Theme Extraction in Bulgarian: Experiments in Supervised and Unsupervised Settings

Valentin Zhikov [*] [†] Ivelina Nikolova* Laura Tolosi*

Yavor Ivanov [‡] [§] Georgi Georgiev*

ABSTRACT

In this work we address the task of automatic keyword and keyphrase extraction from unstructured text, and suit it to the need of a popular Bulgarian media for induction of 'themes'. Themes are defined as text snippets that summarize the essence of an article, facilitating the itemization and retrieval of collections of textual resources and document clustering. We evaluate the performance of several generic methods for keyword and keyphrase extraction on a corpus of articles in Bulgarian, as to the best of our knowledge no such study has been conducted in the past. The methods that we discuss rely on widely accepted information retrieval and machine learning techniques and are language-independent. We also consider the effect of a stemmer component on the keyphrase extraction accuracy. The satisfactory performance of our models in spite of the limited linguistic knowledge incorporated in them recommends our models as a baseline for keyword and keyphrase extraction for Bulgarian language.

1. INTRODUCTION

Themes represent a brief summary and capture the essence of a text document. They are extremely useful for automated and efficient categorization of documents, guided querying, document skimming by visually emphasizing important phrases and offer a powerful basis for measuring document similarity [8, 12, 21]. The popular Bulgarian media resource Svejo.net uses themes for describing documents, for browsing the document collection and as a basis for document clustering.

[*]Ontotext AD., Polygraphia Office Center fl.4, 47A Tsarigradsko Shosse, Sofia 1504, Bulgaria

[†]{valentin.zhikov,ivelina.nikolova,laura.tolosi, georgiev}@ontotext.com

[‡]Xenium Ltd., Polygraphia Office Center fl.4, 47A Tsarigradsko Shosse, Sofia 1504, Bulgaria

[§]yavor@xenium.bg

Svejo.net is a very popular media website, reaching over 20% of auditorium in the segment of Bulgarian news web sites. Every day the users of Svejo.net add over 1 500 news and articles, 3 000 comments and vote over 15 000 times. The content at Svejo.net is managed entirely by the users, the site has no journalists nor moderation at board, therefore relying entirely on the social element. The site allows users to add links of interest with focus on news articles, or multimedia (videos, pictures, etc.). In the case of textual content, users must manually provide a brief description, to categorize and identify the themes of the document. Although the process is partially automated (e.g., a brief description extracted from the article itself is suggested to the user), categorization and theme association are still manual.

In the general case in which theme selection is required for text document collections (for example scientific articles), the possible themes can either be preselected keywords, or unconstrained short text. The themes at Svejo.net are acquired from a meta keyword tag, whenever it exists in the original content, or are assigned by the support team of the website. However, often keywords in the meta tag are generated by tokenization of the article title, which is not very accurate. At the same time, it is time-consuming for the support team to handle all submissions without themes. Consequently, automatic extraction of themes is of great interest to Svejo.net. This task is also known in the scientific literature as *keyword* and *keyphrase extraction*.

This paper discusses automatic extraction of keywords and keyphrases for the Bulgarian language and its application to the documents from Svejo.net.

2. RELATED WORK

There are two general approaches for the extraction or assignment of keywords and keyphrases from unstructured text. The first approach is unsupervised and it is based on the the assumption that keywords appear frequently in a document, but occur less often in the entire document collection. To this end, the popular TFIDF weighting scheme is used. Numerous papers show that TFIDF is very effective for some particular domains [5, 10, 9]. In order to get reliable TFIDF scores, the corpus of documents must be relatively large. [13] propose a competitive method which uses a co-occurrence distribution and a clustering strategy for extracting keywords, which does not rely on a large corpus. Other authors make use of additional knowledge resources

from the web – an idea exploited in this manuscript as well. [17, 11] estimate a point-wise mutual information score in order to select keywords. Graph-based methods similar to Google's PageRank algorithm [1] have also been proposed. [20] adopted a reinforcement learning technique for simultaneous keyword extraction and text summarization, based on the assumption that important sentences usually contain keywords. A related task named *keyword assignment* allows keywords to be assigned only from a predefined dictionary [3]. In this work, we do not make use of a predefined dictionary because we desire flexibility and fast adaptation to new topics, which emerge rapidly at Svejo.net.

