

# Separating Named Entities

Barbora Ulipová and Marek Grác

Computational Linguistics Centre  
Faculty of Arts  
Masaryk University  
Czech Republic  
b.ulipova@gmail.com, grac@mail.muni.cz

**Abstract.** In this paper, we analyze the situation of long sequences of mostly capitalized words which look like a named entity but in fact they consist of several named entities. An example of such phenomena is *hokejista (hockey player) New York Rangers Jaromír Jágr*. Without splitting the sequence correctly, we will wrongly assume that the whole capitalized sequence is a name of the hockey player. To find out how the sequence should be split into the correct named entities, we tested several methods. These methods are based on the frequencies of the words they consist of and their n-grams. The method DIFF-2 proposed in this article obtained much better results than MI-score or logDice.

**Keywords:** text corpus, mutual information, named entities

In the process of creating a new question answering system, we have encountered several problems. One of the problems that has to be solved is extracting the roles, titles or occupation of various people from a free text. This information together with (at least part of the) name of person often occur together and it should be possible to extract them. In this paper we will focus on a situation when a longer capitalized sequence of words is found and it is required to split the sequence into multiple parts. In the next section we will describe examples found during research in Czech corpora. The third section describes possible approaches based on frequencies and co-occurrences of words. The results are presented in the last section of this paper.

## 1 Named entities characterization

In this project, we focus on extracting two types of information. First one, identify person's features such as role, title or occupation of person. In that case we are looking for a list of words that can be prepared in advance. Such words are *actress (herečka)*, *model (modelka)*, *hockey player (hokejista)*, *minister (minister)*. These words are often mentioned in tabloids that are a great resource of such information. The second information type we are looking for is a relationship between two people where both of them are mentioned. Often, they are not referenced by full names but only by the first name or a nickname, e.g. *Brooklyn*,

son of *David Beckham*. This information is sometimes set, sometimes it can change in time (e.g. *wife*, *boss*). The Process of extracting valid information for given time is not a part of this paper.

In the Czech, another issue occurs regularly. In the corpus, we can find clusters of capitalized words where parts of such clusters belong to separate named entities. An example of such phrase is *hráč New York Rangers Jaromír Jágr* (*New York Rangers' player Jaromír Jágr*) or *hra Vladimíra Franze Válka s mloky* (*Vladimír Franz's play Válka s mloky*). These phrases have to be identified and separated into the relevant parts.

In this paper we focus on methods to split such sentences of mostly capitalized words into multiple parts. During our research we have not met many sequences that require to be split to more than two parts, so we will focus mainly on the ones which only need to be split into two parts in next sections. The other cases can be solved by running our methods multiple times.

## 2 Extraction of names

The first step in creating a system for extraction of information about people is finding the names of people. People's names are, obviously, capitalized in a text which we can use. However, there are other words which are capitalized as well – proper names such as names of institutions, nationalities, products and artistic works. Therefore, we need to find a way to distinguish between people and other capitalized words. We have a list of nouns that can represent a person [1] and also a list of words that describe a relationship between people. Our preliminary research shows that if we look for the relationship words in the vicinity of two or more capitalized words which are not at the beginning of a sentence, the majority of capitalized words found will contain names of persons.

## 3 Separating the phrases

To separate the phrases, we have decided to use an approach based on frequencies and co-occurrences of words. There are several ways how to express a co-occurrence between words, we have tried the MI-score [2] and the logDice [3]. In this section, we will present various methods together with examples on real sequences obtained from corpus *czTenTen12\_8* [4]. At first, we tried to write corpus queries in CQL [5] but we have found out that one complex query finished in minutes which makes it almost unusable in the real-world applications. We had to simplify the queries to work just with the n-grams which occur in a corpora that can be found in a different way based on the preprocessed data and now a query took less than 0.1 seconds on the same hardware.

### 3.1 Method based on mutual information between two words

The first, most naive approach is based on counting MI-score or logDice between bigrams in sequence. The sequence is a segment of text that should be divided into separate named entities. Let's have a *sequence* =  $(w_1, w_2, w_3, \dots, w_X)$  then we will count both logDice and MI-score for each pair  $(w_I, w_{I+1})$ . According to the theory of mutual information, it can be expected that lower values of mutual score provide a better identification of the borders as these words should be the words that relate to each other less than others. If the sequence itself or candidate n-gram divisions are not mentioned in corpus, we have to modify the equation a bit and replace the occurrences of zero by one. Our test suite<sup>1</sup> showed that using logDice results had the precision **29.5%** while MI score's precision was only **11.8%**. Examples of selected sequences are shown in table 1 where the text in bold represents the correct division and the numbers in bold represent the results according to used methods.

Table 1: Detailed results of using method based on MI between two words. Sequences shown in table are *PSP Miroslava Němcová* and *D. Cerekve Zdeněk Jirsa*

	MI-score logDice			MI-score logDice	
<b>PSP - Miroslava</b>	<b>8.77</b>	<b>-14.26</b>	D. - Cerekve	<b>-0.49</b>	-18.18
Miroslava - Němcová	10.21	-13.71	<b>Cerekve - Zdeněk</b>	1.08	-18.19
			Zdeněk - Jirsa	2.69	<b>-19.44</b>

### 3.2 Method based on mutual information between n-grams

In order to improve this method, we have decided to extend the above mentioned equations to work directly on n-grams. There will be only two n-grams for each division because we want to split the sequence to just two parts as was explained before. The first n-gram will start at the first word of the sequence and the second one will end with the last word. Such n-grams do not have to be a part of the corpora so we will reuse the modification proposed in the previous subsection and we will replace zeros with ones. Such modifications will also be a part of every other proposed method.

