Improving Topic Classification for Highly Inflective Languages

Jurgita KAPOČIŪTĖ-DZIKIENĖ[1] *Frederik VAASSEN*[2]
Walter DAELEMANS[2] *Algis KRUPAVIČIUS*[1]

(1) KAUNAS UNIVERSITY OF TECHNOLOGY, K. Donelaičio 73, LT-44249 Kaunas, Lithuania
(2) COMPUTATIONAL LINGUISTICS & PSYCHOLINGUISTICS RESEARCH CENTER, Lange Winkelstraat 40-42, B-2000 Antwerp, Belgium
Jurgita.Kapociute-Dzikiene@ktu.lt, Frederik.Vaassen@ua.ac.be, Walter.Daelemans@ua.ac.be, pvai@ktu.lt

ABSTRACT

Despite the existence of many effective methods to solve topic classification tasks for such widely used languages as English, there is no clear answer whether these methods are suitable for languages that are substantially different. We attempt to solve a topic classification task for Lithuanian, a relatively resource-scarce language that is highly inflective, has a rich vocabulary, and a complex word derivation system. We show that classification performance is significantly higher when the inflective character of the language is taken into account by using character n-grams as opposed to the more common bag-of-words approach. These results are not only promising for Lithuanian, but also for other languages with similar properties. We show that the performance of classifiers based on character n-grams even surpasses that of classifiers built on stemmed or lemmatized text. This indicates that topic classification is possible even for languages for which automatic grammatical tools are not available.

TITLE AND ABSTRACT IN LITHUANIAN

Klasifikavimo į temas gerinimas stipriai kaitomoms kalboms

Nepaisant to, jog tokioms plačiai naudojamoms kalboms kaip anglų yra sukurta daug efektyvių metodų, sprendžiančių klasifikavimo į temas uždavinius, neaišku ar šie metodai yra tinkami visiškai skirtingoms kalboms. Siekiame išspręsti klasifikavimo į temas uždavinį gana mažai išteklių šioje srityje turinčiai lietuvių kalbai, kuri yra stipriai kaitoma, turi turtingą žodyną, sudėtingą žodžių darybos sistemą. Pademonstruosime, kad galima pasiekti ženkliai geresnius klasifikavimo rezultatus, kuomet atsižvelgiama į kaitomą kalbos pobūdį: naudojamos simbolių n-gmamos vietoj labiau įprasto žodžių rinkinio. Gauti rezultatai perspektyvūs ne tik lietuvių kalbai, bet taip pat ir kitoms, panašiomis savybėmis pasižyminčioms, kalboms. Pademonstruosime, kad klasifikatorių, naudojančių simbolių n-gramas veikimas netgi efektyvesnis, palyginus su klasifikatoriais, naudojančiais į žodžių kamienus arba lemas transformuotą tekstą. O tai reiškia, kad šį klasifikavimo į temas metodą galima taikyti netgi toms kalboms, kurios neturi specializuotų automatinių gramatinių įrankių.

KEYWORDS : Character n-grams, topic classification, Lithuanian.
KEYWORDS IN LITHUANIAN : Simbolių n-gramos, klasifikavimas į temas, lietuvių kalba.

Proceedings of COLING 2012: Technical Papers, pages 1393–1410,
COLING 2012, Mumbai, December 2012.

1 Introduction

With the exponential growth in the number of electronic documents, automatic text categorization has become one of the key techniques to help control and organize the constant influx of information. The ever-increasing amount of articles, e-mails, comments, and other types of texts leaves no opportunity for manual classification.

Initially, the main focus of text categorization research was on proposing and applying supervised text classification methods on different benchmark data collections, such as Reuters-21578 (Lewis, 1992; Lewis et al., 2004), 20 Newsgroups (Lang, 1995), Ohsumed (Hersh et al., 1994; Lewis et al., 2004), etc. When text classification methods achieved sufficient accuracy, research topics started to diversify, focusing instead on integrating different sources of information (semantic information, world knowledge), developing increasingly autonomous classification methods (semi-supervised or unsupervised classification), applying the approach to diverse classification tasks (genre classification, author classification), etc.

Whereas currently proposed methods solve text classification tasks very effectively, they often require information that is not immediately present in the text. Using part-of-speech information, for instance, requires the availability of accurate part-of-speech taggers. The same can be said about world knowledge, which requires the existence of ontologies or common sense databases. For English and other major languages, a wide range of annotated corpora, grammatical tools and databases are available, but this is not the case for other, more resource-scarce languages. Furthermore, most research currently focuses on a limited selection of languages, and there is no guarantee that the techniques that work best to classify English texts will also be most effective for languages that are substantially different.

In this paper, we investigate whether the widely accepted bag-of-words approach that seems to work well for topic classification in English is also the best method for topic classification in Lithuanian, which is a highly inflectional language. We also assess whether it is possible to perform topic classification on Lithuanian without resorting to external resources such as extra grammatical annotation layers or automatic grammatical parsers, since Lithuanian has a limited number of these resources. We posit that our findings can also be applied to other, similar languages (according to such properties as inflectional morphology, complexity of word derivation, etc.) such as Latvian (the only other living Baltic language) or Slavic languages.

Section 2 contains an overview of related work. In Section 3, we describe the Lithuanian language and its properties. We outline the methodology of our topic classification experiments in Section 4 and present the results in Section 5. We discuss these results in Section 6 and evaluate their implications for text classification procedures in general, and for inflectionally rich languages specifically.

2 Related work

In this paper we focus on supervised machine learning methods (for review see Kotsiantis, 2007) applied to a text categorization task (for review see Sebastiani, 2002).

