VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Magistrinis darbas

Lietuviškų tekstų klasifikavimas į verstinius ir originalius pagal jų žodžių dažnių skirstinius

Classification of Lithuanian Text into Translated and Original Based on Word Frequency Distribution

Gediminas Šimaitis

VILNIUS 2016

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

MATEMATINĖS ANALIZĖS KATEDRA

Darbo vadovas Dr. (HP) Marijus Radavičius _____

Darbo recenzentas _____

Darbas apgintas _____

Darbas įvertintas _____

Registravimo NR. _____

Įrašoma atidavimo į katedrą data _____

Lietuviškų tekstų klasifikavimas į verstinius ir originalius pagal jų žodžių dažnių skirstinius

Santrauka

Tekstas, verstas iš kitų kalbų, turi specifinius bruožus, kuriuos galima identifikuoti statistiniais metodais. Ankstesni tyrimai įvardina kitokį žodyno turtingumą, leksinį tankį ir žodžių ilgių skirstinį kaip bruožus, kuriais išsiskiria verstinis tekstas. Šiame darbe vektorių palaikymo mašinų modeliai, sėkmingai naudoti šiam klasifikavimo uždaviniui ankstesniuose tyrimuose, pritaikomi lietuviškų tekstų rinkiniams. Šie modeliai tuomet papildomi kintamaisiais, atspindinčiais įvardintus verstinio teksto bruožus, taip pagerinant klasifikavimo tikslumą.

Raktiniai žodžiai : Versto teksto klasifikavimas, vektorių palaikymo mašinos, Zipf dėsnis, žodžių dažnių skirstinys.

Classification of Lithuanian Text into Translated and Original Based on Word Frequency Distribution

Abstract

Translated text has certain features which mark it as such, which can be identified using statistical methods. Features such as lexical density, vocabulary richness and word length distribution are some of the marks of translated text identified by existing research. In this work support vector machine models, which were found to be effective for this purpose by previous studies, are applied to corpora of Lithuanian monolingual texts. The models are then augmented using variables constructed to reflect the suggested marks of translated text in an attempt to improve classification performance.

Key words : Translationese classification, support vector machines, Zipf's law, word frequency distribution.

Contents

1 Introduction

It is recognized that translated text has distinct features and style peculiarities which allow it to be distinguished from texts originally written in the language. Language containing these features is called *translationese* by translation researchers (Gellerstam, 1986). The origin of *translationese* is attributed to "fingerprints" carried from the source language onto the translation language or from the translation process itself.

Text classification methods are starting to be adapted for the purpose of identifying these features and classifying monolingual text as translated or originally written in the language, with promising results (Baroni and Bernardini, 2006). Support vector machines on the text data is the most popular approach, with research testing different approaches to text tokenization and applying the methods to different text corpora.

In a separate line of investigation, a growing body of research suggests a variety of statistical differences which set apart translated text from text written in the language. The identified differences are generally based on vocabulary diversity and lexical density (ratio of content words to function words in a text). These features are shown to exist for different origin and translated languages, including Lithuanian (Piaseckienė and Radavičius, 2014), and can be used to assist in the identification of translated text without a reference to the source text.

However, none of the reviewed works attempted a combination of the two aforementioned lines of research. In this work classification of text as translated or written in the language originally is augmented with statistical parameters. Furthermore, to the best of our knowledge, this classification model is applied to a Lithuanian corpora for the first time. Classification of the text is performed using the support vector machines approach. The additional statistical parameters of the text are constructed based on existing research and a brief investigation into their applicability for classification between translated and original text is performed.

In this work it is shown that these additional variables can be used to improve the accuracy of a support vector machine model used for the classification of text as translated or originally written in Lithuanian. This is demonstrated using two different corpora of monolingual Lithuanian text.

Applications of translationese identification are in improving statistical machine translation, both for constructing the training data sets for assessment in parallel corpus extraction, and identifying the direction of translation. Awareness of the statistical features

of translationese can also be used in improving existing algorithms. Other possible uses are as a self-assessment tool for translators and multi-lingual plagiarism detection.

The rest of this work is structured as follows: Section 2 provides a brief overview of previous research into features of translated text and classification of translated text. It is followed by a description of the statistical methods and data used (Section 3). Section 4 presents the empirical results in automatic categorization of translated text, and Section 5 sums up the main results.

2 Literature Review

The hypothesis that translated text contains "fingerprints" of the language it is translated from, called "translationese" by researchers, is first described by Gellerstam (1986). Gellerstam focused on translationese in translations from English to Swedish, however, a more general hypothesis that translated text contains characteristics typical of translation, regardless of original and translation languages, is raised by more recent researchers (Baker et al., 1993). There are a few different ideas of what the differences may be, however most of them lend well to quantification.

One of the more common hypothesis among linguists studying translationese is that translated text is less lexically dense and contains less options, which appears to be a common trait among translationese of different languages (Olohan, 2001; Puurtinen, 2003). Lexical density is defined as the ratio of content words to function words, where function words are words which have little meaning on their own and serve to express grammatical relationships within the sentence. Baroni and Bernardini (2003) similarly find that translated text characteristically contains more sequences of function words.

The methodology used in such studies typically involves constructing a monolingual comparable corpora comprising texts originally written in a language and translations into the same language. An excellent overview of the various properties investigated by translation researchers is provided by Zanettin (2013).

