UDK: 811.163.2’322.2

ENHANCING SOCIAL NEWS MEDIA IN BULGARIAN WITH NATURAL LANGUAGE PROCESSING

Valentin Zhikov, valentin.zhikov@ontotext.com, Ontotext AD
Ivelina Nikolova, iva@lml.bas.bg, IICT, Bulgarian Academy of Sciences and Ontotext AD
Laura Toloşi, laura.tolosi@ontotext.com, Ontotext AD
Yavor Ivanov, yavor@xenium.bg, Xenium Ltd.
Borislav Popov, borislav.popov@ontotext.com, Ontotext AD
Georgi Georgiev, georgiev@ontotext.com, Ontotext AD

Abstract

In this work we introduce a system based on natural language processing techniques which aim is to enhance social news media in Bulgarian. It solves the task of multi-class, multi-label classification of documents. We apply the algorithms to a collection of media articles from Svejo.net, a popular Bulgarian web resource comprising user-generated content. Our algorithms are one-versus-all classification methods widely used in the computational linguistics community. We describe the algorithms, the features employed and we evaluate the impact of the features on the performance of the models. Thereby, we show that knowledge about the user and user behavior can greatly improve performance. Also, despite the fact that our document collection is generated entirely by social media users, the quality of the classification results is comparable to that of previously reported studies. We address also the task of automatic keyword and keyphrase extraction from unstructured text, and suit it to the needs of Svejo.net for induction of’themes’. Themes are defined as text snippets that summarize the essence of an article. We evaluate the performance of several generic methods for keyword and keyphrase extraction on a corpus of articles in Bulgarian. 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.

Keywords: natural language processing, machine learning, language agnostic approaches, keyword extraction, text classification

1. Introduction

The need of Natural Language Processing (NLP) techniques to enhance social web media is indisputable. Social media web sites have access to vast amounts of textual data that need efficient and automated processing. In this paper we discuss two different aspects of the utilization of NLP to improve the social media service Svejo.net in Bulgarian. One of them is the automatic text classification of news collections in order to speed up the addition of new articles to the service. The other aspect is automatic extraction of themes from the articles which can be further used for article clustering.

Svejo.net is very popular and one of the first Bulgarian social media websites, 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, there are no journalists at board but only a handful of moderators with very limited duties regarding the website contents, therefore Svejo.net reliesentirely on the social element. The site allows users to add links of interest with focus on news articles or multimedia (videos, pictures, etc.). Although the articles linked from Svejo.net do not have language restrictions they are mainly in Bulgarian, with occasional submissions in English, French, German, Russian etc.

The popularity of social media is very much related to how intuitive and easy to use their interface is. In the case of adding textual content to Svejo.net, users must manually provide a brief description, 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. One of the steps to facilitate the addition of new articles to Svejo.net is to provide automatic recognition of the article topic and keyterms and free the users from specifying all of these, and allow them to add an article with a single click. Partially this problem could be solved with the classical NLP technique of automatic text classification. Text classification has been studied for many years and it is still challenging in an open domain setting. We apply machine classifiers to multi-label, multi-class and multi-language text categorization task, tailored to the concrete needs of Svejo.net. We demonstrate how traditional machine learning techniques can be enhanced by feature representative of the individual users and their behavior.

Themes represent a brief summary and capture the essence of a text document. They are useful for automated and efficient categorization of documents, guided querying, document skimming by visually emphasizing important phrases. They offer a powerful basis for measuring document similarity (Gutwin et al. 1999, Jones 1998, Witten 2003). The popular Bulgarian media resource Svejo.net uses themes for describing documents, for browsing the document collection and as a basis for document clustering.

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. Therefore, 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.

The rest of the article is structured as follows: related work is presented in section 2, the experimental datasets are described in section 3, a short system overview is given in section 4, section 5 presents the methods, section 6 contains results and error analysis and conclusion and directions for future work are given in section 7.

2. Related Work

A variety of supervised learning algorithms, including naive Bayes, support vector machines, boosting, rule learning demonstrate reasonable performance for text classification (Lewis 1998, McCallum and Nigam 1998, Sahami 1996, Dumais et al. 1998, Joachims 1998, Schapireand Singer 2000, Cohen and Singer 1999, Slattery and Craven 1998, Yang 1999). It is worth to note, that among all the techniques mentioned above, no single method can prove to significantly and consistently outperform the others across many domains and languages.

