Exploring Features for Named Entity Recognition in Lithuanian Text Corpus

Jurgita Kapočiūtė-Dzikienė[1], *Anders Nøklestad*[2],

Janne Bondi Johannessen[2], *Algis Krupavičius*[1]

(1) KAUNAS UNIVERSITY OF TECHNOLOGY, K. Donelaičio 73, LT-44249, Kaunas, Lithuania
(2) UNIVERSITY OF OSLO, P.O. Box 1102, Blindern, N-0317 Oslo, Norway

Jurgita.Kapociute-Dzikiene@ktu.lt, Anders.Noklestad@iln.uio.no,
J.B.Johannessen@iln.uio.no, pvai@ktu.lt

ABSTRACT

Despite the existence of effective methods that solve named entity recognition tasks for such widely used languages as English, there is no clear answer which methods are the most suitable for languages that are substantially different. In this paper we attempt to solve a named entity recognition task for Lithuanian, using a supervised machine learning approach and exploring different sets of features in terms of orthographic and grammatical information, different windows, etc. Although the performance is significantly higher when language dependent features based on gazetteer lookup and automatic grammatical tools (part-of-speech tagger, lemmatizer or stemmer) are taken into account; we demonstrate that the performance does not degrade when features based on grammatical tools are replaced with affix information only. The best results (micro-averaged F-score=0.895) were obtained using all available features, but the results decreased by only 0.002 when features based on grammatical tools were omitted.

KEYWORDS: Named entity recognition and classification, supervised machine learning, Lithuanian.

1 Introduction

The objective of Named Entity Recognition (NER) is to allocate and classify tokens in texts into predefined categories, such as person names, location names, organizations, etc. NER is a subtask of many natural language processing applications, i.e. in information extraction, machine translation, question answering, etc.

The methods used to solve NER task fall into three main categories: hand-crafted, machine learning and hybrid. Hand-crafted approaches are based on an analysis of the application domain and the manual construction of gazetteers and sets of patterns capable of taking named entity recognition and classification decisions. Hand-crafted methods assure high accuracy, but at the same time they have low portability to new domains: i.e. changing domains require expert intervention and manual re-creation of rules. Thus, very often machine learning methods are selected instead of the rule-based ones. For machine learning approaches NER is usually interpreted as classification task, where tokens are classified into categories with non-named and different named entity types. Rules (defined as the model) by these methods are built automatically after observation and generalization over the characteristics extracted from the text. Supervised machine learning methods for model creation use texts prepared by the domain experts with manually assigned classes (identifying particular named entity type or not a named entity) to all tokens. Semi-supervised machine learning methods for NER are usually based on a bootstrapping learning scheme: an initial small set of tokens with predefined categories (types of named entities) is used as the seed to build initial model, then all tokens in the texts are classified using that seed model and the most confident classifications (recognized named entities) are added to the training data and the process is iterated. Supervised machine learning methods outperform semi-supervised methods in terms of accuracy, but the strongest point of semi-supervised methods is that they do not require large annotated corpora. Hybrid methods usually combine the strongest points of both hand-crafted and machine learning methods.

In this paper we are solving the named entity recognition task using a supervised machine learning technique. Our focus is to find the most informative features for the Standard Lithuanian language yielding the best named entity classification results for person, location and organization names. Our experiments include different sets of language independent and dependent features based on orthographic and grammatical information, different windows (tokens before and after a current one), etc. We also assess whether it is possible to perform named entity classification effectively without resorting to external grammatical tools such as part-of-speech taggers, lemmatizers or stemmers, because they have limited availability, and are slow and unreliable when identifying proper names in Lithuanian texts.

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 named entity classification experiments in Section 4 and present the results in Section 5. We discuss these results in Section 6 and conclude in Section 7.

2 Related work

The NER research for English started in early 1990s and this task was successfully solved (for a review see Nadeu & Sekine, 2007), because NER achieved human annotation performance (Sundheim, 1995) in terms of accuracy. Thus current works for English mostly focus on the

improvement of other important metrics, e.g. speed: i.e. Al-Rfou' & Skiena (2012) demonstrated that the speed of a NER tagger can be increased after improvement of tokenizer and part-of-speech tagger that are used as the sub-modules in NER pipeline.

The reason why methods achieve very high accuracy on English texts is because they are based on rich external resources (such as wide range of annotated corpora and gazetteers, accurate part-of-speech taggers, etc.) that are not always available for the other languages. Besides, the English language is not as complex as highly inflective, morphologically rich and ambiguous languages; moreover it does not suffer from lack of capitalization problems for proper nouns. Thus, it can be concluded that NER task for each language requires its detailed analysis and adaptation.

However, some generalizations can be made, e.g. language dependent features (based on gazetteers or morphological information) always increase NER accuracy; morphological information is inevitable for morphologically rich languages in order for the NER method to be comparable to the state-of-the-art methods for English. The importance of accurate morphological analyzers in NER task was revealed by Hasan et al. (2009): NER results for Bengali were significantly improved by simple refining non-accurate part-of-speech tagger (the part-of-speech tagger was improved during extraction of information from Wikipedia that in turn improved recognizer). Morphological information in NER methods can be either incorporated into rules or used as features (part-of-speech tags, lemmas). Examples of such methods are: rule-based methods for Arabic (Elsebai et al., 2009), Danish and Norwegian (Johannessen et al., 2005), Russian (Popov et al., 2004), and Urdu (Singh et al., 2012); supervised machine learning methods for Bulgarian (Georgiev et al., 2009), Dutch (Desmet & Hoste, 2010), Norwegian (Haaland, 2008; Nøklestad, 2009), and Polish (Marcinczuk et al., 2011; Marcinczuk & Janicki, 2012); hybrid method for Arabic (Mai & Khaled, 2012). Hasan et al. (2009) also proved that affixes are very important in solving NER task for morphologically rich languages. Affixes helped to increase NER results for Bengali, Bulgarian, Polish, etc.