However the same researchers tend to indicate genre differences and corresponding translation conventions may play a role, even in a comparable corpora, for example detective novels may be overrepresented in English translations to Swedish (Gellerstam, 1986) and subgenres of children's literature show different lexical and even syntactic features (Puurtinen, 2003).

Research by Piaseckienė and Radavičius (2014) indicates texts in Lithuanian language by

native and foreign authors exhibit different word distributions under Zipf's law - translated texts generally tend to have a more standard vocabulary. On the other hand, they may contain more words specific to other nations which are otherwise rare in the Lithuanian language.

Zipf's law, formulated by Zipf (1935), states that rank-frequency distribution of words in a text is an inverse relation. The most common way to observe Zipf's law is by plotting the data on a log-log graph, if the data conforms to Zipf's law, the plot should be roughly linear. Mathematically this can be expressed as:

$$f(k, s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \tag{2.1}$$

where N is the number of elements, k is their rank and s an exponent characterizing the distribution.

The word length distribution may also be useful for identification of translated text, as shown by Piaseckienė and Radavičius (2014). For a given text, the frequency of words of each length (in letters) is calculated, and the slope of the frequency curve is typically less steep for foreign authors.

Using Support Vector Machines (SVM) to classify text was first proposed by Joachims (1998). SVM have the ability to learn independent of the dimensionality of the feature space. This makes them well suited for text classification tasks, which generally feature a high dimensional input space and the document vectors are sparse.

Additionally, Joachims suggests most text categorization problems are linearly separable, thus using SVM with linear kernels is appropriate. Experimental evidence by him supports these assertions.

Baroni and Bernardini (2006) first suggested identifying translationese using Support Vector Machines. In their research, the authors explore different ways to represent a document, such as unigram, bigram and trigram, as well as using lemmatized (changed to their base form) and original form words.

For single identifiers, unigram representations performed best, with smaller differences between original and base word forms, achieving 74.2%-77.1% accuracy. The highest accuracy reached by an ensemble of identifiers is reported to be 86.7%.

Similar methods are successfully used and augmented by other researchers, however such augmentations generally targeted machine translated text (Kurokawa et al., 2009) (Arase and Zhou, 2013).

3 Methodology

3.1 Statistical Methods

A number of different variables are constructed based on research detailed in the Literature Review section. Vocabulary richness is expressed through the word rank-frequency distribution (Zipf's law) and ratio of words used only a few times in the text. The word length distribution is calculated and included directly. In order to quantize lexical density the ratio of stop words is calculated. Stop words, which are the most common function words, were loaded from an existing list.

Specifically, the constructed variables consist of:

- The ratio of stop words (as a proxy for function words) to all words in a given text. The ratio itself is used as a variable.
- The frequency of each word is calculated and its rank relative to other words. The deciles of this distribution are used as variables. This variable is created to reflect vocabulary richness.
- Additionally, as per Zipf's law, the slope of the frequency-rank curve (expressed logarithmically) is calculated using ordinary least squares (OLS) regression, and used as a variable.
- The ratio of words of each length (in characters) to all words is calculated. Both the numbers directly and their slope estimated using OLS are used as variables. This variable is based on the corresponding evidence in Piaseckienė and Radavičius (2014).
- The ratio of words which occur at most a specified number of times in the text - once (*hapax legomena*), at most twice, and similarly up to five times (five different variables). This variable is also a proxy for vocabulary richness.

In order to determine if the created variables carry relevant information, first a logistic regression is performed on only the created variables, and a support vector machine (SVM) model is trained.

A support vector machine model is then trained only on the text data, to be used as a baseline, and another SVM model is trained on both the text data and the additionally introduced variables, in order to assess whether the additional variables can be used to

increase the precision of the model. The text data is in the form of a document-term matrix, where each variable is the number of occurrences of the word in a document.

A SVM model is selected, as it is the most used model in the reviewed research which performed text classification into translated and original, and is thus the best platform to assess the value of including additional statistical data into the model. This is mainly because of SVM being well-suited for text classification tasks due to capability of coping with features typical of text data such as large feature spaces, few irrelevant features and sparse data (Joachims, 1998).

All tests are performed using 10-fold cross-validation (the data is divided into 10 parts, and the test is then run 10 times, using a different part as the test set each time). All the reported results are the average of results across the folds. The SVM model is used as implemented by Meyer et al. (2015). All other models are used as implemented by R Core Team (2016), unless specified otherwise.

3.2 Support Vector Machine Model

A support vector machine model works by constructing a hyperplane in a way to maximize the separation between the cases in the training set. Classification of the test set is then performed based on which side of the hyperplane the case falls on.

A brief summary is presented below. A detailed construction and solution of the specific C-Support Vector Classification algorithm is presented by Chang and Lin (2011), an implementation of the algorithm described by Cortes and Vapnik (1995).