Maximum entropy models are often used for text classification (Nigam et al.1999). There are works describing the feature selection for such models,as in (Mikheev 1998)(technical abstracts) for the RAPRA corpus. In (Ratnaparkhi1998), maximum entropy and decision trees models are compared and it is shown that the maximum entropy is superior at classifying some of the classes in the Reuters-21 578 data set.

An interesting problem is assigning more than one label to a document, known as multi-class, multi-label text classification, which is the focus of this work. We add yet another source of complexity to the task, namely multiple languages. In (Luoand NurZincir-heywood 2005) two machine learning algorithms are compared:kNN classifiers and “Latent Semantic Indexing”. The authors find out that the first system performs better on multi-labeled documents, while the second one outperforms on uni-labeled documents. They conclude that performance depends on the applied dataset and the objective of the application.

Recent studies address the same task by the application of multiple classifiers that work in one-versus-all settings (Zelaia et al. 2011).

Perhaps more important than the choice of classification method is the choice of features. Studies have shown that for the task of web page classification, features extracted from the semi-structured HTML are more expressive and more predictive than features traditionally used for pure text classification. Such features include families of HTML tags, the web page URL, HTML meta tags like keywords, neighbor pages, anchors, headings etc. (Qi and Davison 2009).

For the task of extraction or assignment of keywords and keyphrases from unstructured text there are two general approaches. The first approach is unsupervised and it is based on the assumption that keywords appear frequently in a document, but occur less often in the entire document collection. To this end, the popular TF-IDF weighting scheme is used. Numerous papers show that TF-IDF is very effective for some particular domains (Frank et al.1999, Hulth 2003, Ha Cohen-Kerner et al. 2005). In order to get reliable TF-IDF scores, the corpus of documents must be relatively large. In (Matsuo and Ishizuka2004) a competitive method is proposed, 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. In (Turney2002, Inkpen and Desilets2004) the authors estimate a point-wise mutual information score in order to select keywords. Graph-based methods similar to Google’s PageRank algorithm (Brinand Page 1998) have also been proposed. In (Wan et al.2007), a reinforcement learning technique for simultaneous keyword extraction and text summarization is adopted, 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 (Dumaiset al. 1998). 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 also be formulated as a supervised classification task and can be addressed by machine learning techniques (Frank et al. 1999, HaCohen-Kerneret al. 2005, Turney 2000, Turney2002, Turney 2003). 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 TF-IDF and its variations, position of the keyphrase from beginning of the document, parts-of-speech, stems, lemmata, relative phrase length of a phrase, etc. (Turney2002).

3. Datasets

Our corpus for the classification task is collected by Svejo.net over several years and sums up to nearly 400 000 documents in Bulgarian and other languages, including English, French, German and Russian (however, under-represented in comparison to Bulgarian). The data is in XML format and each document contains the following elements: *title*, *summary*, *user_id*, *media_type*, *tags*, *categories*, *created_at* and *updated_at*. The last two elements are date tags. The *summary* of each document is extracted from an online article and contains up to 1 000 characters taken from its beginning. All HTML tags are removed, leaving only free text. The *title* contains the article title, the *tags* are free text consisting of short text snippets relevant for the content of the article and the categories are assigned from predefined lists by the user. Each document may have more than one *tag* and *category* among: *society*, *technologies*, *science*, *business*, *politics*, *sport*, *art*, *health*, *fun*, *lifestyle*, *shopping*. More than 9% of the articles have multiple categories assigned. The distribution of the documents among the categories is included in Table 1. The most popular categories are *society* and *fun*. *Lifestyle* follows closely, with 18%. About 10% of the documents are categorized as *technologies*, *sport* and *health*. The least popular category in Svejo.net is *shopping* with only 930 articles, corresponding to less than a quarter percent of the whole document collection.

For the task of theme extraction we employ a different document set prepared for this special purpose. Although our system can process corpora in multiple languages, our evaluation is focused on Bulgarian text, since our application is targeted for Svejo.net. The 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.

Table 1. Distribution of articles by categories

category	# articles	% corpus	avg # words
society	88425	22,53	38,54
fun	82839	21,11	30,44
lifestyle	71151	18,13	41,54
technologies	42399	10,80	22,25
sport	37092	9,45	36,87
health	36180	9,22	39,59
business	24759	6,31	38,11
politics	21692	5,53	44,17
art	17658	4,50	36,86
science	12539	3,19	42,53
shopping	930	0,24	64,56

In addition to the gold-standard dataset, we index a bulky collection of articles obtained from Svejo.net without considering the tags, in order to obtain more representative statistics of the word frequencies.