The most important features when performing topic classifications are usually those lexical features that clearly refer to topic-specific terminology. Function words are less important for this task, and so is most grammatical information. Therefore, prior to topic classification, the

documents are usually pre-processed in such a way that this topic-specific lexical information is easily accessible. This may involve the removal of stop words, function words; stemming or lemmatization; spelling normalization; word segmentation; etc. Stemming and lemmatization are advisable for highly inflective languages, spelling normalization for languages having a highly variable orthography (e.g. Arabic), word segmentation is demanded for languages having concatenative compounding (e.g. Swedish or German) and inevitable for the languages that have no whitespace between words (e.g. some Asian languages such as Chinese or Japanese). However, the gain one can expect from pre-processing steps like stemming or lemmatization seems to be strongly dependent on the language of the dataset. Some comparative experiments have revealed that stemming improves the classification performance of various machine learning algorithms on English texts (Radovanovic & Ivanovic, 2006). However, in Gaustad & Bouma (2002)'s experiments on Dutch data, it had no influence on classification results. Stemming can even adversely affect accuracy, as shown by Wahbeh et al. (2011) on Arabic data. As for lemmatization, Leopold & Kindermann (2002) report that it led to no significant classification improvements on German, sometimes even yielding worse results.

Regardless of these pre-processing steps, the most popular feature representation for topic classification seems to be the bag-of-words approach. In some cases, it may be interesting to add word bigrams, on the condition that these are not entirely redundant. Indeed, adding bigrams does not usually help classification performance (on the contrary, performance often drops), since the feature space becomes even more sparse. Boulis & Ostendorf (2005), however, showed that carefully selected token bigrams can offer improvements over the bag-of-words approach. They propose a measure to evaluate the redundancy of token bigrams based only on their related words. Tan et al. (2002) use a lexical analyzer to determine and include the most important token bigrams and demonstrate a positive influence on classification results. Nastase et al. (2007) use syntactically related word pairs (verb + each of its arguments, noun + each of its modifiers) instead of token n-grams and report an improvement over the pure bag-of-words representation.

Stop word removal, stemming, lemmatization and lexical analysis are language dependent, and are therefore necessarily based on external resources –lists of stop words, grammar tools or dictionaries– resources that resource-scarce languages may not always have. It is however also possible to use character level n-grams. Character n-grams can often capture the language-dependent elements without having to resort to external stemmers, lemmatizers or word segmentation tools. Character n-grams will highlight highly relevant word segments, and since each string is decomposed into small parts, any error can affect only a limited number of these parts, which makes character n-grams ideal to use with informal texts containing many typographical errors.

Peng et al. (2003a, 2003b) skipped language dependent pre-processing and feature selection stages and demonstrated that a language modelling approach on character n-grams gives superior classification results for English and competitive results for Chinese and Japanese. For English, the most accurate performance was achieved with n-grams of 3 characters or more, peaking at 6-grams with Witten-Bells smoothing. Bigrams (or higher order) with different smoothing techniques worked best for Chinese, whereas 6-grams (or more) with absolute or Written-Bell smoothing worked best for Japanese. Wei et al. (2008)'s experimented on Chinese texts, showing good performance for combinations of character unigrams and bigrams, or combinations of unigram, bigrams and trigrams. Peng & Schuurmans (2003) demonstrated that a chain-

augmented Naïve Bayes classifier can work equally well on character n-grams as on bag-of-words for Chinese, Japanese, English and Greek.

Sometimes character level n-grams are applied on pre-processed texts. Khreisat (2006) applied character trigrams on Arabic texts after stop word removal and spelling normalization. A similar approach was applied to Farsi by Bina et al. (2008) who report that 4-grams give the best results.

Unfortunately, we cannot give any examples of topic classification experiments for Lithuanian, since no automatic text categorization tasks have yet been described for this language. Consequently, this paper will be the first attempt at finding a good method to perform topic classification on Lithuanian text, taking into account language-specific characteristics, while keeping the use of external information sources to a minimum.

3 The Lithuanian language

Lithuanian is considered one of the most archaic and conservative living Indo-European languages (Gimbutas, 1963) and it has retained a lot of cognates to many words found in such languages as Sanskrit or Latin.

Lithuanian has a rich vocabulary and word derivation system. The Academic Dictionary of Lithuanian (Naktinienė et al., 2005) has more than 0.6 million headwords (e.g. Oxford English Dictionary has about 0.3 million headwords, Deutsches Wörterbuch – the largest lexicon of the German language – has around 0.33 million). As an example of Lithuanian's rich vocabulary, in the Academic Dictionary of Lithuanian, there are more than 1300 synonyms for the Lithuanian word "eiti" (*to go*) (Piročkinas, 2012) – the majority of which are derived from onomatopeias – and for some of these synonyms it is impossible to find equivalents in other languages. Lithuanian has 25 prefixes for nouns and 78 suffixes for diminutives and hypocoristic words (Ulvydas, 1965). Diminutives and hypocoristic words (which are especially frequent in fiction and spoken language) very often have two/three suffixes, and in rare cases the number of suffixes can go as high as six. E.g. the word "sesuo" (*sister*) can be found in texts as "sesutė", "sesužėlė", "seserytė", etc.

Of all living Indo-European languages, Lithuanian has the richest inflectional morphology, making it more complex than even Latvian or Slavic languages (Savickienė et al., 2009). Different parts-of-speech of a word are expressed through inflections and through different endings. As an example of Lithuanian's inflectional complexity, one can calculate all the possible word forms for the Lithuanian word "aukštas" and its English translation, "*high*" (see FIGURE 1).

