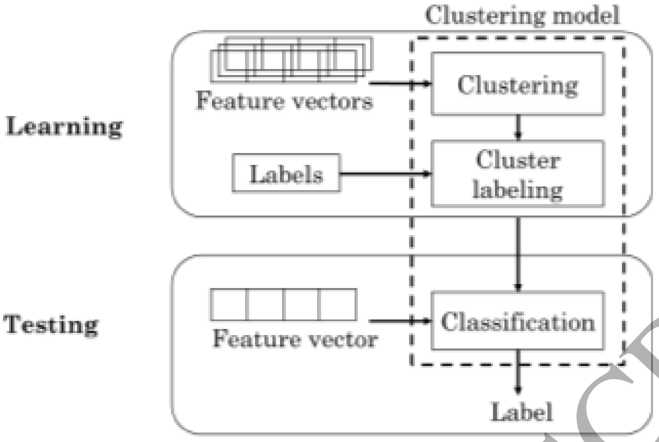


**Figure 7** Clustering-based title extraction.

### 3.5.3. *k-nearest neighbors (k-NN)*

K-NN (Cover & Hart, 1967) does not train any model, but it simply stores the feature vectors of the training data as such. In the testing phase, we compute the feature vector for all candidate phrases and map them in the feature space. The k-nearest training vectors are found for every new unlabeled feature vector, and its class is selected by the majority rule: a class that has the most representatives within the nearest neighbors is chosen. We use Euclidean distance to find the neighbors. We choose the phrase that has the highest number of neighbors with the label *title* as the page title. In the case of a tie, we just select the class of the first neighbor found.

### 3.5.4. *Support Vector Machines (SVMs)*

Support vector machine (Vapnik, 1995) is a binary classifier that represents the feature vectors as points in the feature space so that points of different classes are separated by the so called maximum-margin hyperplane. SVM has shown to be well suited for text categorization ((Joachims, 1998).

In the learning phase, SVM aims at finding an optimal hyperplane that maximally separates the two classes of the training data $y_i \in$ {title, non-title}. We use radial basis function (RBF) kernel as it performs better than linear and polynomial kernels (see subsection 4.3). Given two feature vectors $\boldsymbol{X}$ and $\boldsymbol{X}'$, the RBF kernel is defined as follows:

$$K(\boldsymbol{X}, \acute{\boldsymbol{X}}) = exp(-\gamma\|\boldsymbol{X} - \acute{\boldsymbol{X}}\|^2) \qquad (6)$$

where $\|\boldsymbol{X} - \acute{\boldsymbol{X}}\|^2$ is the squared Euclidean distance between the feature vectors, and $\gamma = 2^{-8}$ is a parameter, which defines how much influence a single training example has.

In the testing phase, a new unlabeled feature vector is mapped into the same space and predicted to belong to a class based on which side of the hyperplane it falls into. We select the one with the largest distance to the support vector as the title.

## 4 Experiments

We perform a set of experiments to evaluate our proposed method. Experiments also reveal the effect of the POS patterns and the best parameter selection for the method. We compare to all existing popular methods such as Title tag (the baseline), Google, Title Tag Analyzer (Gali & Fränti, 2016), TitleFinder (Mohammadzadeh et al., 2012) and Styling (Changuel et al., 2009). We use two datasets in our evaluation: Titler and Mopsi services. The former is used for training and testing by following a k-fold cross validation strategy. The latter is a more difficult, multiple language dataset, which we use to stress test our models. Furthermore, we perform an evaluation of the features. There is no question that the features are useful, however, some features' usefulness is diminished in the presence of others. This complementary aspect varies depending on the method (classifier) used. We conclude our experiments by especially addressing the newly proposed POS feature and its usefulness when combined with the strategy of segmenting the text inside the DOM tree nodes.

### 4.1. Datasets

We use the Titler corpus detailed in subsection 3.1. After the POS patterns had been identified, we extracted 152,163 unique candidate phrases from all the web pages. The number of phrases per page varies from 2 to 3,166 (152 phrases on average). All phrases in a web page that match the given ground truth title are labeled as *title*, the rest as *non-title*. The labeling was done manually. We then divide the data into five folds to conduct a cross-validation. In each iteration, we use four folds (80%) of the data for training and the remaining (20%) for testing.

In addition, we use Mopsi services dataset[10]. Mopsi (Fränti, Chen, & Tabarcea, 2011), implements various location-based services and applications such as mobile search engine, data collection, user tracking, and route recording. It has applications integrated both on web and in mobile phones. Mopsi services contain 414 user collected places. Each service has a web link, title, keywords, photo, additional description (optional), and address that are manually added by the users and further confirmed by an administrator. For the sake of this study, we also downloaded the content of web pages and stored them so that the same content is available for further tests.

Mopsi services include a wide variety of web links such as *home*, *brand*, *business directory*, *Wikipedia*, *Facebook*, *blog* and *information* pages and they cover various domains such as *education*, *health*, *shop*, *bank*, *entertainment*, *sport*, *food & drink*, *hotel & accommodation*, *travel & leisure*, and *news*. The pages include several languages such as *Finnish (375 pages), English (35 pages)*, and few pages in *French*, *Italian*, *Spanish*, *Estonian* and *Swedish*. This will put the developed method also in stress-test since the method covers only English.

In both datasets, the relevant information about the services such as the title, description and opening hours is commonly static because users are expected to rely on this information. However, if the target pages are expected to be fully dynamic, we recommend to use a WebKit browser such as *PhantomJS*[11] to render the web page prior to the application of our method.

### 4.2. Evaluation measures

To evaluate the performance of the title/non-title classifier, we use precision, recall, and F-score, which are widely used to evaluate information extraction systems. We count the following classification results:

*tp* = number of candidate phrases correctly identified as titles
*fp* = number of candidate phrases incorrectly identified as titles
*fn* = number of correct candidate phrases erroneously identified as non-titles.

$$Precision = \frac{tp}{tp + fp} \tag{7}$$

$$Recall = \frac{tp}{tp + fn} \tag{8}$$

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{9}$$

To evaluate the quality of the extracted title phrases, we use four different measures:

1. Rouge-N (Lin, 2004): a well-known N-gram metric that measures the overlap units between the candidate and the ground truth titles. It is defined by the precision, recall, and F-score as follows:

$$Precision = \frac{c_m}{cd} \tag{10}$$

[10] http://cs.uef.fi/mopsi/MopsiSet
[11] http://phantomjs.org/

$$Recall = \frac{c_m}{gt} \tag{11}$$

$$F = \left(alpha \times (\frac{1}{precision}) + (1 - alpha) \times (\frac{1}{recall})\right)^{-1}, alpha = 0.5 \tag{12}$$

Here *cd* and *gt* are the number of n-grams in the candidate titles and the ground truth, and $c_m$ is the number of common n-grams co-occurring in both of them. We use Rouge-1 as was reported to work best for a very short text in (Lin, 2004).

