Named Entity Recognition for Ukrainian: A Resource-Light Approach

Sophia Katrenko
HCSL, University of Amsterdam,
Kruislaan 419, 1098VA Amsterdam,
the Netherlands
katrenko@science.uva.nl

Pieter Adriaans
HCSL, University of Amsterdam,
Kruislaan 419, 1098VA Amsterdam,
the Netherlands
pitera@science.uva.nl

Abstract

Named entity recognition (NER) is a subtask of information extraction (IE) which can be used further on for different purposes. In this paper, we discuss named entity recognition for Ukrainian language, which is a Slavonic language with a rich morphology. The approach we follow uses a restricted number of features. We show that it is feasible to boost performance by considering several heuristics and patterns acquired from the Web data.

1 Introduction

The information extraction task has proved to be difficult for a variety of domains (Riloff, 1995). The extracted information can further be used for question answering, information retrieval and other applications. Depending on the final purpose, the extracted information can be of different type, e.g., temporal events, locations, etc. The information corresponding to locations and names, is referred to as the information about named entities. Hence, named entity recognition constitutes a subtask of the information extraction in general.

It is especially challenging to extract the named entities from the text sources written in languages other than English which, in practice, is supported by the results of the shared tasks on the named entity recognition (Tjong Kim Sang, 2002).

Named entity recognition for the languages with a rich morphology and a free word order is difficult because of several reasons. The entropy of texts in such languages is usually higher than the entropy of English texts. It is either needed to use such resources as morphological analyzers to reduce the data sparseness or to annotate a large amount of data in order to obtain a good performance. Luckily, the free word order is not crucial for the named entity recognition task as the local context of a named entity should be sufficient for its detection. Besides, a free word order usually implies a free order of constituents (such as noun phrases or verb phrases) rather than words as such. For instance, although (1)[1] is grammatically correct and can occur in the data, it would be less frequent than (2).

(1) червону вона тримає квітку
red she holds flower
she holds a red flower

(2) вона тримає червону квітку
she holds red flower
she holds a red flower

The first phrase exemplifies that an adjective 'червону' is in a focus, whereas the second reflects the word order which is more likely to occur. In terms of named entities, an entity consisting of several words is also less likely to be split (consider, e.g., *National saw she Bank* where 'National Bank' represents one named entity of type organization). In the newspaper corpus we annotated, we have observed no examples of split named entities.

In this paper, we study different data representation and machine learning methods to extract the named entities from text. Our goal is two-fold. First,

[1] all examples in the paper are in Ukrainian, for convenience translated and sometimes transliterated

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 88–93,
Prague, June 2007.

we explore the possibility of using patterns induced from the data gathered on the Web. We also consider Levenshtein distance to find the most similar instances in the test data given a training set. Besides, we study the impact of different feature sets on the resulting classification performance.

We start with the short overview of the methods for NER proposed by the IE community. Afterwards, the experiments are described. We conclude with the outlook for the future work.

2 Related work

The existing NER systems use many sources in order to be able to extract NEs from the text data. Some of them rely on hand-written rules and precompiled lists of city names, person names and other NEs in a given language, while others explore methods to automatically extract NEs without prior knowledge. In the first case, the gazetteers will in most cases improve NER results (Carreras et al., 2002) but, unfortunately, they may not exist for a language one is working on. Hand-written rules can also cover more NEs but building such patterns will be a very time-consuming process.

There have been many methods applied to NER, varying from the statistical to the memory-based approaches. Most work on NER has been focused on English but there are also approaches to other languages such as Spanish (Kozareva et al., 2005), German, or Dutch. In addition, several competitions have been organized, with a focus on multilingual NER (Tjong Kim Sang, 2002). While analyzing the results of these shared tasks, it can be concluded that the selected features are of a great importance. In our view, they can be categorized in two types, i.e. *contextual* and *orthographic* [2]. The first type includes words surrounding a given word while the other contains such features as capitalized letters, digits contained within the word, etc. Both types of features contribute to the information extraction task. Nevertheless, orthographic features can already be language-specific. For instance, capitalization is certainly very important for such languages as English or Dutch but it might be less useful for German.