In Lithuanian, nominal words (nouns, adjectives, numerals, pronouns, and participles) are inflected by 7 (+2) cases, 2 (+1) genders, and 2 numbers. There are 7 "common" cases, and 2 additional forms of locative that are still found in spoken language and idiomatic use. Nouns have 5 declensions (12 inflection paradigms), adjectives have 3 (9 inflection paradigms). E.g. the noun "vanduo" (*water*) of a masculine gender can be found in Lithuanian texts written as "vanduo", "vandens", "vandeniui", "vandenį", "vandeniu", "vandenimi", "vandenyje", "vandenie", "vandenys", "vandenų", "vandenims", "vandenis", "vandenimis", "vandenyse". The word "vanduo" is an example of one of the inflection paradigms, where the part "vand" (i.e. the root of word) always remains stable. Nominal words usually have 2 grammatical genders, except for adjectives, which have an additional neuter gender that is not declined.

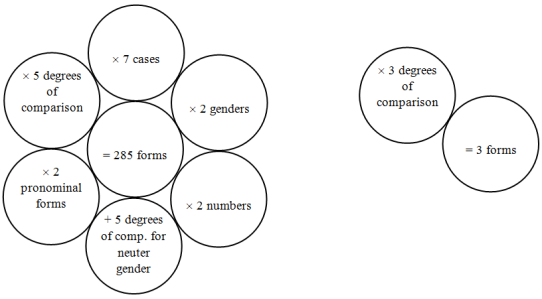

FIGURE 1 – The adjective "aukštas" (*high*) has 285 different forms in Lithuanian (a); and 3 forms in English (b).

Lithuanian also has pronominal forms of adjectives, ordinal numerals, pronouns and active forms of participles. Pronominal forms are unknown to English, German, French, and to hundreds of other languages (they are however present in Russian, for instance). They are used to highlight an object and its properties from other similar objects. Pronominal forms are composed by appending the anaphoric pronoun to the ending of the word and composing more complex dissyllabic endings. E.g., the ordinal number "pirmas" (*the first*) has a simple ending indicating masculine gender in singular number, namely "as". The pronominal form, "pirmasis", is formed by adding an extra ending, "is". Both endings can be inflected further, which can even lead to trisyllabic endings, as is the case of the plural locative "pirmuosiuose", which has an ending made up of three elements: "uos", "iuos" and "e". Despite these extra endings, the root "pirm" is retained.

Adjectives, adverbs, and some forms of verbs have 3 (+2) degrees of comparison, where two of them are usually used only in fiction and spoken language and have no equivalents in English. Degrees of comparison in Lithuanian are expressed by different endings and suffixes, e.g. singular masculine gender noun "gražus" (*beautiful*), "gražesnis" (*more beautiful*), "gražiausias" (*the most beautiful*), where "graž" remains stable.

The most ancient Indo-European languages had 2 or 3 degrees of comparison for demonstratives (pronouns), but in most existing languages only 2 (sometimes 1) are left. Meanwhile, Lithuanian retained 3 degrees of comparison: describing objects located close to the speaker; objects located not close to the speaker, but close to the listener; objects located far from both, i.e. the speaker and the listener. All these words are also inflected, but are too short to retain stable parts longer than one symbol. E.g. the pronoun "šis" (*this/these*) can be found as "šio", "šiam", "šį", "šiuo", "šiame", "šie", "šių", "šiems", "šiuos", "šiais", "šiuose".

Lithuanian language also has compound nouns, consisting of two or even three roots. E.g. "saulėtekis" (*sunrise*) ("saulė", *sun* + "tekėti", *to rise*), "sienlaikraštis" (*wall newspaper*) ("siena", *wall* + "laikas", *time* + "raštas", *writing*), etc. But only the last of the compound words can change its inflection form. E.g. "sienlaikraštis", "sienlaikraščio", "sienlaikraščiui", etc. The same can be said of some fixed combinations of words, e.g. "žemės ūkis" (*agriculture*), "saulės smūgis" (*sunstroke*), where the first word is stable and the second is inflected ("žemės ūkio",

"žemės ūkiui", "žemės ūkį", etc.), whereas both words can be inflected when they occur separately.

Verbs have 3 conjugations, and are inflected by 4 tenses, 3 persons, 2 numbers, and 3 moods. Some conjugated verb forms are also reflexive. This can be expressed by adding particle "si" or "s" at the end of word or after extra added prefixes, e.g. "nusiprausti" (particle "si" after prefix "nu"), "praustis" (*to have a wash*) (particle "s" at the end of word). Both words have the same root "praus". Phrasal verbs in Lithuanian have extra prefixes (14 in total), whereas in English they are expressed by two words: i.e. verb + preposition/adverb. E.g. "nubėgti" (*to run away*), "prabėgti" (*to run by*), "įbėgti" (*to run in*), "išbėgti" (*to run out*), all having different prefixes but the same stable root "bėg". Other forms of verbs are non-conjugative (e.g. participles, adverbial participles, verbal adverbs, and some forms of gerund), but can be inflected by tense, case, gender, number, and have active and passive forms. Non-conjugative forms of verbs retain the same root, but have different suffixes and endings in different inflection forms.

4 Methodology

Our goal is to see if we can carry out automatic topic classification despite the complexity of Lithuanian (and, by extension, of other Baltic and Slavic languages). Since for these relatively resource-scarce languages, we do not always have access to freely available dictionaries, annotated datasets or grammatical tools, we also want to avoid using such external resources and use only the information present in the text's surface representation.

Section 4.1 will describe the datasets we used and Section 4.2 will give a formal description of the classification task. In Section 4.3, we will describe which different feature types we experimented with and formulate two hypotheses we would like to test through these experiments. Section 4.4 describes the actual classification process, including optimization of the classifier and evaluation of the models' performance.

4.1 Datasets

All of our experiments were carried out on three different datasets to make sure our findings generalize over different domains.