Given training vectors $\boldsymbol{x}_i \in R^n, i = 1, \ldots, l$ in two classes with a corresponding indicator vector $\boldsymbol{y} \in R^l$, $y_i \in \{-1, 1\}$ the optimization problem is

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i\left(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = 1, \ldots, l \end{aligned} \tag{3.1}$$

where $C > 0$ is a regularization parameter and $\boldsymbol{w}$ is the solution vector. Due to high dimensionality of $\boldsymbol{w}$ the dual problem is then solved

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{y}^T \boldsymbol{\alpha} = 0, \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, l \end{aligned} \tag{3.2}$$

where Q is an l by l positive semidefinite matrix, $Q_{ij} \equiv y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ is the kernel function and $\boldsymbol{e} = [1, \ldots, 1]^T$ is a vector of all ones.

Equation (3.2) is then usually solved iteratively using decomposition methods, which allow solving smaller optimization sub-problems. In this specific implementation, the sequential minimal optimization type method proposed in Fan et al. (2005) is used.

Using the solution of equation (3.2), the optimal $\boldsymbol{w}$ then satisfies

$$\boldsymbol{w} = \sum_{i=1}^{l} y_i \alpha_i \phi(\boldsymbol{x}_i) \tag{3.3}$$

The decision function is thus

$$\text{sign}\left(\boldsymbol{w}^T \phi(\boldsymbol{x}) + b\right) = \text{sign}\left(\sum_{i=1}^{l} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right) \tag{3.4}$$

In the case of a linear kernel function, which is the recommended kernel function for text data (Joachims, 1998), and scaled data, the feature weights $\boldsymbol{w}$ can be interpreted directly to determine the importance of each variable to the decision function.

3.3 Corpus Construction

For the empirical study two different data sets are used:

The first data set (*delfi.lt*) is taken from the publicly accessible news website *delfi.lt*, and consists of articles regarding events in Lithuania and abroad. While it is not explicitly stated that all articles regarding events abroad are translated, almost all of them come from news agencies such as ELTA and BNS, which, in turn, state in their web pages they are providers of news within Lithuania / Baltics as well as providing news feeds from their partners abroad.

The articles are all written in a fairly similar journalistic style, and cover a variety of topics. They are also relatively short compared to texts used in previous studies. However it is possible the topics covered by original and translated articles are identifiably different.

The second data set (*emokykla*) is taken from the digital library *http://ebiblioteka.mkp.emokykla.lt/* and consists of recommended reading texts in the public school curriculum of Lithuania, grades 5-8. Due to significant style differences,

poetry is excluded from this sample, resulting in a corpus of 58 text documents, of which 19 are translated, and 39 are originally written in Lithuanian.

The texts display an assortment of genres and a SVM model would pick up the genre features easily, e.g. some texts originally written in Lithuanian contain features of Lithuanian folklore. In order to reduce the impact of this for the empirical study the documents are split into segments by paragraphs, each segment of similar size to the median size of an article in the *delfi.lt* corpus. A total of 1000 such chunks were selected randomly to comprise the corpus (500 each translated and original).

Descriptive statistics of the two used corpora are presented in Table 1.

	delfi.lt	*emokykla*
Number of texts	3091	1000
Number of translated texts	1684	500
Number of original texts	1407	500
Average length in words	487.5	215.9
Median length in words	215	204

Table 1: Descriptive statistics of used corpora

Both datasets are sanitized by removing proper nouns, such as person, organization and place names.

Word stemming is performed using a stemming algorithm described by Porter (1997), as implemented by Bouchet-Valat (2014).

Compared to research reviewed in Section 2, the documents in both corpora are shorter, however, considering the expected possible applications such as web mining or plagiarism detection, short texts or excerpts of texts are expected to be the standard rather than the exception.

4 Results

4.1 Assessment of Additional Variables

4.1.1 Overview of Additional Variables

In order to ascertain the applicability of the additional variables, all the variables are calculated and reviewed if they might carry information useful for text classification into

translated text and original text. This is performed by first plotting the means of the variables and their 95% confidence interval, and second, by performing a logistic regression and training a support vector machine model using only the additional variables.

The accuracy (% of correct classifications), sensitivity (true positive rate or recall), specificity (true negative rate) and precision (positive predictive value) are reported for each model. The F score is also reported for each model, which is the harmonic mean of sensitivity and precision.

For reference, a random classifier which knows the ratio of translated to original texts in the corpus and assigns documents as translated or original with this probability would obtain 50% accuracy, 50% precision, 50% sensitivity, 50% specificity and 50% F. A trivial acceptor which would treat all documents as translated in a corpus half of which is original documents would obtain 50% accuracy, 50% precision, 100% sensitivity, 0% specificity and 66.7% F.