Keyword extraction can be also formulated as a supervised classification task and addressed by machine learning techniques [5, 18, 9, 17, 19]. The learning algorithm classifies candidate words and phrases found in a document into positive (keywords) and negative (non-keywords) based on a set of features. Useful features include TFIDF and its variations, position of the keyphrase from beginning of the document, parts of speech, stems, lemmata, relative phrase length, etc. [17].

3. METHODS

We select keywords and keyphrases from a set of candidates comprising n-grams of a predefined set of sizes (in our experiments, unigrams and bigrams). During the preprocessing phase, the documents undergo tokenization, stop-words are removed, tokens are converted to lowercase, and tokens that contain undesirable characters (e.g. numerals and punctuation) are eliminated.

We compare two different approaches for keyphrase extraction in Bulgarian - an unsupervised approach and a supervised approach.

The unsupervised approach is based on the TFIDF heuristics. The TFIDF of a candidate keyword is computed using the traditional formula. The TFIDF score of keyphrase candidates is computed in two ways: *i*) using the traditional TFIDF formula, considering the keyphrase as one token (method called here **mix**) and *ii*) on the basis of the TFIDF scores of its constituents (method **mean**). Specifically, if a phrase is composed of two words, we compute the TFIDF of the two constituent words and the entire bigram, and then average over the three values to get a single score for the phrase. Furthermore, we allow the filtering of constituents whose individual score is below a certain threshold value when calculating the score of the entire phrase.

The candidates are ranked by their TFIDF score. In order to select a threshold for the top ranking candidates, we compute the nearest integer greater than the mean keyphrase count in our evaluation dataset. For the **mix** algorithm, it is necessary that we normalize the TFIDF values obtained within a document to values between 0 and 1.

The second approach is a supervised classification method that predicts keywords from the set of candidates, based on a set of manually labeled training examples. The method is inspired by the KEA algorithm [22], which uses two basic features : the TFIDF score of each of the candidates (denoted by **TFIDF**) and the positional offset (denoted by **pos**), computed as the count of tokens preceding the first occurrence of the candidate phrase in the text. As in the original method, we discretized these features using a supervised method [4], and compared the results with those obtained when with an unsupervised discretization strategy that groups them into equally-sized bins. We added other features to the set proposed in the original article [22], which lead to an improved performance. Specifically, we added the candidate length in tokens (denoted by **len**) and a boolean attribute that indicates whether a token is included in the title of the article or not. Finally, we considered various conjunctions between the features described so far.

For classification we used two of the algorithms implemented in Edlin [6] - multinomial naive Bayes (**MNB**), perceptron (**PER**) [16] and MIRA (**MIRA**) [2]. As in [22], we filter out candidate keywords and phrases that occur in the text of the document analysis only once.

4. EXPERIMENTAL SETTINGS

4.1 Dataset

Although our system can process corpora in multiple languages, our evaluation is focused on Bulgarian text, since our application is targeted for Svejo.net. Our gold-standard dataset contains mostly news documents and analyses, with an accent on political topics. In order to ensure a good quality of annotations, we selected only documents with keywords added by the Svejo.net support team or by the authors of the documents. The final dataset comprises 1 798 articles, divided into training (70%), development (10%) and test (20%) splits, drawn randomly from the entire collection.

In addition to the gold-standard dataset, we index a bulky collection of articles obtained from Svejo.net without controlling the tags, for a more realistic analysis.

Prior to running our methods, we applied some preprocessing (lowercase conversion, numeric tokes removal, stemming, etc.) to the gold-standard keywords and keyphrases in order to ensure compatibility with our set of candidates, which were presented in the Methods section.

4.2 Implementation

The TFIDF scores and some of the preprocessing steps are implemented using the Lucene[1] framework. All machine learning algorithms and the experiments with supervised discretization are implemented in Edlin[2] - a machine learning framework for linear models [6]. The stemmer[3] is used as described in [15]. The system is exposed to Svejo.net as Ontotext's KIM Enterprise[4] services.

5. RESULTS AND DISCUSSION

Experimental results are presented in Tables 1 and 2. The evaluation metrics that we use are precision, recall and F1-score, computed for the target class (true keywords and keyphrases). A candidate is a true positive only if it is an exact match of an entity found in the gold standard set. In the

[1] lucene.apache.org/core/
[2] edlin.sourceforge.net/
[3] lml.bas.bg/ nakov/bulstem/index.html
[4] www.ontotext.com/kim

Table 1: **Experiments in the unsupervised setting.**

Method	n	Filter	Limit	F1	P %	R %
1. mix	1, 2	-	7	**11.95**	8.28	21.44
2. mix	1, 2	-	5	**13.78**	10.61	19.66
3. mix	1	0%	5	12.76	09.82	18.21
4. mean	1, 2	0%	5	11.52	8.87	16.44
5. mean	1, 2	25%	7	11.54	8.00	20.71
6. mean	1, 2	50%	4	12.76	9.82	18.21
7. mean	1, 2	75%	5	13.63	10.49	19.46
8. mix	1, 2	-	5	**13.79**	10.61	19.67

The upper and lower sections of the table contain the results with words and stems, respectively.

