Classification of Short Legal Lithuanian Texts

Vytautas Mickevičius[1,2] **Tomas Krilavičius**[1,2]
Vaidas Morkevičius[3]
[1]Vytautas Magnus University, [2]Baltic Institute of Advanced Technologies,
[3]Kaunas University of Technology, Institute of Public Policy and Administration
vytautas.mickevicius@bpti.lt, t.krilavicius@bpti.lt,
vaidas.morkevicius@ktu.lt

Abstract

Statistical analysis of parliamentary roll call votes is an important topic in political science because it reveals ideological positions of members of parliament (MP) and factions. However, it depends on the issues debated and voted upon. Therefore, analysis of carefully selected sets of roll call votes provides a deeper knowledge about MPs. However, in order to classify roll call votes according to their topic automatic text classifiers have to be employed, as these votes are counted in thousands. It can be formulated as a problem of classification of short legal texts in Lithuanian (classification is performed using only headings of roll call vote).

We present results of an ongoing research on thematic classification of roll call votes of the Lithuanian Parliament. The problem differs significantly from the classification of long texts, because feature spaces are small and sparse, due to the short and formulaic texts. In this paper we investigate performance of 3 feature representation techniques (*bag-of-words*, *n-gram* and *tf-idf*) in combination with Support Vector Machines (with different kernels) and Multinomial Logistic Regression. The best results were achieved using *tf-idf* with SVM with linear and polynomial kernels.

1 Introduction

Increasing availability of data on activities of governments and politicians as well as tools suitable for analysis of large data sets allows political scientists to study previously under-researched topics. As parliament is one the major foci of attention of the public, the media and political scientists, statistical analysis of parliamentary activity is becoming more and more popular. In this field, parliamentary voting analysis might be discerned as getting increasing attention (Jackman, 2001; Poole, 2005; Hix et al., 2006; Bailey, 2007).

Analysis of the activity of the Lithuanian parliament (the Seimas) is also becoming more popular (Krilavičius and Žilinskas, 2008; Krilavičius and Morkevičius, 2011; Mickevičius et al., 2014; Užupytė and Morkevičius, 2013). However, overall statistical analysis of the MP voting on all the questions (bills etc.) during the whole term of the Seimas (four years) might blur the ideological divisions that arise from the differences in the positions taken by MPs depending on their attitudes towards the governmental policy or topics of the votes (Roberts et al., 2009; Krilavičius and Morkevičius, 2013). Therefore, one of the important tasks is creating tools to compare the voting behavior of MPs with regard to the topics of the votes and changes in the governmental coalitions.

One of the options to assign a thematic category to each topic is manual annotation. However, due to a large amount of voting data and constantly increasing database (there are up to 10000 roll call votes in each term of the Seimas) it becomes complicated. Better solution may be introduced by using automatic classification with machine learning and natural language processing methods.

Some attempts to classify Lithuanian documents were already made (Kapočiūtė-Dzikienė et al., 2012; Kapočiūtė-Dzikienė and Krupavičius, 2014; Mickevičius et al., 2015), but they pursue different problems, i.e., the first one works with full text documents, the second tries to predict faction from the record and the last one is quite sparse (only the basic text classifiers are examined). This paper presents a broader research which aims to find an optimal automatic text classifier for short political texts (topics of parliamentary votes) in Lithuanian. The methods used are rather well known and standard with other languages than

Proceedings of the 5th Workshop on Balto-Slavic Natural Language Processing, pages 106–111,
Hissar, Bulgaria, 10–11 September 2015.

Lithuanian. However, due to specific type of analyzed short legal texts and high inflatability of Lithuanian language (Kapočiūtė-Dzikienė et al., 2012) these methods must be tested under different conditions.

New tasks tackled in this paper include experiments with: (1) different features, namely bag-of-words, *n-gram* and *tf-idf*; (2) different classifiers: Support Vector Machines (Harish et al., 2010; Vapnik and Cortes, 1995; Joachims, 1998), including different kernels (Shawe-Taylor and Cristianini, 2004), and Multinomial Logistic Regression (Aggarwal and Zhai, 2012); (3) identifying the most efficient combinations of text classifiers and feature representation techniques.

Automatic classification of Seimas' voting titles is a part of an ongoing research dedicated to creating an infrastructure that would allow its user to monitor and analyze the data of roll call voting in the Seimas. The main idea of the infrastructure is to enable its users to compare behaviors of the MPs based on their voting results.