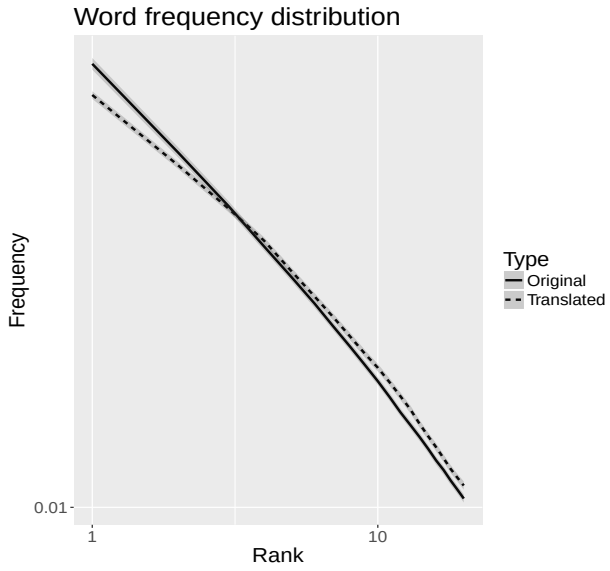

Figure 4.1: *delfi.lt* data set rank-frequency distribution

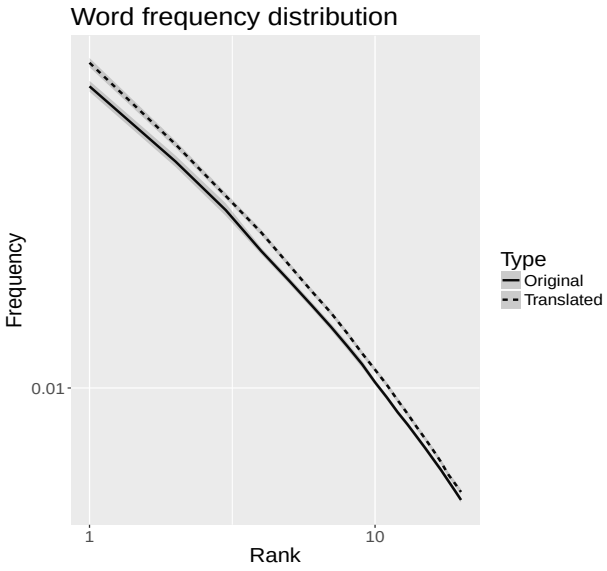

Figure 4.2: *emokykla* data set rank-frequency distribution

As can be seen from Figures 4.1 and 4.2, the means of the word frequency distribution are statistically significantly different. However the direction of the difference is different in the beginning of the curve for the two text corpora used, which may limit cross-corpus application of the trained model.

The word length distribution is very close together for the *emokykla* data set (Figure 4.4), however, there are pronounced kinks and differences in the word distribution by length in the *delfi.lt* data set (Figure 4.3). The average slope of the curve is steeper for translated

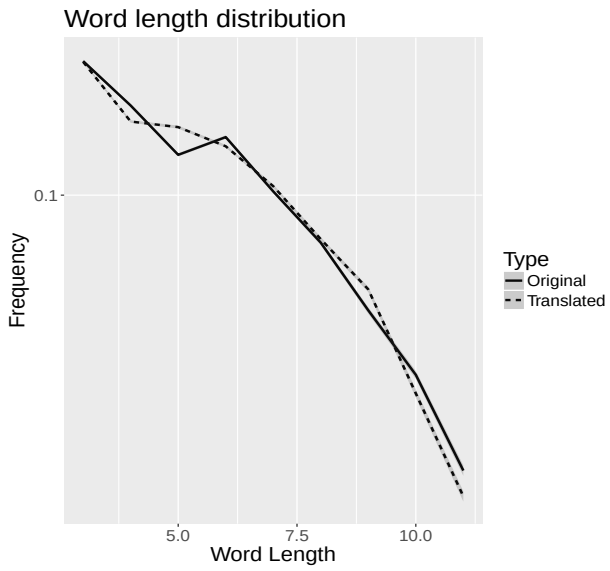

Figure 4.3: *delfi.lt* data set length-frequency distribution

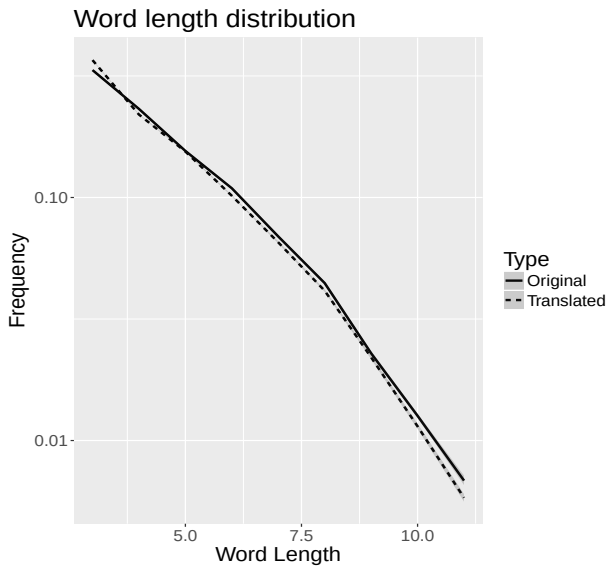

Figure 4.4: *emokykla* data set length-frequency distribution

text for both data sets.