NER task for Lithuanian was previously solved using two different methods: rule-based (for person and location names, abbreviations, date and time) (Kapočiūtė & Raškinis, 2005) and semi-supervised machine learning (bootstrapping) (for person, location and organization names, product, date, time and money) (Pinnis, 2012). Unfortunately, we cannot give any examples of experiments based on supervised machine learning techniques for NER. Consequently, this paper will be the first attempt at finding an accurate supervised machine learning method (in terms of the most informative features) for NER in Lithuanian texts.

3 Lithuanian language challenges

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). Lithuanian nominal words (nouns or adjectives that are used as person, location or organization names) can be inflected by 7 cases, 2 genders, and 2 numbers. Various inflection forms are expressed by different endings. Person names can have either masculine or feminine gender and are inflected by case and number. To exemplify: the possible inflection forms of masculine person name *Jonas* (John) are: *Jonas* (singular nominative), *Jono* (singular genitive), *Jonui* (singular dative), *Joną* (singular accusative), *Jonu* (singular ablative), *Jone* (singular locative), *Jonai* (singular and plural vocative; plural nominative), *Jonų* (plural genitive), *Jonams* (plural dative), *Jonus* (plural accusative), *Jonais* (plural ablative), *Jonuose* (plural locative). The location

names or organizations can have either masculine or feminine gender and singular or plural number, and are inflected by case. E.g. *Lietuva* (Lithuania) is singular feminine noun; *Aukštieji Tatrai* (High Tatres) is composed of plural masculine adjective *Aukštieji* (High) and plural masculine noun *Tatrai* (Tatres), where all words are inflected only by 7 cases. In the Lithuanian language there are 12 inflection paradigms for nouns, and 9 inflection paradigms for adjectives.

The same inflection rules are used to inflect foreign person names in Standard Lithuanian texts. Endings can either be directly attached to the end of the noun e.g. *Tonis Blairas* (Tony Blair) or can be added after an apostrophe e.g. *Tony'is Blair'as* (Tony Blair).

In Lithuanian, male and female surnames are different; moreover, in the surnames of married and unmarried women the suffixes are also different. For example, given the male surname *Karalius*, the appropriate female surname of an unmarried woman is *Karaliūtė*, and of a married woman – *Karalienė*. Besides, women can have surnames that do not reveal their marital status, e.g. *Karalė*, or keep both surnames – typically written with a hyphen, e.g. *Petraitytė-Karalienė* or *Petraitytė-Karalė*.

In Lithuanian organization names only the first word is capitalized in organization names (the other words are capitalized only if they happen to be a person or location name). E.g. *Lietuvos mokslo taryba* (Research Council of Lithuania), *Kauno Maironio gimnazija* (Kaunas Maironis Gymnasium) *Maironio* is a person name that is capitalized. The names of the companies in Lithuanian texts are written in quotation-marks. Despite the fact that the words in organization names are written in lower case (starting from the second word); acronyms are written in upper case, e.g. the acronym for *Kauno technologijos universitetas* (Kaunas University of Technology) is abbreviated to *KTU*.

Lithuanian is a synthetic language; and the word order in the sentence is absolutely free. As in ancient Indo-European languages, in Lithuanian the word order does not perform a grammatical, but a notional, function: i.e., the same sentence will be grammatically correct regardless of word order, but the meaning (things that have to be highlighted) will be a bit different. Thus, any named entity can appear in any place of the sentence. Despite this, the words that act as triggers (words that directly signal a named entity) or modifiers (words that limit or qualify the sense of named entity) are located close to the named entity they describe. E.g. in *dr. Jonas Basanavičius* (dr. Jonas Basanavičius) *dr.* is the trigger that identifies the following word as the person name; in *gyventi Lietuvoje* or *Lietuvoje gyventi* (to live in Lithuania) *gyventi* (to live in) is the modifier that qualifies the named entity *Lietuvoje* as a location name.

47% of Lithuanian words or word forms are ambiguous (Zinkevičius et al., 2005); named entities are no exception. Ambiguities are often between common nouns/adjectives written at the beginning of a sentence and person/location/organization names, e.g. *Raudonoji* (Red) can be either a common word in *Raudonoji arbata* (Red tea) or a location name in *Raudonoji jūra* (Red sea); *Rūta* can be either the name of the person or the common word "rue". Ambiguities can also emerge between person names and location names, e.g. *Roma* is the name of the person or location name "Rome"). Depending on the meaning, named entities can change their type, e.g. *Maironis* is a person name, but in *Maironio gatvė* (Maironis street) the word is treated as a location; *Vilnius* is a location name, but in *Vilniaus universitetas* (Vilnius University) it is treated as part of an organization name. This analysis reflects what is called "Function over Form" in Johannessen et al. (2005).

4 Methodology

4.1 Feature extraction

Supervised machine learning systems cannot be directly trained on a corpus annotated with named entities. The corpus has to be transformed into a collection of instances. Usually instances are generated for consecutive tokens excluding punctuation marks (e.g. from string *J. Petraitytė-Karalienė* would be generated three instances "J", "Petraitytė", and "Karalienė"), sometimes punctuation marks are stored as part of tokens (e.g. from string *J. Petraitytė-Karalienė* would be generated: "J.", "Petraitytė-", and "Karalienė"). Since punctuation marks carry a lot of information about named entities in Lithuanian language and their loss would be harmful (e.g. when recognizing classified named entities in plain text) in our experiments we treat punctuation marks as tokens, too – i.e. as separate instances (thus, e.g. from string *J. Petraitytė-Karalienė* would be generated five instances: "J", ".", "Petraitytė", "-", and "Karalienė"). During such transformation the order of tokens and punctuation marks in the collection is not changed.