[2]Sometimes, these types of features are referred to as *word-external* and *word-internal* (Klein et al., 2003)

The feature set of some NER methods (Wu, 2002) also includes part-of-speech information and/or word prefixes and suffixes. Although this information (and especially lemmas) is very useful for the languages with rich morphology, it presupposes the existence of POS taggers for a given language.

Another conclusion which can be drawn relates to the machine learning approaches. The best results have been received by applying ensemble methods (Wu, 2002; Florian, 2002; Carreras et al., 2002).

A very interesting work on named entity recognition task was reported by Collins et al. (1999) who used only few named entities to bootstrap more. The other approach proposed recently makes use of the data extracted from the Web (Talukdar et al., 2006). By restricting themselves to the fixed context of the extracted named entities and by employing grammar inference techniques, the authors filter out the most useful patterns. As they show, by applying such approach precision can already be largely boosted.

Pastra et al. (2002) focused on the applicability of already existing resources in one language to another. Their case study was based on English and Romanian, where a system, originally developed for NER in English was adapted to Romanian. Their results suggest that such adaptation is easier than developing a named entity recognition system for Romanian from scratch. However, the authors also mention that not all phenomena in Romanian have been taken into account which resulted in low recall.

3 Methodology

Ukrainian belongs to the languages where the named entities are usually capitalized, which makes their detection relatively easy. In this paper we focus on using minimal information about the language in combination with the patterns learnt from the Web data, features extracted from the corpus and Levenshtein similarity measure.

Our hypothesis behind all three components is the following. We expect orthographic features be useful for a named entity detection but not sufficient for its classification. Contextual information may already help but as we do not intend to use lemmas but words instead, it will likely not boost recall of the named entity recongnition. To be able to detect more named enities in the text, we propose to use pat-

terns collected from the Web and Levenshtein similarity measure. Patterns from the Web should provide more contextual information than can be found in a corpus. In addition, a similarity measure gives us an opportunity to detect the named entities which have the same stem. The latter is especially useful when the same entity was mentioned in the training set as well as in the test data but its flections differ.

The intention of our study is, therefore, to start with a standard set of features (contextual and orthographic) as used for the many languages in the past and to add some means which would account for the fact that Ukrainian is a highly-inflected language.

3.1 Classification

First, we consider the features which can be easily extracted given the data, such as contextual and orthographic ones as described below in Table 1. For each word in a corpus its context (2 tokens to left and to the right) and its orthographic features are extracted. Orthographic features are binary features which, for instance, indicate whether a word is capitalized (1 or 0), etc. We have selected the following machine learning methods: k-nearest neighbor (knn) and voting and stacking as the ensemble methods which have been successfully applied to the named entity recognition task in the past.

contextual	-2/+2 words
	orthographic
CAP	capitalized
ALLCAP	all elements of a token capitalized
BSENT	first token in a sentence
NUM	contains digits
QUOTE	contains quotes

Table 1: Features

To overcome data sparseness and to increase recall, we make use of two techniques. First, we apply the patterns extracted from the Web.

3.2 Patterns

If we wish to collect patterns for a certain category C of the named entities (e.g.), we first collect all named entities that fall into it. Then, for each $X \in C$, we use X as a query term for Google (for this purpose we used the Google API). The queries we constructed were mainly based on the locations, such as 'Київ', 'Львів', 'Харків', 'Чернівці' etc. For each of these words we created queries by declining them (as there are 7 cases in Ukrainian language which causes the high variability). Consequently, we get many snippets where X occurs. To extract patterns from snippets, we fix a context and use 2 words to the left and to the right of X as in the classification approach above. The patterns which only consist of a named entity, closed-class words (e.g., prepositions, conjunctions, etc.) and punctuation are removed as such that do not provide enough evidence to classify an instance.

Intuitively, if there are many patterns acquired from the large collection of data on the Web, they must be sufficient (in some sense even redundant) to recognize named entities in a text. For instance, such pattern as *was located in* X in English can correspond to three patterns in Ukrainian *was located (fem., sing.) in* X, *was located (masc., sing.) in* X, *was located (neut., sing.) in* X. Even though these patterns could be embraced in one, we are rather interested in collecting all possible patterns avoiding this way stemming and morphological analysis.

