Automatic Thematic Classification of the Titles of the Seimas Votes

Vytautas Mickevičius[1,2] **Tomas Krilavičius**[1,2]
Vaidas Morkevičius[3] **Aušra Mackutė-Varoneckienė**[1]
[1]Vytautas Magnus University, [2]Baltic Institute of Advanced Technology,
[3]Kaunas University of Technology, Institute of Public Policy and Administration
vytautas.mickevicius@bpti.lt, t.krilavicius@bpti.lt,
vaidas.morkevicius@ktu.lt, a.mackute-varoneckiene@if.vdu.lt

Abstract

Statistical analysis of parliamentary roll call votes is an important topic in political science as it reveals ideological positions of members of parliament and factions. However, these positions depend on the issues debated and voted upon as well as on attitude towards the governing coalition. Therefore, analysis of carefully selected sets of roll call votes provides deeper knowledge about members of parliament behavior. 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.

In this paper we present results of an ongoing research on thematic classification of roll call votes of the Lithuanian Parliament. Also, this paper is a part of a larger project aiming to develop the infrastructure designed for monitoring and analyzing roll call voting in the Lithuanian Parliament.

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 science researchers to study previously under-researched subjects. 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 prominent. In this field, parliamentary voting analysis might be discerned as getting increasing attention (Jackman, 2001; Poole, 2005; Hix et al., 2006; Bailey, 2007; Jakulin et al., 2009; Lynch and Madonna, 2012). Analysis of the activity of the Lithuanian parliament (the Seimas) is also becoming more popular. Voting of Lithuanian members of parliament (MPs) has been analyzed using various methods from both political science as well as statistical perspectives. Importantly, quite many different methods of statistical analysis have already been applied, such as multidimensional scaling (Krilavičius and Žilinskas, 2008), homogeneity analysis (Krilavičius and Morkevičius, 2011), cluster analysis (Mickevičius et al., 2014), and social networks analysis (Užupytė and Morkevičius, 2013).

This paper present results 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. However, overall statistical analysis of the MP voting on all the questions (bills etc.) during the whole term of the Seimas (4 years) might blur the ideological divisions that arise from 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 the possibility to compare the voting behavior of MPs with regard to the topics of the votes and changes in the governmental coalitions. The latter objective is rather unproblematic as changes in the government are closely monitored by the media and information on the Seimas website (www.lrs.lt) allows extracting the information about MPs' belonging to factions, which can easily be matched with their position regarding the governmental coalition.

The other feature – possibility to monitor MPs' voting with regard to the topic of the vote – is more problematic to implement. (1) Votes on the floor of the Seimas are not thematically annotated by the Office of the Seimas, nor are there interest groups that are doing this (as in the US). Therefore, it is not possible to use any of such sources in classifying the votes. (2) Political science literature abounds with rather different approaches to the classification of political texts into thematic categories,[1] which requires making difficult subjective choices in selecting among them if one is about to include any of them into the infrastructure. (3) Even more problematic aspect is related to the vast quantities of votes in the parliament (counted in thousands) and the resulting requirement of automatic classification of them according to some selected topic scheme.

This paper presents research in progress which aims to find an optimal automatic text classifier for political texts (topics of parliamentary votes) in Lithuanian. The tasks tackled in the paper include: (1) To test the two most popular methods of natural language processing and feature selection – bag-of-words and *n*-gram; (2) To test the two most popular text classifiers – Support Vector Machines (SVM) and *k* nearest neighbors (*k*-NN); (3) To compare the efficiency of the selected text classifiers when using binary and non-binary feature matrices. Some attempts to classify Lithuanian documents were already made (Kapočiūtė-Dzikienė et al., 2012; Kapočiūtė-Dzikienė and Krupavičius, 2014), but they pursue a different problem, i.e. the first one works with full text documents, while the latter tries predicting faction from the record, not classify it.

The research is ongoing and the results are described in section 5 are partial. Future plans (see section 6) will cover more experiments with Lithuanian political texts.

2 Data

2.1 Data extraction

The data used for the study was extracted from the official Lithuanian parliament web site (`www.lrs.lt`). It consists of the titles of debates and votes that took place in the Seimas from 2008-11-17 to 2014-03-25 (`www3.lrs.lt/pls/inter/w5_sale.kad_ses`). The following rules were applied when collecting data: (1) debates from 2008-11-17 to 2014-03-25 were examined; (2) only debates with roll call votes were included; (3) in cases when single roll call votes were associated with several (usually very similar) titles of the debates (the so-called 'package voting'), these titles were merged and treated as one case.