All instances that are used in the classification process are represented as vectors, each composed of the class label (identifying none or a particular type of named entity) and the list of features (where the first one describes original token – "J", "Petraitytė", "Karalė" or punctuation mark – ".", "-"). Considering Lithuanian language properties and available resources, we have made the list of features that should be effective in solving named entity classification task. All features can be grouped into two main categories: language independent and language dependent. Language independent features are very general, based only on the orthographic information directly available in the corpus; language dependent features resort to external resources such as special purpose grammatical tools (part-of-speech tagger, lemmatizer, stemmer) or gazetteers (gazetteer of person names, gazetteer of location names).

Below we present a list of all the features that were used in our experiments.

Language independent (basic and orthographic) features:

- *t* – the original token or punctuation mark.
- *Lowercase* (*t*) – *t* in lower case letters (e.g. *Lowercase* (*Lietuva*) = *lietuva*). The usage of this feature decreases the number of tokens that can be treated differently because of case sensitivity (especially relevant for common words at the beginning of the sentence).
- *IsFirstUpper*(*t*) – Boolean indicator that determines if the first character of *t* is capitalized (e.g. *IsFirstUpper*(*Lietuva*) = *true*).
- *Acronym*(*t*) – Boolean indicator that determines if all characters in *t* are capitalized (e.g. *Acronym*(*KTU*) = *true*). This feature should help to identify organizations abbreviated as acronyms.
- *Number*(*t*) – Boolean indicator that determines if *t* is a number (e.g. *Number*(*Sk1*) = *false*).
- *Length* (*t*) – numeral indicator that determines the length of *t*.
- *Prefix-n*(*t*) – affix that determines the first *n* (in our experiments $3 \leq n \leq 5$) characters of *t* in lower case (e.g. *Prefix*-5(*Šiauliai*) = *šiaul*). This feature should help to extract stable parts of tokens and thus get rid of inflectional Lithuanian endings.
- *Suffix-n*(*t*) – affix that determines the last *n* (in our experiments $3 \leq n \leq 5$) characters of *t* in lower case (e.g. *Suffix*-5(*Šiauliuose*) = *iuose*; *Suffix*-5(*Šakiuose*) = *iuose*). This feature should help to extract inflectional Lithuanian endings. E.g. the roots in "Šiauliuose" and "Šakiuose" are different, but they share the same ending in locative case "iuose".

Language dependent features:

- *Lemma*(*t*) – the lemma of *t* (e.g. *Lemma*(*Lietuvoje*) = *Lietuva*). The lemmatizer transforms recognized words into their major form (by replacing appropriate endings, but not touching affixes); the common words also are transformed into lower-case letters.
- *POS*(*t*) – the part-of-speech tag of *t* (e.g. *POS*(*Lietuvoje*) = *noun*).
- *Stem*(*t*) – the stem of *t* (e.g. *Stem*(*Lietuvoje*) = *Lietuv*). The stemmer eliminates inflectional ending (and some other suffixes) of the input word.
- *IsPERGaz*(*t*) – A Boolean indicator that determines if *t* is in the gazetteer of person names.
- *IsLOCGaz*(*t*) – A string indicator that determines if *t* is in the gazetteer of location names, and if so, then if it is the beginning or any other word in the location name (e.g. for *gyvenu Didžiojoje Britanijoje* (I live in Great Britain), *IsLOCGaz*(*gyvenu*) = *no*; *IsLOCGaz*(*Didžiojoje*) = *yesB*; *IsLOCGaz*(*Britanijoje*) = *yesI*).

Class labels:

The corpus contains three types of named entities for person names, location names and organizations. Class labels are represented in BIO notation, thus B-PER, B-LOC, B-ORG indicate the first token in person, location, organization named entity, respectively; I-PER, I-LOC, I-ORG indicate any other token (except the first one) in person, location, organization named entity, respectively; O indicates any other token (not in named entities).

4.2 Dataset

All the texts for the NER task were taken from the Vytautas Magnus University corpus (Marcinkevičienė, 2000). We aimed to avoid domain adaptation problem, thus texts were taken from several domains (as it was done by Kitoogo et al. (2008)). The created dataset in total contained ~ 0.5 million running words and was composed from the texts representing Standard Lithuanian and taken from 5 different domains: popular periodic, local newspapers, republican newspapers, parliamentary transcripts and legislative texts (the distribution of running words over different domains is presented in FIGURE 1).

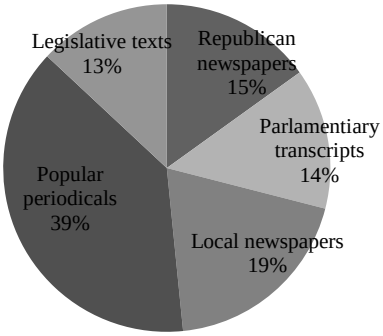

FIGURE 1 – The distribution of words (used in our NER experiments) over 5 different domains.