As in Talukdar's approach (Talukdar et al., 2006), we expect patterns to provide high precision. We are, however, concerned about the size of Ukrainian Web which is much smaller than English part of the Web. As a consequence, it is not clear whether recall can be improved much by using the Web data.

3.3 Levenshtein distance

Yet another approach to address rich morphology of Ukrainian, is to carry out a matching of probable named entities in a test set against a list of named entities in a training set. It can be done by using string edit distances, such as Levenshtein.

Levenshtein (or edit) distance of two strings, x and y is measured as the minimal number of insertions, deletions, or substitutions to transform one string into the other. Levenshtein distance has become popular in the natural language processing field and was used for the variety of tasks (e.g., semantic role labeling).

Definition 1 (Levenshtein distance) *Given two sequences* $x = x_1x_2 \dots x_n$ *and* $y = y_1y_2 \dots y_m$ *of a length* n *and* m *respectively, Levenshtein distance is defined as follows*

$$lev(i,j) = min \begin{cases} lev(i-1, j-1) + d(x_i, y_j) \\ lev(i-1, j) + 1 \\ lev(i, j-1) + 1 \end{cases}$$

In the definition above, $d(x_i, y_j)$ is a cost of substituting one symbol in x by a symbol from y. The insertion and deletion costs are equal to 1.

Let $\mathcal{A}$ be a candidate named entity and $\mathcal{L}$ a list of all named entities found in the training set. By computing the Levenshtein distance between $\mathcal{A}$ and each element from $\mathcal{L}$, the nearest neighbor to $\mathcal{A}$ will be a NE with the lowest Levenshtein score. It might, however, happen that there are no named entities in a training set that correspond to the candidate in a test set. Consider, for instance the Levenshtein distance of two words 'Юрій' (George) and 'Крім' (besides) which is equal to 2. Even though the distance is low, we do not wish to classify 'Крім' as a named entity whose type is PERSON because it is simply a preposition. The problem we described can be solved in several ways. On the one hand, it is possible to use a list of stop words with most frequent prepositions, conjunctions and pronouns listed. On the other hand, we can also set a threshold for the Levenshtein distance. In the experiments we present below, we avoid setting threshold by using a simple heuristics. We align the first letters of $\mathcal{A}$ with its nearest neighbor. If they do not match (as in example above), we conclude that no variants of $\mathcal{A}$ belong to the training set.

4 Experiments and Evaluation

We have conducted three types of experiments using different feature sets, patterns extracted from the Web and Levenshtein distance. We expect that both types of experiments can shed a light on usefulness of the features that we defined for NER on Ukrainian data.

4.1 Data

Initially, several articles of the newspaper Mirror Weekly (year 2005)[3] were annotated. During the annotating process we considered the following named instances: PERSON (person names), LOC (location), ORG (organization).In total, there were 10,000 tokens annotated, 514 of which are named entities. All named entities have been annotated according to the IOB annotation scheme (Ramshaw and Marcus, 1995). The annotated corpus can be downloaded from `http://www.science.uva.nl/~katrenko/Corpus`

The corpus was split into training and test sets of 6,606 and 3,397 tokens, respectively. The corpus is relatively small but we hope to study whether such features as orthographic are sufficient for the NER task alone or it is needed to add more sources to approach this task.

4.2 Classification

The results of our experiments on classification of named entities are provided in Table 2. Baseline B_1 was defined by the most frequent tag in the data (ORG). Similarly to Conll shared task (Tjong Kim Sang, 2002), we also calculated a baseline by tagging all named entities which occurred in the training set (B_2). Although there are many names of organizations detected, there are only 1,92% of person names recognized.

	precision	recall	F-score
B_1	0.32	0.32	0.32
B_2	0.29	0.18	0.22
M^{2-knn}_{ortho}	0.31	0.44	0.36
$M^{2-knn}_{ortho+cont}$	0.38	0.46	0.42
$M^{Voting}_{ortho+cont}$	0.47	0.38	0.42
$M^{Stacking}_{ortho+cont}$	0.40	0.43	0.41
$M^{Voting}_{ortho+cont+pat}$	0.46	0.39	0.42
$M^{Voting}_{ortho+cont+pat+lev}$	**0.50**	**0.46**	**0.48**