Following these rules, the titles for 12211 roll call votes were identified in the time period analyzed and accordingly 12211 text documents (consisting of the titles of these votes) generated for further processing and analysis.

2.2 Preprocessing

In order to eliminate the influence of functional characters in the text analysis, the documents were normalized in the following way: (1) all punctuation marks were removed with no exceptions; (2) all multiple space characters (either intentional or not) were merged into one space character; (3) all numbers were removed; (4) all uppercase letters were converted to lowercase in order to eliminate the influence of word capitalization.

After the preprocessing a dictionary consisting of 2762 different words from the texts was generated. Here the word is defined as a set (or a substring) of symbols which is separated from the rest of text by one (in the beginning or the end of text) or two (in the middle) non-consecutive space characters.

Descriptive statistics of the text documents can be seen in table 1.

Length	In words	In characters
Minimum	2	19
Average	31	247
Maximum	775	6344

Table 1: Descriptive statistics of text documents.

Figures 1 and 2 show the frequencies of words and characters in the text documents.

2.3 Training and testing data

A set of 750 text documents (titles of votes) was selected out of the original data set to be used for training and testing of the classifiers. 500 documents were used for training of the classifiers and 250 documents were used to test the results.

These 750 titles of votes (text documents) were manually classified[2] into 7 aggregate

[1] Two major attempts are Manifesto Research Group (`manifestoproject.wzb.eu`) and Policy Agendas/Comparative Agendas (`www.comparativeagendas.info`) projects

[2] For the help in performing the classification authors thank Giedrius Žvaliauskas, researcher at the KTU Institute of Public Policy and Administration.

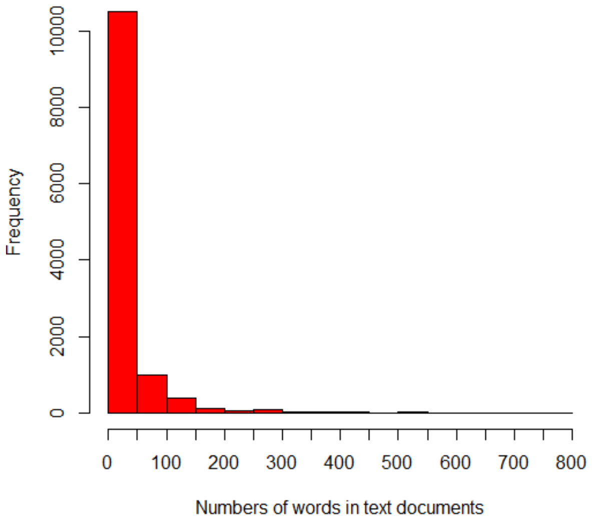

Figure 1: Distribution of words in the text documents.

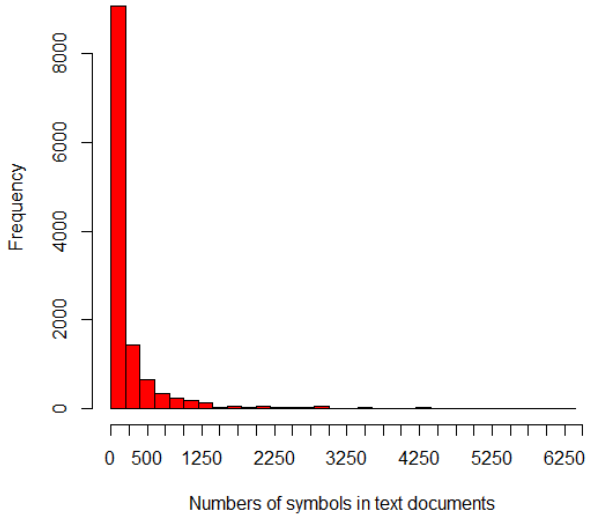

Figure 2: Distribution of characters on the text documents.

classes using the classification scheme of the Danish Policy Agendas project (`http://www.agendasetting.dk`). In order to avoid bias in automatic classification towards a more populous classes, the amounts of texts belonging to classes should not be significantly different, therefore titles of votes consisting the data set were not selected randomly: around 100 of votes for each class (aggregate topic) were selected 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 and the names of the classes.