All texts used in our named entity classification experiments were manually annotated with person, location and organization names (the distribution of words into named entities over different domains is presented in FIGURE 2.). Named entities were annotated considering "Function over Form" annotation manner, thus the same named entity could be tagged differently

in different context, e.g. in *Maironis skaito savo poeziją* (Maironis reads his poetry) *Maironis* is annotated as person name; in *gyvenu Maironio gatvėje* (live in Maironis street) *Maironio* is annotated as location name; in *Skaitau knygą "Apie Maironį"* (I read the book "About Maironis") *Maironį* is not annotated as named entity because it is neither person, nor location, nor organization name, but it is the title of the book. Besides, inner named entities were not annotated. E.g. in *Vilniaus universitetas* (Vilnius University) inner location name *Vilniaus* was ignored; instead of it both words were annotated as single organization name.

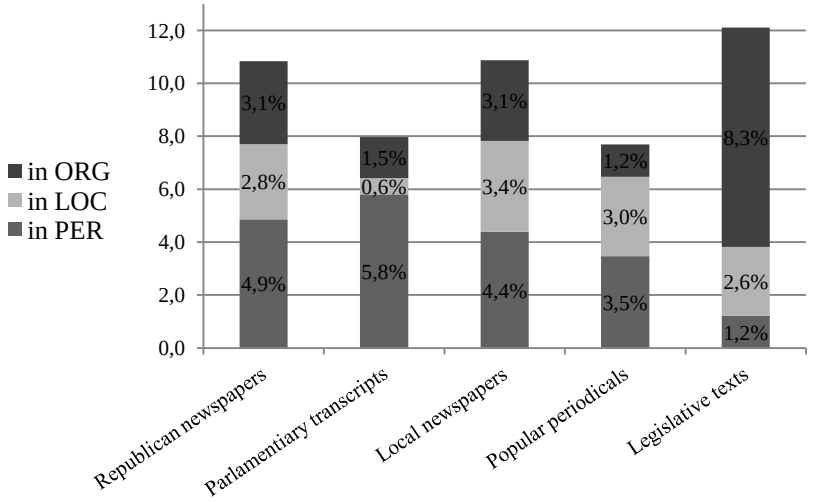

FIGURE 2 – The distribution of words into named entities (person names – *in PER*, location names – *in LOC*, organization names – *in ORG*) over 5 different domains.

Annotated texts were transformed into collection of instances. Statistics about number of instances (tokens/punctuation marks) belonging to their appropriate classes is presented in TABLE 1.

Class label	Number of instances
B-PER	10902
I-PER	13861
B-LOC	12152
I-LOC	1322
B-ORG	5937
I-ORG	10482
All NEs	54656
O	577771
Total	632427

TABLE 1 – Number of instances for all classes in the dataset used in our NER experiments.

4.3 Formal description of the task

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

Let D be our dataset containing instances I, where $\forall I \in D$.

Let C be a finite set of classes: $C = \{c_1, c_2, \ldots, c_N\}$. In our case $2 < N << \infty$ – i.e. we have a multi-class classification problem, because C = {B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, O} and N=7.

Let I be the instance described by feature vector v with the appropriate class label c, thus all instances in the dataset are represented as $I=\langle v, c\rangle$. We do not consider inner named entities; therefore our dataset contains single-labeled instances only: i.e. any instance cannot be attached to more than one class c.

Let function γ be a classification function mapping instances to classes, $\gamma: I \rightarrow C$. Function γ determines how I was labeled with c. In our dataset annotation of named entities was performed by the domain expert.

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

4.4 Experimental setup

In order to find the most informative features for the NER task on Lithuanian texts, we performed experiments based on:

- Different windows $[t_{x\text{-}m}, t_{x+m}]$. E.g. window $[t_{x=0}]$ contains only one element – current instance $t_{x=0}$; window $[t_{x-1}, t_{x=0}]$ contains current instance $t_{x=0}$ and instance t_{x-1} that is before $t_{x=0}$; window $[t_{x=0}, t_{x+1}]$ contains current instance $t_{x=0}$ and instance t_{x+1} that is after $t_{x=0}$, etc. We restricted m to 3 and thus experimented with 12 different windows in total.
- 9 different sets of features (see TABLE 2). The sets of features No. 1 and No. 5 contain only language independent features; all other sets of features contain both language independent and language dependent features.
  Sets No. 2 and No. 6 – No. 9 also contain gazetteer lookup features. These feature values were generated considering information stored in the gazetteers. To avoid redundancy when storing named entities in all their inflectional forms as separate instances, distinct named entities were represented by stem and their appropriate inflectional paradigm identifier. Thus, the gazetteer of person names consists of ~7 thousand individual person names (e.g. *Jon:*1 retains named entity in all its inflectional forms: *Jonas*, *Jono*, *Jonui*, etc.); the gazetteer of location names consists of ~2.2 thousand location names: individual (the same as for person names) or compound (e.g. *Didži*:2 *Britanij*: 3 retains this named entity in all its inflectional forms: *Didžioji Britanija*, *Didžiosios Britanijos*, *Didžiajai Britanijai*, etc.).
  Sets No. 3, No. 6, and No. 9 also contain part-of-speech tags and lemmas as features. The tokens were part-of-speech tagged and lemmatized using the dictionary-based morphological analyzer-lemmatizer "Lemuoklis" (Zinkevičius, 2000; Daudaravičius et al., 2007). "Lemuoklis" also solves morphological disambiguation problems and achieves ~95% accuracy on standard texts. Despite "Lemuoklis" being very accurate on common words; its weakness is unknown proper names (Zinkevičius et al., 2005). In our experiments "Lemuoklis" was unable to recognize 30.6% of the words as either named entities or common words, and classified them as "unknown"; it recognized only 34.4% of words as proper names, the rest 35% words it recognized as common words (for the detailed statistics see FIGURE 3).