- "*Lietuvos rytas*". The information for this dataset was crawled from the internet forum of the largest daily newspaper in Lithuania, "Lietuvos rytas" (Lietuvos rytas, 2012). The structure of the forum is hierarchical: it contains a number of topics (root categories), each of which may contain several sub-topics (sub-categories). Within a topic, the discussion is determined by a collection of posted messages. The data was crawled automatically, resulting in a collection of documents, each containing a single forum post, and grouped according to their root category. Root categories (topics) are very general including such areas as "Business", "Politics", "Sports", etc. The "Lietuvos rytas" dataset is thus composed of 8,936 text documents grouped into 11 topics.
- "*Supermamos*". This dataset contains information from the largest internet forum for women in Lithuania (Supermamos, 2012). The structure of this forum is analogous to the "Lietuvos rytas" forum, and was processed similarly. The root categories are more specific compared to "Lietuvos rytas", and concern topics such as "Parenting and Education", "Maternity and Paternity", "Health and Beauty", etc. The created dataset contains 11,353 text documents grouped into 14 topics.

- "*Rinkimų programos*'04". This dataset contains political information, extracted from the programs of major political parties for the Lithuanian parliamentary elections of 2004. The programs were manually processed by a domain expert following classification rules determined in Volkens (2002): each program was split into small phrases and labeled with appropriate categories indicating policy domains, such as "External Relations", "Freedom and Democracy", "Political system", "Economy", etc. The "Rinkimų programos'04" dataset contains 2,388 text documents grouped into 8 topics (7 are related to policy domains and one is neutral).

As we already mentioned, the "Lietuvos rytas" and "Supermamos" datasets contain messages from internet forums, and thus represent spoken Lithuanian language in written form. "Rinkimų programos'04" contains texts extracted from formal documents and represents normative written Lithuanian.

The classification of "Lietuvos rytas" and "Supermamos" is a challenging task, even for a domain expert, because some posts are very general, and attaching them to any of the possible topics is very difficult. Moreover, internet forums have their own specific jargon full of informal words, foreign language insertions, barbarisms, and other expressions that are outside standard language norms. Besides, "Lietuvos rytas" and especially "Supermamos" are full of diminutives and hypocoristic words. Typographical and grammatical errors can also not be avoided. A particularly difficult problem is that informal written language often imitates spoken language. In the case of Lithuanian, some grammatical forms are very often pronounced without the ending vowel (such as "mergyt" → "mergyte" (*lassie*, in vocative case); "eit" → "eiti" (*to go*, infinitive); "schemoj" → "schemoje", (*schema*, in locative case)). This trend is reflected in informal written language. Moreover, Lithuanian uses the Latin script supplemented with diacritics (ą, č, ę, ė, į, š, ų, ū, ž), but in internet messages, diacritics are very often replaced with matching Latin letters (e.g.: ą → a, č → c, ę → e, etc.) or pairs of letters expressing the same sounds as in English (e.g.: č [tʃ] → ch, š [ʃ]→ sh, etc.). "Rinkimų programos'04", consisting of formal written language only, avoids the above-mentioned problems.

TABLE 1 shows the number of topics (classes) and the majority and random baselines (determined by calculating the sum of the squared probability of the classes, or $\sum_i P(c_i)^2$)) for each dataset.

Dataset	Number of topics	Number of text documents	Random baseline	Majority baseline
Lietuvos rytas	11	8,936	0.145	0.247
Supermamos	14	11,353	0.093	0.137
Rinkimų programos'04	8	2,388	0.167	0.231

TABLE 1 – Number of topics and text documents, random and majority baselines for all datasets.

4.2 Formal description of the task

The mathematical formulation of the topic classification task we are attempting to solve is given below.

Let $d \in D$ be a text document, belonging to a document space D. In our case we have three document spaces, related to different datasets: "Lietuvos rytas", "Supermamos", and "Rinkimų programos'04".

Let C be a finite set of classes (topics): $C=\{c_1, c_2, \ldots, c_N\}$. In our case $2 < N << \infty$ – i.e. we have multi-class classification problem, because "Lietuvos rytas" has N=11 classes; "Supermamos" – N=14 classes; "Rinkimų programos'04" – N=8 classes.

Let D^L be a training set, containing instances I – i.e. document feature vectors d' (where d' corresponds to document d) with their appropriate class labels: $I=\langle d', c\rangle$. "Lietuvos rytas", "Supermamos", "Rinkimų programos'04" contain single-labeled instances only: i.e. the text document d cannot be attached to more than one class c.

Let function γ be a classification function mapping text documents to classes, $\gamma : D \rightarrow C$. Function γ determines how d was labelled with c. In "Lietuvos rytas" and "Supermamos", the labeling was performed automatically by the users by replying to forum messages; in "Rinkimų programos'04", the annotation was performed by a domain expert.

Let Γ denote a supervised learning method which given training D^L as the input, can return a learned classification function γ' (defined as a model) as the output: $\Gamma(D^L)\rightarrow\gamma'$.

4.3 Experimental setup

The datasets "Lietuvos rytas", "Supermamos" and "Rinkimų programos'04" were pre-processed using different pre-processing techniques (TABLE 2):

Dataset	Total number of tokens	Number of distinct tokens	Number of distinct tokens after stemming	Number of distinct tokens after lemmatization
Lietuvos rytas	331,068	70,760	41,291	37,743
Supermamos	703,220	116,144	67,811	63,919
Rinkimų programos'04	29,745	7,426	4,474	3,320

TABLE 2 – Statistics about pre-processed datasets.

