ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 3

Predicting Party Group from the Lithuanian Parliamentary Speeches

Jurgita Kapočiūtė–Dzikienė

Vytautas Magnus University, Faculty of Informatics
Vileikos st. 8-409, LT-44404 Kaunas, Lithuania
Kaunas University of Technology, Faculty of Social Sciences
K. Donelaičio st. 20, LT-44239 Kaunas, Lithuania
e-mail: jurgita.k.dz@gmail.com

Algis Krupavičius

Kaunas University of Technology, Institute of Public Policy and Administration
K. Donelaičio st. 20-218, LT-44239 Kaunas, Lithuania

crossref http://dx.doi.org/10.5755/j01.itc.43.3.5871

Abstract. A number of recent research works have used supervised machine learning approaches with a bag-of-words to classify political texts –in particular, speeches and debates– by their ideological position, expressed with a party membership. However, our classification task is more complex due to the several reasons. First, we deal with the Lithuanian language which is highly inflective, has rich morphology, vocabulary, word derivation system, and relatively free-word-order in a sentence. Besides, we have more classes, as the Lithuanian Parliament consists of more party groups if compared to e.g. the European Parliament or the US Senate. Moreover, classes are not stable, because a considerable number of the Lithuanian parliamentarians migrate from one party group to another even within the same parliamentary term. In this research we experimentally investigated the influence of different pre-processing techniques and feature types on two datasets composed of the texts taken from two parliamentary terms. A classifier based on the bag-of-words and token bigrams interpolation gave the best results: i.e. it outperformed random and majority baselines by more than 0.13 points and achieved 0.54 and 0.49 accuracy on the 1st and the 2nd dataset, respectively. The error analysis revealed that the same confusion patterns stand for both datasets, besides, majority of these confusions can be explained on the basis of the ideological or pragmatic similarities between those party groups.

Keywords: computational linguistics; supervised machine learning; text classification into party groups.

1. Introduction

With an exponential growth in the number of electronic documents, an automatic text classification has become one of the key techniques able to organize the constant influx of the information. Therefore since the 80's automatic text classification has been an important research topic, but the first text classification approaches were based on an analysis of the application domain, and the manual construction of rules capable of taking classification decisions (as e.g. [1]). Although rule-based text classification methods assure high classification accuracy, but suffer from a domain adaptation problem: i.e. domain changes require expert intervention and manual recreation of rules. Therefore in the early 90's a supervised machine learning approach gained interest and became the dominant paradigm. Supervised machine learning does not require any manual creation of rules, because rules (defined as a model) are built automatically by observing (and generalizing) the characteristics of the text documents in a training set whose class labels were manually assigned beforehand by the domain expert. After the model is created, it can already be applied to predict the class labels of the unseen text documents automatically.

Thus, supervised machine learning techniques applied for the text classification help solving many different tasks, such as topic classification, opinion mining, authorship identification, ideology detection, etc. which, in turn, if used in practice, can facilitate the work of the experts from various fields. E.g. topic classification helps distributing e-mails to the competent recipients according to the discussed topics; opinion mining helps companies to measure the feedback about their products or services; authorship identification helps forensic experts in a criminal investigation

by comparing the writing style of unknown criminal text with the written texts of potential criminals; ideology detection helps political scientists to follow the speeches of politicians in the press or in the parliament and detect if they are consistent with their officially declared political views.

The ideological position detection (usually expressed as the party membership) from the text (or ideology-based classification) is one of the most complicated classification tasks. In this research we are using the transcripts of the speeches and debates taken from the sessions of the Lithuanian Parliament and for the first time trying to tackle ideology-based classification problem for the Lithuanian language. However, our classification task is more complex if compared to the related research due to the several reasons. First, we deal with the Lithuanian language which compared to e.g. English is highly inflective, has rich morphology, vocabulary, word derivation system; and relatively free-word-order in a sentence. Besides, we have more classes (up to 12), as the Lithuanian Parliament consists of more party groups if compared to e.g. 8 in the European Parliament or 2 in the US Senate. Moreover, the classes are not stable, because a considerable number of the Lithuanian parliamentarians migrate from one party group to another even within the same parliamentary term. We posit that from the computational linguistics perspective the findings of our research should be useful to the other, similar languages (according to such properties as inflectional morphology, complexity of word derivation, etc.) as Latvian or Slavic languages. Moreover, the results would help to reveal how much Lithuanian political parties are ideologically consistent and differ from each other according to the talks at the parliament, whereas it would be interesting findings for the political scientists.

2. Related work

In this paper we focus on the supervised machine learning methods (for a review see [2]) that are applied to text classification tasks (for a review see [3]). We narrow down the area of surveyed methods to a single-label (each text document can have only one class label), multi-class (dataset contains more than two classes), and flat (no hierarchical structure is considered) classification only.

Comparative analysis of the text classification methods performed by Joachim [4] revealed that Support Vector Machines (SVMs) and k-Nearest Neighbor (k-NN) are top-notch classifiers, compared to Decision Trees (DTs) or Naïve Bayes (NB). Dumais et al. [5] also demonstrated that SVMs work very well, followed by DTs and lastly by NB. However, Gabrilovich and Markovitch [6] on the contrary, claim that DTs (C4.5) are significantly superior to SVMs on their solving task. Besides, Pak and Paroubek [7] reported that Naïve Bayes Multinomial (NBM) can even outperform popular SVM. Such contradictory findings are due to the fact that classification results can also be affected by the different pre-processing techniques, feature representation types, and solving tasks; therefore method selection requires more comprehensive analysis.

The pre-processing techniques (such as spelling normalization, word segmentation, stemming or lemmatization, etc.) involve pre-treatment of the dataset, anticipating that this could help to get rid of the redundant information and to increase the classification results. The spelling normalization is advisable for the languages (e.g. Arabic) having very highly variable orthography; word segmentation is demanded for the languages having a lot of compound words (e.g. Swedish or German) and inevitable for the languages having no white spaces between the words (e.g. some Asian languages as Chinese or Japanese). Stemming or lemmatization is used in many classification experiments for the different languages, but is especially advisable for the languages that are highly inflective. Some comparative experiments revealed that stemming improved the text classification performance of NB, SVMs, k-NN and had extremely strong positive impact on DTs (C4.5) on the English texts [8], but had no influence on classification results for Dutch when using Mutual Information with NB [9] and even dropped down SVMs classification accuracy for Arabic [10]. Lemmatizition led to no significant SVMs classification improvements on German, and sometimes even yielded worse results [11].

