

Assigning Inflectional Paradigms to Named Entities by Linear Successive Abstraction

Nikola Ljubešić, Nikola Bakarić, Tomislava Lauc
Department of Information Sciences
Faculty of Humanities and Social Sciences
Ivana Lučića 3, Zagreb, Croatia
Email: {nljubesi,nbakaric,tlauc}@ffzg.hr

Abstract—This paper describes how a supervised learning method is used for assigning inflectional paradigms to organizational named entities as the main prerequisite for generating a morphological lexicon of these entities. An inflectional paradigm consists of a set of rules for generating all forms of a lexicon entry. A morphological lexicon consists of lexicon entries and their corresponding forms. This type of language resource is crucial in tasks such as natural language generation (generating natural language business news from database data and news templates) and named entity identification (necessary step in data mining and business intelligence). The basic resource used in this research is a list of 106.530 named entities of organizations given in basic form (nominative) and ranked by relevance. On the first 5.000 manually tagged named entities 59 inflectional paradigm classes are defined. Using linear successive abstraction, a suffix model is trained, validated and tested on this tagged dataset. Morphological lexica of general language, personal names and settlements are used as additional resources in the decision process. The achieved accuracy on the test set is 98,70%.

I. INTRODUCTION

Information extraction (IE) is the problem of deriving structured factual information from text. One of information extraction tasks deals with finding names of entities in unstructured or partially structured text and determining the relationships between these entities. This task is called named entity recognition (NER) and is known as a subtask of information extraction. It tries to locate and classify atomic elements in text into predefined categories such as names of persons, organizations, locations, etc. [1]

There are two main research approaches in the field of entity extraction: the approach based on stochastic methods and deterministic approach. In stochastic approaches the name entity models are trained on a large amount of manually annotated data. The disadvantage of this approach is the acquisition bottleneck, i.e. the need for large amounts of manually annotated data. Deterministic methods consist of manually crafted rules mainly written in form of regular expressions, i.e. finite-state automata and transducers. The disadvantage of this approach is the complexity of producing hand-crafted rules that require a full understanding of the problem for a given language. There is also a middle way between these two approaches where stochastic methods are used to generate human-readable rules that can be corrected and expanded by hand. [2]

It should be stressed that the task of entity extraction gets more difficult when dealing with highly inflected languages such as Croatian. That is because there are many word forms in which one named entity can occur. For example, the rich nominal inflection in Croatian includes seven cases, which means 14 suffixes concerning singular and plural. It is different for poorly inflected languages such as English where the entity extraction task is usually backed up by simple morphological normalizers in order to generate lexicons of named entities that are necessary for entity identification. Therefore, in cases of Slavic languages like Croatian there are often no freely available resources that could provide a morphological treatment of named entities which are a necessary prerequisite for entity extraction task. [3]

This task deals with problem of grouping different forms of one named entity. In order to group different forms of one named entity it is possible to create lexicon of named entities that will generate all possible word forms for a given named entity normal form, i.e. for a given lemma. In that case, the process of building a lexicon is not corpus-based, but rule-based where rules describe possible inflectional paradigms for named entities. In general, this approach usually deals with creating a morphological lexicon database that associates an inflected word to a set of lemmas and a set of features and vice versa for generation. However, the problem is how to generate all possible word forms for a given lemma if there is not enough prescriptive linguistic work done and therefore there are many ways in which names are inflected in practice. It is the problem concerning organization name inflection for the Croatian language. [4] In order to solve the problem we describe generating the morphological lexicon of organization entity names for purpose of organization name identification and information extraction for business intelligence.

II. ANALYZING THE PROBLEM

The aim of this research is to generate a language resource - morphological lexicon of organizational entity names. The primary resource is a list of entities which consists of 106.530 organization names given in basic form (nominative). These entities are ranked by relevance that is calculated through criteria like number of employees, frequency of occurrence in corpora, business performance etc.

The use of this resource is twofold:

- 1) language generation, i.e. business news generation from patterns and numerical database information
- 2) named entity identification for information extraction i.e. business intelligence.