When we use this method, we can expect that higher values of mutual score will represent the correct division. This should happen because the mutual information between two incorrectly selected n-grams should be lower as they do not occur together as frequently as the correct ones. The precision results when logDice was used is **41.2%** and MI score's precision is **29.5%**.

<sup>1</sup> Test suite was manually created from 17 phrases that were selected according to estimation of frequency of different types in corpus

Table 2: Detailed results of using method based on MI between n-grams.

	MI-score logDice	
<b>PSP - Miroslava Němcová</b>	10.07	-11.23
PSP Miroslava - Němcová	<b>10.89</b>	<b>-9.86</b>
D. - Cerekve Zdeněk Jirsa	-1.49	-13.83
<b>D. Cerekve - Zdeněk Jirsa</b>	<b>10.64</b>	<b>-4.30</b>
D. Cerekve Zdeněk - Jirsa	4.44	-8.01

### 3.3 Method extended with negative n-grams

The previous method did not take into account that if a particular word always follows the n-gram, it should be a part of it, too. Our next method is based on very same idea. In order to count only the *clean* value we will subtract the number of longer n-grams from the frequency of the shorter one. Because we are working with the n-gram model, finding negative occurrences is not possible like it is in CQL itself, so we have simplified this process to testing the first adjacent word to our n-gram. For example when if we have a sequence *New York Rangers Jaromír Jágr* then we will count the modified frequency for n-gram *New York* as follows:  $freq(New York) - freq(New York Rangers)$ .

In the ideal case, the modified frequency will be zero which will in the case of this sequence happen only if there is not any other *Jaromír* in this hockey team as he will have different surname (last word of sequence). In such case the modified frequency will be the frequency of occurrences of the same sequence with just the last word replaced. The number subtracted from frequency provides additional information that can be used for evaluation too.

Table 3: Detailed results of using method based on MI between n-grams with modified frequencies

	MI-score logDice	
<b>PSP - Miroslava Němcová</b>	10.17	<b>-10.62</b>
PSP Miroslava - Němcová	<b>11.19</b>	N/A
D. - Cerekve Zdeněk Jirsa	-1.49	-13.62
<b>D. Cerekve - Zdeněk Jirsa</b>	<b>11.19</b>	<b>-3.26</b>
D. Cerekve Zdeněk - Jirsa	4.46	N/A

When modified frequencies will be used in the n-gram equations, the results show that there are too many results in intermediate steps that can not be counted

as the modified frequency drops to zero. As the results were heavily corrupted by this, we have decided not to count the mutual information of n-grams but to use only the value subtracted in equations. We have created two equations that make sense to us.

The first one, is based on the idea that when we sum modified frequencies (*DIFF-4* – use four elements in equation) of n-grams the lowest value will show the ideal borderline between them. The other one is just simplification of previous one where we will sum together just subtracted values (*DIFF-2* – use two elements in equation). Unlike the previous methods these numbers are not normalized at all but they work correctly on a selected sequence. In order to compare values across the sequences, some normalization will have to be done which will be a part of following research.

According to our test suite, the *DIFF-4* has a precision **35.2%** which ranks it above all usage of the MI-score which could not be expected before the experiment. The *DIFF-2* method has a precision **94.1%** with only one error on the test suite. This example is *D. Cerekve Zdeněk Jirsa* where borders were properly identified by n-gram methods based on both MI-score and logDice. When we expand *D.* to *Dolní* then division will be found correctly.

## 4 Results and Future Work

In this paper, we have presented several methods that should divide a sequence of words into two semantically correct parts. According to our results, using bi-gram model does not provide the best precision on the test suite. Extending equations for MI-score and logDice rapidly increases obtained precision. It was shown that in both of these cases using logDice resulted into improved precision against MI-score, so we can suggest to test it also for other applications.

The best method is *DIFF-2* with the precision **94.1%**. It is surprising that it does not take into account the frequency of the final n-grams. One of the reasons why they are not used yet is that the results are not normalized so they cannot be compared across various sequences like MI-score and logDice allows. Obtained results are not really representative yet, so we will prepare a larger dataset for Czech.

Although people tend to split such phrases quite easily, it is often a result of detecting patterns of either n-grams or word-class like first names. We did not use first names or any other source of information to be able to improve our methods in the real applications.

In the future, we would like to focus on a way to detect sequences that have to be divided. Also the idea of normalization of *DIFF-2* and *DIFF-4* is very interesting as then they can also be tested against MI-score and logDice in different applications and for different languages.

**Acknowledgements** This work has been partly supported by the Masaryk University within the project *Čeština v jednotě synchronie a diachronie – 2014* (MUNI/A/0792/2013).

## References

1. Grác, M.: Rapid Development of Language Resources. PhD thesis, Dissertation, Masaryk University in Brno (2013)
2. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. *Computational linguistics* **16**(1) (1990) 22–29
3. Rychlý, P.: A lexicographer-friendly association score. *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN* (2008) 6–9
4. Jakubíček, M., Kilgarrieff, A., Kovář, V., Rychlý, P., Suchomel, V., et al.: The tenten corpus family. In: *Proc. Int. Conf. on Corpus Linguistics*. (2013)
5. Arasu, A., Babu, S., Widom, J.: Cql: A language for continuous queries over streams and relations. In: *Database Programming Languages*, Springer (2004) 1–19