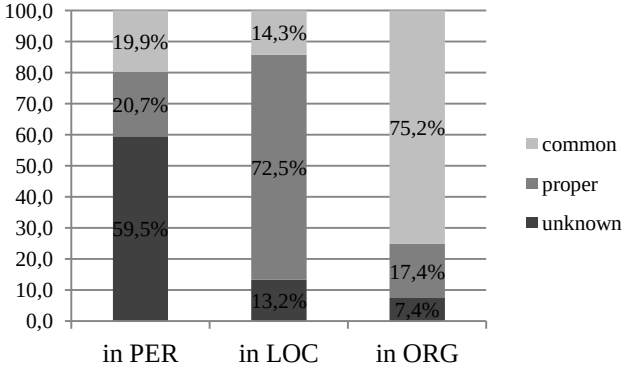

FIGURE 3 – The percentage distribution of words in named entities (person names – *in PER*, location names – *in LOC*, organization names – *in ORG*) that were recognized by "Lemuoklis" as proper names (*proper*), common words (*common*) or not recognized at all (*unknown*).

		Feature sets								
		***No.*1**	***No.*2**	***No.*3**	***No.*4**	***No.*5**	***No.*6**	***No.*7**	***No.*8**	***No.*9**
Features	***t***	×	×	×	×	×	×	×	×	×
	***Lowercase*(*t*)**	×	×	×	×	×	×	×	×	×
	***IsFirstUpper*(*t*)**	×	×	×	×	×	×	×	×	×
	***Acronym*(*t*)**	×	×	×	×	×	×	×	×	×
	***Number*(*t*)**	×	×	×	×	×	×	×	×	×
	***Length*(*t*)**	×	×	×	×	×	×	×	×	×
	***Prefix*-3-5(*t*)**					×			×	×
	***Suffix*-3-5(*t*)**					×			×	×
	***Lemma*(*t*)**			×			×			×
	***POS*(*t*)**			×			×			×
	***Stem*(*t*)**				×			×		×
	***IsPERGaz*(*t*)**		×				×	×	×	×
	***IsLOCGaz*(*t*)**		×				×	×	×	×

TABLE 2 – Different sets of features used in our NER experiments.

Sets No. 4, No. 7, and No. 9 also contain stems of tokens as features. The tokens were stemmed using a Lithuanian stemmer (Krilavičius & Medelis, 2010) based on the Porter stemming algorithm (Willet, 2006). The Lithuanian stemmer eliminates all endings and also some suffixes (but do not touch prefixes). The stemmer is rule-based, thus it can cope with proper names as well as with the other words. But in some cases stemming can cause the loss of meaning and ambiguity, e.g. *Lietuva* (Lithuania), *Lietuvis* (Lithuanian) will both be stemmed to the same *Lietuv*.

Sets No. 5, No. 8, and No. 9 contain affix features: the first and the last 3, 4 and 5 characters of each token.

As previously described, in our experiments we used 9 sets of features and 12 windows, that is 9×12=108 different experiments in total.

Before the experiments we made two hypotheses.

Our first hypothesis is that language dependent features (morphological and gazetteer lookup) should increase the accuracy of NER on Lithuanian texts (the same as for Bengali ((Hasan et al., 2009), Polish (Marcinczuk & Janicki, 2012), and other languages). We expect that gazetteers should solve the disambiguation problems between named entities and not named entities or between different types of named entities; and grammatical tools (part-of-speech tagger/lemmatizer and stemmer) will cope with the inflections of the Lithuanian language (that in turn should reduce the sparseness of feature space and should lead to the construction of more reliable model).

The grammatical tools seem necessary due to the importance of inflection in Lithuanian, but they are unreliable when identifying proper names (Zinkevičius et al., 2005). Therefore, our second hypothesis is that the stable parts of the words and the influence of inflection can be captured in a different way – i.e. by affix information. We expect that with affix features (*Prefix*-3-5(*t*) and *Suffix*-3-5(*t*)), the NER method will achieve similar accuracy as with features based on part-of-speech information, lemmas or stems.

4.5 Classification

Our attempt was to find a method Γ which could create the model γ' the best approximating γ. Since a NER task using supervised machine learning approach has never been undertaken for Lithuanian, we had no knowledge which method is the most appropriate. Therefore we selected Conditional Random Fields (CRFs) method (a detailed description of CRFs is presented in Lafferty et al., 2001) as Γ, because it is one of the popular techniques utilized for the NER task (Nadeau & Sekine, 2007) outperforming many other popular methods, such as Hidden Markov Models (HMMs), Maximum Entropy Markov Models (MEMMs), and etc.

The implementation of CRFs we used in our experiments is CRF++ version 0.57[1], which requires a template file (with the determined features that are considered) and datasets for training and testing.

5 Results

All results reported below are based on 10 fold cross-validation (the biggest class *O* for non-named entities is treated as the negative class in all calculations).

The experiments with different windows revealed that the best results are obtained with window $[t_{x-2}; t_{x+1}]$, except for the feature sets No. 2 and No. 3, where the best results were obtained with $[t_{x-1}; t_{x+1}]$. Despite that, the improvement on micro-averaged F-scores was less than 0.0006. FIGURE 4 reports the results obtained using different windows with the feature set No. 1.

We experimented with different *n* values of *Prefix*-*n*(*t*) and *Suffix*-*n*(*t*) also, but the best results were reported with $3 \le n \le 5$: with feature set No. 5 micro-averaged F-score using n=3 was 0.8556, with $3 \le n \le 4$ – 0.8614, the peak 0.8645 was with $3 \le n \le 5$, the higher *n* values degraded the results. Besides, using only *Prefix*-3-5(*t*) or only *Suffix*-3-5(*t*) was not effective also: e.g. with the feature set No. 5 micro-averaged F-score when using only *Prefix*-3-5(*t*) was 0.8549, when using only *Suffix*-3-5(*t*) – 0.8517, when using both – 0.8645.