One of the main problems for these tasks is the fact that there is more than one way to inflect an organizational named entity. For language generation the most preferred inflectional paradigm has to be used. Since there are more inflectional paradigms used for one named entity in everyday use, entity identification task requires all possible paradigms of a named entity to be used.

For these reasons a distinction has to be made between paradigms for language generation and paradigms used only for entity identification. The task of finding the most preferred inflectional paradigm is much harder than finding any inflectional paradigm. On the other hand, it is to be expected that only most prominent named entities should appear in automatically generated news. Since the lexicon generation task is achieved by a supervised method (which assumes a tagged dataset), the decision is to manually annotate 5.000 most relevant named entities and thereby make a distinction between the preferred paradigm and all other paradigms. Therefore, the problem of preferable paradigm is not present when training the model.

As already stated, most entities can be inflected with more than one paradigm. A decision which paradigm is preferable is made during manual annotation. It depends on grammatical rules and use in everyday language. Therefore certain categories consist of more than one possible paradigm ("INA-e", "Ine").

Several problems were encountered when assigning paradigms to entities. Grammar rules are often disregarded in everyday use, especially when it comes to entities which contain non-Croatian words ("Atlantic trade") as well as entities which are orthographically or phonetically similar or equal to Croatian general language lexemes ("Konzum", "Zvijezda"). The grammar rules are often vague or do not exist when applied to cases stated above. Therefore more than one paradigm was included when annotating such entities because the task of named entity identification requires all possible forms. However, the task of language generation requires only one paradigm and therefore a decision has to be made which one is preferable. This decision was based on the comparison of frequencies of different forms for a specific named entity through Google.

During annotation 39 basic paradigms were defined and combined into 68 paradigm combinations. Since the order of paradigms is not relevant for building the model (the category "51" with the paradigm "5" as the preferred one and "1" as the non-preferred equals the category "15") the number of paradigm combinations is reduced to 59. In the next chapter the process of building the model for classifying named entities into these 59 categories is described.

III. BUILDING THE MODEL

Most models for classifying lexemes into morphological categories are supervised and are based on n-grams. Because of the sparse data problem in natural language processing there is a need for combining evidence from different size n-grams. Two basic techniques are commonly used: linear interpolation and smoothing by redistributing a part of the probability mass to unseen n-grams. [5] The method applied in this research uses linear interpolation. It was introduced in [6] and used in [7] and is called linear successive abstraction. The values calculated in the model are conditional probabilities of a specific tag t given the last m letters of an n letter word. The algorithm combines that conditional probability $P(t|l_{n-m+1}...l_n)$ with the conditional probabilities of more general contexts $P(t|l_{n-m+2}...l_n)$, $P(t|l_{n-m+3}...l_n)$... $P(t)$. The recursion formula is

$$P(t|l_{n-i+1}, \dots, l_n) = \frac{\hat{P}(t|l_{n-i+1}, \dots, l_n) + \Theta_i P(t|l_{n-i+2}, \dots, l_n)}{1 + \Theta_i} \quad (1)$$

for $i = m..0$ using the maximum likelihood estimates \hat{P} from frequencies in the training set, weights Θ_i and the initialization

$$P(t) = \hat{P}(t) \quad (2)$$

The maximum likelihood estimate for a suffix of length i is derived from the training set by

$$\hat{P}(t|l_{n-i+1}, \dots, l_n) = \frac{C(t, l_{n-i+1}, \dots, l_n)}{C(l_{n-i+1}, \dots, l_n)} \quad (3)$$

where $C()$ is the count function.

The weights proven to get best results are standard deviations of unconditioned maximum likelihood estimates of n-grams in the training set [6] by

$$\Theta_i = \sqrt{\frac{1}{N} \sum_{j=1}^N (\hat{P}(l_{n-i+1}, \dots, l_n) - \bar{P})^2} \quad (4)$$

$$\bar{P} = \frac{1}{N} \sum_{j=1}^N \hat{P}(l_{n-i+1}, \dots, l_n) \quad (5)$$

In the decision process besides the model morphological lexica of general language, personal names and settlements are used. For 33,20% of all named entities longer than three characters ("Zvijezda") and 51,19% of latter parts of named entities not ending on "e", "i" or "u" ("Tehnocentar", "centar"), entries are found and inflected by lexica. The above decisions are based on manual observation of possible results of the method.