2 Data

2.1 Data Extraction

All data used in the research is available on the Lithuanian Parliament website[1]. In order to convert data into suitable format for storage and analysis, a custom web crawler was developed and used. The corpus used in the research was generated applying the following steps: (1) The object of analysis are the titles of debates in Lithuanian Parliament; (2) Following a unique ID (which is assigned to every debate in Seimas) every debate title was examined (no titles were skipped); (3) The analyzed time span goes from 2007-09-10 to 2015-04-14; (4) Only titles of debates that included at least one roll call voting were selected for the analysis. Using such approach 11521 text documents were retrieved.

2.2 Preprocessing and Descriptive Statistics

In order to eliminate the influence of functional words and characters (as well as spelling errors), the documents were normalized in the following way: (1) Punctuation marks and digits removed; (2) Uppercase letters converted to lowercase; (3) 185 stop words (out of 3299 unique words) were removed.

[1]URL: http://www.lrs.lt

Descriptive statistics of the preprocessed text documents are provided in Table 1.

Length	Words	Characters
Minimum	2	19
Average	33	264
Maximum	775	6412

Table 1: Descriptive statistics of the corpus.

2.3 Classes

In order to achieve proper results of automatic text classification, clearly defined classes must be used. To fulfill this requirement classification scheme of Danish Policy Agendas project[2] was followed. Regarding the size of the analyzed corpus, 21 initial thematic categories were aggregated into 7 broader classes.

A set of 750 text documents were selected (see below) and manually classified to build a gold standard. To avoid bias in automatic classification towards populated classes, the amounts of documents belonging to classes should not be significantly different, therefore the text documents were not selected randomly. Instead approximately 100 of objects for each class (aggregate topic) were picked from the debates of the last term of the Seimas (from 2012-11-16). See Table 2 for the number of text documents in each class.

Class	No. of docs
Economics	126
Culture and civil rights	121
Legal affairs	106
Social policy	107
Defense and foreign affairs	82
Government operations	104
Environment and technology	103
Total	**750**

Table 2: Corpora.

3 Tools and Methods

3.1 Feature Representation Techniques

Bag-of-words. When using this method, the terms are made of single and whole words. Therefore,

[2]URL: http://www.agendasetting.dk

the dictionary of all unique words in the corpus needs to be produced. Then a feature vector of length m is generated for each text document in the data, where m is a total number of unique words in the dictionary. Feature vectors contain the frequencies of terms in the text documents.

N-grams. Using this method text documents are divided into character sets (substrings) of length n insomuch as the first substring contains all the characters of the documents from the 1st to n-th inclusive. Second substring contains all characters of the document from 2nd to $(n+1)$-th inclusive. This principle is used throughout the whole text document, the last substring containing characters from $(k-n+1)$-th to k-th, where k is the number of characters in the text document. This process is applied to each text document and a dictionary of unique substrings (considered as terms) of length n (n-grams) is generated. Character sets is one of several ways to use n-grams. However, character n-grams tend to show significantly better results in this case (Mickevičius et al., 2015) than word n-grams, therefore it was decided to discard word n-grams in the study.

tf-idf. The idea of *tf-idf* (term frequency - inverse document frequency) method is to estimate the importance of each term according to its frequency in both the text document and the corpus). Suppose t is a certain term used in a document d, which belongs to corpus D. Then each element in the feature vector of d is calculated using (1), (2) and (3) formulas:

$$tf(t,d) = 0.5 + \frac{0.5 \cdot f(t,d)}{\max\{f(w,d) : w \in d\}} \quad (1)$$

$$idf(t,d,D) = \log \frac{N}{1 + |\{d \in D : t \in d\}|} \quad (2)$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,d,D), \quad (3)$$

where $f(t,d)$ is a *raw term frequency* (count of term appearances in the text document), $\max\{f(w,d) : w \in d\}$ is a *maximum raw frequency* of any term in the document, N is a *total number of documents* in the corpus, and $|\{d \in D : t \in d\}|$ is a *number of documents* where the term t appears. The base of the logarithmic function does not matter, therefore natural logarithm was used. The term itself was defined as a single separate word (identically to *bag-of-words* method).