Class	**No. of text documents**
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: Manual classification of documents.

3 Tools and methods

The research was performed using statistical package R (Team, 2013), a free software for statistical computing and graphics.

3.1 Features

Several popular feature representation techniques were used.

Bag-of-words is arguably the simplest and one of the most popular techniques for natural language processing. First of all, the dictionary of all unique words (for a definition of a word, see 2.2) in all of text documents is generated. 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. Every element in the feature vector represents the count of appearance of a word in a text document for which the feature vector is generated. For example, if the 5th element of a feature vector is equal to 3, this indicates that the 5th word of the constructed dictionary occurs 3 times in a document under consideration.

***N*-gram.** Using this method documents are divided into character sets (substrings) of length n insomuch as the first substring contains all characters of the document 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 through the whole text document, the last substring containing characters from $(k-n+1)$ to k, where k is the number of characters in the text document. This process is applied to each given text document and a dictionary of unique substrings of length n (called n-grams) is generated. The set of feature vectors (feature matrix) is generated using the same principle as in the bag-of-words method, the only difference is that feature vectors contain counts of *n*-grams in a given text document instead of full words.

Sets containing series of characters is only one of several ways to use *n*-grams. Substrings can also be constructed of whole words, phonemes, syllables and other morphological units. The technique of using *n*-grams is advantageous in terms of flexibility as it does not require intensive data preprocessing, such as stemming, lemmatizing or removal of stop-words.

3.2 Text classifiers

Support Vector Machines (SVM) (Harish et al., 2010). This is a supervised classification algorithm (Vapnik and Cortes, 1995) that has been extensively and successfully used for the text classification tasks (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 two 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 adapted 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 input space. Such non-linear transformations define extensions of scalar products between input vectors, which are called kernels (Shawe-Taylor and Cristianini, 2004). In this paper linear kernel is examined, while analysis of non-linear kernels is included in the future plans (see section 6).

***K* Nearest Neigbors (*k*-NN) (Harish et al., 2010).** Let X be a document to classify. Using *k*-NN method distances between every document in a training dataset and document X are found. Out of all, k least distances are selected, considering the corresponding k documents nearest neighbors to document X. Document X is then assigned to a class that dominates in a set of k nearest neighbors.

This method has two modifiable parameters: dissimilarity measure (distance) and the number of nearest neighbors k. Euclidean distance is one of the most popular dissimilarity measure, calculated using formula 1.

$$d(X,Y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}, \qquad (1)$$

here d is a distance between text documents X and Y, m is a number of features (length of feature vector), x_i and y_i – i-th feature (i-th element of feature vectors) of documents X and Y respectively.

The optimal number k of neighbors may be estimated from training data by cross validation (Hotho et al., 2005).

3.3 Testing results evaluation

As the actual classes of text documents in a training data set are known, it is possible to compare predicted classes with the actual ones. In order to evaluate testing results generated by a text classifier, formula 2 is applied.

$$ACC = \frac{\sum_{i=1}^{k} q_i}{k} \cdot 100\%, \quad x_i = \begin{cases} 1, & a_i = p_i \\ 0, & a_i \neq p_i \end{cases}, \qquad (2)$$

here ACC is the accuracy of the examined classifier, k is the number of documents in a testing data set, a_i is the i-th element of a vector that contains actual classes of the documents in a testing data set, p_i is the i-th element of a vector that contains predicted classes of the documents in a testing data set.

4 Experimental evaluation

4.1 Feature selection

For the analysis 5 dictionaries were generated out of 12211 text documents employing several variations of 2 natural language processing methods. While bag-of-words method is more or less straightforward and does not depend on changeable parameters, *n*-grams were analyzed in more depth – *3*-grams and *4*-grams were selected for the research discussed in this paper. Also, differences in classification effectiveness of *n*-grams as character sets and *n*-grams as word sets were analyzed. Descriptive statistics of the dictionaries generated can be seen in table 3.

Dictionary	**No. of entries**
Bag-of-words	2762
3-gram, chars	3730
4-gram, chars	10004
3-gram, words	12006
4-gram, words	16541

Table 3: Descriptive statistics of the dictionaries.