- Stemming. This pre-processing technique includes case normalization, the replacement of numbers with a general placeholder, and stemming. Words were stemmed using a Lithuanian stemmer (Krilavičius & Medelis, 2010) based on the Porter stemming algorithm (Porter, 1980; Willett, 2006). The Lithuanian stemmer eliminates endings and some suffixes. It is rule-based and can therefore cope with some irregular words (barbarisms, words with replaced diacritic letters, and words with grammatical errors). In some cases, stemming can cause the loss of meaning (e.g. "sala" (*island*), "salė" (*hall*), and "salti" (*to malt*) will be stemmed to the same "sal"). This pre-processing technique allows reducing the number of distinct tokens by ~42% for "Lietuvos rytas" and "Supermamos", and by ~40% for "Rinkimų programos'04".
- Lemmatization. This preprocessing technique includes the replacement of numbers by a placeholder. Documents were lemmatized using the Lithuanian morphological analyzer-lemmatizer "Lemuoklis" (Zinkevičius, 2000; Daudaravičius et al., 2007), which also solves morphological disambiguation problems. Its accuracy on normative Lithuanian texts is ~94% (Rimkutė & Daudaravičius, 2007). "Lemuoklis" transforms words into their lemmas by replacing the words' endings by their main lemma ending (but it does not touch suffixes and prefixes). However, the lemmatizer is dictionary-based, and can therefore not cope with unknown words. Case normalization was not necessary, because

"Lemuoklis" transforms the letters of generic words into lower case and leaves proper nouns capitalized. Unrecognized words are not modified in any way. "Lemuoklis" was unable to recognize ~15% of tokens in "Lietuvos rytas"; ~26% in "Supermamos", and ~2.5% in "Rinkimų programos'04". These numbers are in line with research by Giesbrecht & Evert (2009), who showed that for German data, every seventh token is incorrectly recognized when working with internet forum data, whereas almost no errors occur in (standard-language) fiction texts. Lemmatization allowed reducing the number of distinct tokens by ~47% for "Lietuvos rytas", by ~45% for "Supermamos", and by ~55% for "Rinkimų programos'04".

It should be noted that even though it is a common pre-processing step, we chose not to remove stop words from our datasets. This choice was made because of the following two reasons: 1) there is no list of stop words available for Lithuanian; 2) stop words are not a big problem in Lithuanian given the inflective nature of the language (e.g. Lithuanian does not have articles; phrasal verbs are expressed with a single token; negations are usually expressed by single token also, where participle "ne" (*not*, *no*) is attached as the prefix, etc.).

We now have three versions of the "Lietuvos rytas", "Supermamos" and "Rinkimų programos'04" datasets: one version with unmodified words, one with stemmed words, and one in which the words are lemmatized. Both the lemmatizer and the stemmer significantly reduce the number of distinct features, mapping several forms of the same word to a single feature. This should improve topic classification accuracy, but both the lemmatizer and the stemmer are in fact external resources, which may not be available for many resource-scarce languages, or which may not be accurate enough. Therefore, we will not only perform experiments with unigrams of lemmatized or stemmed words, but we will also attempt to classify the documents by using character n-grams. It is our expectation that the character n-grams will capture intrinsically what the lemmatizer and stemmer do explicitly, i.e. the terms present in the text, regardless of their inflection or derivational affixes.

The different feature types we will compare for all three datasets are thus the following:

- Unigrams based on word tokens.
- Unigrams based on stemmed words.
- Unigrams based on lemmatized words.
- Character n-grams based on word tokens.

The results of the corresponding experiments are described in Section 5.

Our first hypothesis is that a simple bag-of-words approach, which has shown its efficiency in topic classification tasks in English and in related languages, will not be the best technique for Lithuanian. We expect stemming and lemmatization to improve the classification results for Lithuanian significantly, due to the language's complex morphology and the importance of inflection.

Our second hypothesis is that it may not be necessary to use grammatical tools such as stemmers or lemmatizers – which may not be available for many resource-scarce languages – to capture the influence of this complex morphology and inflectional system. Character n-grams can capture these influences intrinsically. We expect character n-grams to perform significantly better than the bag-of-words approach, and we anticipate they will reach similar performance as the stemmed or lemmatized words.

4.4 Classification

We attempt to find a method Γ which could create the model γ' that best approximates γ. Since topic classification of text has never been solved for Lithuanian, we selected Support Vector Machines (a detailed description of SVMs is presented in Cortes & Vapnik (1995)) as Γ, based on comparative research indicating that SVMs perform best on a variety of text classification experiments. Additionally, SVMs can cope with the following problems which typically occur in text classification tasks:

- High-dimensional feature spaces. A very large feature set is not uncommon when dealing with text documents. E.g. if all token unigrams in a dataset (when $D^L = D$) would be considered as features (following the very common bag-of-words approach), "Lietuvos rytas" would have 70,760 features, "Supermamos" – 116,144; "Rinkimų programos'04" – 7,426.
- Sparseness of feature vectors. Each *d'* usually contains only a small number of features with non-zero values; these vectors are therefore considered sparse. E.g. in a token unigram approach (assuming $D^L = D$), each *d'* would contain ~14 non-zero feature values (among 70,760) in "Lietuvos rytas"; ~18 (among 116,144 features) in "Supermamos"; ~5 (among 7,426 features) in "Rinkimų programos'04".
- Heterogeneous use of features: i.e. when instances belonging to the same class do not share any common features. This is often the case when classifying short texts. Each *I* contains ~36 words in "Lietuvos rytas"; ~60 in "Supermamos"; only ~12 in "Rinkimų programos'04".

Additionally, SVMs do not require one to perform aggressive feature selection, which may result in a loss of information. This is especially important in our case, i.e. when working on Lithuanian, where there may be important information in both its rich vocabulary and in its complex word derivation system. The implementation of Support Vector Machines we use in our experiments is LIBSVM (Chang & Lin, 2011), using the radial basis kernel function. Most parameters were left to their default values, except for the cost parameter and the gamma parameter in the radial basis function, which were optimized using a grid search.