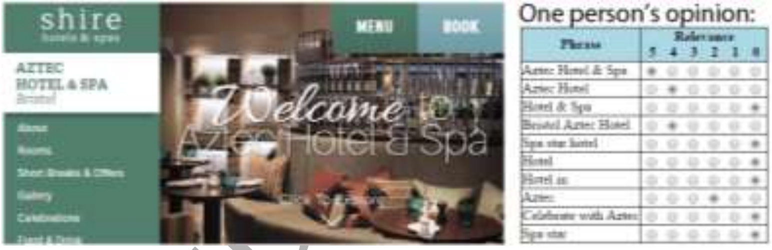
2. Jaccard index: It counts the number of common character bigrams divided by the total number of unique bigrams in both strings:

$$Jaccard\ (cd, gt) = \frac{|bigrams(cd) \cap bigrams(gt)|}{|bigrams(cd) \cup bigrams(gt)|} \tag{13}$$

where cd stands for candidate title and gt stands for ground truth title.

3. Dice coefficient: It counts the number of shared character bigrams divided by the total number of bigrams in both strings; see Eq.1.
4. Human judgment: we developed a tool[12] (see Figure 8) to allow users to rank the candidate phrases. Phrases that have either high character-level or high word-level similarity (>0.8) with the ground truth titles are shown. The user then rates the candidates using a scale from 0 (irrelevant) to 5 (exact match). The quality of the chosen candidate phrase is measured as the average of the human ratings. We refer the human judgment as *accuracy* in the rest of the paper. It counts the number of titles having average ratings > 1.

We further apply the Mann Whitney U-test for significance testing.



One person's opinion:

| Phrase | Relevance 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|
| Aztec Hotel & Spa | ● | ○ | ○ | ○ | ○ | ○ |
| Aztec Hotel | ○ | ● | ○ | ○ | ○ | ○ |
| Hotel & Spa | ○ | ○ | ○ | ○ | ○ | ● |
| Bristol Aztec Hotel | ○ | ● | ○ | ○ | ○ | ○ |
| Spa star hotel | ○ | ○ | ○ | ○ | ○ | ● |
| Hotel | ○ | ○ | ○ | ○ | ○ | ● |
| Hotel in | ○ | ○ | ○ | ○ | ○ | ● |
| Aztec | ○ | ○ | ○ | ● | ○ | ○ |
| Celebrate with Aztec | ○ | ○ | ○ | ○ | ○ | ● |
| Spa star | ○ | ○ | ○ | ○ | ○ | ● |

**Figure 8** Evaluation tool[13] for human ratings.

### 4.3. Classifiers setup

The parameters of the classifiers were optimized by brute force (grid search) using the training set. The tested parameter values and the best obtained combinations are shown in Table 3.

**Table 3** Best combination of parameters obtained for each classifier

| Classifier | Best obtained parameters | Testing range |
|---|---|---|
| Bayes | Laplace smoothing α=1 | Lidstone smoothing α = 0.1 – 0.9,<br>Laplace smoothing α =1 |
| Clustering | clusters = 2048 | clusters = 256 – 8192 by doubling |
| KNN | k = 30<br>Euclidean distance<br>Uniform | k = 1 – 40<br>Manhattan distance, Euclidean distance<br>Weight of neighbors: uniform, weighted |
| SVM | c = $2^{10}$<br>$\gamma = 2^{-8}$<br>Kernel= RBF | c = $2^5$ – $2^{20}$ , increment by 5<br>$\gamma = 2^{-6} - 2^{-13}$<br>Kernel = linear, polynomial, RBF |

In clustering, the problem of overfitting happens after 4096 (see Figure 9); therefore, we select 2048. For SVM, we subsample the training set by randomly discarding examples from the majority class (non-title) until the training set becomes

[12] http://cs.uef.fi/mopsi_dev/Titler/TitleRater.php
[13] http://www.shirehotels.co.uk/aztec/

balanced. Table 4 shows the results with linear, polynomial and RBF kernels. The RBF kernel provides slightly better result in the recall and the F-score in comparison with polynomial and linear kernels respectively and is therefore selected. We use RBF with c=$2^{10}$ and $\gamma = 2^{-8}$ for the further experiments. For all learning models, we conduct a five-fold cross-validation. All the results reported here are averaged over five trials.

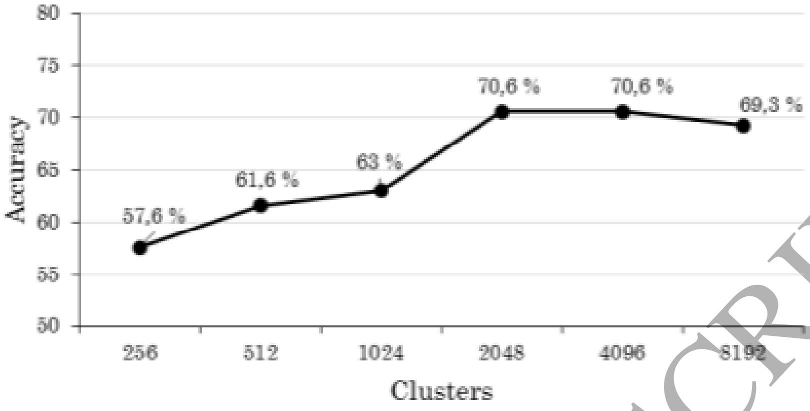


**Figure 9** Titling accuracy with different numbers of clusters.

**Table 4** Performance of SVM with different kernel functions

| SVM model type | Precision | Recall | F-score |
|---|---|---|---|
| Linear kernel | 0.44 | 0.84 | 0.57 |
| Polynomial kernel | 0.47 | 0.79 | 0.59 |
| RBF kernel | 0.44 | 0.84 | 0.58 |

### 4.4. Methods evaluated

We compare the following methods:

| | |
|---|---|
| • Title tag | (baseline) |
| • Google | (search engine) |
| • Title Tag Analyser (TTA) | (Gali & Fränti, 2016) |
| • TitleFinder | (Mohammadzadeh et al., 2012) |
| • Styling | (Similar to Changuel et al., 2009) |
| • Titler (BAYES) | (proposed) |
| • Titler (CLUS) | (proposed) |
| • Titler (KNN) | (proposed) |
| • Titler (SVM) | (proposed) |

As a baseline, we use the content of the title tag as such. We compare to the titles provided by Google search engine in the results page. We also compare our method with TTA which is a simplified version of the proposed method where only title and meta tag segmentation is applied but without POS tag patterns and other linguistic features, and with TitleFinder which uses the content of the title tag as a feature. We re-implemented a Styling method similar to Changuel et al. (2009) with the following features:

Tags

- Level of headings: h1…h6;
- Div: division or section;
- Span: group inline elements;
- P: paragraph;
- A: anchor;

- Strong: important text;
- B: bold;
- U: underline;
- I: list item;
- Em: emphasized text.