3.2 Text Classifiers

Support Vector Machines (SVM) (Harish et al., 2010; Vapnik and Cortes, 1995; Joachims, 1998). A document d is represented by a vector $x = (w_1, w_2, \ldots, w_k)$ of the counts of its words (or *n-grams*). A single SVM can only separate 2 classes: a positive class $L1$ (indicated by $y = +1$) and a negative class $L2$ (indicated by $y = -1$). In the space of input vectors x a hyperplane may be defined by setting $y = 0$ in the linear equation $y = f_\theta(x) = b_0 + \sum_{j=1}^{k} b_j w_j$. The parameter vector is given by $\theta = (b_0, b_1, ..., b_k)$. The SVM algorithm determines a hyperplane which is located between the positive and negative examples of the training set. The parameters b_j are estimated in such a way that the distance ξ, called margin, between the hyperplane and the closest positive and negative example documents is maximized. The documents having distance ξ from the hyperplane are called support vectors and determine the actual location of the hyperplane.

SVMs can be extended to a non-linear predictor by transforming the usual input features in a non-linear way using a feature map. Subsequently a hyperplane may be defined in the expanded (latent) feature space. Such non-linear transformations define extensions of scalar products between input vectors, which are called kernels (Shawe-Taylor and Cristianini, 2004).

Multinomial Logistic Regression (Aggarwal and Zhai, 2012). An early application of regression to text classification is the Linear Least Squares Fit (LLSF) method, which works as follows. Let the predicted class label be $p_i = \overline{A} \cdot \overline{X_i} + b$, and y_i is known to be the true class label, then our aim is to learn the values of A and b, such that the LLSF $\sum_{i=1}^{n} (p_i - y_i)^2$ is minimized.

A more natural way of modeling the classification problem with regression is the logistic regression classifier, which differs from the LLSF method by optimizing the likelihood function. Specifically, we assume that the probability of observing label y_i is:

$$p(C = y_i|X_i) = \frac{exp(\bar{A} \cdot \bar{X}_i + b)}{1 + exp(\bar{A} \cdot \bar{X}_i + b)}. \quad (4)$$

In the case of binary classification, $p(C = y_i|X_i)$ can be used to determine the class label. In the case of multi-class classification, we have $p(C = y_i|X_i) \propto exp(\bar{A} \cdot \bar{X}_i + b)$, and the class label with the highest value according to $p(C = y_i|X_i)$ would be assigned to X_i.

3.3 Testing and Quality Evaluation

Training and testing of the classifiers was performed using 750 selected text documents with training:testing data ratio being 2:1. All selected documents were ordered randomly and a non-exhaustive 6-fold cross validation was applied.

Standard evaluation measures of *precision* $\left(P_n = \frac{\mathrm{TP}_n}{\mathrm{TP}_n+\mathrm{FP}_n}\right)$, *recall* $\left(R_n = \frac{\mathrm{TP}_n}{\mathrm{TP}_n+\mathrm{FN}_n}\right)$ and *F-score* $\left(F_n = \frac{2 \cdot P_n \cdot R_n}{P_n+R_n}\right)$ were used for each class and overall, and where

- *True positive (TP)*: number of documents correctly assigned class C_n;
- *False positive (FP)*: number of documents incorrectly assigned to class C_n;
- *False negative (FN)*: number of documents that belong, but were not assigned to C_n;
- *True negative (TN)*: number of documents correctly assigned to class, different than C_n.

Baseline accuracy was calculated using the following equation $ACC_B = \frac{1}{N^2} \sum_{i=1}^{m} {N_i}^2$, where N is the total number of documents in the training dataset, N_i is the number of documents in the training dataset that belong to class C_i, and m is the number of classes. In this case: $ACC_B = 0,151$.

4 Experimental Evaluation

4.1 Method Selection

3 variations of the most popular feature selection methods were used, see statistics in Table 3.

Feature set	Unique terms	
	Overall	Per doc
bag-of-words	3130	0,27
3-gram	3995	0,35
tf-idf	3130	0,27

Table 3: Descriptive statistics of the feature sets.

Due to good performance (Mickevičius et al., 2015) SVM classifier was examined more in depth. Multinomial Logistic Regression was selected as a second classifier in order to test its suitability to Lithuanian political texts.

Logistic Regression is a powerful method with no parameters that would be crucial to adjust. SVM is quite the opposite with the following changeable parameters: *kernel* function, *degree* (for polynomial kernel only), *cost* and *gamma* (for all kernels except linear).