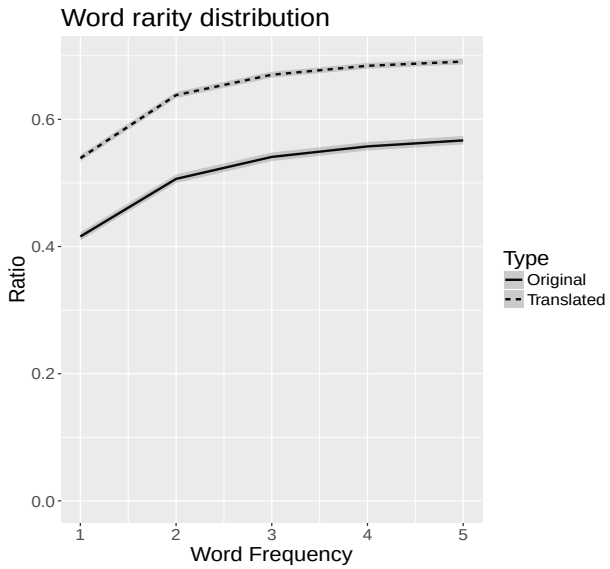

Figure 4.5: *delfi.lt* data set word rarity distribution

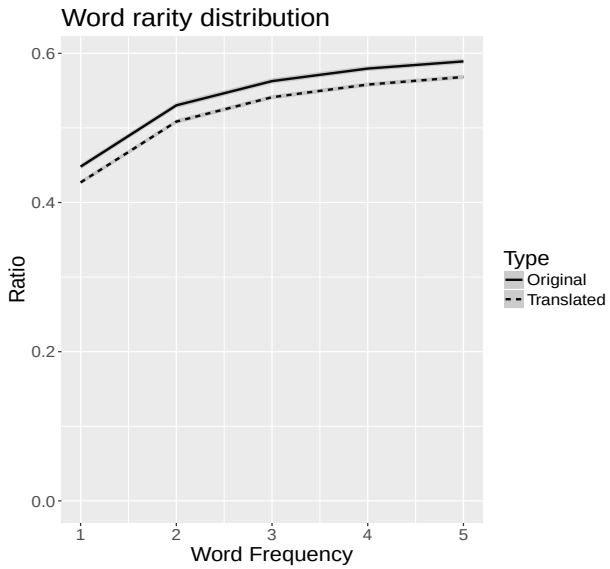

Figure 4.6: *emokykla* data set word rarity distribution

Figures 4.5 and 4.6 show the proportion of words in the text which occur only a few times in the text, as another proxy for richness of vocabulary. In the *emokykla* data set the original texts have slightly more words used only a few times, which is in line with the expectation of a richer vocabulary. In the *delfi.lt* data set the difference is in the opposite direction, with translated texts having significantly more words used only a few times, again limiting cross-corpus application of trained models.

Generally, the parameters of the *emokykla* data set behave as predicted by previous research, however the *delfi.lt* data set does not completely conform to expectations. This may be due to style differences between the two corpora (e.g. foreign news articles might be using richer article templates), however the exact origin is not certain. Regardless, the variables do have differences between translated and original texts, indicating they may be useful for classification.

4.1.2 Predictive Power of Additional Variables

To determine if the statistical variables have notable predictive power, a support vector machine model was trained for classification using only the additional variables. A logistic regression was also constructed for comparison purposes.

Tables 2 and 3 show the results (using 10-fold cross-validation).

	Accuracy	Precision	Sensitivity	Specificity	F
Logistic regression	77.1%	83.6%	72.1%	83.0%	77.4%
Support vector machines	79.4%	79.6%	83.6%	74.2%	81.6%

Table 2: *delfi.lt* data set additional variable predictive power assessment

	Accuracy	Precision	Sensitivity	Specificity	F
Logistic regression	64.5%	72.0%	46.8%	82.2%	56.7%
Support vector machines	65.6%	65.1%	67.4%	63.8%	66.2%

Table 3: *emokykla* data set additional variable predictive power assessment

In both of the cases the support vector machine model performed better than the logistic regression. The classification accuracy is notably lower in the *emokykla* data set, presumably due to style differences within the corpus.

In order to determine variables with the highest predictive power, the feature weights of the support vector machine model were extracted. The top ten variables with highest feature weights are presented in Tables 4 and 5. All variables were rescaled for the support vector machine model, thus the feature weights are directly comparable.

The word length variables are the ratios of words of that specific length to all words in the text (after stemming), word instances variables are the ratios of words which appear at most that many times in the text to all words in the text, the word frequency quantile is the

Variable	Feature weight
Word length 5	8.646
Word length 6	-3.867
Word length 4	-3.694
Word length 12	-3.527
Word length 3	3.133
Word length 9	2.732
Word frequency 10% quantile	-2.156
Word length 10	-2.117
Word instances 5	1.818
Word instances 2	1.782

Table 4: *delfi.lt* additional variable feature weights

Variable	Feature weight
Word length 3	6.780
Stop words ratio	4.075
Word length 4	-3.804
Word length 6	-1.995
Word length 10	-0.391
Word length 11	-0.387
Word length 5	0.375
Word length 8	-0.345
Word instances 1	-0.306
Word length 7	-0.271

Table 5: *emokykla* additional variable feature weights

corresponding quantile of the word frequency-rank distribution (10% quantile is the most frequent words) and the stop words ratio is the ratio of stop words to all words in the text.

As can be seen, the word length variables were assigned quite a few of the highest weights in both data sets, indicating the word length-frequency distribution has significant correlation with whether the text is translated or not, however other variables are also present.

4.2 Classification with Full Text Data

With relevance of the additional parameters established, an assessment is performed of whether the inclusion of statistical data on word distribution improves the classification. First a support vector machine model is trained using only the text data (Text only SVM), and compared to a support vector machine model trained using both text data and the additional statistical variables (Augmented SVM). This allows to determine if augmentation of the SVM model on only the text data used in previous research can yield classification benefits.