Formats

- Font size: level 1-7;
- Font weight: bold;
- Font family: Arial, Calibri…
- Font color: RGB values converted to the YUV color space;
- Alignment: top, left…

Format and tag changes

- All formats and tags change with the previous nodes;
- All formats and tags change with the next nodes.

We use decision tree and random forest classifiers as in Changuel et al. (2009). We did not compare with Xue et al. (2007), which is an updated version of Hu et al. (2005) as the authors mentioned 254 features, but only a few are explained to allow reproducing. However, Changuel et al. (2009) uses very similar features to (Xue et al., 2007). Both methods use visual and formatting features to extract the titles. In Section 4.9, we demonstrate that these kinds of features are disadvantageous for service-based web pages.

### 4.5. The POS patterns

After applying the POS patterns in the text of the web pages, we observed that the exact titles are found by the patterns in 93% of the web pages and approximate titles in 4.7% of the cases according to the human judgment. The patterns fail to extract correct title phrases in 2.3% of the pages (see Table 5). The failures happen with web pages that have the exact title as an image and do not provide useful text (see Figure 10, right).

**Table 5** Summary of the titles extracted by POS patterns

| Type of title | Number of web pages | Proportion (%) |
|---|---|---|
| Perfect match | 932 | 93 |
| Approximate match | 47 | 4.7 |
| Not found | 23 | 2.3 |

A further analysis of the patterns revealed that 36% of the extracted title phrases match the pattern *<NNP><NNP>*, 17% match *<NNP><NNP><NNP>*, and 14% match *<NNP>*, which are all proper nouns (see Table 6). Patterns *<JJ><VBG>NNP>* and *<DT><CD><NNPS>* are rare as only 0.3% of the title phrases match these patterns. However, they are effective because only titles appear in this format. About 2% of the title phrases are extracted by the pattern *<VB>*, while it extracts 5% of the non-title phrases, which makes it less important. No significant observations were found from the rest of the patterns. We conclude that noun patterns have the highest impact in comparison to the other type of patterns such as verbs, which extract only 2% of the titles when they are used separately.

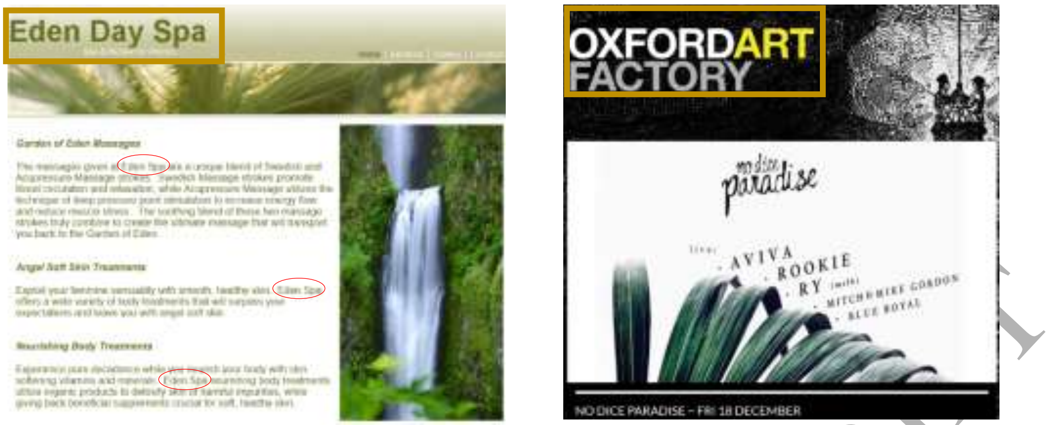


**Figure 10** Exact title appears as an image and partially in the text (left)[14] and only as an image (right)[15].

**Table 6** Summary of the effectiveness of individual patterns

| Type of pattern | Example of titles | Proportion (%) |
|---|---|---|
| <NNP><NNP> | Aqua Dining | 36 |
| <NNP><NNP><NNP> | Woolwich Pier Hotel | 17 |
| <NNP> | Ventuno | 14 |
| <VB> | Recreate | 2 |
| <JJ><VBG>NNP> | Functional Training Ireland | 0.3 |
| <DT><CD><NNPS> | The Five Bells | 0.3 |
| Others | Spice Me UP | 30 |

### 4.6. Feature selection

We next investigate the importance of the individual features to evaluate their potential benefit. We use a wrapper method called greedy backward elimination (Kragh, Jørgensen, & Pedersen, 2015). It starts by using all the features in combination, and then gradually removes one feature at a time. At each step, the classifier is trained with the current combination of the features and the average of five-fold cross validation is used as a score. The feature whose removal improves the performance most is eliminated for the next iteration. The process continues iteratively until no further improvement. Results are reported in Table 7.

Although all features are relevant, some of them are not useful in the presence of others. Capitalization frequency is disadvantageous in both BAYES and SVM. Title length seems to be a bad feature in BAYES and CLUS but helps the others. Likewise, heading and frequency features seem to be bad in CLUS and SVM respectively, but useful in BAYES and KNN. Both CLUS and KNN require fewer features (6 in CLUS and 8 in KNN) to provide the best results (82.4% and 84.8%). This is because, in high dimensional space, the distance to the nearest data point approaches the distance to the farthest data point when the dimension increases. Consequently, distance-based classifiers become less accurate when too many features are used. The same phenomena were also observed in (Beyer, Goldstein, Ramakrishnan, & Shaft, 1999).