Parameters were tuned using cross-validation to find the best performance thus determining the most suitable values for each parameter. *Cost* and *gamma* parameters were picked of a range from 0.1 to 3 by a step of 0.1, and 6 different kernel functions were tested: linear, 2 to 4 degree polynomial, Gaussian radial basis and sigmoid function.

4.2 Classification Results

After the parameter tuning phase the most suitable parameter values were found and maximal classification quality (*F-score*) was achieved with each tested classifier and feature representation method, see Table 4.

Classifier	b-o-w	3-gram	tf-idf
SVM linear	0.716	0.683	**0.825**
SVM pol. 2 deg.	0.701	0.613	**0.815**
SVM pol. 3 deg.	0.646	0.593	**0.815**
SVM pol. 4 deg.	0.589	0.567	**0.815**
SVM radial	0.610	0.169	0.728
SVM sigmoid	0.325	0.091	0.057
LogReg	0.696	0.667	**0.793**

Table 4: Best performing classifiers, F-score.

Five classifier and feature representation method combinations produced exceptionally good results in comparison to other combinations. It is easy to see that *tf-idf* features are superior to *bag-of-words* and *n-gram* regardless of the classifier.

The aforementioned classifiers were subjected to deeper analysis where *precision*, *recall* and *F-score* measures were estimated for each class. The results are shown in Tables 5, 6, 7, 8 and 9 while averaged *F-score* for each of the 5 best classifiers are depicted in Table 4.

Best performing classifier for each class is depicted in Figure 1. Further analysis did not yield information about certain classifier being unsuitable due to neglect of one or more classes. Considering a narrow margin that separates the quality of tested classifiers (the highest *F-score* is 0.825, the lowest is 0.793) it would be fair to consider all 5 of them being equally suitable for classifying roll call votes headings of the Lithuanian Parliament.

Class	Prec.	Rec.	F-score
1	**0.978**	0.913	**0.944**
2	0.936	0.835	0.883
3	0.649	0.710	0.678
4	0.846	**0.846**	**0.846**
5	0.863	**0.824**	**0.843**
6	**0.777**	0.732	**0.754**
7	0.591	0.898	0.713

Table 5: SVM, linear kernel, tf-idf.

Class	Prec.	Rec.	F-score
1	0.973	0.892	0.931
2	0.936	**0.839**	0.885
3	0.699	**0.757**	0.727
4	0.810	0.813	0.811
5	**0.893**	0.765	0.824
6	0.698	**0.750**	0.723
7	**0.612**	0.867	**0.718**

Table 6: SVM, 2 degree polynomial kernel, tf-idf.

Class	Prec.	Rec.	F-score
1	0.973	0.895	0.932
2	**0.940**	**0.839**	**0.887**
3	0.703	**0.757**	0.729
4	0.805	0.813	0.809
5	0.886	0.765	0.821
6	0.701	**0.750**	0.725
7	0.609	0.857	0.712

Table 7: SVM, 3 degree polynomial kernel, tf-idf.

Class	Prec.	Rec.	F-score
1	0.973	0.895	0.932
2	**0.940**	**0.839**	**0.887**
3	0.703	**0.757**	0.729
4	0.805	0.813	0.809
5	0.880	0.765	0.818
6	0.700	0.746	0.722
7	0.609	0.857	0.712

Table 8: SVM, 4 degree polynomial kernel, tf-idf.

5 Results, Conclusions and Future Plans

1. *Tf-idf* feature matrix produced significantly better results than any other feature matrix.

Class	Prec.	Rec.	F-score
1	0.911	**0.934**	0.922
2	0.905	**0.839**	0.871
3	**0.837**	0.698	**0.761**
4	**0.874**	0.774	0.821
5	0.826	0.654	0.730
6	0.725	0.693	0.709
7	0.428	**0.939**	0.588

Table 9: Multinomial Logistic Regression, tf-idf.

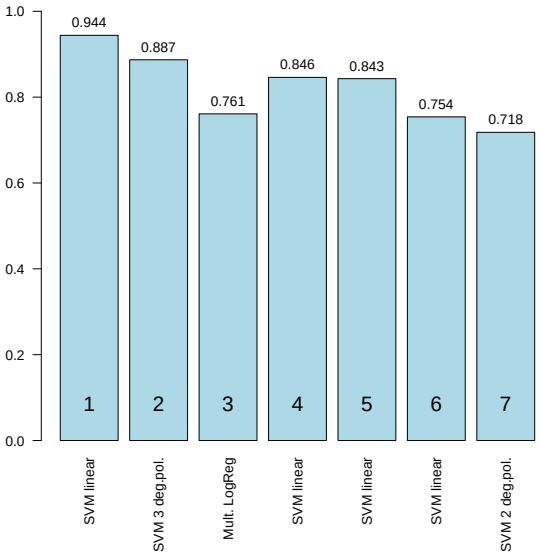

Figure 1: Best classifier for each class, F-score.