Table 2: **Experiments in supervised setting.**

Features	Algorithm	F1 %	P %	R %
1. TFIDF+pos	MNB	9.32	47.62	5.16
2. TFIDF+pos	MNB	10.31	32.31	6.14
3. +len	MNB	25.6	26.90	24.42
4. + TFIDF&pos	MNB	29.73	22.81	42.70
5. + len&pos	MNB	**30.02**	22.15	46.58
6. TFIDF+pos	PER	11.58	36.68	6.88
7. TFIDF+pos	MIRA	10.37	27.35	6.36
8. TFIDF+pos	MNB	10.26	36.65	5.97
9. +len	MNB	**27.35**	19.40	46.33
10. TFIDF+pos	PER	11.58	36.68	6.88
11. TFIDF+pos	MIRA	15.86	29.07	10.90
12. All conjunctions	MIRA	20.69	44.92	13.44

The first section represents the results obtained with words, and the second with stems. "+"denotes the addition of a feature to the set from the previous line;

unsupervised setting, we only report the scores yielded with an optimal setting of the parameter that limits the count of returned results.

The performance of unsupervised methods is not influenced significantly by stemming (Table 1). This is probably due to the numerous constraints on the candidate set (ignoring low-frequency candidates, filtering longer candidates containing a stop word, etc.). The **mix** method outperforms the **mean** method both with stemming (line 8, F1=13.79) and without stemming (line 1, F1=13.78%). Among the variants of **mean**, the best score is achieved if the 75% least important tokens are filtered out (line 7, F1=13.63%).

The models using the basic features of [22], namely **TFIDF +pos**, did not outperform the best unsupervised methods, regardless of the use of stemming. If **len** is added as a feature, the supervised approach outperforms the unsupervised baseline by a considerable margin (compare Table 1, line 2 and Table 2, line 2). A remarkable increase in performance is observed if feature conjunctions are introduced into the model, and the best scores that we have obtained include the conjunctions **TFIDF**&**pos** and **pos**&**len**.

In our experiments, we discretized the continuous-valued **TFIDF** and **pos** features as explained in the Methods section. Supervised discretization did not outperform the unsupervised discretization when using only basic features and the models including feature conjunctions gave slightly better results with unsupervised discretization (an F1 increase of >1%).

In Table 2, lines 1 and 2 show the performance of the models when using the gold-standard training set and the complete collection of articles (see Data section), respectively, for the purpose of evaluation of frequency statistics. Since the performance with the whole collection is superior, we use the complete corpus for obtaining frequency statistics in the following experiments (models 3-12).

As in the unsupervised case, using a stemmer does not influence much the performance of the models, except for the case of the **MIRA** classifier, where we note an improvement of 5% in F1 (Table 2, lines 7 and 11). Adding many conjunctions to the feature set increases further the performance (Table 2, line 12).

As the number of features increases, the precision of these algorithms measured against the positive class improves and eventually reaches levels above 44%, but this growth is accompanied by a steep decline in recall and F1-score. When comparing the performance of the three classifiers, we notice that both **PER** and **MIRA** outperform **MNB** on the basic feature set. Both **MIRA** and **PER** outperform the unsupervised baseline in terms of F1-score, and the best results surpass 20%. We achieve our best result (F1=30.02%) with the **MNB** algorithm using the features **TFIDF**, **pos**, **len** and conjunctions **TFIDF**&**pos** and **pos**&**len**.

6. ERROR ANALYSIS

Error analysis is performed on the development set. Many errors are due to the fact that family names of Bulgarian politicians tend to occur more frequently in the documents than their full names and are therefore preferred by our unsupervised algorithm. On the other hand, the gold standard recommends their full names. For example, the gold standard suggests *Sergei Stanishev* and *Traicho Traikov* as keywords for some documents. Our algorithm returns *Traicho Traikov*, which is a true positive, but also *Traicho*, *Stanishev* and *Traikov*, which are false positives according to our evaluation scheme. This problem is addressed by the addition of **len** feature to our supervised learner.