The experiments show that the similarity with the link of the web page has the highest impact on all models. By removing this feature, the accuracy would decrease significantly: BAYES: 54.8% → 41.1%, CLUS: 70.6% → 57.8%, KNN: 79.2% → 62.8%, SVM: 84.9% → 56.4%. Surprisingly, meta feature is harmful in all models and its removal always improves the performance. This is because the meta tag is rarely found on the page and even when existing, the appearance in meta tag feature fails. The rest of the features show a positive effect on the overall accuracy.

**Table 7** Impact of the features on the accuracy (%) after eliminating the corresponding feature

| BAYES | CLUS | KNN | SVM |
|---|---|---|---|
| All features (54.8) | All features (70.6 ) | All features (79.2) | All features (84.9) |
| − Capt. Freq. (58.5) | − POS (77.4) | − POS (82.8) | − Capt. Freq. (85.6) |
| − Title length (59.6) | − Heading (80.9) | − Meta (**84.8**) | − Frequency (85.7) |
| − Meta (**60.1**) | − Title length (80.9) | | − Meta (**85.9**) |
| | − Meta (**82.4**) | | |

[14] http://www.edensauna.com/
[15] http://oxfordartfactory.com/

### 4.7. The classifiers

We evaluate the performance of the classifiers using the best set of features found in subsection 4.6. The results in Figure 11 show that CLUS (0.80) and KNN (0.81) outperform BAYES (0.60) and SVM (0.44) in the precision, whereas SVM provides better results in the recall (0.84) and F-score (0.58). SVM classifies several non-title phrases incorrectly in the title class, which harms the precision, but it still classifies the majority of the title phrases correctly in the title class. A less number of non-title phrases are classified in title class by other models, which provides better precision, but the percentage of the title phrases correctly classified in title class are smaller when compared with SVM, and therefore the recall is low.

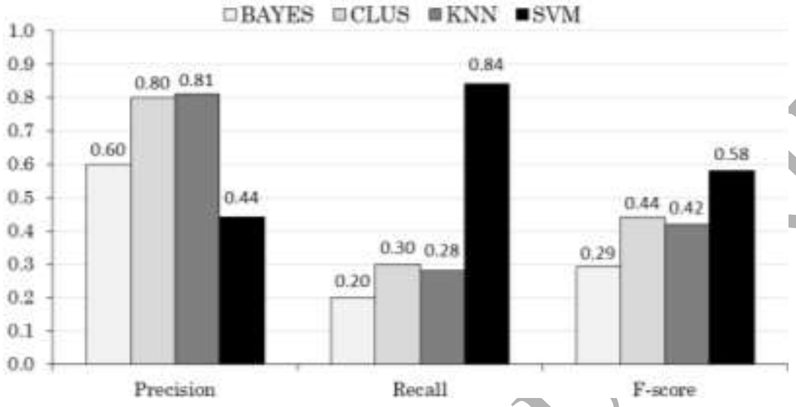


**Figure 11** Performance results for the classifiers.

### 4.8. Title selection

Among the phrases that are classified as a title, we select the one with the highest confidence. We compare this against random choice. Figure 12 show that there is only a small difference between these two approaches in case of BAYES, CLUS, and KNN. In the case of SVM, however, the random choice works poorly (50%). This is because several non-title phrases (FP) are classified in the title class by SVM. The probability of choosing a non-title phrase by random approach is, therefore, high. The title class in other models is more balanced; therefore a random selection performs well.

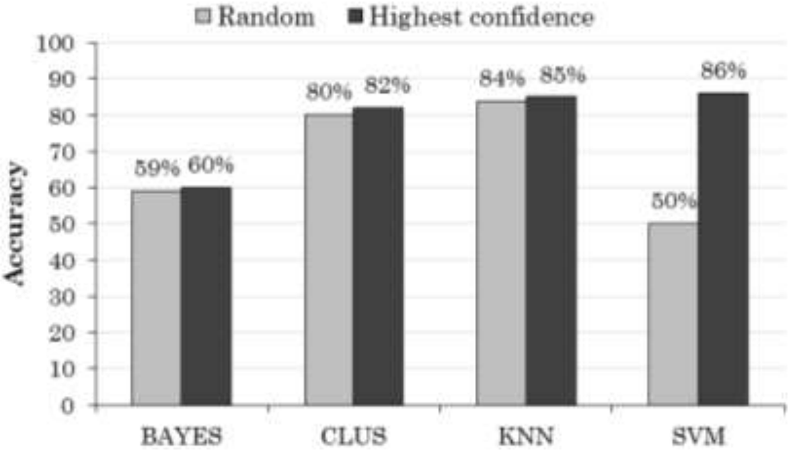


**Figure 12** Accuracy of title selection methods.

We also compare the quality of the correctly extracted titles to the human judgment. In Figure 13, we observe that the majority of the titles match perfectly with the ground truth titles for all models. BAYES tends to provide shorter titles, such as *Cava* for *Cava restaurant* while SVM, CLUS, and KNN tend to provide more descriptive titles such as *Home Leisure Direct*.

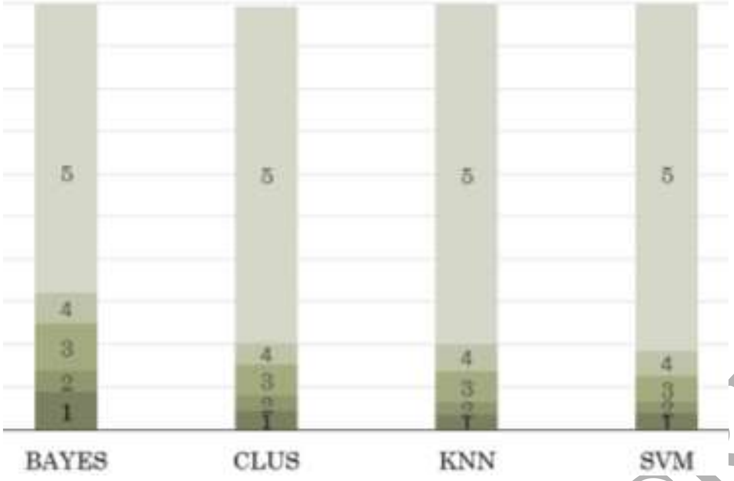


**Figure 13** Quality of titles (confidence 5 is the perfect match and confidence 1 is satisfactory).

### 4.9. Comparative results for all methods

The overall results are compared to those existing methods that were available, and to the baseline (title tag) using the Titler corpus and Mopsi service dataset. The results of the Rouge-1, Jaccard, and Dice measures are summarized in Tables 8 and 9. The proposed method clearly outperforms the Baseline, Google, TitleFinder, Styling and TTA except for BAYES, for the Titler corpus. One reason is that many web pages do not have their titles appear independently in the text nodes, and therefore, selecting the entire text node as in TitleFinder and Styling is harmful. This decrease their overall performance (TitleFinder 0.47) and (Styling 0.35). Our method extracts only potential substrings from each node.