The different feature types (such as bag-of-words, token n-grams, stems or lemmas, character n-grams, etc.) used with the classification method have strong influence on the results. Nevertheless, the most common feature type remains bag-of-words interpretation, especially that Pang et al. [12] showed it can beat other feature types (based on token bi-grams, parts-of-speech information and word position in the text) with SVM. But on the contrary, Dave et al. [13] report that higher order token n-grams (up to trigrams) can improve the performance compared with the unigrams (bag-of-words) approach. Cui et al. [14] also claim that higher order token n-grams ($n = 3, 4, 5, 6$) and Passive Aggressive classifier outperform unigrams and bi-grams. Pak and Paroubek [7] demonstrated that token bigrams can outperform both token unigrams and trigrams with NBM method. Nastase et al. [15] instead of simple token bigrams used syntactically related word pairs (verb + its arguments, noun + its modifiers) and classification results both with SVM and DTs were better compared with the simple bag-of-words representation. Dave et al. [13] report that stems improve classification accuracy over the simple bag-of-words baseline, but other linguistic features on the contrary – hurt the performance. Hartmann et al. [16] claim that document-level character n-grams used, namely, with NB are even better choice compared to token n-grams (because the probability of finding character n-gram is much higher, besides, the relations between consecutive words are still considered). Peng et al. [17] demonstrated that a language modeling approach with the

character n-grams gives superior classification results for English and competitive results for Chinese and Japanese over the bag-of-words. Character 3-grams and 4-grams also outperformed bag-of-words for Greek [18] with SVMs.

The method selection also depends on the solving task and here our focus is on the ideology-based classification research works. However this area is not thoroughly researched, but the most common approach used to solve ideology-based classification task is the bag-of-words representation with SVMs. Diermeier et al. [19] classified members of the U.S. Senate focusing on the most indicative conservative and liberal positions of the legislative speeches. Yu et al. [20] done a similar work, but instead of using only the most indicative speeches they used all of them. Similar work was done on the Canadian Parliament data (English and French), but the classifier was trained on one parliamentary term and tested on another [21]. Jiang and Argamon [22] before the classification of the political blogs were firstly trying to select only the subjective sentences. The majority of ideology-based classification works are done with 2 classes (party groups), but Hoyland and Godbout [23] used the European Parliament data with the 8 classes.

Unfortunately, the ideology-based classification has never been done for the Lithuanian language[1]. Consequently, this paper will be the first attempt at finding a good classification method for this task.

3. The Lithuanian language

In this section we discuss the Lithuanian language properties focusing on the aspects that might be important in method selection for solving ideology-based classification task:

- **Rich inflectional morphology.** The Lithuanian language morphology is more complex compared with e.g. Latvian or Slavic languages [25]. Besides, various inflection forms in the Lithuanian language are expressed by the different endings (and suffixes).
- **Rich vocabulary.** The Academic Dictionary of Lithuanian [26] has more than 0.5 million headwords, e.g. Oxford English Dictionary or the largest lexicon of German language have only about 0.3 million and 0.33 million headwords, respectively.
- **Rich word derivation system.** In order to derive the new Lithuanian words, prefixes, suffixes and participles are used. 19 prefixes are used to derive verbs, e.g. prefix *į* (in) attached to the simple verb *eiti* (to go) turns it into the phrasal verb *įeiti* (to come in); participle *ne-* or *nebe-* (no, not) attached as the prefix reverses the polarity of a noun, verb, adjective or adverb; participles *-s* or *-si-* attached to the ending e.g. *praustis* (to have a wash) or as the second prefix *nusiprausti* (to have a wash), respectively, turn a simple verb into reflexive; 78 suffixes are used to derive diminutives and hypocoristic nouns [27].
- **Relatively free word-order in a sentence.** The word order in the Lithuanian language performs notional function, i.e. the sentences can be grammatically correct regardless of the word order, but their meaning will be slightly different (because different word order emphasizes different things).

4. The data

All our experiments were carried out on two datasets to make sure that the findings generalize over the different domains. Both datasets were composed of the text transcripts made of the Lithuanian parliamentary speeches and debates, thus represent normative Lithuanian language[2]:

- **"2008–2012"** dataset was composed of the transcripts taken from the 6th parliamentary term, i.e. from 17/11/2008 to 14/11/2012 (see Table 1[3]);
- **"2012–2013"** dataset contains transcripts taken from the 7th parliamentary term. Since this parliamentary term is not finished yet, for our experiments we used transcripts from 16/11/2012, but only to 19/09/2013 (see Table 2).

The transcripts of the solemn sessions (held in the special occasions as The National Day, Lithuania's Independence Day, etc.) were eliminated from the datasets, leaving only the transcripts of the regular plenary sessions.

The author (parliamentarian) of each speech or debate was known, moreover, the information about which party group parliamentarian belonged when spoke was also available; therefore speeches and debates were automatically related with the appropriate party groups. "2008–2012" and "2012–2013" datasets contain 12 party groups and 8 party groups, respectively. We used the following abbreviations for the party groups: HU-LCD stands for the Homeland Union – the Lithuanian Christian Democrats; LSD – for the Lithuanian Social Democrats; O&J – for the party Order and Justice; LM – for the Liberal Movement; LP – for the Labor Party; MPG – for the Mixed Parliamentary Group; LCU – for the Liberal and Central Union; RN – for the party of the Rising Nation; OP – for the Oak party group; OL – for One Lithuania party group; CP – for the Christian Party group; LCU&RN – for the Joint Party group of LCU and RN; CR – for the party Courage Road; EAPL – for the Electoral Action of Poles in Lithuania.

[1] Ideological analysis was done only theoretically from the historical perspective [24].

[2] Downloaded from the official page of the Lithuanian Parliament: http://www3.lrs.lt/pls/inter/w5_sale.kad_ses.

