Automatic Thematic Classification of the Titles of the Seimas Votes

Vytautas Mickevičius¹,² Tomas Krilavičius¹,² Vaidas Morkevičius³ Aušra Mackutė-Varoneckienė⁴

¹Vytautas Magnus University, ²Baltic Institute of Advanced Technology, ³Kaunas 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). There-
fore, it is not possible to use any of such sources
in classifying the votes. (2) Political science lit-
erature abounds with rather different approaches
to the classification of political texts into thematic
categories,\footnote{Two major attempts are Manifesto Research Group (manifestoproject.wzb.eu) and Policy Agen-
das/Comparative Agendas (www.comparativeagendas.
info) projects} which requires making difficult sub-
jective choices in selecting among them if one is
about to include any of them into the infrastruc-
ture. (3) Even more problematic aspect is related
to the vast quantities of votes in the parliament
(counted in thousands) and the resulting require-
ment 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 in-
clude: (1) To test the two most popular methods
of natural language processing and feature selec-
tion – 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 fea-
ture matrices. Some attempts to classify Lithua-
nian documents were already made (Kapočiūtė-
Dzikiene et al., 2012; Kapočiūtė-Dzikiene 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 de-
scribed 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.
1rs.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/c
ter/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 asso-
ciated 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 ana-
lyzed and accordingly 12211 text documents (con-
sisting 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 punc-
tuation 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 gener-
ated. Here the word is defined as a set (or a sub-
string) 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 char-
acters.

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

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Length} & \textbf{In words} & \textbf{In characters} \\
\hline
Minimum       & 2           & 19          \\
Average       & 31          & 247         \\
Maximum       & 775         & 6344        \\
\hline
\end{tabular}
\caption{Descriptive statistics of text documents.}
\end{table}

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 docu-
ments 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\footnote{For the help in performing the classification authors
thank Giedrius Žvaliauskas, researcher at the KTU Institute
of Public Policy and Administration.} into 7 aggregate
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.

<table>
<thead>
<tr>
<th>Class</th>
<th>No. of text documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Economics</td>
<td>126</td>
</tr>
<tr>
<td>Culture and civil rights</td>
<td>121</td>
</tr>
<tr>
<td>Legal affairs</td>
<td>106</td>
</tr>
<tr>
<td>Social policy</td>
<td>107</td>
</tr>
<tr>
<td>Defense and foreign affairs</td>
<td>82</td>
</tr>
<tr>
<td>Government operations</td>
<td>104</td>
</tr>
<tr>
<td>Environment and technology</td>
<td>103</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>750</strong></td>
</tr>
</tbody>
</table>

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 \( y = f(x) = b_0 + \sum_{j=1}^{k} b_j w_j \). The parameter vector is given by \( \theta = (b_0, b_1, \ldots, b_k) \). 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 \( \xi \) – called margin – between the hyperplane and the closest positive and negative example documents is maximized. The documents having distance \( \xi \) 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 Neighbors (\( k \)-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.

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

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.

\[
ACC = \frac{\sum_{i=1}^{k} q_i}{k} \cdot 100\%, \quad x_i = \begin{cases} 1, & a_i = p_i \\ 0, & a_i \neq p_i \end{cases}, \tag{2}
\]

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.
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.

<table>
<thead>
<tr>
<th>No. of iteration</th>
<th>Training set</th>
<th>Testing set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–500</td>
<td>501–750</td>
</tr>
<tr>
<td>2</td>
<td>51–550</td>
<td>1–50, 551–750</td>
</tr>
<tr>
<td>3</td>
<td>101–600</td>
<td>1–100, 601–750</td>
</tr>
<tr>
<td>4</td>
<td>151–650</td>
<td>1–150, 651–750</td>
</tr>
<tr>
<td>5</td>
<td>201–700</td>
<td>1–200, 701–750</td>
</tr>
<tr>
<td>6</td>
<td>251–750</td>
<td>1–250</td>
</tr>
</tbody>
</table>

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.

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%.
<table>
<thead>
<tr>
<th>Classifier</th>
<th>Features</th>
<th>Binary</th>
<th>Testing accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Bag-of-words</td>
<td>No</td>
<td>70.7</td>
</tr>
<tr>
<td>SVM</td>
<td>Bag-of-words</td>
<td>Yes</td>
<td>70.5</td>
</tr>
<tr>
<td>1-NN</td>
<td>3-gram, chars</td>
<td>Yes</td>
<td>58.5</td>
</tr>
<tr>
<td>SVM</td>
<td>3-gram, chars</td>
<td>Yes</td>
<td>58.3</td>
</tr>
<tr>
<td>SVM</td>
<td>3-gram, chars</td>
<td>No</td>
<td>56.7</td>
</tr>
</tbody>
</table>

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**