The recall of the Baseline and Google is high (0.89) because many web pages include the correct titles in their title tag, but also include other irrelevant text which causes a decrease in the precision (0.41) and (0.48) respectively. The Baseline, Google, TitleFinder, TTA and Titler all provide better results than Styling method because the majority of the web pages (88.8%) has their titles integrated within the logo image and presented in the text only much later with no styling differences from the other parts of the text. In this kind of web pages, the more visual focus is given to the advertisements or products (see Figure 14). Therefore, style-dependent features do not work well in general.

We also conducted statistically significance tests (Mann Whitney U-test) for the Rouge-1. The results indicate that the advantages of our method over the Baseline, Google, TitleFinder, TTA and Styling method are statistically significant (*P* value < 0.05). No significance difference is noticed between SVM, KNN, and CLUS. We also count the number of titles that perfectly match with the ground truth titles. The methods find the perfect match as follows: SVM 700, KNN 677, CLUS 644, and BAYES 406 times. These are significantly more than that of the TTA 555, TitleFinder 212, Google 192, Styling 162, and Baseline 149.

When testing with Mopsi dataset, we use KNN classifier because it performs well without the linguistic feature (POS) as shown in subsection 4.6. We extract all n-grams as candidate phrases because we do not have POS taggers for all the languages in the set. From Table 9, we observe that the proposed method still outperforms other methods even though the overall result decreases compared to Titler corpus. One reason is that Mopsi set is more challenging due to the high diversity of the domains, functions, and templates of the web pages. Another reason is that, unlike Titler corpus, the titles in Mopsi services are annotated and may not always appear on the page in the text at all. To test the effect of this, we added alternative ground truth titles by manual extraction. This improved the F-scores of all methods: Titler (0.55 → 0.70), TTA (0.52 → 0.67), Google (0.43 → 0.52), Baseline (0.41 → 0.52), TitleFinder (0.37 → 0.47), and Styling (0.15 → 0.20). These results show that the real problem is more challenging than just extracting the best phrase from the text.

The existence of multiple languages in the dataset also decreases the performance since the linguistic model was designated for English only. TTA performs fairly well, which indicates that the title tag itself is still useful, but its segmentation is important. The performance of the Styling method is still the lowest among all the methods tested.

We have also tested the Styling method with two small sets of web pages from Eureka[16] an online educational portal used by Changuel et al. (2009) and Wikipedia. The accuracies were 85% and 70% respectively. This confirms that the methods depend only on the styling information are suitable for web pages that follow a standard format such as news and education pages, but not for more general pages as in our case.

[16] http://eureka.ntic.org/

**Table 8** Comparative results for Titler corpus

| Method | Rouge-1 | | | Jaccard | Dice |
|---|---|---|---|---|---|
| | Precision | Recall | F-score | | |
| Baseline | 0.41 | 0.89 | 0.52 | 0.50 | 0.58 |
| Google | 0.48 | 0.89 | 0.58 | 0.56 | 0.64 |
| TitleFinder (Mohammadzadeh et al., 2012) | 0.43 | 0.61 | 0.47 | 0.43 | 0.50 |
| Styling (Changuel et al., 2009) | 0.36 | 0.41 | 0.35 | 0.38 | 0.43 |
| TTA (Gali and Fränti, 2016) | 0.73 | 0.80 | 0.75 | 0.75 | 0.78 |
| Titler (BYAES) | 0.46 | 0.40 | 0.42 | 0.64 | 0.70 |
| Titler (CLUS) | 0.87 | 0.81 | 0.82 | 0.82 | 0.86 |
| Titler (KNN) | 0.87 | 0.82 | 0.83 | 0.84 | 0.87 |
| Titler (SVM) | 0.88 | 0.84 | 0.85 | 0.85 | 0.88 |

**Table 9** Comparative results for Mopsi services

| Method | Rouge-1 | | | Jaccard | Dice |
|---|---|---|---|---|---|
| | Precision | Recall | F-score | | |
| Baseline | 0.33 | 0.71 | 0.41 | 0.44 | 0.54 |
| Google | 0.34 | 0.74 | 0.43 | 0.46 | 0.56 |
| TitleFinder (Mohammadzadeh et al., 2012) | 0.35 | 0.47 | 0.37 | 0.37 | 0.43 |
| Styling (Changuel et al., 2009) | 0.14 | 0.21 | 0.15 | 0.22 | 0.28 |
| TTA (Gali and Fränti, 2016) | 0.52 | 0.59 | 0.52 | 0.54 | 0.62 |
| Titler (KNN) | 0.59 | 0.56 | 0.55 | 0.59 | 0.66 |