The results are provided in Tables 6 and 7.

	Accuracy	Precision	Sensitivity	Specificity	F
Text only SVM	85.1%	79.7%	95.9%	74.6%	86.9%
Augmented SVM	80.0%	79.1%	86.2%	72.7%	82.5%

Table 6: *delfi.lt* data set augmented SVM performance

	Accuracy	Precision	Sensitivity	Specificity	F
Text only SVM	77.7%	85.3%	67.0%	88.4%	75.1%
Augmented SVM	67.0%	66.5%	68.6%	65.4%	67.5%

Table 7: *emokykla* data set augmented SVM performance

The text only support vector machine model outperforms the augmented support vector machine model in both cases. However reviewing the top feature weights of the text only models (Tables 8 and 9) it can be noted both models picked up a lot of content trends, such as the local parliament election, the presidential election of the USA and domestic weather forecasts for news articles. While this is not as obvious in the top features by weight of the *emokykla* corpus, it similarly contains content words referring to concepts such as spirits

Variable	Feature weight	Variable	Feature weight
rinkim	-4.936	pirminink	-2.697
prezident	4.611	nar	-2.696
darb	-4.225	buv	2.637
partij	-3.784	kad	2.238
valstiet	-3.725	pat	-2.238
kur	3.721	sav	2.143
laipsn	-3.704	koalicij	-2.031
bus	-3.344	kov	2.015
tur	-2.923	pare	1.98
apygard	-2.725	temperatūr	-1.954

Table 8: *delfi.lt* text only SVM feature weights

Variable	Feature weight	Variable	Feature weight
man	2.3519913	būt	0.7000959
buv	1.9400128	tar	0.6885604
kad	1.8714959	bet	0.6811001
jis	1.5654227	nes	0.6119852
taip	1.3595746	pasak	0.5638752
jūs	1.0320651	pon	0.5535543
kai	0.9876899	lab	0.5510278
vyr	-0.8550096	atrod	0.4972833
kur	0.8426029	jeig	0.4910799
ant	-0.8053306	nei	-0.4880243

Table 9: *emokykla* text only SVM feature weights

of Lithuanian folklore (kipš, laum, ragan) and nobility titles (kunigaikšt) as well as topics generally attributable to foreign literature such as slavery (verg, negr).

4.3 Classification without Content Words

Considering that the support vector machine model picks up on content trends in the text, an additional model is constructed keeping only the stop words in the text. This approach ensures that no content trends are used in the model, as identification of translated text by content is not the goal of this work.

This is supported with a review of least 50 features with the highest weights, the features consisted of stop words, words containing an obvious content trend and, in the case of augmented SVM, the additional variables.

As an additional support for this approach, in their study Baroni and Bernardini (2006) similarly indicate that the main features used in the classification of translated text are pronouns and adverbs, a subset of words which largely overlaps with stop words.

Tables 10 and 11 show the results of these models.

	Accuracy	Precision	Sensitivity	Specificity	F
Text only SVM	71.7%	75.6%	64.4%	79.0%	69.6%
Augmented SVM	77.6%	76.9%	84.2%	69.6%	80.4%

Table 10: *delfi.lt* data set with stop words only SVM performance

	Accuracy	Precision	Sensitivity	Specificity	F
Text only SVM	60.7%	80.6%	28.2%	93.2%	41.8%
Augmented SVM	65.8%	65.4%	67.4%	64.2%	66.4%

Table 11: *emokykla* data set with stop words only SVM performance

After excluding content words, a gain of about 5 percentage points of accuracy and an increase of over 10 percentage points in the F-score is noted on both corpora from the inclusion of additional variables, which is considered a significant improvement in the model performance.

The performance of the text only model on the *delfi.lt* corpus is slightly lower than previous research, e.g. 74%-77% in Baroni and Bernardini (2006), however as the corpus consists of only short articles (average length 215 words compared to 3572 used by Baroni

Variable	Feature weight	Variable	Feature weight
Word length 5	8.452	Word instances 2	1.789
Word length 6	-3.808	Word frequency slope	1.778
Word length 4	-3.538	Word instances 5	1.772
Word length 12	-3.515	Word length 11	-1.714
Word length 3	3.051	per	1.662
kur	2.799	Word instances 4	1.61
Word length 9	2.694	Stop words ratio	1.549
Word frequency 10% Q	-2.123	pat	-1.452
Word length 10	-2.068	kai	1.281
kad	2.023	Word frequency 20% Q	-1.22

Table 12: *delfi.lt* augmented SVM with stop words feature weights

Variable	Feature weight	Variable	Feature weight
Word length 3	7.2140663	ant	-0.7834825
Word length 4	-3.8468142	kai	0.7803245
Stop words ratio	3.731759	kur	0.7284102
Word length 6	-1.5053784	Word length 7	-0.6657842
man	1.4771795	nes	0.5127298
jis	1.1484048	nei	-0.4815024
taip	1.1287052	pasak	0.4367507
kad	1.0959533	net	-0.4336442
Word length 8	-1.0273199	Word length 5	0.416792
Word instances 1	-0.9263284	Word instances 3	0.3805412

Table 13: *emokykla* augmented SVM with stop words feature weights

and Bernardini) this is to be expected due to less data being available for the classification of each document. The weaker performance on the *emokykla* corpus is also attributable to the presence of texts of different styles.