At this point named entities found in lexica are removed from the dataset which shrinks down to 2.423 data points.

The remaining dataset is divided into a 9/10 training and validation set and a 1/10 test set (2.180, ie. 243 data points).

TABLE I
ACCURACY REGARDING THE VALUE OF m

m	accuracy
1	.9125
2	.9469
3	.9458
4	.9465
5	.9448
6	.9436
7	.9429
8	.9447
9	.9438
10	.9434

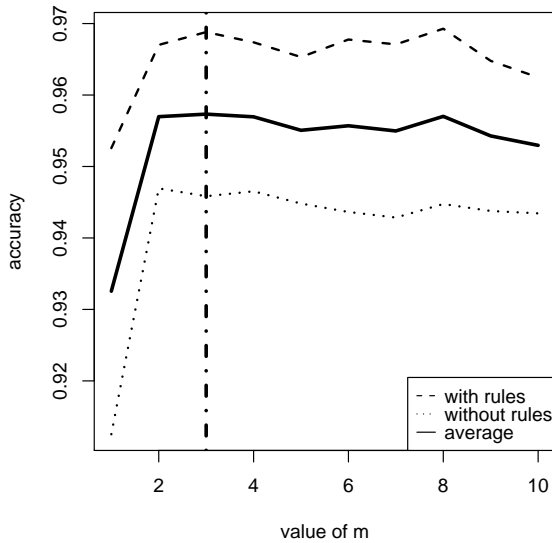


Fig. 1. Accuracy regarding the the value of m

Since the parameter t has to be empirically tuned, holdout validation with 100 iterations is used to estimate the parameter value more accurately.

The loss function used is

$$L(P_a, P_m) = 1 - \frac{C(P_a \cap P_m)}{C(P_m)} \quad (6)$$

where P_a are the paradigms assigned by the model, P_m the manually assigned paradigms and $C()$ the count function. Two examples of the loss function would be $L('15', '159') = 0.33$ and $L('15', '1') = 0.0$. Overgeneration is not penalized since this data will be used for named entity identification only and all the 59 categories consist of paradigms that correspond in the category of number and gender (1, 5 and 9 are all masculinum, singular) which makes a homonymy clash with another lexeme very improbable. It should be stressed that in this research accuracy equals recall while precision is neglected because the overgeneration problem is out of interest.

TABLE II
ACCURACY REGARDING THE VALUE OF m WITH RULES APPLIED

m	accuracy
1	.9526
2	.9670
3	.9688
4	.9674
5	.9653
6	.9678
7	.9671
8	.9692
9	.9647
10	.9624

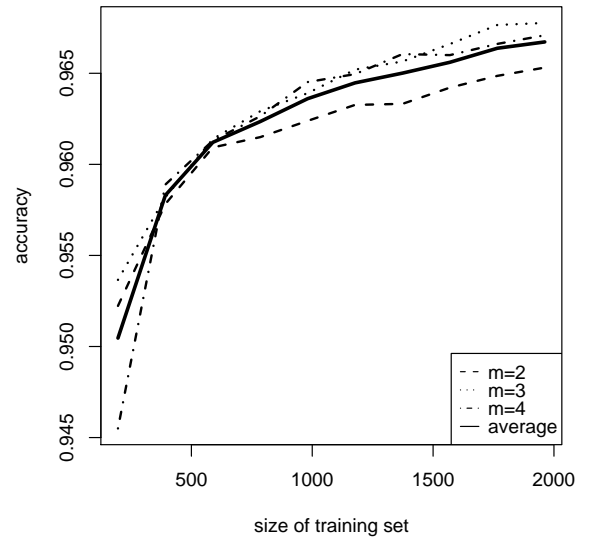


Fig. 2. Accuracy regarding the size of the training set

At this point most frequent errors are manually checked and, since overgeneration is out of interest, two general expansion rules are introduced:

- 1) mutually expand paradigms 1('TVIN', 'TVIN-a') and 5('Tvin', 'Tvina')
- 2) mutually expand paradigms 2('INA', 'INA-e') and 3('Ina', 'Ine')

For example, by these rules the class "5" is expanded to class "15" and class "2" to class "23".