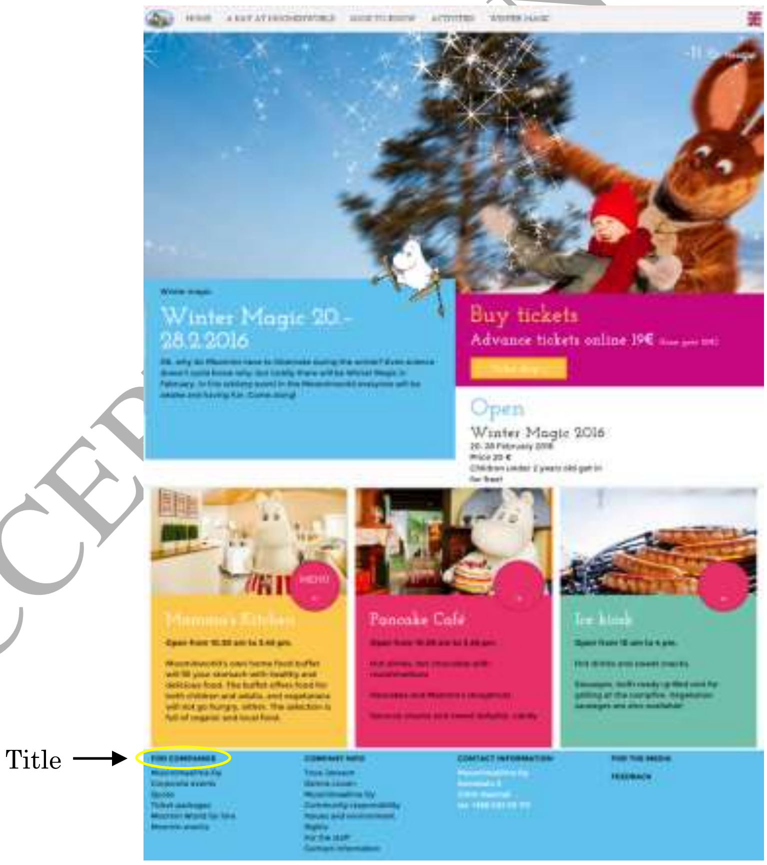


**Figure 14** Example of a web page gives visual focus on the products and events (Muumimaailma)[17].

### 4.10. Effect of segmentation

The biggest deficiency of the existing methods is that they use the content of the text nodes (or title tag) as such. We tested how much the methods can be improved by segmenting the content of the nodes (N-grams), and how much by using the POS patterns to filter out the unwanted phrases. We consider two methods: TitleFinder and Titler (with k-NN). We exclude Styling method because it mainly depends on the features within the surrounding text, which would be the same for all candidate phrases in the same node, and therefore, segmentation would not make any difference.

Results in Table 10 show that both TitleFinder and Titler are improved by the segmentation, and the effect is significant. When using N-gram phrases, Titler is improved to 0.73 whereas TitleFinder to 0.55. Using POS patterns improves further to 0.83 and 0.60. We conclude that segmentation plays an important role in title extraction task.

We further observe that the POS patterns have another important effect that is not shown in the numbers. After segmentation, the structure of the titles often becomes incorrect, such as "*in Bristol Aztec Hotel & Spa*" and "*to Essen restaurant*" but these are still rated as correct in the numerical evaluation. The use of POS pattern is, therefore, important processing step needed with the segmentation to avoid grammatically incorrect titles.

**Table 10** Effect of phrase segmentation (N-grams) and POS patterns with TitleFinder and Titler methods

| Method | Pre-processing | Rouge-1 | | | Jaccard | Dice |
|---|---|---|---|---|---|---|
| | | Precision | Recall | F-score | | |
| TitleFinder (Mohammadzadeh et al., 2012) | Complete node | 0.43 | 0.61 | 0.47 | 0.43 | 0.50 |
| | N-grams | 0.53 | 0.65 | 0.55 | 0.53 | 0.60 |
| | POS patterns | 0.61 | 0.63 | 0.60 | 0.58 | 0.63 |
| Titler (KNN) | Complete node | 0.46 | 0.54 | 0.47 | 0.51 | 0.58 |
| | N-grams | 0.79 | 0.73 | 0.73 | 0.74 | 0.80 |
| | POS patterns | 0.87 | 0.82 | 0.83 | 0.84 | 0.87 |

## 5 Conclusion

We have proposed a new method to extract the title from HTML web pages using text segmentation, statistical features of the main content, and linguistic knowledge. The proposed method outperforms the best existing method (Google) by a large margin; F-score of the Rouge-1 measure is improved from 0.58 to 0.85 in the case of Titler corpus, and from 0.43 to 0.55 in the case of Mopsi services. More detailed analysis revealed the following findings that explain the results:

- Segmentation of the text nodes is most important. Results showed that it is not enough to find the correct text node from the DOM tree, or use the title tag as such. Segmenting the content to n-grams and further analysis by POS tagging improved both TitleFinder (0.47→0.60) and the proposed method Titler (0.47→0.83) significantly.
- The link of a web page has the highest impact on the overall performance of the method. It works well because the majority of the web pages have the title also appear in the link. Meta feature is not useful possibly because it is rarely specified and given a correct value.
- SVM model achieved the highest F-score (0.85) for the title extraction task, but with no significance difference from k-NN (0.83) and clustering (0.82) models. Both k-NN and clustering are simple to implement and easy to understand in comparison with the underlying theory behind SVM. Naive Bayes achieved the lowest F-score (0.42).
- The POS patterns models can be generalized to any language provided that a corpus and a POS tagger for the specified language are available. Our k-NN model works reasonably well with other languages; however, we did not experiment extensively in this regard.
- Noun phrases are more effective than adjective, determiner, propositional, adverb and verb phrases.
- A drawback of the POS tagging of the web pages is the slowness, especially when the amount of text is big.
- A large number of pages (89%) show the title within a logo image. The title in the text content is therefore not highly emphasized or might even be missing completely. This gives the biggest challenge to the title extraction. An alternative approach would be to use image analysis, but this raises even bigger challenges. First, one should detect which one is the logo image (see Gali, Tabarcea, & Fränti, 2015) for a possible approach). Second, the content of the image are highly complex, and standard OCR approach would not work as such.

[17] http://www.moominworld.fi/

The proposed method works well on both static and dynamic web pages because the most relevant content is usually static. If the target pages are expected to be fully dynamic, we recommend to render the web page prior to the application of our method.

## References

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999, January). When is “nearest neighbor” meaningful? In *International conference on database theory* (pp. 217-235). Springer Berlin Heidelberg.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Brew, C., & McKelvie, D. (1996, September). Word-pair extraction for lexicography. In *Proceedings of the 2nd International Conference on New Methods in Language Processing* (pp. 45-55).

Cai, D., Yu, S., Wen, J. R., & Ma, W. Y. (2003). *VIPS: a vision-based page segmentation algorithm*. Microsoft technical report, MSR-TR-2003-79.

Changuel, S., Labroche, N., & Bouchon-Meunier, B. (2009, July). A general learning method for automatic title extraction from html pages. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition* (pp. 704-718). Springer Berlin Heidelberg.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, *29*(2-3), 103–130.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification. 2nd.*Edition. New York, NY*.

Fan, J., Luo, P., & Joshi, P. (2011, February). Identification of web article pages using HTML and visual features. In *IS&T/SPIE Electronic Imaging* (pp. 78790K-78790K). International Society for Optics and Photonics.

Fränti, P., Chen, J., & Tabarcea, A. (2011, May). Four aspects of relevance in sharing location-based media: content, time, location and network. *International Conference on Web Information Systems and Technologies*, 413–417.