For every dictionary 2 feature matrices (10 feature matrices in total) were generated – one containing the counts of words in the feature vectors (as described in 3.1) and the other binary. Binary feature matrix is a variation of regular feature matrix where the feature is not the number of words in a document but the presence of a word in a document. Binary feature matrices were generated by converting all the elements greater than 0 (a word is not present in a document) to 1 (word is present in a document).

4.2 Automatic classification of documents

Out of every (10) feature matrices 750 documents were selected for training and testing of the classifiers (see 2.3 for the details). In order to achieve greater effectiveness training and testing was implemented in 6 iterations using cross-validation. First, all 750 selected documents were listed randomly. Then during each iteration document set was split 500 : 250 for training and testing classifiers, respectively. See table 4 for the details about data selection for each iteration.

No. of iteration	**Training set**	**Testing set**
1	1–500	501–750
2	51–550	1–50, 551–750
3	101–600	1–100, 601–750
4	151–650	1–150, 651–750
5	201–700	1–200, 701–750
6	251–750	1–250

Table 4: Data selection for cross-validation.

See results of experiments, in tables 5 and 6, for SVM and k-NN, correspondingly. The results show that *n*-grams representing sets of characters produce significantly better classification accuracy than *n*-grams representing sets of full words for both SVM and *k*-NN classifiers.

For SVM classifier, bag-of-words method of feature selection produced significantly better results than any of the analyzed *n*-gram variations, whereas *k*-NN classifier did not indicate any feature matrix as superior to the others. It is notable that increasing the number of nearest neighbors used in *k*-NN classifier produces worse results, therefore, 1-NN variation might be considered optimal.

Features	**Binary**	**Testing accuracy (%)**
Bag-of-words	No	**70.7**
3-gram, chars	No	56.7
4-gram, chars	No	55.5
3-gram, words	No	48.5
4-gram, words	No	39.7
Bag-of-words	Yes	**70.5**
3-gram, chars	Yes	58.3
4-gram, chars	Yes	55.7
3-gram, words	Yes	48.3
4-gram, words	Yes	40.1

Table 5: Classification accuracy (%) with SVM.

Features	**Binary**	**No. of nearest neighbors (accuracy, %)**		
		1	3	5
Bag-of-words	No	**55.3**	46.3	45.5
3-gram, chars	No	54.1	47.1	43.8
4-gram, chars	No	52.7	47.4	43.5
3-gram, words	No	35.9	27.6	24.5
4-gram, words	No	30.9	22.9	21.6
Bag-of-words	Yes	57.6	46.7	43.7
3-gram, chars	Yes	**58.5**	51.8	48.8
4-gram, chars	Yes	54.8	47.8	45.4
3-gram, words	Yes	35.3	28.1	24.4
4-gram, words	Yes	30.3	22.3	21.8

Table 6: Classification accuracy (%) with *k*-NN.

The 5 best results achieved by the used classifiers are presented in table 7.

5 Results and conclusions

1. **Support Vector Machines (SVM) classifier is more suitable for automatic classification of Lithuanian political texts (titles of the Seimas votes) than *k* nearest neighbors (*k*-NN) method.** During the experiments a maximum of 70.7% classification accuracy was achieved using SVM, with a maximum of *k*-NN method being 58.5%.

Classifier	Features	Binary	Testing accuracy (%)
SVM	Bag-of-words	No	**70.7**
SVM	Bag-of-words	Yes	**70.5**
1-NN	*3*-gram, chars	Yes	58.5
SVM	*3*-gram, chars	Yes	58.3
SVM	*3*-gram, chars	No	56.7

Table 7: Summary of the best classifiers.

2. **Bag-of-words method of feature representation is more suitable than *n*-grams while using SVM classifier.** The maximum accuracy combining SVM with bag-of-words technique was 70.7%, while the maximum accuracy combining SVM with any variation of *n*-gram was 58.3%.

3. **There is no significant difference between feature selection method when using *k*-NN classifier.** The maximum accuracies combining bag-of-words and *n*-gram with *k*-NN were 57.6% and 58.5% respectively.