Prior to running the experiments, we applied some preprocessing (lowercase conversion, numeric tokens removal, stemming, etc.) to the gold-standard keywords and keyphrases in order to ensure compatibility with our set of candidates.

4. System Description

The aim of this study is to support a real world system which will facilitate the content sharing in a social media website. The process of development consists of several iterations of training, test and validation, which are costly in terms of associated manual labour and computational resources, and are executed in collaboration with the Svejo.net support team. Each update iteration is handled by them through a specialized interface, exposed via an array of web services.

In the development phasemodel and dataset updates occur often. The development cycle includes analysis of the classification errors against unseen documents, revising the gold-standard datasets, acquiring additional annotated articles and retraining of the models.

Under these considerations, we have designed a batch-learning algorithmic solution, which supports iterative updates and makes the system easy-to-use by non-experts. The system incorporates web methods for labeling of unseen documents, model retraining and system status retrieval. The labeling method accepts article submissions in XML format, and generates a machine-processable XML response containing categorical predictions. Labeled document collections for model development are uploaded in advance to a repository folder accessible by the server via generic file transfer protocols. Collections can be provided as either collections of XML documents residing in a subfolder of the repository, or .zip archives containing a batch of documents of an arbitrary size. The state and contents of the document repository, the count of active labeling models, along with the count of available permits for parallel access to the system are reported upon calls to a specialized system status method.

System retraining can be triggered by calling a service method by specifying the path to a particular dataset.

5. Method

5.1 Automatic Text Classification

Considering that the most important factor influencing the model performance is the set of features used for training, one of the valuable contributions of this work is the feature engineering. We defined a number of features, some depending on the textual content (bag of words), others on the meta-data supplied along with the document: *media type*, *user identifier* and *tags* provided by the user. We also experimented with conjunctions between the *tag* and *user identifier* and calculated character n-grams over the tags. We evaluated the contribution of each feature type to the system performance. The features are language agnostic and we do not make use of any linguistic knowledge or resources for Bulgarian (the main language representative in our data set). We designed our algorithm as hard classification, e.g., a document is classified using a one versus all approach. We train a binary classifier for each category and collect all positive classifications to allow assignment of multiple labels per document. The classification is performed with Edlin[1] with DSL and software layer for feature engineering (Ganchev and Georgiev 2009). The system is exposed to Svejo.net as Ontotext's KIM Enterprise[2] services.

The classification methods which are used are naive Bayes, maximum entropy, perceptron (Crammer et al.2006) and MIRA (Rosenblatt1958). We use 70% of the entire collection for training, reserve 15% for a development set, and keep another 15% for assessment of the classifier output. For naive Bayes classifiers, we

[1] http://www.edlin.sourceforge.net

[2] http://www.ontotext.com/kim

optimize the hyper-parameter that controls the extent of smoothing (we use Laplacian smoothing) against the development set. Since our goal is to produce a real world system there are features enabling real time training by the end users. For this system the smoothing parameter is set in advance, stratified training and test splits are dynamically built for training of each classifier, and the ratio between training and test data is set to 9:1. Training and evaluation takes place via an automated routine that extracts all classes present in the provided document collection, prepares randomized and stratified training and test splits for each class, analyses and stores the results, and saves the generated models.

5.2 Theme Extraction/Assignment

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 TF-IDF heuristics. The TF-IDF of a candidate keyword is computed using the traditional formula. The TF-IDF score of keyphrase candidates is computed in two ways: (*i*) using the traditional TF-IDF formula, considering the keyphrase as one token (method called here **mix**) and (*ii*) on the basis of the TF-IDF scores of its constituents (method **mean**). Specifically, if a phrase is composed of two words, we compute the TF-IDF 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 TF-IDF (or average TF-IDF) 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 TF-IDF 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 (Witten1999), which uses two basic features: the TF-IDF 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 (Fayyad and Irani1993), 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 (Wittenet al. 1999), 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 (Ganchev and Georgiev 2009) - multinomial naive Bayes (**MNB**), perceptron (**PER**) (Rosenblatt 58) and MIRA (**MIRA**) (Crammer et al.2006). As in (Witten et al. 1999), we filter out candidate keywords and phrases that occur in the text of the document analysis only once.