Fränti, P., & Kivijärvi, J. (2000). Randomised local search algorithm for the clustering problem. *Pattern Analysis & Applications*, *3*(4), 358–369.

Gali, N. & Fränti, P. (2016). Content-based title extraction from web page. *International Conference on Web Information Systems and Technologies*, 204-210.

Gali, N., Tabarcea, A., & Fränti, P. (2015). Extracting Representative Image from Web Page. In *International Conference on Web Information Systems and Technologies*.

Gibson, D., Punera, K., & Tomkins, A. (2005, May). The volume and evolution of web page templates. In *Special interest tracks and posters of the 14th international conference on World Wide Web* (pp. 830-839). ACM.

Hulth, A. (2003, July). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 216-223). Association for Computational Linguistics.

Hu, Y., Xin, G., Song, R., Hu, G., Shi, S., Cao, Y., & Li, H. (2005, August). Extraction from bodies of html documents and its application to web page retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 250-257). ACM.

Jeong, O. R., Oh, J., Kim, D. J., Lyu, H., & Kim, W. (2014). Determining the titles of Web pages using anchor text and link analysis. *Expert Systems with Applications*, *41*(9), 4322-4329.

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Springer Berlin Heidelberg.

Kragh, M., Jørgensen, R. N., & Pedersen, H. (2015, July). Object Detection and Terrain Classification in Agricultural Fields Using 3D Lidar Data. In *International Conference on Computer Vision Systems* (pp. 188-197). Springer International Publishing.

Lafferty, J., McCallum, A., & Pereira, F. (2001, June). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML* (Vol. 1, pp. 282-289).

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, *31*(4), 721-735.

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop* (Vol. 8).

Li, Y., Zaragoza, H., Herbrich, R., Shawe-Taylor, J., & Kandola, J. (2002, July). The perceptron algorithm with uneven margins. In *ICML* (Vol. 2, pp. 379-386).

Lopez, C., Prince, V., & Roche, M. (2010, December). Automatic titling of electronic documents with noun phrase extraction. In *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of* (pp. 168-171). IEEE.

Lopez, C., Prince, V., & Roche, M. (2011, December). Automatic titling of articles using position and statistical information. In *RANLP'11: Recent Advances in Natural Language Processing* (pp. 727-732).

Lopez, C., Prince, V., & Roche, M. (2014). How can catchy titles be generated without loss of informativeness? *Expert Systems with Applications*, *41*(4), 1051–1062.

Manning, C. D., Prabhakar, R., & Hinrich, S. (2008). Introduction to information retrieval, volume 1 Cambridge University Press. *Cambridge, UK*.

Marchlonini, G. (1992). Interfaces for end-user information seeking. *Journal of the American Society for Information Science (1986-1998)*, *43*(2), 156.

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, pp. 41-48).

Mohammadzadeh, H., Gottron, T., Schweiggert, F., & Heyer, G. (2012, November). TitleFinder: extracting the headline of news web pages based on cosine similarity and overlap scoring similarity. In *Proceedings of the twelfth international workshop on Web information and data management* (pp. 65-72). ACM.

Quinlan, J. R. (1993). Machine Learning, C4. 5: Programs for machine learning. San Francisco: Morgan Kaufmann Publishers Inc.

Song, R., Xin, G., Shi, S., Wen, J. R., & Ma, W. Y. (2006, April). Exploring URL hit priors for web search. In *European Conference on Information Retrieval* (pp. 277-288). Springer Berlin Heidelberg.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003, May). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 173-180). Association for Computational Linguistics.

Vapnik, V. (1995): The nature of statistical learning theory. New York: Springer.

Wang, C., Wang, J., Chen, C., Lin, L., Guan, Z., Zhu, J., Zhang, Ch., & Bu, J. (2009, July). Learning to extract web news title in template independent way. In *International Conference on Rough Sets and Knowledge Technology* (pp. 192-199). Springer Berlin Heidelberg.

Wang, J., Li, G., & Feng, J. (2014). Extending string similarity join to tolerant fuzzy token matching. *ACM Transactions on Database Systems (TODS)*, *39*(1), 7.

Xue, Y., Hu, Y., Xin, G., Song, R., Shi, S., Cao, Y. & Li, H. (2007). Web page title extraction and its application. *Information Processing & Management*, *43*(5), 1332-1347.

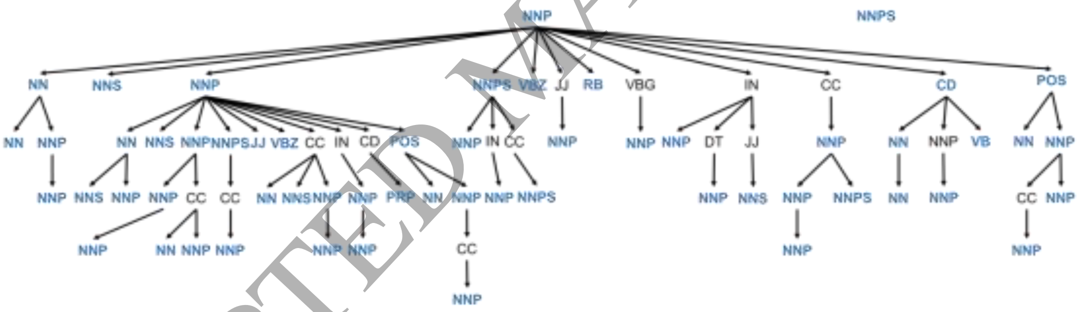
## Appendix A. Part-of-speech tags

| Index | Tag | Description | Index | Tag | Description |
|---|---|---|---|---|---|
| 1 | CC | Coordinating conjunction | 11 | NNP | Proper noun, singular |
| 2 | CD | Cardinal number | 12 | NNPS | Proper noun, plural |
| 3 | DT | Determiner | 13 | POS | Possessive ending |
| 4 | FW | Foreign word | 14 | PRP | Personal pronoun |
| 5 | IN | Preposition or subordinating conjunction | 15 | RB | Adverb |
| 6 | JJ | Adjective | 16 | RP | Particle |
| 7 | JJR | Adjective, comparative | 17 | VB | Verb, base form |
| 8 | JJS | Adjective, superlative | 18 | VBG | Verb, gerund or present participle |
| 9 | NN | Noun, singular or mass | 19 | VBP | Verb, non-3rd person singular present |
| 10 | NNS | Noun, plural | 20 | VBZ | Verb, 3rd person singular present |
|  |  |  | 21 |  | Others |