Table 2: Experiments: precision and recall

Since we are interested in how much each type of the feature sets contributes to the classification accuracy, we have conducted experiments on contextual features only, on orthographic features only (model M^{2-knn}_{ortho} in Table 2) and on the combinations of both (model $M^{2-knn}_{ortho+cont}$ in Table 2). When used alone, contextual features do not provide a high performance. However, their combination with the orthographic features already results in a higher precision (at expense of recall) and in a higher F-score. It is worth noting that all results given in Table 2 were obtained either by using memory-based learning (in particular, k-nearest neighbor as in M^{2-knn}_{ortho} and in $M^{2-knn}_{ortho+cont}$) or by ensemble methods (as in $M^{Voting}_{ortho+cont}$ and $M^{Stacking}_{ortho+cont}$). The latter option was particularly interesting to explore because it proved to provide accurate results for the

[3] can be found at `http://www.zn.kiev.ua`

named entity recognition task in the past. The results in Table 2 also seem to support a claim that the ensemble methods perform better. It can be seen when comparing $M^{2-knn}_{ortho+cont}$, $M^{Voting}_{ortho+cont}$ and $M^{Stacking}_{ortho+cont}$. Despite of using the same feature sets, Voting (based on Naive Bayes, decision trees and 2-knn) and Stacking (2-knn as a meta-learner applied to Naive Bayes and decision tree learner) both provide higher precision but lower recall.

By using χ^2 test on the training set, we determined which attributes are the most informative for the classification task. The most informative turned out to be a word itself followed by the surrounding context (one token to the right and to the left). The least informative feature is NUM, apparently because there have been not many named entities containing digits.

4.3 Patterns

As a next step, we employed the patterns extracted from the Web data. Some of the patterns accompanied with the translation and information on case are given in Table 3. It can be noticed that not all of the patterns are accurate. For instance, a pattern *together with a city mayor LOC* can also be used to extract a name of a mayor (hence, PERSON) and not a location (LOC). Patterns containing prepositions (so, mostly patterns containing a named entity in locative case) 'in', 'with', 'nearby' are usually more accurate but they still require additional context (as a word 'town' in *in a little town LOC*).

The results we obtained by employing such patterns did not significantly change the overall performance (Table 2, model $M^{Voting}_{ortho+cont+pat}$). However, the performance on some categories such as ORG or LOC (Table 5 and Table 6, model *ALL+P*) was positively affected in terms of F-score.

4.4 Levenshtein distance

Finally, we compare all capitalized words in a test set against the named entities found in the training data. The first 6 examples in Table 4 show the same nouns but in different cases. The distance in each case is equal 1. Since we did not carry out the morphological analysis of the corpus, many such occurrences of the named entities in the test data were found given the information from the training set using orthographic and contextual features only (as they do not match exactly). However, Levenshtein distance helps to identify the variants of the same named entity. The results of applying Levenshtein distance (together with the patterns and Voting model on all features) for each category are given in Table 5 and Table 6 (model *ALL+P+L*). LOC and ORG are two categories whose performance is greatly improved by using Levenshtein distance. In case of PERSON category, recall gets slightly higher, whereas precision does not change much.

PATTERN	case
у містечку LOC in a little town LOC	locative
з містом LOC with a city LOC	instrumental
карта LOC a map of LOC	genitive
спільно з мером LOC together with a city mayor LOC	instrumental
мій рідний LOC my dear/native LOC	vocative
У LOC виявлено in LOC was found	locative
мандруючи LOC travelling in LOC	instrumental
живе десь під LOC lives somewhere nearby LOC	instrumental

Table 3: Patterns for LOC category

The last three examples in Table 4 are very interesting. They show that sometimes the nearest neighbor of the candidates for NEs in the test data is a named entity of the same category but it cannot be found by aligning. Having noticed this, we decided to exclude aligning step and to consider a nearest neighbor of every capitalized token in the test set. Although we extracted few novel person names and locations, performance in terms of precision dropped significantly. The very last example in Table 4 demonstrates a case when applying Levenshtein measure fails. In this case 'БЮТ' is of type ORG (a political party) and 'БЮТівці' are people who belong to the party. Given the nearest neighbor and the successful alignment, it is predicted that 'БЮТівці' belongs to the category ORG but it is not true. In the other example involving the same entity 'БЮТ', 'БЮТу' is correctly classified as ORG (it is the same named entity as in the training data but in dative case).