The TF-IDF scores and some of the preprocessing steps are implemented using the Lucene[3] framework. All machine learning algorithms and the experiments with supervised discretization are implemented in Edlin. The stemmer Bulstem[4] which is described in (Nakov1998) is also used. The system is exposed to Svejo.net as Ontotext's KIM Enterprise services.

6. Results and Discussion

6.1 Automatic Text Classification

We observed that the naive Bayes strongly outperforms the other classification methods. Below we show only experiments using the naive Bayes classifier.

Table 2. summarizes the performance of the classifiers for each of the target categories. The reported precision, recall and the macro F1 scores represent the mean score values obtained over each category for 10 independent experiments.

Our baseline comprises a bag-of-words (BOW) over the title and body, and results in macro F1-scores smaller than 60% across all categories. The poor performance is probably due to the limited amount of text contained in each of the documents. The inclusion of tags leads to a noticeable improvement of performance, F1-score increasing up to 67%. The feature *media_type* does not improve the F-score, neither when used alone, nor in a conjunction with other meta-data attributes. In our models, the most informative feature among the three meta-data features is the *user_id*, particularly in conjunction with the document tags. We explain this by either a tendency of certain users to assign certain specific tags, or by the interest of each user towards a certain article category. The highest accuracy in our experiments is achieved by the system that uses a feature set comprising bag-of-words over the textual contents, all meta-data features and the conjunction *user_id&tags*. On average the system scores nearly 5% higher than the baseline model. Extending the feature set further by the addition of character trigrams over the tags assigned to a particular document leads to a reduction in the average score. The reason for including n-grams over *tags* is to address issues like the occurrence of *tags* in plural and singular, with or without article, as *tags* are free text written out by the user upon addition of the new resource to Svejo.net. Although we observed some improvement when using n-gram and particularly trigrams over tags in comparison to the cases in which tags are included as words, this feature does not seem to lead to an improvement in the presence of other, more informative attributes, like *user_id*.

As expected, *shopping* is the lowest-scoring category; however, when using this model, we witness nearly 12% combined improvement on precision and recall in comparison to the second best model, and overall score increasing to nearly 42% in this setting. The model for the *sport* category produces the best F1 score among all models in/from the set (92%).

Error Analysis

The main source of errors is the limited amount of text in the documents to be classified, affecting all categories regardless of their abundance in the corpus.

Table 2. Text classification results

	Precision	**Recall**	**F-measure**
AVG baseline (BOW)	0,53	0,61	0,57
+ tags			
shopping	0,25	0,35	0,29
sport	0,90	0,87	0,89
art	0,61	0,65	0,63
business	0,52	0,68	0,59
politics	0,50	0,75	0,60
society	0,69	0,81	0,75
fun	0,81	0,80	0,80
science	0,54	0,47	0,50
lifestyle	0,70	0,77	0,73
technologies	0,80	0,85	0,82
health	0,78	0,84	0,81
other	0,98	0,96	0,97
AVG all	**0,64**	**0,71**	**0,67**

[3] http://lucene.apache.org/core/

[4] http://lml.bas.bg/~nakov/bulstem/index.html

+ media_type			
shopping	0,23	0,32	0,27
sport	0,90	0,87	0,89
art	0,61	0,66	0,64
business	0,51	0,68	0,58
politics	0,50	0,74	0,60
society	0,69	0,81	0,75
fun	0,80	0,80	0,80
science	0,53	0,48	0,51
lifestyle	0,69	0,77	0,73
technologies	0,80	0,85	0,82
health	0,78	0,83	0,81
other	0,98	0,96	0,97
AVG all	**0,64**	**0,71**	**0,67**
+user_id + user_ id&tags			
shopping	0,37	0,45	0,41
sport	0,93	0,91	0,92
art	0,72	0,74	0,73
business	0,63	0,75	0,69
politics	0,60	0,81	0,69
society	0,77	0,87	0,82
fun	0,85	0,86	0,86
science	0,66	0,57	0,61
lifestyle	0,79	0,83	0,81
technologies	0,85	0,88	0,87
health	0,83	0,87	0,85
other	0,98	0,97	0,98
AVG all	**0,73**	**0,78**	**0,75**
+ 3-grams over tags			
shopping	0,21	0,47	0,29
sport	0,91	0,88	0,90
art	0,60	0,73	0,66
business	0,53	0,71	0,61
politics	0,49	0,80	0,61
society	0,69	0,85	0,76
fun	0,83	0,82	0,82
science	0,52	0,56	0,54
lifestyle	0,72	0,80	0,76
technologies	0,81	0,85	0,83
health	0,78	0,86	0,82
other	0,98	0,96	0,97
AVG all	**0,64**	**0,76**	**0,69**

We also note that the categories are difficult to differentiate by their lexical features only. Many categories, including the largest ones (*society* and *fun*) contain rather general lexicons and little specific terminology. This leads to poor performance if BOW is used for prediction. The terminology is more explicit for only two categories: *sport* and *technologies*.