Another issue is the tendency of authors to add names of politicians and political organizations, even when they are not explicitly mentioned in the article. For instance, a report about the new *minister of health* is tagged by the actual name of the minister (gold standard), whereas we return *minister of health*, *new*, *health*, *minister*, etc. The Svejo.net estimates that the accuracy can be increased by up to 30% if we include part-of-speech analysis and remove verbs from our selection.

When our model is built without stemming, many true positives are not considered during the evaluation, because the keywords returned by the algorithm contain for example the article and the gold ones do not. For instance, the gold standard keyword *medal* is returned by the algorithm *the medal*. Our algorithms also return both the plural and singular

forms of a keyword, which is not the case of the gold standard. This suggests that working with lemmata (canonical word forms) would improve the performance of our models.

7. CONCLUSIONS AND FUTURE WORK

In this work we addressed the task of automatic keyword and keyphrase extraction and its application to the popular Bulgarian web resource Svejo.net. To the best of our knowledge, this is the first study on keyword and keyphrase extraction for Bulgarian, a resource-poor and under-explored language.

We presented two simple approaches which do not rely on costly tools for linguistic analysis. We explored different candidate selection strategies and evaluated the effect of several types of features and their conjunctions on our models. We also assessed the accuracy gain achieved by the introduction of a stemmer component.

In the future, we plan to increase the involvement of linguistic knowledge. Many studies suggest vast improvements in performance with addition of part-of-speech tags. Recent work [7] suggests that adding morphological features can improve supervised classification. The error analysis suggests that orthographic features, (e.g. "word comprises only upper case characters") can isolate names and organizations in the sentence, since these phrases coexist with a prominent change in case. Another useful direction of development could be the automated induction of feature conjunctions [14].

8. REFERENCES

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, Apr. 1998.

[2] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:551–585, 2006.

[3] S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, CIKM '98, pages 148–155, New York, NY, USA, 1998. ACM.

[4] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029, 1993.

[5] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and et al. Domain-specific keyphrase extraction. In *PROC. SIXTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pages 668–673. Morgan Kaufmann Publishers, 1999.

[6] K. Ganchev and G. Georgiev. Edlin: an easy to read linear learning framework. In *Recent Advances in Natural Language Processing*, 2009.

[7] G. Georgiev, K. Simov, P. Osenova, V. Zhikov, and P. Nakov. Feature-rich part-of-speech tagging for morphologically complex languages: Application to bulgarian. In *Preceedings of European chapter of the Association for Computational Linguistics (ACL)*, 2012.

[8] C. Gutwin, G. Paynter, I. Witten, C. Nevill-Manning, and E. Frank. Improving browsing in digital libraries with keyphrase indexes, 1998.

[9] Y. HaCohen-Kerner, Z. Gross, and A. Masa. Automatic extraction and learning of keyphrases from scientific articles. In *Proceedings of the 6th international conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 657–669, Berlin, Heidelberg, 2005. Springer-Verlag.

[10] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[11] D. Inkpen and A. Desilets. Extracting semantically-coherent keyphrases from speech. *Canadian Acoustics*, 32:130–131, 2004.

[12] S. Jones. Link as you type: Using key phrases for automated dynamic link generation, 1998.

[13] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:2004, 2004.

[14] A. McCallum. Efficiently inducing features of conditional random fields. *Nineteenth Conference on Uncertainty in Artificial Intelligence UAI03*, pages 403–410, 2003.

[15] P. Nakov. Design and evaluation of inflectional stemmer for bulgarian. In *IN PROCEEDINGS OF WORKSHOP ON BALKAN LANGUAGE RESOURCES AND TOOLS (1ST BALKAN CONFERENCE IN INFORMATICS*, 1998.

[16] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[17] P. Turney and P. D. Turney. Mining the web for lexical knowledge to improve keyphrase extraction: Learning from labeled and unlabeled data, 2002.

[18] P. D. Turney. Learning algorithms for keyphrase extraction. *INFORMATION RETRIEVAL*, 2:303–336, 2000.

[19] P. D. Turney. Coherent keyphrase extraction via web mining. In *Proceedings of the 18th international joint conference on Artificial intelligence*, IJCAI'03, pages 434–439, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[20] X. Wan, J. Yang, and J. Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction, 2007.

[21] I. H. Witten. Browsing around a digital library. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '03, pages 99–99, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[22] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.