4. **Using *n*-gram feature representation with political texts in Lithuanian language (titles of the Seimas votes), *3*-grams and *4*-grams should represent sets of consecutive characters, not sets of consecutive words.** *3*-grams consisting of characters produced maximum accuracy of 58.5%, while using *3*-grams consisting of words only 48.5% maximum accuracy was achieved. The corresponding maximums when using *4*-grams were 55.7% and 40.1%. Combined with *k*-NN classifier, *n*-grams consisting of words showed notably poorer results.

5. **Optimal number of nearest neighbors using *k*-NN method is 1.** Increasing number of nearest neighbors corresponds with deteriorating classification accuracy.

6. **No significant difference between the types of feature matrix (binary and non-binary) was detected.** Slightly better results were achieved using binary feature matrices with *k*-NN method, while the same matrices with SVM classifier produced nearly identical results.

6 Future plans

The results presented in this research paper are partial results of work-in-progress of creating a larger infrastructure of monitoring activities of the Lithuanian Seimas. The plans of further research in the field of automatic text classification are as follows:

1. Experiments with other classifiers, such as Multinomial Naive Bayes, Artificial Neural Networks, Logistic Regression, etc;

2. Experiments with other feature representation and selection techniques, such as tf-idf, w-shingling;

3. To use linguistically preprocessed data sets, such as stemmed or lemmatized dictionaries.

There are also plans to perform text classification on larger sets of data, including:

1. Analysis of titles of debates from all the sessions of the Lithuanian Parliament, regardless of the presence of roll call votes;

2. Employing additional documents (such as texts of the debated laws, bills, resolutions etc.) attached to the debates and votes.

It was also discovered that the problem of misclassification might be related with the fact that certain titles of the Seimas debates present classification challenge even for human coders. In other words, titles of the Seimas debates (and especially votes) can not be clearly assigned to one of the classes using only the title itself. More information about the debates and votes might be required. Also, classes (aggregate topics of Policy Agendas) themselves might require a critical review and stricter definitions.

The ultimate plan remains the same – to combine the results of automatic classification of debates (votes) with the analysis of roll call votes in the Seimas. This should result in a completion of the infrastructure designed for monitoring and analysis of the activity of the Lithuanian Parliament.

References

M.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.

B.S. Harish, D.S. Guru, and S. Manjunath. 2010. Representation and Classification of Text Documents: a Brief Review. *IJCA,Special Issue on RTIPPR*, (2):110–119.

S. Hix, A. Noury, and G. Roland. 2006. Dimensions of Politics in the European Parliament. *American Jrnl. of Political Science*, 50(2):494–520.

A. Hotho, A. Nürnberger, and G. Paaß. 2005. A Brief Survey of Text Mining. *Jrnl for Comp. Linguistics and Language Technology*, 20:19–62.

S. Jackman. 2001. Multidimensional Analysis of Roll Call. *Political Analysis*, 9(3):227–241.

A. Jakulin, W. Buntine, T.M. La Pira, and H. Brasher. 2009. Analyzing the U.S. Senate in 2003: Similarities, Clusters and Blocs. *Political Analysis*, 17:291–310.

T. 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.

J. Kapočiūtė-Dzikienė and A. Krupavičius. 2014. Predicting Party Group from the Lithuanian Parliamentary Speeches. *ITC*, 43(3):321–332.

J. Kapočiūtė-Dzikienė, F. Vaasen, A Krupavičius, and W. Daelemens. 2012. Improving Topic Classification for Highly Inflective Languages. In *Proc. of COLING 2012*, pages 1393–1410.

T. Krilavičius and V. 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.

T. Krilavičius and V. 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*.

T. Krilavičius and A. Žilinskas. 2008. On Structural Analysis of Parlamentarian Voting Data. *Informatica*, 19(3):377–390.

M.S. Lynch and A.J. Madonna. 2012. Viva Voce: Implications from the Disappearing Voice Vote, 1865-1996. *Social Science Quarterly*, 94:530–550.

V. Mickevičius, T. Krilavičius, and V. 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.

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

J.M. Roberts, S.S. Smith, and S.R. Haptonstahl. 2009. The Dimensionality of Congressional Voting Reconsidered.

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

R Core Team, 2013. *R: A Language and Environment for Statistical Computing*. R Found. for Stat. Comp., Vienna, Austria.

R. Užupytė and V. 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.

V. Vapnik and C. Cortes. 1995. Support-Vector Networks. *Machine Learning*, 2:273–297.