A document from the *business* category that speaks about the Bulgarian finance minister is classified by our system as *politics*. The reason may be that the Bulgarian finance minister is mentioned three times in the 1 000 characters textual limit, which is provided to the system. Another similar example is the following: a story about a student who made a trailer for his favorite football team is classified as *sport*, whereas the true class is *lifestyle*. Another document from the class *lifestyle*, talking about a popular Bulgarian football team relaxing in a restaurant in Burgas after winning the Bulgarian football super cup is classified as *sport*. Another source of errors is a set of documents containing advertisement, which we classify as *shopping*, but which belong to the *health* group. In our view, the above mentioned mistakes of our system are in fact acceptable suggestions for categorization. They are a consequence of the overlap between the topics in different categories and we believe that the optimal assignment would admit both the predicted and the gold standard categories.

A clear mistake, which cannot be explained by overlapping terminology, refers for instance to an article reporting on Lichtenstein joining the Schengen space, which is incorrectly classified as *shopping*. We believe that this is due to the extensive faith of the algorithm in some meta-features like the *user_id* and other tags. This occurs mostly for the shopping category, which is under-represented in the corpus.

Another source of errors refers to documents written in foreign languages which are underrepresented in the corpus.

6.2 Theme Extraction/Assignment

Experimental results from the keyword extraction/assignment are presented in Tables 3 and 4. 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 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 3). 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 (Wittenet al. 1999), 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 3, line 2 and Table 4, 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%).

Table 3. Experiments in the unsupervised setting

	Method	n	F	L	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 separated by a double linecontain the results with words and stems, respectively. **Legend**: L (limit), F (filter), n (n-grams)

Table 4. Experiments in supervised setting

	Features	Algo	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	25.60	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 conj	MIRA	20.69	44.92	13.44

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

In Table 4 lines 1 and 2 show the performance of the models when using the gold-standard training set and the complete collection of articles correspondingly (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 4, lines 7 and 11). Adding many conjunctions to the feature set increases further the performance (Table 4, 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**.

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. Conclusion

We have presented an overview of a system based on NLP techniques that facilitates the sharing of content over a social web space.

The first task is attached as a multi-class, multi-label, multi-language classification. The implemented modules allow the automation of a task that demands a significant amount of manual effort, and provide capabilities for improvement of the accuracy of modeling that does not require an insight into the details concerning the functioning of the system. We have described the algorithms and features employed, evaluated the impact of the features that participate in the modeling process and show that knowledge about the user can greatly improve performance.

We addressed also 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 (Georgiev et al.2012) 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 conjuctions (McCallum2003).

References

Brin, Sergey and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30:107-117, ISSN 0169-7552, DOI: 10.1016/S0169-7552(98)00110-X.

Cohen, William W. and Yoram Singer. 1999. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems* 17:141-173 ISSN 1046-8188, DOI: 10.1145/306686.306688.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research* 7:551-585.

Dumais, Susan, John Platt, David Heckerman and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 148-155, ISBN 1-58113-061-9, Bethesda, Maryland, United States, DOI: 10.1145/288627.288651.

Fayyad, Usama M. and Keki B. Irani. 1993. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, 1022-1029.

Frank, Eibe, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, et al. 1999. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 668-673.

Ganchev, Kuzman and Georgi Georgiev. 2009. Edlin: an easy to read linear learning framework. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 94-98.

Georgiev, Georgi, Kiril Simov, Petya Osenova, Valentin Zhikov and Nakov, Preslav. 2012. Feature-Rich Part-of-speech Tagging for Morphologically Complex Languages: Application to Bulgarian. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France*, 492-502.

Gutwin, Carl, Gordon Paynter, Ian Witten, Craig Nevill-Manning and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems* 27:81-104.