[3] The total value for *Number of authors* in Table 1 is above 141 (i.e. above defined number of the Lithuanian parliament members) because a few parliamentarians quit during this parliamentary term, thus were replaced with the new members.

5. Computational linguistic analysis

5.1. Formal description of the task

Let $d \in D$ be a text document –in particular, one speech or debate of a single parliamentarian– belonging to a document space D. In our task there are two document spaces, i.e. "2008–2012" and "2012–2013".

Let C be a finite number of classes –in particular, political party groups. Since $C = \{c_1, c_2, \dots cN\}$, where $2 < N \ll \infty$, we have multi-class classification problem, because "2008–2012" dataset has 12 classes (see Table 1) and "2012–2013" – has 8 (see Table 2).

Due to the fact that parliamentarians can migrate from one party group to another, but cannot belong to more than one party group at the same time, we have a single-label classification task, where each d is labelled with only one class label c_i. d and c_i altogether form a classification instance $\langle d, c_i \rangle$.

Let function η be a classification function mapping text documents to classes: i.e. $\eta : D \to C$. We hypothesize that function η is ideology, which indeed distinguishes party groups from each other.

Let Γ denote a method, which given D as the input, could return a learned classification function η' (defined as a model) as the output: $\Gamma(D) \to \eta'$.

5.2. Classification method

Our goal is to find a classification method Γ which could create a model η' (as accurately as possible approximating η, i.e. capturing ideology), able to predict a class label of each unseen text document automatically.

For solving our problem we selected Naïve Bayes Multinomial – a supervised machine learning approach, introduced by Lewis and Gale [28]. It is based on a fact that text document d' with unknown class label has to be attached to the particular class c, whose conditional probability $P(c|d')$ is the highest.

This conditional probability is calculated as:

$$P(c|d') \propto P(c) \prod_{k=1}^{n_d} P(t_k|c), \quad (1)$$

where n_d is a number of tokens (words and numbers) in d'; t_k is a k^{th} token in d'.

Prior probability $P(c)$ and conditional probability $P(t_k|c)$, both used in eq. 1, are calculated considering known information (about each d labeled with c) stored in D.

$P(c)$ is calculated as:

$$P(c) = \frac{N_c}{N}, \quad (2)$$

where N_c is a number of documents belonging to particular class c in the document space D; N is a total number of documents in D.

$P(t_k|c)$ is calculated as:

$$P(t_k|c) = \frac{count(t_k|c)+1}{count(c)+|V|}, \quad (3)$$

where $count(t_k|c)$ is a number of particular token t_k belonging to particular class c in the document space D; $count(c)$ is a number of all tokens belonging to c in D; $|V|$ is a number of distinct tokens in D (vocabulary size).

Since the ideology-based classification task has never been solved for the Lithuanian language, we do not know which classification method could work the best. Despite it Naïve Bayes Multinomial method was selected due to the following reasons:

- **Robust to the irrelevant features** that cancel each other without affecting the results. We assume that ideology of any party group can be expressed with its separate vocabulary (i.e. set of features characterizing only that group), which distinguishes it from the others. Therefore it is important that our method would not overestimate and bind to the features that are outside this vocabulary.
- **Performs well in the domains with many equally important features** (e.g. compared with such classification methods as Decision Trees [29]). We assume that in the vocabulary characterizing any party group the features cannot be strictly ranked one-by-one according to their importance, because the attention of the party group can be focused to the many different topics at the same time. Hence, we need the method that could cope with many equally important features at the same time.
- **Very fast** (e.g. compared with Support Vector Machines [30]) and **has low storage requirements** (e.g. compared with the Memory-Based Learning methods [31]). It is especially important when dealing with a huge amount of data as it is in our task: i.e. "2008–2012" dataset contains 7,958,058 tokens (194,322 distinct); "2012–2013" datasets – 1,128,564 (74,221 distinct).
- **Dependable for the text classification experiments in general.** Naïve Bayes Multinomial is used in many text classification tasks as the baseline approach. Besides, it sometimes outperforms popular Support Vector Machine when solving opinion mining tasks (classifying texts due to the positive, negative or objective point of view of their authors) for English [7] and for Lithuanian [32].

In all our experiments we used Naïve Bayes Multinomial method implementation in WEKA [33] machine learning toolkit, version 3.6[4]. All parameters were set to their default values.

5.3. Applied pre-processing techniques

Due to a specificity of the ideology-based classification problem, the texts required special pre-treatment.

[4] Downloaded from: http://www.cs.waikato.ac.nz/ml/weka/.

Table 1. "2008–2012" dataset statistics. The last row represents total sum of values distributed over the party groups except for *Number of authors* and *Number of distinct tokens* which are distributed over the entire dataset

Party group	Numb. Of authors	Numb. of speeches & debates	Numb. of tokens (words, numbers)	Numb. of distinct tokens
CP	12	1,661	138,738	21,788
HU-LCD	47	55,884	3,023,001	118,041
LCU	16	8,638	369,255	29,944
LCU&RN	13	2,234	98,417	15,69
LM	13	14,57	640,7	50,019
LP	12	15,248	782,394	54,758
LSD	30	20,628	1,647,513	88,559
MPG	18	2,565	244,021	31,533
O&J	20	8,745	833,049	66,276
OL	11	210	16,776	5,786
OP	4	462	16,441	4,093
RN	18	3,851	147,753	18,489
		In total:		
12	49	134,696	7,958,058	194,322

Table 2. "2012–2013" dataset statistics

Party group	Numb. of authors	Numb. of speeches & debates	Numb. of tokens (words, numbers)	Numb. of distinct tokens
CR	7	523	57,666	14,002
EAPL	8	822	31,336	7,149
HU-LCD	33	3,468	279,92	36,701
LM	10	1,118	98,309	18,285
LP	29	5,481	254,495	26,93
LSD	39	5,381	290,497	33,488
MPG	10	237	21,613	6,634
O&J	12	1,809	94,728	16,07
		In total:		
8	141	18,839	1,128,564	74,221

In order to detect the effect on the text classification accuracy, we explored different pre-processing techniques at the dataset-level (*no soft classification instances, no outside the domain instances*) and at the document-level (*no digits, no case sensitivity*). The influence of different pre-processing techniques is summarized in Table 3 and Table 4 for "2008–2012" and "2012–2013" datasets, respectively.