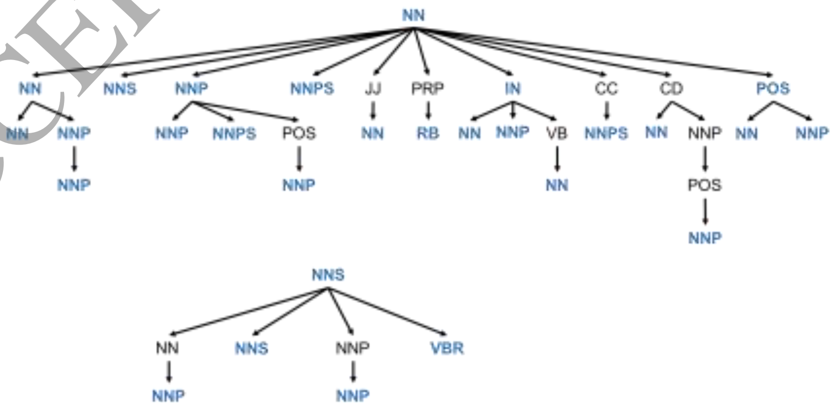
## Appendix B. POS pattern trees

In this appendix, we show a few examples on how to generate the patterns from the trees. In all trees, each blue node is a possible end of a pattern. For example, in *proper noun tree*, the root <*NNP*> is an individual pattern; <*NNPS*> is another individual pattern. From the root node, we can follow the arrows to generate longer patterns. For example, if we start from the left most side of the tree, we can generate <*NNP*>< *NN*>, <*NNP*><*NN*><*NN*> and so forth. However, a pattern from the root *NNP* cannot end at the node *JJ* because it is black. For example we do not generate a pattern <*NNP*>< *JJ*> but we generate <*NNP*><*JJ*><*NNP*>.

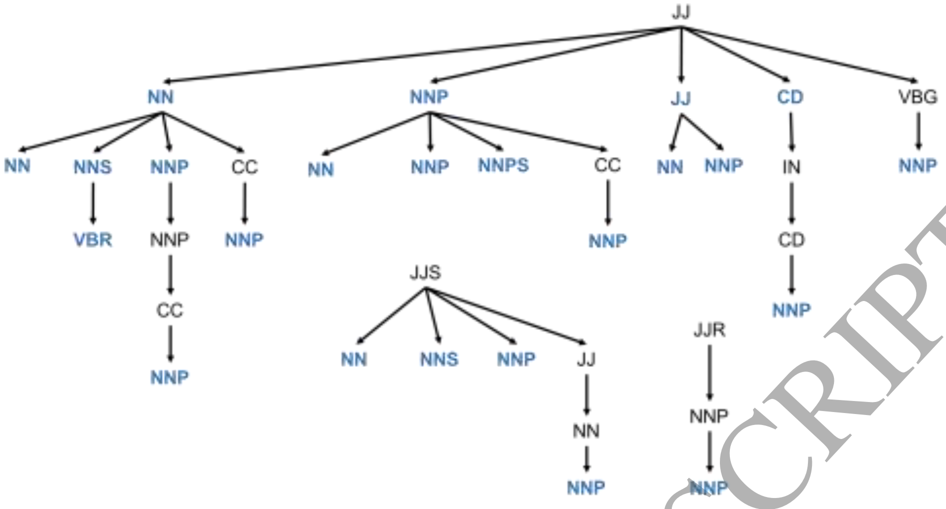
**Proper noun tree**



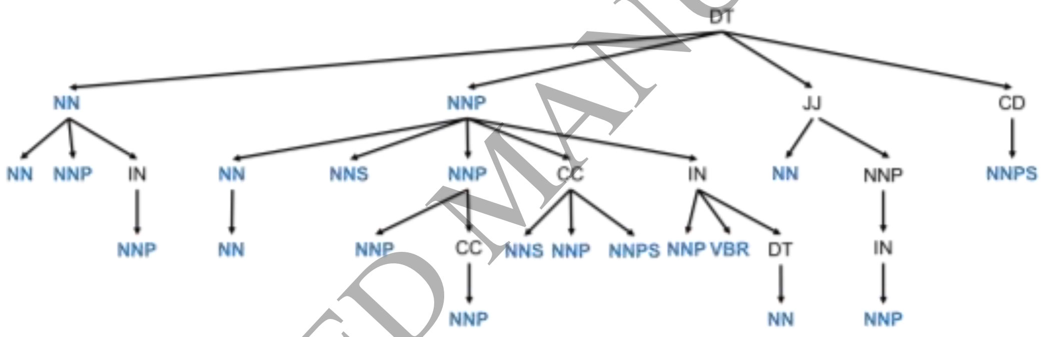
**Noun trees**



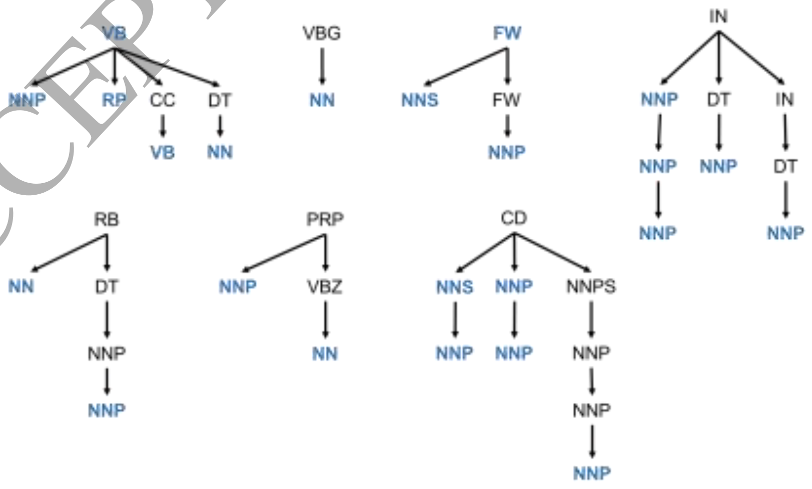
**Adjective trees**



**Determiner tree**



**Other POS trees**


VB
NNP
RP
CC
VB
DT
NN
VBG
NN
FW
NNS
FW
NNP
IN
NNP
NNP
NNP
DT
NNP
IN
DT
NNP
RB
NN
DT
NNP
NNP
PRP
NNP
VBZ
NN
CD
NNS
NNP
NNP
NNP
NNPS
NNP
NNP
NNP