It can be concluded that, in general, Levenshtein distance helps to identify many named entities which were either misclassified or not detected at all. However, it is sometimes unable to distinguish between the variant of the same named entity and a true negative. Additional constraints such as the upper threshold of the Levenshtein distance might solve this problem.

Category	Test set	Training set	L-score
PERSON	Юлію	Юлії	1
PERSON	Лисенком	Лисенко	1
ORG	БЮТу	БЮТ	1
LOC	Львові	Львова	1
LOC	Києва	Києві	1
PERSON	Віктором	Віктор	2
PERSON	Роман	Іван	3
PERSON	Домбровський	Гошовський	4
WRONG	БЮТівці	БЮТ	4

Table 4: The nearest neighbors

As can be seen from Table 2, the best overall performance is achieved by combining contextual and orthographic features together with the patterns extracted from the Web and entities classified by employing the Levenshtein distance.

Model	PERSON	LOC	ORG
ORTHO	0.25	0.34	**0.52**
ALL	0.47	0.37	0.49
ALL+P	0.48	0.31	0.47
ALL+P+L	**0.49**	**0.55**	0.51

Table 5: Performance on each category: precision

Model	PERSON	LOC	ORG
ORTHO	**0.49**	0.26	0.42
ALL	0.36	0.15	**0.51**
ALL+P	0.36	0.27	0.49
ALL+P+L	0.42	**0.49**	0.56

Table 6: Performance on each category: recall

5 Conclusions and Future work

In this paper, we focused on standard features used for the named entity recognition on the newswire data which have been used on many languages. To improve the results that we get by employing orthographic and contextual features, we add patterns extracted from the Web and use a similarity measure to find the named entities similar to the NEs in the training set. The results we received are, in general, lower than the performance of NER systems in other languages but higher than both baselines. The former might be explained by the size of the corpus we use and by the characteristics of the language. As Ukrainian language is a language with a rich morphology, there are several directions we would like to explore in the future.

From the language-oriented perspective, it would be useful to determine to which extent stemming and morphological analysis would boost performance. The other problem which we have not considered up to now is the ambiguity of some named entities. For example, a word 'Ukraine' can belong to the category LOC as well as to the category ORG (as it is a part of a complex named entity).

In addition, we would also like to explore the semi-supervised techniques such as co-training and self-training (Collins and Singer, 1999).

References

Carreras et al. 2002. Named Entity Extraction using AdaBoost. *In the Proceedings of CoNLL-2002, Taipei, Taiwan*.

Michael Collins and Yoram Singer 1999. Unsupervised Models for Named Entity Classification. *In Proccedings of EMNLP/VLC-99*.

Radu Florian. Named Entity Recognition as a House of Cards: Classifier Stacking. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 2002*.

Dan Klein et al. Named Entity Recognition with Character-Level Models. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 2003*.

Zornitsa Kozareva, Boyan Bonev, and Andres Montoyo. 2005. Self-training and Co-training Applied to Spanish Named Entity Recognition. *In MICAI 2005: 770-779*.

Katerina Pastra, Diana Maynard, Oana Hamza, Hamish Cunningham and Yorick Wilks. 2002. How feasible is the reuse of grammars for Named Entity Recognition? *In LREC'02.*

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking Using Transformation-Based Learning *In ACL'95.*

Ellen Riloff. 1995. Information Extraction as a Basis for Portable Text Classification Systems. *PhD Thesis. Dept. of Computer Science Technical Report, University of Massachusetts Amherst.*

P. P. Talukdar, T. Brants, M. Liberman and F. Pereira. 2006. A Context Pattern Induction Method for Named Entity Extraction. *In the Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-2006)*.

Erik Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 155–158.*

Dekai Wu et al. 2002. Boosting for Named Entity Recognition. *In the Proceedings of CoNLL-2002, Taipei, Taiwan*.