- **No pre-processing:** i.e. all texts remained untouched.
- **No outside domain instances.** The texts, whose authors are the chairpersons of the parliamentary sessions, commonly are very technical (related with the giving a voice to the speakers, controlling voting procedures, etc.) thus do not reveal any political views at all. We assume that elimination of such instances that actually are outside the ideology-based domain should improve our classification results. Besides, this pre-processing technique decreased the number of tokens by 19.67% and by 19.26% for "2008–2012" and "2012–2013" datasets, respectively, compared to the unprocessed text (see Table 1 – Table 4).
- **No soft classification instances.** The texts whose authors are the disloyal parliament members –in particular, members who changed the party group at least once during the same parliamentary term–can be hardy attached to one specific class. We assume that elimination of all such soft instances will transform classes into more stable; hence, machine learning method will create more robust model which in turn should positively impact the classification accuracy. Besides, this pre-processing technique decreased the number of tokens by 13.77% and only by 1.69% for "2008–2012" and "2012–2013" datasets, respectively, compared to the datasets after no

Table 3. Pre-processed dataset "2008–2012" statistics (see Table 1 for unprocessed)

Dataset-level pre-processing:				
	Numb. of classes	Numb. of instances	Numb. of tokens	Numb. of distinct tokens
No outside domain inst.	12	71,085	6,392,829	184,426
No soft & no outside domain inst.	7	61,062	5,512,594	171,275
Document-level pre-processing:				
Words & numbers (bag-of-words)			5,512,594	171,275
Words			5,443,286	169,237
Words in lowercase & numbers			5,512,594	157,436
Words in lowercase			5,443,286	155,379

Table 4. Pre-processed dataset "2012–2013" statistics (see Table 2 for unprocessed).

Dataset-level pre-processing:				
	Numb. of classes	Numb. of instances	Numb. of tokens	Numb. of distinct tokens
No outside domain inst.	8	10,394	911,182	70,073
No soft & no outside domain inst.	8	10,231	895,762	69,317
Document-level pre-processing:				
Words & numbers (bag-of-words)			895,762	69,317
Words			884,194	68,563
Words in lowercase & numbers			895,762	64,661
Words in lowercase			884,194	63,904

outside domain instances pre-processing (see Table 3 and Table 4).

- **No digits.** We assume that digits are not related with the ideology domain, thus elimination of the redundant information should slightly boost the classification results. This pre-processing technique decreased the number of tokens only by 1.26% and by 1.29% for "2008–2012" and "2012–2013" datasets, respectively, compared to the datasets after dataset-level pre-processing (see Table 3 and Table 4).
- **No case sensitivity.** We assume that the orthographic information (words replaced with the lowercase letters) should not have any influence on the classification results.

5.4. Explored feature types

Naïve Bayes Multinomial method which was chosen to solve text classification task has to be applied on some text elements t_k (see eq. 1 and eq. 3), named features. However, the most common feature representation type is a bag-of-words, where tokens t_k are words or numbers. This feature type usually achieves relatively high classification accuracy on English (for topic classification, opinion mining, etc.) and even outperforms the others, therefore is often chosen without any considerations. Consequently, this feature type is also used in the majority of the ideology-based classification experiments for English [19], French [21], etc. Despite that ideology-based text classification task has never been solved for Lithuanian, but topic classification [34] and sentiment classification [32] applied on the forum data and internet comments, respectively, proved that the simple bag-of-words approach is outperformed by other more sophisticated feature types. Since the data (political domain and normative language) used in this research significantly differ from the data used in just mentioned classification tasks it is still not clear which feature type is the best. We could answer this question only after experimental investigation of the following feature types:

- **Bag-of-words (or token unigrams).** Each text is split into tokens –in particular, words and numbers– using whitespaces and punctuation symbols as separators.
- **Token lemmas.** The same tokenization procedure as using the bag-of-words approach, but words are replaced with the appropriate main grammatical form, e.g. *Europos* (Europe, in genitive) would be replaced with *Europa* (Europe, in nominative); *modifikuotas* (modified) would be replaced with *modifikuoti* (to modify). For lemmatizing the text we used Lithuanian part-of-speech tagger and lemmatizer "Lemuoklis" [35, 36]. It should be emphasized that this feature type is strongly recommended for the highly inflective languages, because lemmatization significantly decreases the sparseness of the data. Besides, it decreased the number of distinct tokens by 70.37% and by 67.73% for "2008–2012" and "2012–2013" datasets, respectively, compared to bag-of-words representation.
- **Token n-grams.** Using sliding window of size n, the text is split into parts containing collocations of the consecutive tokens. E.g. if using token bigrams phrase *Lietuvos Respublikos parlamento nariai* (parliament members of the Lithuanian Republic) would be split into these pairs of tokens: *Lietuvos Respublikos, Respublikos parlamento, parlamento nariai*; if using token trigrams – into these triplets of tokens: *Lietuvos Respublikos parlamento, Respublikos parlamento nariai*. As higher order token n-gram as rarer it occurs in the text; therefore higher order token n-grams are often used only as interpolation (complement) to the lower order n-grams (usually to the token unigrams).
- **Document-level character n-grams.** Using sliding window of size n the text is split into units containing collocations of the consecutive characters (besides all punctuation marks are removed beforehand, but whitespaces are treated as characters). E.g. if using character 4-gram, *žemės ūkis* (agriculture)

would be split into these n-grams: *žemė, emės, mės_, ės_ū, s_ūk, _ūki, ūkis*. The probability of finding character n-gram is much higher, compared with the token n-grams: e.g., nouns *valdymas* (management), *valdžia* (authority), *pavaldumas* (subordination); verb *valdyti* (to manage); phrasal verbs *įvaldyti* (to master), *suvaldyti* (to suppress); or even compound nouns as *savivaldybė* (self-government), share the same 4 characters *vald*. It is very important that using character n-grams the relations between consecutive words are still considered. Moreover, Hartmann et al. [16] proved that document-level character n-grams used, namely, with Naïve Bayes method are better choice than token n-grams.