5 Results

All results reported in TABLE 3 – TABLE 5 are based on 10-fold cross-validation. For each experiment, the learner parameters were determined using a grid search, using macro-averaged F-scores as the optimization target.

	Lietuvos rytas		
	acc.	macro-F	micro-F
Tokens-1	0.271	0.190	0.265
Stems-1	0.305	0.221	0.300
Lemmas-1	0.303	0.232	0.300
Characters-4	0.324	0.237	0.317
Random baseline	*0.091*		
Majority baseline	*0.247*		

TABLE 3 – Accuracies, macro-averaged and micro-averaged F-scores for the different feature types on "Lietuvos rytas".

	Supermamos		
	acc.	macro-F	micro-F
Tokens-1	0.327	0.277	0.316
Stems-1	0.360	0.320	0.353
Lemmas-1	0.365	0.323	0.358
Characters-4	0.398	0.356	0.392
Random baseline	*0.071*		
Majority baseline	*0.137*		

TABLE 4 – Classification results for "Supermamos".

	Rinkimų programos'04		
	acc.	macro-F	micro-F
Tokens-1	0.298	0.238	0.282
Stems-1	0.326	0.262	0.306
Lemmas-1	0.363	0.291	0.346
Characters-4	0.376	0.290	0.351
Random baseline	*0.126*		
Majority baseline	*0.231*		

TABLE 5 – Classification results for "Rinkimų programos'04".

To determine whether the performances of the classifiers trained with each feature type were significantly different from each other, we performed approximate randomization testing (Noreen, 1989; Yeh, 2000).

For "Lietuvos rytas", all differences are significant to a very high degree ($p <= 0.0001$), with the following exceptions: the difference between the "Stems-1" and the "Lemmas-1" results are not statistically significant; the difference between "Characters-4" and "Stems-1" is significant to a high degree (accuracy: $p = 0.001$, macro-F: $p = 0.006$, micro-F: $p = 0.003$); the difference between "Characters-4" and "Lemmas-1" is only significant in terms of accuracy ($p = 0.0005$) and micro-F ($p = 0.001$).

For "Supermamos", all differences are significant to a very high degree ($p <= 0.0001$), with the exception of the difference between "Stem-1" and "Lemmas-1", which is not significant.

Finally, for "Rinkimų programos'04", all differences are significant to a very high degree ($p <= 0.0001$), with the following exceptions: the difference between "Token-1" and "Stems-1" in terms of macro-F is slightly less significant ($p = 0.007$), as is the difference between "Characters-4" and "Stem-1" ($p = 0.002$); the differences between "Characters-4" and "Lemmas-1" are not statistically significant.

6 Discussion

Before we could compare the character n-gram performance to the performance of the word unigrams (tokens, stemmed words or lemmatized words), it was necessary to determine the optimal size of the n-grams. As reported by Peng et al. (2003a, 2003b) (see Section 2), for English, the character n-gram performance is competitive starting from trigrams, and peaks at 6-grams. Chinese, on the other hand, performed best with bigrams. For classification of Farsi texts,

character 4-grams seem to be the recommended feature type (Bina et al., 2008). The optimal size of the character n-grams thus appears strongly dependent on the language of the dataset. No research regarding the optimal n-gram size for Lithuanian has yet been published.

We expect word roots to give us the most useful information for a topic classification task, and n-grams of a length that manages to capture these roots most often should therefore be the most powerful. Based on the fact that Lithuanian has an average word length of 6 characters, and that most endings are 2 characters long, we expect that 4-grams will yield the best classification performance.

To test this hypothesis, we carried out experiments on all three datasets using all n-gram sizes ranging from trigrams to 7-grams. FIGURE 2 shows a graph mapping the n-gram performance (in terms of micro-averaged F-score) on all three datasets.

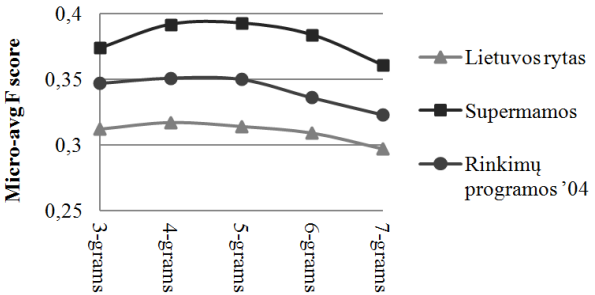

FIGURE 2 – Micro-averaged F-scores for n-gram sizes from 3-grams to 7-grams on all three datasets.

As expected, performance reaches a peak for character 4-grams, though the difference with 5-grams is subtle. Shorter string segments (trigrams) do not appear to capture enough information, while the longer segments (6-grams, 7-grams) fail to generalize sufficiently. TABLE 6 contains a more detailed performance breakdown.

	Lietuvos rytas	Supermamos	Rinkimų programos'04
3-grams	0.312	0.374	0.347
4-grams	0.317	0.392	0.351
5-grams	0.314	0.393	0.350
6-grams	0.309	0.384	0.336
7-grams	0.297	0.361	0.323

TABLE 6 – Micro-averaged F-scores for n-gram sizes from 3-grams to 7-grams on all three datasets.

A glance at TABLE 3 – TABLE 5 in Section 5 shows that our first hypothesis is confirmed: the simple bag-of-words approach (Tokens-1) is easily beaten when using a stemmer or a lemmatizer. For both the "Lietuvos rytas" and "Supermamos" datasets, stemming and lemmatization both account for an increase of 0.04 in micro-averaged F-score. On "Rinkimų programos'04", stemming does cause a small boost in performance, but lemmatization seems to

perform even better. The reason why lemmatization improves performance over stemming on the "Rinkimų programos'04" dataset, while it only matches the stemmer's scores on the other datasets, is that "Rinkimų programos'04" contains formal written text. As we described in Section 4.1, "Lietuvos rytas" and "Supermamos" both contain conversations from Internet forums, which are much more informal and contain non-standard language. The lemmatization package "Lemuoklis" is unable to deal with much of this "chat" data, while it does a better job lemmatizing the formal language in the political dataset.