HaCohen-Kerner, Yaakov, Zuriel Gross and Masa, Asaf. 2005. Automatic extraction and learning of keyphrases from scientific articles. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico*, 657-669, ISBN 3-540-24523-5, ISSN 0302-9743, DOI: 10.1007/978-3-540-30586-6_74.

Hulth, Anette. 2003. Improved automatic

keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 216-223, DOI: 10.3115/1119355.1119383.

Inkpen, Diana and Alain Desilets. 2004. Extracting Semantically-Coherent Keyphrases from Speech, *Canadian Acoustics* 32:130-131.

Joachims, Thorsten. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning*, 137-142, ISBN:3-540-64417-2,

Jones, Steve. 1998. Link as you type: using key phrases for automated dynamic link generation (Working paper 98/16). Hamilton, New Zealand: University of Waikato, Department of Computer Science.

Lewis, David D.. 1998. Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In *Proceedings of the 10th European Conference on Machine Learning*, 4-15.

Luo, Xiao and A. Nur Zincir-heywood. 2005. Evaluation of Two Systems on Multi-class Multi-label Document Classification. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*, 161-169, DOI:10.1007/11425274_17.

Matsuo, Y. and M. Ishizuka. 2004. Keyword Extraction From A Single Document Using Word Co-Occurrence Statistical Information, *International Journal on Artificial Intelligence Tools* 13:157-170

McCallum, Andrew and Kamal Nigam. 1998. A comparison of event models for Naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization. Technical report WS-98-05*.

McCallum, Andrew. 2003. Efficiently Inducing Features of Conditional Random Fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence*, 403-410, ISBN:0-127-05664-5

Mikheev, Andrei. 1998. Feature lattices for maximum entropy modeling. In *Proceedings of the 17th International Conference on Computational linguistics, Vol.2, Montreal, Quebec, Canada*, 848-854, DOI: 10.3115/980432.980709

Nakov, Preslav. 1998. Design and Evaluation of Inflectional Stemmer for Bulgarian. In *Proceedings of Workshop on Balkan Language Resources and Tools (1st Balkan Conference in Informatics)*

Nigam, Kamal, John Lafferty and Andrew Mccallum. 1999. Using Maximum Entropy for Text Classification. In *IJCAI Workshop on Machine Learning for Information Filtering*, 61-67.

Qi, Xiaoguang and Brian D.Davison. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys* 41(2)1-31, ISSN 0360-0300, DOI: 10.1145/1459352.1459357

Ratnaparkhi, Adwait. 1998. Maximum Entropy Models for Natural Language Ambiguity Resolution, PhD dissertation. University of Pennsylvania Philadelphia, PA, USA, ISBN:0-591-94112-0

Rosenblatt, Frank. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Psychological Review* 65:386-408.

Sahami, Mehran. 1996. Learning Limited Dependence Bayesian Classifiers. In *KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 335-338.

Schapire, Robert E. and Yoram Singer. 2000. BoosTexter: A Boosting-based System for Text Categorization, *Machine Learning* 39:135-168

Slattery, Sean and Mark Craven. 1998. Combining Statistical and Relational Methods for Learning in Hypertext Domains, In *Proceedings of the 8th International Conference on Inductive Logic Programming*, 38-52.

Turney, Peter D. 2000. Learning Algorithms for Keyphrase Extraction, *Information Retrieval* 2:303-336.

Turney, Peter D. 2002. Mining the web for lexical knowledge to improve key phrase extraction: Learning from labeled and unlabeled data. In *ERB-1096NRC#44947, National Research Council, Institute for Information Technology*

Turney, Peter D. 2003. Coherent keyphrase extraction via web mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, 434-439.

Wan, Xiaojun, Jianwu Yang and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of theAssociation of Computational Linguistics, Prague, Czech Republic, June,* 552–559.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, and Carl Gutwin and Craig G. 1999. Nevill-Manning, KEA: practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on Digital libraries, Berkeley, California, United States*, 254-255, ISBN 1-58113-145-3, DOI: 10.1145/313238.313437

Witten, Ian H.. 2003. Browsing around a digital library. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, Baltimore, Maryland*, 90-99, ISBN 0-89871-538-5.

Yang, Yiming. 1999. An evaluation of statistical approaches to text categorization, *Journal of Information Retrieval* 1:67-88

Zelaia A. and I. Alegria and O. Arregi and B. Sierra. 2011. A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension, *Applied Soft Computing* 11: 4981-4990