- **Words of indicated part-of-speeches.** All the words except of the indicated part-of-speeches are removed from the text. E.g. if indicated part-of-speech is a noun, only *argumentus* (arguments) would remain in the sentence *Dėkoju už argumentus* (Thanks for the arguments). This feature type is not common in the text classification tasks, but it was proved to be effective in the ideology-based classification task [19] for English.

6. Experiments and Results

The results reported in Fig. 1 – Fig. 6 are obtained with the Naïve Bayes Multinomial classification method (see Section 5.2) and are based on 10-fold cross-validation. The figures represent accuracy (eq. 4) and f-score (eq. 5) values over random (eq. 6) and majority (eq. 7) baselines.

$$accuracy = \frac{N_{correct}}{N_{all}}, \quad (4)$$

where $N_{correct}$ is a number of correctly classified instances; N_{all} is a number of all instances.

$$f - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (5)$$

where $precision = tp/(tp + fp)$ and $recall = tp/(tp + fn)$, tp is a number of correctly classified instances of c_i (c_i classified as c_i: correct result); fp is a number of incorrectly classified instances with c_i (c_j was incorrectly classified as c_i: unexpected result); fn is a number of incorrectly classified instances with c_j (c_i was incorrectly classified as c_j : missing result).

$$random\ baseline = \sum_i P(c_i)^2, \quad (6)$$

where $c_i \in C$ and $P(c_i)$ is calculated with eq. 2.

$$majority\ baseline = max(P(c_i)), \quad (7)$$

$c_i \in C$ and $P(c_i)$ is calculated with eq. 2.

Our experiments involved exploration of the different pre-processing techniques (Fig. 1 – Fig. 4) and feature types (see Fig. 5 and Fig. 6) on "2008–2012" and "2012–2013" datasets, respectively. All experiments were performed in a greedy manner: i.e. the best discovered technique was used in the following experiments.

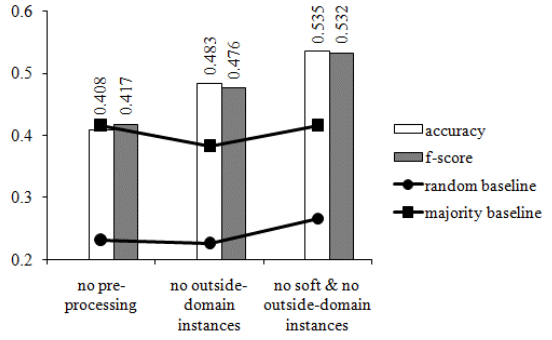

Figure 1. Dataset-level pre-processing on "2008–2012" with *bag-of-words*

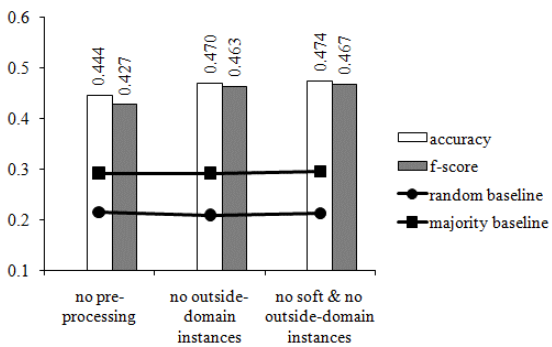

Figure 2. Dataset-level pre-processing on "2012–2013" with *bag-of-words*

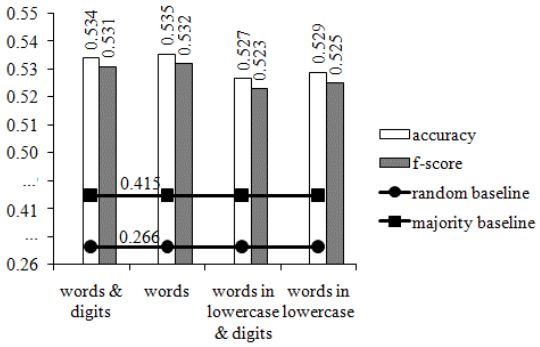

Figure 3. Document-level pre-processing on "2008–2012" with *no soft & no outside domain instances* and *bag-of-words*

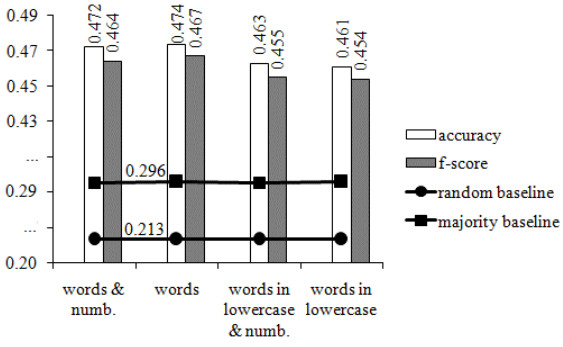

Figure 4. Document-level pre-processing on "2012–2013" with *no soft & no outside domain instances* and *bag-of-words*

To measure whether the differences between obtained results are statistically significant we performed McNemar's test [37] with one degree of freedom.

All differences between the results in "2008–2012" dataset (see Fig. 1) and between no *pre-processing* and *no outside domain instances* pre-processing techniques in "2012–2013" dataset are statistically significant ($p \ll 0.05$), except for the results between *no outside domain* and *no soft & no outside domain instances* ($p = 0.243$) in "2012–2013" dataset (see Fig. 2). Despite that the positive impact of *no soft & no outside domain instances* pre-processing on "2012–2013" dataset is only marginal; however, this technique in general is still the best obtained dataset-level pre-processing technique.

The differences between the best (*words*) and the worst (*words in lowercase & numbers*) document-level pre-processing techniques are statistically significant on "2008–2012" dataset ($p = 0.02$), but not on "2012–2013" dataset ($p = 0.29$) (see Fig. 3 and Fig. 4), respectively. Despite that differences between the best (*words*) and the second best (*words & numbers*) document-level pre-processing techniques for both datasets are not statistically significant; we consider that in general the best document-level pre-processing technique is *words*.

The best feature type (*bag-of-words & token bigrams*) beats the second best feature type (*bag-of-words* or *bag-of-words & token bigrams & token trigrams*) on "2008–2012" dataset and the differences are statistically significant ($p = 0.01$) (see Fig. 5[5]. The best feature type (the same as in "2008–2012" dataset) outperforms the second best feature type (*bag-of-words*) on "2012–2013" dataset, but the differences are not statistically significant ($p = 0.14$) (see Fig. 6).