[1] http://crfpp.sourceforge.net

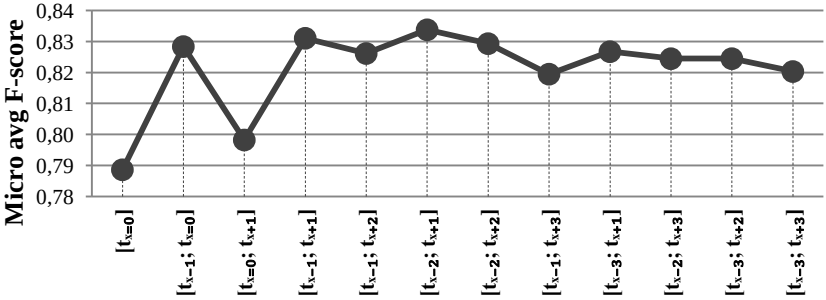

FIGURE 4 – Micro-averaged F-scores obtained with different windows when feature set No. 1.

The results obtained using window $[t_{x-2}; t_{x+1}]$ and $3 \leq n \leq 5$ for all 9 feature sets are reported in TABLE 3.

		Precision		Recall		F-score	
		micro avg	***macro avg***	***micro avg***	***macro avg***	***micro avg***	***macro avg***
Feature sets	***No.*1**	0.8655	0.8648	0.8043	0.8061	0.8338	0.8342
	***No.*2**	0.9159	0.9174	0.8587	0.8643	0.8864	0.8897
	***No.*3**	0.8738	0.8745	0.8353	0.8380	0.8541	0.8556
	***No.*4**	0.8748	0.8744	0.8122	0.8141	0.8424	0.8429
	***No.*5**	0.8933	0.8923	0.8375	0.8391	0.8645	0.8647
	***No.*6**	0.9141	0.9169	0.8732	0.8785	0.8932	0.8970
	***No.*7**	***0.9177***	***0.9195***	0.8636	0.8691	0.8899	0.8933
	***No.*8**	0.9168	0.9176	0.8708	0.8754	0.8932	0.8957
	***No.*9**	0.9140	0.9161	***0.8773***	***0.8817***	***0.8953***	***0.8984***

TABLE 3 – Micro/macro averaged precision/recall/f-scores for all sets of features with window $[t_{x-2}; t_{x+1}]$ and $3 \leq n \leq 5$.

To determine whether the performances of the classifier trained on different feature sets were significantly different from each other, we performed approximate randomization testing (Yeh, 2000). In all approximate randomization testing experiments we used 1000 shuffles.

The differences between many experiments are statistically significant to a very high degree ($p = 0.00099$) with some exceptions, presented in FIGURE 5: i.e. differences are statistically significant to a high degree when $0.00099 < p \leq 0.01$; statistically significant when $0.01 < p \leq 0.05$; and not significant when $p > 0.05$.

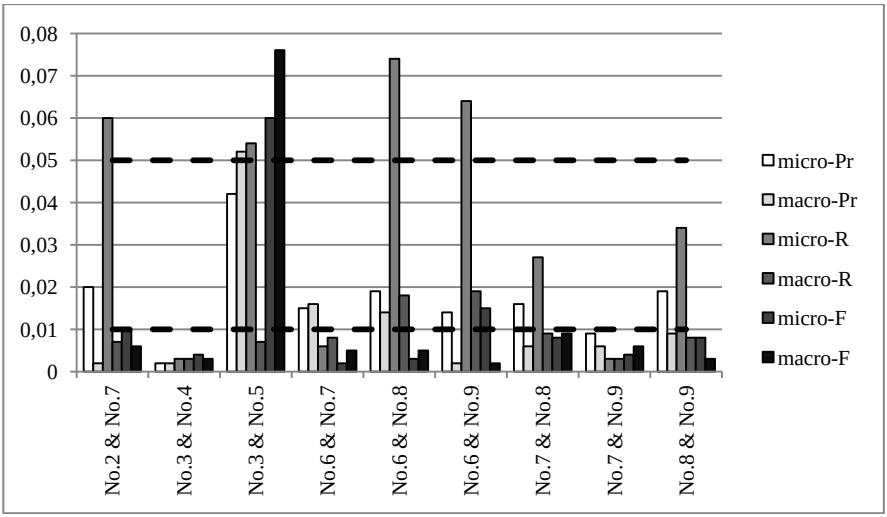

FIGURE 5 – *p* scores indicating statistical differences between the results (micro/macro precision (Pr)/recall (R)/F-score (F)) obtained on different feature sets.

6 Discussion

A glance at TABLE 3 in Section 5 shows that our first hypothesis is confirmed: language independent features are easily beaten with language dependent features. These findings agree with the findings obtained for other languages: Bengali (Hasan et al., 2009), Bulgarian (Georgiev et al., 2009), etc.