2. Linear and polynomial kernels produced the best results when using SVM classifier.

3. Support Vector Machines and Multinomial Logistic Regression are suitable for classification of titles of votes in the Seimas.

These results are part of a work-in-progress of creating an infrastructure for monitoring activities of the Lithuanian Parliament (Seimas). Future plans include investigation of other text classifiers, feature preprocessing and selection techniques.

Certain titles of the Seimas debates present a challenge even for human coders due to ambiguity. For that reason multi-class classification and analysis of larger datasets (additional documents attached to the debates and votes) are planned in the future. A critical review and stricter definitions of classes, as well as qualitative error analysis are also included in the future plans.

References

Charu C. Aggarwal and ChengXiang Zhai. 2012. *A Survey of Text Classification Algorithms*. Springer US.

Michael A. Bailey. 2007. Comparable preference estimates across time and institutions for the court, Congress, and presidency. *American Jrnl. of Political Science*, 51(3):433–448.

Bhat S. Harish, Devanur S. Guru, and Shantharamu Manjunath. 2010. Representation and classification of text documents: a brief review. *IJCA,Special Issue on RTIPPR*, (2):110–119.

Simon Hix, Abdul Noury, and Gérard Roland. 2006. Dimensions of politics in the European Parliament. *American Jrnl. of Political Science*, 50(2):494–520.

Simon Jackman. 2001. Multidimensional analysis of roll call. *Political Analysis*, 9(3):227–241.

Thorsten Joachims. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proc. of ECML-98, 10th European Conf. on Machine Learning*, pages 137–142, DE.

Jurgita Kapočiūtė-Dzikienė and Algis Krupavičius. 2014. Predicting party group from the Lithuanian parliamentary speeches. *ITC*, 43(3):321–332.

Jurgita Kapočiūtė-Dzikienė, Frederik Vaasen, Algis Krupavičius, and Walter Daelemans. 2012. Improving topic classification for highly inflective languages. In *Proc. of COLING 2012*, pages 1393–1410.

Tomas Krilavičius and Vaidas Morkevičius. 2011. Mining social science data: a study of voting of members of the Seimas of Lithuania using multidimensional scaling and homogeneity analysis. *Intelektinė ekonomika*, 5(2):224–243.

Tomas Krilavičius and Vaidas Morkevičius. 2013. Voting in Lithuanian Parliament: is there anything more than position vs. opposition? In *Proc. of 7th General Conf. of the ECPR Sciences Po Bordeaux*.

Tomas Krilavičius and Antanas Žilinskas. 2008. On structural analysis of parlamentarian voting data. *Informatica*, 19(3):377–390.

Vytautas Mickevičius, Tomas Krilavičius, and Vaidas Morkevičius. 2014. Analysing voting behavior of the Lithuanian Parliament using cluster analysis and multidimensional scaling: technical aspects. In *Proc. of the 9th Int. Conf. on Electrical and Control Technologies (ECT)*, pages 84–89.

Vytautas Mickevičius, Tomas Krilavičius, Vaidas Morkevičius, and Aušra Mackutė-Varoneckienė. 2015. Automatic thematic classification of the titles of the Seimas votes. In *Proc. of the 20th Nordic Conference of Computational Linguistics (NoDaLiDa 2015)*, pages 225–232.

Keith T. Poole. 2005. *Spatial Models of Parliamentary Voting*. Cambridge Univ. Press.

Jason M. Roberts, Steven S. Smith, and Steve R. Haptonstahl. 2009. The dimensionality of congressional voting reconsidered.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Rūta Užupytė and Vaidas Morkevičius. 2013. Lietuvos Respublikos Seimo narių balsavimų tyrimas pasitelkiant socialinių tinklų analizę: tinklo konstravimo metodologiniai aspektai. In *Proc. of the 18th Int. Conf. Information Society and University Studies*, pages 170–175.

Vladimir N. Vapnik and Corinna Cortes. 1995. Support-vector networks. *Machine Learning*, 2:273–297.