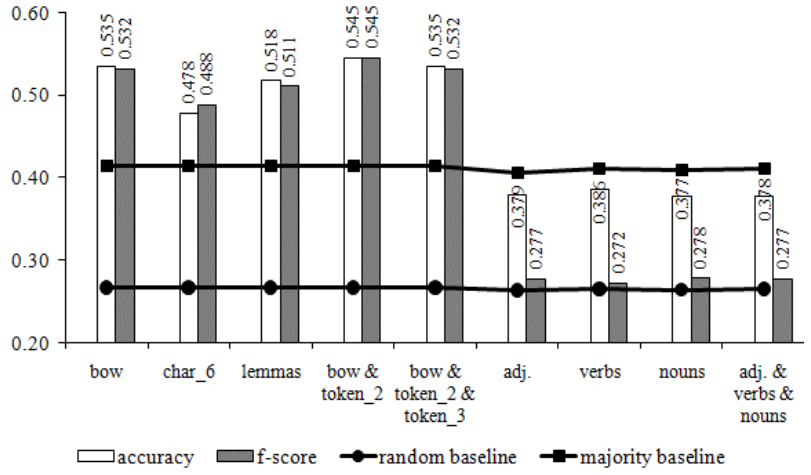

Figure 5. Explored features on "2008–2012" (*bow* stands for bag-of-words (only words), *char*_6 – character 6-grams, *token_n* – token n-grams) using *no soft and no outside domain instances* and *words*

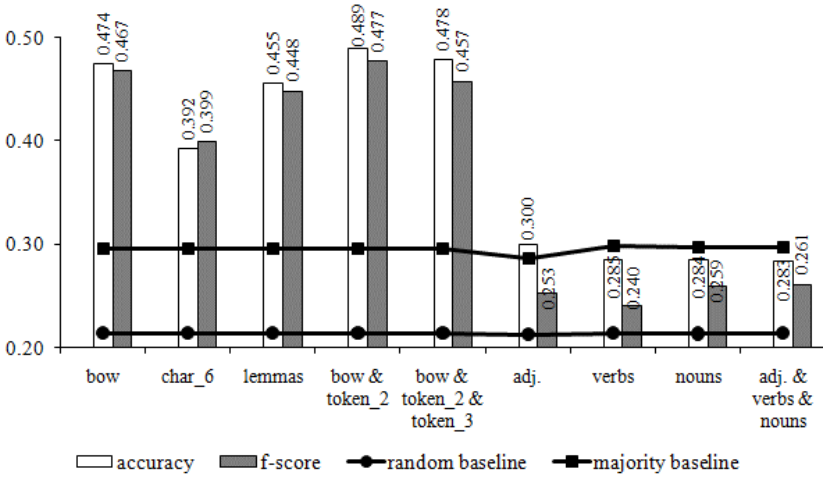

Figure 6. Explored features on "2012–2013" using *no soft and no outside domain instances* as the dataset-level pre-processing technique and *words* as the document-level pre-processing technique

[5] Despite given features, we experimentally investigated character 4-grams, 5-grams and 7-grams, but 6-grams gave marginally the best results.

Table 5. Confusion matrix (in percentage points %) for "2008–2012" dataset. Each row shows the percentage distribution of the classified instances over all classes

		Predicted class						
		HU-LCD	LSD	O&J	LM	LP	MPG	LCU
Real class	HU-LCD	65.24	9.60	12.56	3.83	6.89	1.60	0.29
	LSD	24.28	51.03	9.16	3.45	10.72	1.20	0.16
	O&J	27.24	12.69	48.67	2.40	7.43	1.48	0.09
	LM	34.93	11.25	7.49	36.59	8.19	1.28	0.27
	LP	28.60	14.05	9.17	2.51	44.52	1.02	0.14
	MGP	35.28	11.27	8.55	3.30	7.58	33.72	0.29
	LCU	28.24	4.71	6.47	3.53	4.71	0.00	52.35

Table 6. Confusion matrix (in percentage points %) for "2012–2013" dataset

		Predicted class						
		HU-LCD	LSD	O&J	LM	LP	MPG	CR
Real class	HU-LCD	65.22	19.88	3.02	2.00	8.05	0.47	1.35
	LSD	25.82	58.05	2.98	1.59	10.25	0.50	0.83
	O&J	27.90	26.18	31.12	1.81	11.68	0.10	1.21
	LM	36.23	20.00	2.99	30.65	8.57	0.26	1.30
	LP	29.47	28.13	3.31	1.76	36.04	0.05	1.19
	MGP	29.44	19.63	2.34	6.54	9.81	27.57	4.67
	CR	42.07	17.78	2.10	4.40	8.60	0.19	24.86

7. Discussion

The best achieved accuracy and f-score is 0.545 (which beats random and majority baselines by 0.279 and 0.13, respectively) on "2008–2012" dataset. The best achieved accuracy is 0.489 (which beats random and majority baselines by 0.276 and 0.193, respectively) and the best f-score is 0.477 on "2012–2013" dataset. Since these classification results are still rather low, we made an error analysis (see Table 5 and Table 6).

Despite that the majority of confusions can be explained on the basis of the ideological or pragmatic similarities between attitudes of the confused party groups, we will not go into those details. Especially that the purpose of this research was absolutely different: i.e. using state-of-the art machine learning techniques to achieve as higher classification accuracy as possible on this specific political domain and normative Lithuanian texts. However, it is important to notice that the same confusions (see Table 5 and Table 6) are not accidental: i.e. the majority of the same confusion patterns (e.g. the majority of LSD or LM confusions are with HU-LCD; the majority of O&J or LP confusions are with HU-LCD then with LSD; etc.) stand for both datasets.

We also performed one more control experiment (using the best previously determined pre-processing techniques and feature type) to see if ideology still holds if classifier is trained on one data-set ("2008–2012") and tested on another ("2012–2013") leaving only the instances of those 5 classes which exist in both datasets. We made an assumption that the ideology of each party group should remain stable through the parliamentary terms. Despite that we obtained rather low 0.283 accuracy (still slightly surpassing random baseline, but descending majority baseline by 0.04) and 0.274 f-score for this experiment. The same confusion patterns as in Table 5 and Table 6 still remained (see Table 7). Consequently it allows us to claim that the results are consistent and the ideology can be captured.