IV. RESULTS

The basic task is to find the optimal value of the parameter m which determines the length of the longest n-gram observed. As mentioned before, in this research accuracy is identical to recall. In Table 1 the results of the holdout validation process regarding the value of m are shown without using the lexicon or applying the expansion rules. The data shows that a lot of information is contained in just the last character of a word. When unigram and digram information is combined, there is

a low increase in accuracy (3,77%). In case of m greater than 2, no significant advance is obtained.

Table 2 contains the accuracy measures regarding the value of m with expansion rules applied. The increase in accuracy regarding $m = 1$ and $m = 2$ in this case is even smaller (1,51%). The maximum accuracy is obtained with $m = 3$. Therefore, the value of m is set to 3. A graphical representation of accuracy regarding the value m and the method used is given in figure 1.

Figure 2 depicts accuracy regarding the size of the training dataset. Accuracy increases rapidly up to the training set size of 700 points after which the increase starts to drop gradually. At the dataset size used in this research (1.962 for training during validation) the slope is still positive which indicates that a larger training set could provide even better results.

For $m = 3$ and with expansion rules applied, accuracy is 96,88%. In case of using morphological lexica, accuracy rises up to 98,49% since 2.321 of the data points in the training set are found in the lexicon $((2179 * 0,9688 + 2321)/4500)$. When the method is applied on the test set, an accuracy of 97,33% is achieved on entities not found in the lexicon. With 51,19% of entities found in lexica, the final accuracy of the method on the test set is 98,70%.

V. CONCLUSION

This paper presents a method for building a morphological lexicon with a supervised learning algorithm. The algorithm combines weighted evidence from different length endings using linear interpolation.

The results of the experiment show that a large amount of information is already encoded in the category probability and

the ending of length 1.

Inspecting errors in a confusion matrix is proven to be useful for hand-coding rules that can improve the performance of the method.

Morphological lexica can also be used efficiently in the decision process for inflecting entities which are partially or completely identical to lexicon entries.

With hand-coded rules applied and lexica used, information from last three characters yields best results. On the test set the method achieves accuracy of 98,70%.

It has to be stressed that assigning inflectional categories to organizational named entities is a hard task for humans in cases not covered by grammar rules. This fact makes an automated approach like the one presented in this paper a very appealing one.

REFERENCES

- [1] R. Gaizauskas and Y. Wilks, "Information extraction: Beyond document retrieval," *Journal of Documentation*, vol. 54, no. 1, pp. 70–105, 1998.
- [2] C. Cardie, "Empirical methods in information extraction," *AI Magazine*, vol. 18, no. 4, pp. 65–80, 1997. [Online]. Available: citeseer.ist.psu.edu/cardie97empirical.html
- [3] B. Bekavac and M. Tadić, "Implementation of croatian nerc system," in *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing 2007*, Prague, Czech Republic, 2007.
- [4] M. Kržak and D. Boras, "Lexical database of the croatian literary language," *Informatologia Yugoslavica*, vol. 17, no. 3-4, pp. 223–242, 1985.
- [5] I. Dagan, L. Lee, and F. C. N. Pereira, "Similarity-based models of word cooccurrence probabilities," *Machine Learning*, vol. 34, no. 1-3, pp. 43–69, 1999. [Online]. Available: citeseer.ist.psu.edu/article/dagan99similaritybased.html
- [6] C. Samuelsson, "Handling sparse data by successive abstraction," in *Proceedings of COLING-96*, Kopenhagen, Denmark, 1996.
- [7] T. Brants, "Tnt – a statistical part-of-speech tagger," 2000.