Multi- Class, Label and Language Document Classification: System and Features

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

Yavor Ivanov [‡] Georgi Georgiev*

ABSTRACT

In this work we introduce a system which solves the task of multi-class, multi-label classification of documents, to which we add another source of complexity: multiple languages, mainly Bulgarian, with occasional submissions in English, French, German, Russian etc. 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 results is comparable to that of previously reported studies.

1. INTRODUCTION

The need of natural language processing techniques in the social web media is indisputable. Social media web sites have access to vast amounts of textual data that needs efficient and automated processing. In this paper we discuss the application of text classification to collections from the Bulgarian social media service Svejo.net, and demonstrate how traditional machine learning techniques can be enhanced by features referring to users and user behavior.

*Ontotext AD., Polygraphia Office Center fl.4, 47A Tsarigradsko Shosse, Sofia 1504, Bulgaria {valentin.zhikov,laura.tolosi,georgiev}@ontotext.com

†Institute of Information and Communication Technologies, 25A, Acad. G. Bonchev Str., 1113 Sofia, Bulgaria iva@lml.bas.bg

‡Xenium Ltd., Polygraphia Office Center fl.4, 47A Tsarigradsko Shosse, Sofia 1504, Bulgaria yavor@xenium.bg

Svejo.net is a very popular and one of the first Bulgarian 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, 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.). 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.

In this work we apply machine classifiers to the multi-label, multi-class and multi-language text categorization task, tailored to the concrete needs of Svejo.net.

2. RELATED WORK

A variety of supervised learning algorithms, including naive Bayes, support vector machines, boosting, rule learning, etc., demonstrated reasonable performance for text classification [6, 8, 14, 3, 5, 15, 1, 16, 17]. 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 have been used often for text classification [10]. Feature selection for maximum entropy models have also been described, for example for the RAPRA corpus (technical abstracts) [9]. In [12], 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-21578 data set.

An interesting problem is assigning more than one label to a document, known as the a 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 [7] two machine learning algorithms are applied, namely kNN classifiers and "Latent Semantic Indexing", both aware of the co-occurrences in documents of multiple categories. Recent studies address this task by the application of multiple classifiers that work in one-versus-all settings [18].

Perhaps more important than the choice of classification method for performance 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

Table 1: Distribution of the documents among categories and their percentage of the whole collection

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

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. [11].

3. DATASETS

Our corpus 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 are in .XML format and each document contains the following elements: *title*, *summary*, *user_id*, *media_type*, *tags*, *categories*, *created_at* and *updated_at*.

The *summary* of each document is extracted from the 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 a 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.

4. METHOD

In the development phase, model and dataset updates are expected to 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. Each update iteration is handled by the Svejo.net support team, through a specialized interface exposed via an array of web services, and it is time-consuming.

Under these considerations, we have designed a batch-learning algorithmic solution, which supports iterative updates and makes our 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 category 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

Table 2: Results of distinct experiments

	Precision	Recall	F-measure
AVG baseline (BOW)	0.53	0.61	0.57
+ tags			
shopping	0.25	0.35	0.29
sport	0.9	0.87	0.89
art	0.61	0.65	0.63
business	0.52	0.68	0.59
politics	0.5	0.75	0.6
society	0.69	0.81	0.75
fun	0.81	0.8	0.8
science	0.54	0.47	0.5
lifestyle	0.7	0.77	0.73
technologies	0.8	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
+ *media_type*			
shopping	0.23	0.32	0.27
sport	0.9	0.87	0.89
art	0.61	0.66	0.64
business	0.51	0.68	0.58
politics	0.5	0.74	0.6
society	0.69	0.81	0.75
fun	0.8	0.8	0.8
science	0.53	0.48	0.51
lifestyle	0.69	0.77	0.73
technologies	0.8	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* + *iser_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	0.6	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	0.9
art	0.6	0.73	0.66
business	0.53	0.71	0.61
politics	0.49	0.8	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	0.8	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

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 another service method and specifying the path to a particular dataset. As training is a resource-intensive operation, calls to the method for re-training leads to a temporary suspension of the labeling functionality. At the beginning of each training phase, all classifiers are shut down, and the latest models are backed up. At the end of each training iteration, evaluation results are stored and classifiers are re-instantiated from the newly built models.

Considering that the most important factor influencing model performance is the set of features used for training, one of the most important 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 calculate 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 [4]. The system is exposed to Svejo.net as Ontotext's KIM Enterprise[2] services.

The classification methods that we used are naive Bayes, maximum entropy, perceptron [2] and MIRA [13]. 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 optimize the hyper-parameter that controls the extent of smoothing (we use Laplacian smoothing) against the development set. In production setting, 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 changes 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. RESULTS AND DISCUSSION

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

[1]edlin.sourceforge.net/

[2]www.ontotext.com/kim

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, the F1-score increasing 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, etc., 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 to the set (92%).

6. 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.

We also note that the categories are difficult to differentiate by their lexical features only. Many categories, including the largest (*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 is 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 that 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 Arabic that are classified as *sport* documents. This can be a mistake of the person that added the Arabic content (marked it manually as *sport*) or a mistake of the system, which relies too much on the *user_id*, in the conditions in which the respective the person predominantly shares sport news.

7. CONCLUSION AND FUTURE WORK

We have presented an overview of a system for multi-class, multi-label, multi-language classification that facilitates the sharing of content over a social web space. 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 believe that the system will benefit from language identification and application of language-specific analysis, such as lemmatization and part-of-speech tagging. Such resources will facilitate the recognition of named entities in the documents, which is expected to improve performance. However, introducing such resources will decrease the speed of the system.

Another direction of improvement is providing access to the original HTML pages, in order to increase the amount of available text, and allow the incorporation of structural meta-data from the original HTML page into the models – two steps that will hopefully lead to a better overall performance by providing a richer representation of the content.

Soft classification, i.e. a document instance being assigned probabilities of belonging to various classes, could also improve model performance.

8. REFERENCES

[1] W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Trans. Inf. Syst.*, 17(2):141–173, Apr. 1999.

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

[5] T. Joachims. Text categorization with support vector machines: Learning with many relevant features, 1998.

[6] D. D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998.

[7] X. Luo and A. N. Zincir-heywood. Evaluation of two systems on multi-class multi-label document classification', paper presented to. In *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems*, 2005.

[8] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification, 1998.

[9] A. Mikheev. Feature lattices for maximum entropy modelling. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 848–854, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[10] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification, 1999.

[11] X. Qi and B. D. Davison. Web page classification: Features and algorithms. *ACM Comput. Surv.*, 41(2):12:1–12:31, Feb. 2009.

[12] A. Ratnaparkhi. Maximum entropy models for natural language ambiguity resolution. Technical report, 1998.

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

[14] M. Sahami. Learning limited dependence bayesian classifiers. In *In KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 335–338. AAAI Press, 1996.

[15] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning*, pages 135–168, 2000.

[16] S. Slattery and M. Craven. Combining statistical and relational methods for learning in hypertext domains. In *In Proceedings of the 8th international Conference on Inductive Logic Programming*, pages 38–52. Springer Verlag, 1998.

[17] Y. Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1999.

[18] A. Zelaia, I. Alegria, O. Arregi, and B. Sierra. A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. *Applied Soft Computing*, 11(8):4981–4990, 2011.