As it is presented in TABLE 3, the usage of gazetteer lookup features in No. 2 increased the results of No. 1 by 0.0526 in micro-averaged F-score, despite the fact that a gazetteer of organization names is not available for Lithuanian (only of person and location names). The features obtained using grammatical tools also increased NER results (compared with No. 1 which is based on orthographic information only): the increment in micro-averaged F-score using part-of-speech tagger/lemmatizer in No. 3 is 0.0203; whereas stemmer in No. 4 caused only a small boost in performance – increment in micro-average F-score is only 0.0086. The lemmatizer and stemmer cope with inflectional endings of Lithuanian language by reducing the number of distinct tokens. Thus a reduced number of feature values leads to the construction of more reliable and, as it can be concluded from the results, to more accurate models. The reason why part-of-speech tagging/lemmatization improves performance over stemming is that the part-of-speech tagger/lemmatizer is more accurate compared with stemmer, besides it preserves inflectional information that is important in Lithuanian language (the ending is not lost during lemmatization, only changed into the major form). The results obtained in No. 3 (with the part-of-speech tagger/lemmatizer) are worse than in No. 2 (with the gazetteer lookup), because the part-of-speech tagger/lemmatizer is not robust in recognizing proper names, as well as not being capable of determining types of named entities.

Experiments with gazetteer lookup + part-of-speech tagging/lemmatizing in No. 6, or gazetteer lookup + stemming in No. 7, yielded even better results compared with the results when they

were used separately (as in No. 2, No. 3, No. 4). Despite the result in No. 6 exceeded No. 7 by 0.0033 in micro-average F-score, the difference was proved to be statistically significant.

More interestingly, the results also support our second hypothesis: grammatical tools (part-of-speech tagger/lemmatizer and stemmer) can be replaced with affix information. The same effect can be achieved by automatically slicing *n* characters from the beginning and the end of the token as it is done with *Prefix*-3-5(*t*) and *Suffix*-3-5(*t*). Experiments with *Prefix*-3-5(*t*) and *Suffix*-3-5(*t*) in No. 5 outperformed No. 4 (with stemming) by 0.0221 in micro-average F-score (the difference was proved to be statistically significant to a very high degree for all measures). No. 5 yielded similar results as No. 3 (with part-of-speech taggering/lemmatization): despite the increment in No. 5 was by 0.0102 in micro-average F-score, but the difference was proved to be statistically significant in terms of micro-precision and macro-recall only and not statistically significant for all the other measures.

Experiments with gazetteer lookup + affixes as it is in No. 8 yielded even better results compared with the results obtained separately (as in No. 2, No. 5) (all differences were proved to be statistically significant to a very high degree), besides No. 8 outperformed gazetteer lookup + stemming in No. 7 (the differences were proved to be statistically significant) and No. 8 yielded the same micro-average F-score as with gazetteer lookup + part-of-speech tagging/lemmatization in No. 6 (the differences were proved to be statistically significant for all measures, except for micro-recall). The best results were obtained using all available features in No. 9. Despite that the difference between sets of features No. 8 and No. 9 was proved to be statistically significant, the increment is only by 0.0021 in micro-averaged F-score.

The observations obtained from the experiments allow us to conclude that simple automatic prefix and suffix information capture enough of the Lithuanian language inflection information; therefore it can replace features generated by part-of-speech taggers, lemmatizers, and stemmers. It is indeed very good news, because the Lithuanian part-of-speech tagger/lemmatizer has limited availability, is not very convenient and slow to use; the Lithuanian stemmer is low in accuracy.

If we take a close look at the confusion matrix (see TABLE 4) obtained with feature set No. 9 and window [t_{x-2}; t_{x+1}], we would notice the main errors made by named entity classifier. Below we discuss main named entity classification error types (summarized by Nadeu & Sekine, 2007).

		Predicted class						
		B-PER	***I-PER***	***B-LOC***	***I-LOC***	***B-ORG***	***I-ORG***	***O***
Actual class	***B-PER***	***8,719***	155	789	4	25	7	1,203
	I-PER	63	***13,354***	14	15	17	25	373
	B-LOC	263	169	***10,192***	23	234	36	1,235
	I-LOC	8	138	24	***939***	17	97	99
	B-ORG	82	31	557	6	***3,762***	157	1,342
	I-ORG	29	171	33	48	54	***6,674***	3,473
	O	449	709	547	47	475	1,551	***57,3993***

TABLE 4 – Confusion matrix obtained with the feature set No. 9 and window [t_{x-2}; t_{x+1}].

The amount of correct predictions (see TABLE 4) is presented in diagonal where the actual class label is equal to predicted, thus from 632,427 instances, only 14,794 were predicted incorrectly: 3,788 times the classifier hypothesized as named entities where there was none; 7,725 times a named entity was completely missed by the classifier; 2,444 times classifier correctly recognized

the boundary of named entity, but got the wrong type; 476 times the classifier recognized the type of named entity, but got its boundary wrong; 371 times the classifier recognized that it is a named entity, but got wrong type and boundary.

The major problem is the 25.54% of all errors where the classifier determined named entity where there was none and 52.22% of all errors where named entity was completely missed. These problems arose because of different ambiguities in Lithuanian texts between proper and common words: i.e. a lot of person and location names are ambiguous with common words (especially at the beginning of the sentence), but the major problem is ambiguity between organization names and common words (typically only the first word in an organization name is capitalized and all other words are written in lower-case), because in our experiments we could not use a gazetteer for organization names, because such gazetteer is not yet available.

The experiments with different windows revealed, that the best results for Lithuanian were obtained using a rather small window $[t_{x-2}; t_{x+1}]$ (compared with the results obtained e.g. for Polish $[t_{x-2}, t_{x+2}]$ (Marcinczuk & Janicki, 2012) or Turkish – $[t_{x-3}, t_{x+3}]$ (Gokhan & Gulsen, 2012)): i.e. two instances before and one after the current. Thus, we can conclude, that even if the word order in the sentence in Lithuanian is free, the nearest information is more useful in NER process.

7 Conclusion and Outlook

In this paper we were solving NER task for Lithuanian language using supervised machine learning approach: i.e. we explored different sets of features in terms of orthographic and grammatical information, different windows, etc.