More interestingly, the results also support our second hypothesis: the character 4-gram scores (which were the best n-gram scores, as shown in TABLE 3) are consistently higher than the scores for the other feature types. This means that not only is it possible to perform topic classification for Lithuanian using features that require no external resources, this approach even outperforms the standard classifiers trained on unigrams of stemmed or lemmatized words. The experiments were performed using different datasets, representing both formal and informal Lithuanian, with different particularities and different topic distributions. It can therefore be concluded that the hypothesis does not depend on the corpus or on a specific set of topics, but that it is applicable to Lithuanian in general. Assuming this pattern holds for other languages with similar properties (Latvian and some Slavic languages), this is good news indeed for resource-scarce languages where external tools such as lemmatizers or stemmers may not be available.

If we look more closely at why character n-grams work, we see that they capture patterns which are not always captured by alternative feature types. The lemmatizer reduces the word ending to their base form, but it does not remove suffixes ("mama", *mother*, and "mamytė", *mother* in diminutive, for instance, have the same root "mam", but are treated differently) and prefixes (e.g. "išbėgti", *to run out*, and "bėgti", *to run*, have the same root "bėg", but are treated differently). This means that phrasal verbs are not decomposed and diminutives or hypocoristic words remain untouched by the algorithm. The stemmer, then, is purely rule-based and captures even less of the intricacies of the Lithuanian language. It does not consider the function or meaning of a word in a sentence, and instead mechanically eliminates word endings (and some of the suffixes), resulting in an output that is all but perfect. The word "sveikas" (*healthy*), for instance, is stemmed to "sve" but should be stemmed to "sveik": i.e. ending "as" is correctly eliminated, but "ik" is erroneously treated as a suffix. With character n-grams, on the other hand, one does not need to explicitly define how words are formed. Character n-grams will implicitly capture the relevant inflectional patterns and the base word forms, which make them excellent features for this topic classification task.

The top character 4-gram features for the "Rinkimų programos'04" dataset, for instance, mostly contain the stable parts of words that are not influenced by inflection. In many cases, even, the n-grams capture the semantics of a term, which is stable over different parts of speech. The n-gram "vald", for instance, corresponds to the root of several, related words: the nouns "valdymas" (*management*), "valdžia" (*authority*) and "pavaldumas" (*subordination*); the verb "valdyti" (*to govern*); the phrasal verbs "įvaldyti" (*to master*), "suvaldyti" (*to manage*), etc. It is also one of the roots in the compound nouns "savivaldybė" (*municipality*) ("savas", *own* + "valdyti", *to govern*) and "žemėvalda" (*land-ownership*) ("žemė", *land* + "valdyti", *to govern*). Similarly, the character n-gram "kari" captures the nouns "karininkas" (*officer*) and "kariuomenė" (*army*); the adjective "karinis" (*military*). It captures the part of the noun "kariai" (*soldiers*) that remains stable in all inflected plural forms.

The examples above are lemmas (with the exception of "kariai", which is the plural of "karys", *soldier*), meaning they would not be transformed further by the lemmatizer. The stemmer would shorten the lemmas by eliminating endings and some suffixes, but prefixes and compound words would still remain.

By capturing those parts of words that are stable over semantically related words across different parts of speech, character n-grams help resolve feature sparseness, which is especially problematic for Lithuanian. Since Lithuanian has such a rich vocabulary, the standard bag-of-words approach results in an extremely sparse feature space, which can significantly complicate the construction of a good classification model. Stemming and lemmatization help reduce sparseness by limiting the number of possible realizations of the same lemma, but they do nothing to reduce the number of lemmas. Character n-grams, however, segment text into more frequent, more general substrings, which results in more reliable models.

Character n-grams also don't throw away valuable information as a stemmer or lemmatizer might. Some character n-grams, such as "emės", capture a string that remains stable in fixed combinations of two words, e.g. "žemės ūkis" (*agriculture*), "žemės grąžinimas" (*land restitution*), "žemės nuosavybė" (*land property*), etc. Both the stemmer and the lemmatizer would change "žemės" to "žemė", thus discarding valuable information.

Conclusion and perspectives

We have formulated and experimentally confirmed two hypotheses regarding topic classification for morphologically rich languages such as Lithuanian:

- The standard bag-of-words approach which works best for English and related languages, is not the best approach for these languages. Where English barely benefits from stemming and lemmatization, Lithuanian (and, by extension, likely also other Baltic and Slavic languages) benefit significantly from these pre-processing steps.
- It is possible to capture the intricate morphology of Baltic and Slavic languages without resorting to external tools such as stemmers or lemmatizers, which may not be available (or which may simply not work sufficiently well). Character n-grams implicitly capture relevant patterns and can outperform classifiers trained on stemmed or lemmatized data.

Our assumptions that these hypotheses also hold for Slavic languages are based on the characteristics that they share with Lithuanian. In future research, it will be interesting to experiment with datasets in these languages to see if these assumptions hold.

Acknowledgments

This research is funded by European Union Structural Funds project "Postdoctoral Fellowship Implementation in Lithuania" (No. VP1-3.1-ŠMM-01) and was initiated when the first author was visiting the Computational Linguistics & Psycholinguistics Research Center, Antwerp, Belgium. Frederik Vaassen is funded by the IWT (the Flemish government agency for Innovation by Science and Technology), TETRA project deLearyous.