The analysis of the most informative words revealed that the discussed topics differ in the different parliamentary terms. E.g. in the "2008–2012" dataset the most informative words are *energetika* (energetics), *šeima* (family), etc., whereas the most informative words in "2012–2013" dataset are *žmogaus gyvybė* (human life), *abortų draudimas* (abortion prohibition), *genetiškai modifikuoti* (genetically modified). Since the most informative words differ, the discussed topics differ also: i.e. this fact is the most likely to result the drop of accuracy in the control experiment.

As can be seen from the results in Fig. 1 and Fig. 2, our assumption that elimination of the outside domain instances will increase the classification results was confirmed. The assumption that the elimination of soft instances will increase the classification results was confirmed on "2008–2012" dataset too (see Fig. 1), whereas positive impact of this pre-processing techni-

Table 7. Confusion matrix summarizing the results (in percentage points %) of the control experiment

		Predicted class				
		HU-LCD	LSD	O&J	LM	LP
Real class	HU-LCD	46.83	19.19	20.94	3.46	9.58
	LSD	56.03	20.86	9.45	4.83	8.83
	O&J	44.01	13.80	27.19	2.62	12.39
	LM	38.05	23.12	17.01	13.38	8.44
	LP	51.71	15.62	9.88	2.69	20.11

que is not that obvious on "2012–2013" dataset (see Fig. 2). This happened due to the fact that only a very small part of the parliamentarians of "2012–2013" parliamentary term changed their party group (see Section 5.3).

We assumed that the digit elimination should increase the classification accuracy and it was experimentally proved. Whereas the assumption that orthographic information should not have any influence on the classification results was rejected. It seems that we lost important information about the word position in the sentence by transforming all words into lowercase letters. This might happened due to the fact that missing the word order in the sentence we also loose information about the notional shade (things that are emphasized) of the sentence (see Section 3). All assumptions about document-level pre-processing techniques (see Fig. 3 and Fig. 4) are not very robust, because changes are too little to make a significant influence.

As we assumed, bag-of-words approach is not the best feature type for our task (see Fig. 5 and Fig. 6). Document level character 6-grams underperform all the rest token level feature types, except for indicated part-of-speeches. It seems that the advantage to capture relations between consecutive words is suppressed by the loss of important information about the Lithuanian suffixes, prefixes and compound words. The results obtained with the identified part-of-speeches are very low (in most cases are even below majority baseline), therefore this feature type is not sufficient to capture the ideology. Especially that even not all of the words belonging to those part-of-speeches are recognized by the Lithuanian lemmatizer. Surprisingly, but token lemmas are outperformed by the bag-of-words approach. This might be explained by the same fact that lemmatizer is not very accurate; moreover, lemmatized text losses morphological information (information about the word endings) which seems to be essential. E.g. *žemė* (land) in nominative and *ūkis* (farm) is not the same as *žemės ūkis* (agriculture), where *žemės* (land) is in genitive. Probably for all of these reasons, interpolation of bag-of-words and token bigrams is the best feature type.

8. Conclusions

In this paper we were solving an ideology-based (expressed with the party membership) text classification task for the morphologically rich Lithuanian language. We have experimentally proved that the most effective dataset-level pre-processing technique is elimination of soft and outside the domain instances; document-level pre-processing technique leaving the words (and eliminating digits), but not touching their orthographic information and using bag-of-words interpolation with token bigrams is the most accurate feature type. The best achieved accuracy is 0.545 (which beats random and majority baselines by 0.279 and 0.13, respectively) on "2008–2012" dataset and 0.489 (which beats random and majority baselines by 0.276 and 0.193, respectively) on "2012–2013" dataset. Despite that obtained results are not very high, our ideology-based classification task was more complicated if compared to the similar tasks solved for English: i.e. our datasets contained more classes (if compared with e.g. the European Parliament of the US Senate), moreover, these classes were not such stable (because of the relatively high migration of parliamentarians between the party groups).

The ideology-based text classification task has never been solved for the Lithuanian before; therefore obtained results are interesting in both computational linguistics and political point of view. Besides, obtained results should be promising even for the other languages having similar properties as Lithuanian.

Future research includes detailed error analysis and exploration of the other classification approaches (e.g. rule-based, clustering, etc.) that might increase classification results. Besides, the experiments were performed only with two parliamentary terms, thus, analysis of all seven could allow us to make more robust generalizations and even explore ideology changes, if any, over time. Besides, for the comparison purposes would be useful to measure manual classification accuracy achieved by the human-experts.

9. Acknowledgments

This research was is funded by European Union Structural Funds project "Postdoctoral Fellowship implementation in Lithuania" (No. VP1-3.1-ŠMM-01).

References

[1] **P. J. Hayes, S. P. Weinstein.** CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories. *Proceedings of the 2nd Conference on Innovative Applications of Artificial Intelligence (IAAI-90)*, 1990, pp. 49-64.

[2] **S. B. Kotsiantis.** Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 2007, Vol. 31, 249-268.

[3] **F. Sebastiani.** Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 2002, Vol. 34, 1-47.

[4] **T. Joachims.** Text categorization with support vector machines: learning with many relevant features. *Proceedings of ECML-98, 10th European Conference on Machine Learning*, 1998, pp. 137-142.

[5] **S. Dumais, J. Platt, D. Heckerman, M. Sahami.** Inductive learning algorithms and representations for text categorization. *Proceedings of the 7th international conference on Information and knowledge management*, 1998, pp. 148-155.

[6] **E. Gabrilovich, S. Markovitch.** Text Categorization with Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5. *Proceedings of the 21st International Conference on Machine Learning*, 2004, pp. 321-328.

[7] **A. Pak, P. Paroubek.** Twitter for Sentiment Analysis: When Language Resources are Not Available. *Proceedings of Database and Expert Systems Applications (DEXA 2011)*, 2011, pp. 111-115.