The top feature weights of the augmented SVM model using only stop words are provided in Tables 12 and 13. All feature weights of the two models are included in Appendix A. The top variables by feature weight are fairly similar to the models in the initial assessment of the variables in Section 4.1.2.

The word length variables are the ratios of words of that specific length (after stemming) to all words in the text, word instances variables are the ratios of words which appear at most that many times in the text to all words in the text. The word frequency quantile is the corresponding quantile of the word frequency-rank distribution (10% quantile is the most frequent words), the word frequency slope is the estimated slope of the logarithm of the word rank-frequency distribution, and the stop words ratio is the ratio of stop words to all words in the text. Other variables are the frequency of that specific stop word in the text.

5 Conclusion

The aim of this study was to investigate whether addition of statistical variables can improve the accuracy of the currently used methods for monolingual text classification as translated or originally written in the language. For this purpose support vector machine models were constructed on two different Lithuanian text corpora, as support vector machines are currently the most widely applied model to this problem. Another aim was to demonstrate the applicability of such models to Lithuanian texts.

This study introduced statistical variables into the SVM models with an aim to thus improve the accuracy of these models. After controlling for classification by content, the additional variables were shown to improve the accuracy of such models by 5-6%. Additionally, the variables carry enough information that the performance of a model using only these variables achieves accuracy comparable to a model using both the stop word text data and the additional variables.

Furthermore, the models were shown to be applicable to Lithuanian corpora in classification of text as translated or originally written in Lithuanian.

Establishing the applicability of the new variables to other languages remains for future research. The development of the additional statistical variables is based on research which

is largely based on non-Lithuanian corpora and can thus be believed to transcend at least some language barriers, giving some credibility to the expectation that such research would be fruitful.

6 References

Yuki Arase and Ming Zhou. Machine translation detection from monolingual web-text. In *ACL*, 2013.

Mona Baker, Gill Francis, and Elena Tognini-Bonelli. *'Corpus Linguistics and Translation Studies: Implications and Applications'*. John Benjamins Publishing Company, Netherlands, 1993.

Marco Baroni and Silvia Bernardini. A preliminary analysis of collocational differences in monolingual comparable corpora. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 82–91, 2003.

Marco Baroni and Silvia Bernardini. A new approach to the study of translationese: Machine-learning the difference between original and translated text. *Literary and Linguistic Computing*, 21(3):259–274, 2006.

Milan Bouchet-Valat. *SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library*, 2014. R package version 0.5.1.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working set selection using second order information for training support vector machines. *J. Mach. Learn. Res.*, 6:1889–1918, December 2005.

Martin Gellerstam. Translationese in swedish novels translated from english. In L. Wollin and H. Lindquist, editors, *Translation Studies in Scandinavia*, Lund Studies in English, pages 88–95. CWK Gleerup, 1986.

Thorsten Joachims. *Text categorization with Support Vector Machines: Learning with many relevant features*, pages 137–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

David Kurokawa, Cyril Goutte, and Pierre Isabelle. Automatic detection of translated text and its impact on machine translation. In *Proceedings of MT-Summit XII*, pages 81–88, 2009.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2015. R package version 1.6-7.

Maeve Olohan. Spelling out the optionals in translation: a corpus study. *UCREL Technical Papers*, 13:423–432, 2001.

Karolina Piaseckienė and Marijus Radavičius. Empirical bayes estimators of structural distribution of words in lithuanian texts. *Nonlinear Analysis: Modelling and Control*, 19:611–625, 2014.

M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

Tiina Puurtinen. Genre-specific features of translationese? linguistic differences between translated and non-translated finnish children's literature. *Literary and Linguistic Computing*, 18(4):389–406, 2003.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

Federico Zanettin. Corpus methods for descriptive translation studies. *Procedia - Social and Behavioral Sciences*, 95:20 – 32, 2013.

George Zipf. *The Psychobiology of Language: An Introduction to Dynamic Philology*. M.I.T. Press, Cambridge, Mass., 1935.

Appendices

A SVM Feature Weights

Feature weights of the augmented SVM model trained on *delfi.lt* corpus

Variable	Feature weight	Variable	Feature weight
Word length 5	8.452	jog	-0.1465
Word length 6	-3.808	vos	0.1379
Word length 4	-3.538	pas	-0.1334
Word length 12	-3.515	mus	-0.1146
Word length 3	3.051	ties	-0.1086
kur	2.799	lyg	-0.1007
Word length 9	2.694	dar	-0.09872
Word frequency 10% Q	-2.123	iki	-0.09489
Word length 10	-2.068	tad	-0.09329
kad	2.023	abu	0.08822
Word instances 2	1.789	bent	-0.07
Word frequency slope	1.778	ant	0.06937
Word instances 5	1.772	tol	-0.06409
Word length 11	-1.714	tiesiog	-0.05503
per	1.662	kiek	-0.04902
Word instances 4	1.61	juo	-0.0447
Stop words ratio	1.549	pernelyg	0.04402
pat	-1.452	pirm	-0.04174
kai	1.281	Word length 8	-0.04097
Word frequency 20% Q	-1.22	gal	0.03966
Word instances 1	1.056	link	-0.0379
Word instances 3	1.016	abi	0.03357
Word frequency 40% Q	-1.015	kuomet	-0.02922
tai	-1.01	apie	0.02767
Word frequency 30% Q	-0.9985	tegul	-0.02646
bei	-0.9974	kuri	-0.02392
vien	0.9936	jei	-0.0214