References

Bina, B., Ahmadi, M. H., and Rahgozar, M. (2008). Farsi Text Classification Using N-Grams and Knn Algorithm, A Comparative Study. In *Proceedings of International Conference on Data Mining*, DMIN 2008, pages 385–390.

Boulis, C., and Ostendof, M. (2005). Text Classification by Augmenting the Bag-of-Words Representation with Redundancy-Compensated Bigrams. In *Proceedings of the SIAM International Conference on Data Mining at the Workshop on Feature Selection in Data Mining* (SIAM-FSDM 2005), Newport Beach, California.

Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology* (TIST 2011), 2(3), pages 1–27.

Cortes C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20: 273–297.

Daudaravičius, V., Rimkutė, E. and Utka, A. (2007). Morphological annotation of the Lithuanian corpus. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies* (ACL'07), Prague, Czech Republic, pages 94–99.

Gaustad, T. and Bouma, G. (2002). Accurate Stemming of Dutch for Text Classification. *Language and Computers*, pages 104–117.

Giesbrecht, E. and Evert, S. (2009). Part-of-Speech (POS) Tagging - a solved task? An evaluation of POS taggers for the Web as corpus. In *Proceedings of the Fifth Web as Corpus Workshop* (WAC5), pages 27–35.

Gimbutas M. (1963). *The Balts*. Thames and Hudson, London.

Hersh, W., Buckley, C., Leone, T. J. and Hickman, D. (1994). OHSUMED: an interactive retrieval evaluation and new large text collection for research. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 94), pages 192–201.

Khreisat, K. (2006). Arabic Text Classification Using N-Gram Frequency Statistics. A Comparative Study. In *Proceedings of International Conference on Data Mining*, DMIN 2006, pages 78–82.

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, 31: 249–268.

Krilavičius, T. and Medelis, Ž. Lithuanian stemmer. (2010). May, 2012. <https://github.com/tokenmill/ltlangpack/tree/master/snowball/>.

Lang K. (1995). NewsWeeder: Learning to Filter Netnews. In *Proceedings of the 12th International Machine Learning Conference* (ML95), Tahoe, CA, USA, pages 331–339.

Leopold, E. and Kindermann, J. (2002). Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? *Machine Learning*, 46 (1–3): 423–444.

Lewis, D. D. (1992). An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (SIGIR 92), pages 37–50.

Lewis, D. D., Yang, Y., Rose, T. G. and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research archive*, 5: 361–397.

Lietuvos rytas. (2012). Internet forum of a newspaper "Lietuvos rytas". May, 2012. <http://forum.lrytas.lt/>. (In Lithuanian).

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

Nastase, V., Sayyad, J., and Caropreso, M. F. (2007). Using Dependency Relations for Text Classification. Technical Report TR-2007-12, University of Ottawa, Canada.

Noreen, E. W. (1989). *Computer-intensive Methods for Testing Hypotheses: An Introduction*. John Wiley & Sons, New York, NY, USA.

Peng, F., and Schuurmans, D. (2003). Combining Naive Bayes and n-Gram Language Models for Text Classification. In *Proceedings of 25th European Conference on Information Retrieval Research* (ECIR), pages 335–350.

Peng, F., Huang, X., Schuurmans, D., and Shaojun, W. (2003a). Text classification in Asian languages without word segmentation. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages*, 11, pages 41–48.

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

Piročkinas, A. (2012). Dar keli išskirtiniai lietuvių kalbos bruožai [Several distinctive features of Lithuanian language]. *Dialogas*, 15. <http://www.dialogas.com/laikrastis/dar-keli-isskirtiniai-lietuviu-kalbos-bruozai/>. (in Lithuanian).

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3): 130–137.

Radovanovic, M. Ivanovic, M. (2006). Document representations for classification of short web-page descriptions. In *Proceedings of the 8th international conference on Data Warehousing and Knowledge Discovery*, DaWaK'06, Krakow, Poland, pages 544–553.

Rimkutė E. and Daudaravičius V. (2007). Morfologinis Dabartinės lietuvių kalbos tekstyno anotavimas [Morphological annotation of the Lithuanian corpus]. *Kalbų studijos*, 11: 30–35. (in Lithuanian).

Savickienė, I., Kempe, V. and Brooks, P. J. (2009). Acquisition of gender agreement in Lithuanian: exploring the effect of diminutive usage in an elicited production task. *Journal of Child Language*, 36: 477–494.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34: 1–47.

Supermamos. (2012). Internet forum. May, 2012. <http://www.supermama.lt/forumas/>. (In Lithuanian).

Tan, C. M., Yuan-Fang, W., and Chan-Do, L. (2002). The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4): 529–546.

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

Volkens, A. (2002), *Manifesto coding instructions (second Revised Edition)*. Social Science Research Center Berlin (WZB), Wissenschaftszentrum Berlin, Germany.

Wahbeh, A., Al-Kabi, M., Al-Radaideh, Q. A., Al-Shawakfa, E. M., and Alsmadi, I. (2011). The Effect of Stemming on Arabic Text Classification: An Empirical Study. *International Journal of Information Retrieval Research*, 1(3): 54–70.

Wei, Z., Chauchat, J. H. and Miao, D. (2008). Comparing different text representation and feature selection methods on Chinese text classification using character n-grams. *Journées Internationnales d'Analyse des Données Textuelles*, Lyon, France, pages 1175–1186.

Willett, P. (2006). The Porter stemming algorithm: then and now. *Program: electronic library and information systems,* 40 (3): 219–223.

Yeh, A. (2000). More Accurate Tests for the Statistical Significance of Result Differences. In *Proceedings of the 18th International Conference on Computational Linguistics*, 2, pages 947–953.

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