[8] **M. Radovanović, M. Ivanović.** Document representations for classification of short web-page descriptions. *Proceedings of the 8th international conference on Data Warehousing and Knowledge Discovery (DaWaK'06)*, 2006, 544-553.

[9] **T. Gaustad, G. Bouma.** Accurate Stemming of Dutch for Text Classification. *Language and Computers*, 2002, 104-117.

[10] **A. Wahbeh, M. Al-Kabi, Q. A. Al-Radaideh, E. M. Al-Shawakfa, I. Alsmadi.** The Effect of Stemming on Arabic Text Classification: An Empirical Study. *International Journal of Information Retrieval Research*, 2011, Vol. 1, No. 3, 54-70.

[11] **E. Leopold, J. Kindermann.** Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? *Machine Learning*, Vol. 46, No.1-3, 432-444.

[12] **B. Pang, L. Lee, S. Vaithyanathan.** Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, 2002, pp. 79-86.

[13] **K. Dave, S. Lawrence, D. M. Pennock.** Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proceedings of the 12th international conference on World Wide Web (WWW'03)*, 2003, pp. 519-528.

[14] **H. Cui, V. Mittal, M. Datar.** Comparative experiments on sentiment classification for online product reviews. *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-2006)*, 2006, pp. 1265-1270.

[15] **V. Nastase, J. Sayyad, M. F. Caropreso.** Using Dependency Relations for Text Classification. *Technical Report TR-2007-12, University of Ottawa, Canada*, 2007.

[16] **T. Hartmann, S. Klenk, A. Burkovski, G. Heidemann.** Sentiment Detection with Character n-Grams. *Proceedings of the 7th International Conference on Data Mining (DMIN'11)*, 2011, pp. 364-368.

[17] **F. Peng, D. Schuurmans, S. Wang.** Language and task independent text categorization with simple language models. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL'03)*, 2003, Vol. 1, pp. 110-117.

[18] **I. Kanaris, K. Kanaris, I. Houvardas, E. Stamatatos.** Words vs. Character N-grams for Anti-spam Filtering. *International Journal on Artificial Intelligence Tools*, 2007, Vol. 16, No. 6, 1047-1067.

[19] **D. Diermeier, J. F. Godbout, B. Yu, S. Kaufmann.** Language and Ideology in Congress. *British Journal of Political Science*, 2011, Vol. 42, No. 1, 31-55.

[20] **B. Yu, S. Kaufmann, D. Diermeier.** Classifying party affiliation from political speech. *Journal of Information Technology in Politics*, 2008, Vol. 5, No. 1, 33-48.

[21] **G. Hirst, Y. Riabinin, J. Graham.** Party status as a confound in the automatic classification of political speech by ideology. *Proceedings of the 10th International Conference on Statistical Analysis of Textual Data (JADT 2010)*, 2010, pp. 731-742.

[22] **M. Jiang, S. Argamon.** Political Leaning Categorization by Exploring Subjectivities in Political Blogs. *Proceedings of 4th International Conference on Data Mining (DMIN 2008)*, 2008, pp. 647-653.

[23] **B. Hoyland, J. F. Godbout.** Lost in Translation? Predicting Party Group Affiliation form European Parliament Debates. Unpublished Manuscript, 2008.

[24] **M. Tamošaitis.** Historiography of Lithuanian political parties and ideological streams (up to 1940) (Lietuvių politinių partijų ir ideologinių srovių (iki 1940 m.) istoriografija). *Istorija*, 2011, 84 (in Lithuanian).

[25] **I. Savickienė, V. Kempe, P. J. Brooks.** Acquisition of gender agreement in Lithuanian: exploring the effect of diminutive usage in an elicited production task. *Journal of Child Language*, 2009, Vol. 36, 477-494.

[26] **G. Naktinienė, J. Paulauskas, R. Petrokienė, V. Vitkauskas, J. Zabarskaitė, editors.** Lithuanian language dictionary (vol. 1-20, 1941–2002): electronic version (Lietuvių kalbos žodynas (t. 1-20, 1941–2002): elektroninis variantas), *Institute of the Lithuanian Language, Vilnius, Lithuania*, 2005, <http://www.lkz.lt/dzl.php> (in Lithuanian).

[27] **K. Ulvydas, editor.** Phonetics and morphology (noun, adjective, numeral, pronoun (Fonetika ir morfologija (daiktavardis, būdvardis, skaitvardis, įvardis)), 1, *Mintis, Vilnius*, 1965 (in Lithuanian).

[28] **D. D. Lewis, W. A. Gale.** A sequential algorithm for training text classifiers. *Proceedings of 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR-94)*, 1994, pp. 3-12.

[29] **J. R. Quinlan.** Induction of Decision Trees. *Machine Learning*, 1986, Vol. 1, No. 1, 81-106.

[30] **C. Cortes, V. Vapnik.** Support-vector networks. *Machine Learning*, 1995, Vol. 20, 273-297.

[31] **W. Daelemans, A. van den Bosch.** Memory-Based Language Processing. *Cambridge University Press*, 2005.

[32] **J. Kapočiūtė-Dzikienė, A. Krupavičius, T. Krilavičius.** A Comparison of Approaches for Sentiment Clas-

sification on Lithuanian Internet Comments. *Proceedings of ACL – the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing (ACL-BSNLP)*, 2013, pp. 2-11.

[33] **M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten.** The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 2009, Vol. 11, No. 1, 10-18.

[34] **J. Kapočiūtė-Dzikienė, F. Vaassen, W. Daelemans, A. Krupavičius.** Improving topic classification for highly inflective languages. *Proceedings of 24th International Conference on Computational Linguistics (COLING 2012)*, 2012, pp. 1393-1410.

[35] **V. Zinkevičius.** Morphological analysis with Lemuoklis (Lemuoklis – morfologinei analizei). *Gudaitis, L. (ed.) Darbai ir Dienos*, 2000, Vol. 24, 246-273 (in Lithuanian).

[36] **V. Daudaravičius, E. Rimkutė, A. Utka.** Morphological annotation of the Lithuanian corpus. *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies (ACL'07)*, 2007, pp. 94-99.

[37] **Q. M. McNemar.** Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 1947, Vol. 12, No. 2, 153-157.

Received December 2013.