We have formulated and experimentally confirmed two hypotheses:

- The language dependent features (based on dictionaries, part-of-speech tagger, lemmatizer, and stemmer) increase the NER results, especially due to the importance of inflection information in Lithuanian language.
- The features based on external grammatical tools (part-of-speech tagger, lemmatizer, and stemmer) can be replaced with affixes that can capture relevant patterns intrinsically.

The best results were obtained using all available features and the window $[t_{x-2}; t_{x+1}]$. The decrease of micro-average F-score was only 0.0021 when features generated by the grammatical tools were eliminated. This result should be promising for other resource-scarce languages with similar properties as Lithuanian.

In future research, it would be interesting to experiment with different named entity types and classification methods.

Acknowledgments

This research was funded by European Union Structural Funds project “Postdoctoral Fellowship Implementation in Lithuania” (No. VP1-3.1-ŠMM-01) and initiated when the first author was visiting the Department of Linguistics and Nordic Studies at the University of Oslo, Norway.

References

Al-Rfou', R. and Skiena, S. (2012). SpeedRead: A Fast Named Entity Recognition Pipeline. In *Proceedings of the 24th International Conference on Computational Linguistics* (*COLING* 2012), pages 51–66.

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), pages 94–99.

Desmet, B. and Hoste, V. (2010). Dutch named entity recognition using ensemble classifiers. In *Computational Linguistics in the Netherlands 2010: selected papers from the twentieth CLIN meeting* (*CLIN* 2010), pages 29–41.

Elsebai, A., Meziane, F. and Belkredim, F. Z. (2009). A Rule Based Persons Names Arabic Extraction System. In *Proceedings of the 11th International Conference on Innovation and Business Management* (*IBIMA*), pages 53–59.

Georgiev, G., Nakov, P., Ganchev, K., Osenova, P. and Simov, K. (2009). Feature-Rich Named Entity Recognition for Bulgarian Using Conditional Random Fields. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (*RANLP*-2009), pages 113–117.

Gokhan, A. S. and Gulsen, E. (2012). Initial Explorations on using CRFs for Turkish Named Entity Recognition. In *Proceedings of the 24th International Conference on Computational Linguistics* (*COLING* 2012), pages 2459–2474.

Haaland, Å. (2008). *A Maximum Entropy Approach to Proper Name Classification for Norwegian.* PhD thesis, University of Oslo.

Hasan, K. S., Rahman, A., and Ng, V. (2009). Learning-based named entity recognition for morphologically-rich, resource-scarce languages. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 354–262.

Johannessen, J. B., Hagen, K., Haaland, Å., Nøklestad, A., Jónsdottir, A. B., Kokkinakis, D., Meurer, P., Bick, E. and Haltrup, D. (2005). Named Entity Recognition for the Mainland Scandinavian Languages. *Literary & Linguistic Computing*, 20(1): 91–102.

Kapočiūtė, J. and Raškinis, G. (2005). Rule-based annotation of Lithuanian text corpora. *Information technology and control*, Kaunas, Technologija, 34 (3): 290–296.

Kitoogo F. E., Baryamureeba, V, and De Pauw, G. (2008). Towards Domain Independent Named Entity Recognition. *International Journal of Computing and ICT Research*, 2 (2): 84–95.

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

Lafferty, J. D., McCallum, A. and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (*ICML*'01), pages 282–289.

Mai, M. O. and Khaled, S. (2012). A Pipeline Arabic Named Entity Recognition Using a Hybrid

Approach. In *Proceedings of the 24th International Conference on Computational Linguistics* (*COLING* 2012), pages 2159–2176.

Marcinczuk, M. and Janicki, M. (2012). Optimizing CRF-Based Model for Proper Name Recognition in Polish Texts. In *Proceedings of the 13th international conference on Computational Linguistics and Intelligent Text Processing* (*CICLing*'12), (1): 258–269.

Marcinczuk, M., Stanek, M., Piasecki, M. and Musial, A. (2011). Rich Set of Features for Proper Name Recognition in Polish Texts. *SIIS*, Lecture Notes in Computer Science, 7053: 332–344.

Marcinkevičienė, R. (2000). Tekstynų lingvistika (teorija ir paktika) [Corpus linguistics (theory and practice)]. *Darbai ir dienos*, 24: 7–63. (in Lithuanian).

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30 (1): 3–26.

Nøklestad A. (2009). *A Machine Learning Approach to Anaphora Resolution Including Named Entity Recognition, PP Attachment Disambiguation, and Animacy Detection*. PhD Thesis, University of Oslo.

Pinnis, M. (2012). Latvian and Lithuanian Named Entity Recognition with TildeNER. In *Proceedings of the Eight International Conference on Language Resources and Evaluation* (*LREC*'12), pages 1258–1265.

Popov, B., Kirilov, A., Maynard, D. and Manov, D. (2004). Creation of Reusable Components and Language Resources for Named Entity Recognition in Russian. In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (*LREC* 2004), pages 309–312.

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.

Singh, U., Goyal, V. and Lehal, G. S. (2012). Named Entity Recognition System for Urdu. In *Proceedings of the 24th International Conference on Computational Linguistics* (*COLING* 2012), pages 2507–2518.

Sundheim, B. (1995). Overview of results of the muc-6 evaluation. In *Proceedings of the 6th Conference on Message Understanding* (*MUC*-6), pages 13–31.

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* (*COLING*'00), 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).

Zinkevičius, V., Daudaravičius, V. and Rimkutė, E. (2005). The Morphologically annotated Lithuanian Corpus. In *Proceedings of the Second Baltic Conference on Human Language Technologies*, pages 365–370.