mes	-0.984	sau	-0.01993
man	-0.955	jus	-0.01468
tarp	0.9336	arba	-0.01308
jis	0.8383	abipus	-0.013
Word frequency 50% Q	-0.8368	ana	0.01275
Word frequency 90% Q	-0.8007	dėl	0.01137
Word frequency 70% Q	-0.7832	toks	0.008279
Word frequency 80% Q	-0.7614	gan	0.008279
Word frequency 60% Q	-0.7325	argi	0.006482
tik	-0.6962	nebent	0.005211
prie	0.6466	joks	0.004787
bet	-0.6181	tau	-0.004767
nuo	0.525	juk	-0.003423
kas	-0.488	antai	0.002437
Word length 7	0.4869	arti	0.002141
Word length slope	-0.4845	beveik	-0.002045
nes	-0.4817	nekaip	-0.001927
vis	-0.4426	mano	0.001852
pagal	-0.4091	pro	-0.001787
pasak	-0.3597	nejau	0.001674
taip	-0.3281	palei	-0.001549
mat	-0.3156	abiem	0.001346
aplink	-0.274	idant	0.001345
nei	-0.239	pati	-0.001227
anot	-0.2066	lai	-0.001201
tas	-0.2047	ogi	0.001178
net	-0.1815	sulig	0.0009375
kol	-0.1778	savo	0.0007937
lig	-0.1728	vėl	-0.0004622
kaip	0.1521	anaiptol	0.0004589
itin	0.1495		

Feature weights of the augmented SVM model trained on *emokykla* corpus

Variable	Feature weight	Variable	Feature weight
Word length 3	7.2140663	Word frequency 40% Q	-0.0733455
Word length 4	-3.8468142	ties	0.0726511
Stop words ratio	3.731759	jus	0.0657136
Word length 6	-1.5053784	arba	-0.0607075
man	1.4771795	juk	0.0604361
jis	1.1484048	gal	-0.0575576
taip	1.1287052	nuo	0.0562169
kad	1.0959533	lig	0.0561943
Word length 8	-1.0273199	pro	0.0548206
Word instances 1	-0.9263284	paskum	-0.0537423
ant	-0.7834825	mus	-0.0512752
kai	0.7803245	Word frequency 60% Q	-0.0502833
kur	0.7284102	argi	0.0467182
Word length 7	-0.6657842	anei	0.0435274
nes	0.5127298	pat	-0.0363562
nei	-0.4815024	link	0.035713
pasak	0.4367507	vos	0.0319594
net	-0.4336442	aplink	-0.0306707
Word length 5	0.416792	iki	-0.0265476
Word instances 3	0.3805412	kiek	-0.0261805
Word length 10	-0.3740669	Word instances 2	-0.0260111
kaip	-0.3697702	ogi	-0.0244959
bet	0.3612027	apie	0.0243285
Word length 11	-0.3604751	Word frequency 50% Q	-0.0221349
vien	-0.3457962	sulig	0.0215145
pas	-0.3247605	itin	-0.0207395
tad	0.3221373	irgi	-0.0153391
per	-0.2892904	kuomet	0.0138155
Word instances 4	0.2737599	nejau	-0.0132968
tik	0.2710726	vai	-0.012729

Word frequency 10% Q	-0.2604353	kuri	0.0123015
tai	-0.2590382	oho	0.0121548
lyg	-0.2505333	nekaip	0.0115019
Word frequency 20% Q	-0.2480446	pats	-0.0111897
Word length slope	-0.2419064	tegul	-0.0100107
mes	-0.2251832	pirm	-0.0092568
vis	-0.2199306	nebent	0.0087956
sau	-0.2088229	palei	0.0085261
jei	0.2011363	anaiptol	0.0081303
dar	-0.1858997	pernelyg	-0.0070333
prie	-0.1810224	Word length 12	-0.0065967
Word frequency 30% Q	-0.1791982	mano	0.0061728
Word length 9	0.1555771	arti	-0.005
Word frequency 90% Q	-0.1508245	anot	0.0048527
kas	-0.1473132	bent	-0.00483
tol	-0.1388714	ana	0.0047619
kol	0.1368373	juodu	0.004712
Word frequency 80% Q	-0.1245176	toks	0.0046675
Word instances 5	0.1189442	abipus	-0.0046512
Word frequency 70% Q	-0.1176146	koks	0.0042254
ane	0.1151902	pati	-0.0042105
Word frequency slope	0.1140447	vau	-0.0040984
tiesiog	0.1074615	abiem	0.0040486
jog	-0.1038408	viduj	-0.0035124
mat	-0.1016556	lai	-0.002834
pagal	0.1009532	abi	0.0019778
tas	-0.0977151	abu	0.0016649
tau	-0.0961615	aha	0.0009178
bei	-0.0935628	ech	-0.0008418
juo	0.0920938	anas	-0.0006552
tarp	-0.091313	antai	0.0001295
gan	-0.0835126		