Separating Named Entities

Barbora Ulipová, 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 that looks like 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 proper split of this sequency, we will wrongly assume that whole capitalized sequence is a name of the hockey player. We have tested several methods that can propose split to correct named entities. These methods are based on frequencies of incorporated words and their n-grams. Method DIFF-2 proposed in this article obtained results much better than MI-score or logDice.
Keywords: text corpus, mutual information, named entities

In the process of creating a new question answering systems, we have encountered several problems. One of the problems that has to be solved is extracting roles, titles or occupation of various people from 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 situation when longer capitalized sequence of words is found and it is required to split sequence into multiple parts. In the next section we will describe examples found during research on Czech corpora, the third section describes possible approaches based. Results are presented in the last section of this paper.

1 Name 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 word 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 relationship between two people where both of them are mentioned. More often than not, they are not referenced by full names but only by first name or nickname, e.g. *Brooklyn, son of David Beckham*. Some of these information are set, the others

can change in time (e.g. *wife*, *boss*). 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 belongs to separate name 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 relevant parts.

In this paper we focus on methods to split such sentence of mostly capitalized words into multiple parts. During our research we have not met many sequence that requires split to more than two parts, so we will focus mainly on them 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 – the 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 represents a person [3] and also a list of words that describes 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 approach based on frequencies and co-occurrences of words. There are several ways how to express co-occurrence between words, we have tried MI-score [2] and logDice [5]. In this section, we will present various methods together with examples on a real sequences obtained from corpus czTenTen12_8 [4]. At first, we have tried to write corpus queries in CQL [1] but we have found out that one complex query finished in minutes what makes it almost unusable in the real-world applications. We had simplify queries to work just with n-grams that occur in corpora that can be found in different way based on preprocessed data and now query took less than 0.1 seconds on 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 entitites. Let's have a $sequence = (w_1, w_2, w_3, ..., w_X)$ then we will count both logDice and MI-score for each pair (w_I, w_{I+1}). According to theory of mutual information, it can be expected that lower values of mutual score provides better identification of borders as these words shouls be the word that relates to each other less than others. If sequence itself or candidate n-gram divisions are not mentioned in corpus, we have to modify an equation a bit and replace occurencies of zero by one. On our test suite[1] shown that using logDice results in precision **29.5%** while MI score's precision was only **11.8%**. Examples of selected sequences are shown in table 1 where text in bold represents correct division and numbers in bold represent 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
PSP - Miroslava	**8.77**	**-14.26**	D. - Cerekve	**-0.49**	-18.18
Miroslava - Němcová	10.21	-13.71	**Cerekve - Zdeněk**	1.08	-18.19
			Zdeněk - Jirsa	2.69	**-19.44**

3.2 Method based on mutual information between n-grams

In order, to improve this method, we have decided to extend used equations to work directly on n-grams. There will be only two n-grams for each division because we want to split sequence to just two parts as was explained before. The first n-gram will start at first word of sequence and the second one will end with the last word. Such n-grams does not have to be part of corpora so we will reuse modification proposed in previous subsection and we will replace zeros with ones. Such modifications will be also 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 happends because mutual information between two incorrectly selected n-grams should be lower as they do not occur together as frequently as correct ones. The precision results when logDice was used is **41.2%** and MI score's precision is **29.5%**.

3.3 Method extended with negative n-grams

Previous method did not take into account that if particular word always follow n-gram that it should be part of it, too. Our next method is based on very same idea. In order to count only the *clean* value we will substract the number of longer n-grams from the frequency of the shorter one. Because we are working

[1] 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
PSP - Miroslava Němcová	10.07	-11.23
PSP Miroslava - Němcová	**10.89**	**-9.86**
D. - Cerekve Zdeněk Jirsa	-1.49	-13.83
D. Cerekve - Zdeněk Jirsa	**10.64**	**-4.30**
D. Cerekve Zdeněk - Jirsa	4.44	-8.01

with n-gram model, finding negative occurencies 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 we will have 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 what will happend for this sequence only if there are not any other *Jaromír* in this hockey team as he will have different surname (last word of sequence). In such case the modified frequence will be the frequency of occurencies of same sequence with just last word replaced. The number substracted 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
PSP - Miroslava Němcová	10.17	**-10.62**
PSP Miroslava - Němcová	**11.19**	N/A
D. - Cerekve Zdeněk Jirsa	-1.49	-13.62
D. Cerekve - Zdeněk Jirsa	**11.19**	**-3.26**
D. Cerekve Zdeněk - Jirsa	4.46	N/A

When modified frequenceis will be used in n-gram equations, the results shows that the are too many results in intermediate steps that can not be counted as modified frequency will drop to zero. As the results were heavilly corrupted by this, we have decided to not count mutual information of n-grams but to use only value substracted in equations. We have created two equations that makes 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 then lowest value will show the ideal borderline between them. The other one is just simplification of

previous one where we will sum together just substracted 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 sequences, some normalization will have to be done what will be part of following research.

According to our test suite, the *DIFF-4* has precision **35.2%** what ranks it above all usage of MI-score what could not be expected before experiment. The *DIFF-2* method has 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 will expand *D.* to *Dolní* then division will be find properly.

4 Results and Future Works

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 bigram model does not provide best precision on 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-4 with precision **94.1%**. It is suprising that it does not take into account the frequency of final n-grams. One of the reasons why they are not used yet is that the results are not normalized so they can not be compared across various sequences like MI-score and logDice allows. Obtained results are not really representative yet, so we will prepare a larger datatest for Czech.

Although people tend to split such phrases quite easily, it is often results 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 how to detect sequences that have to divided. Also the idea of normalization of DIFF-2 and DIFF-4 is very interesting as then they can be test against MI-score and logDice also in different applications and for different languages.

Acknowledgments

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. Arvind Arasu, Shivnath Babu, and Jennifer Widom. Cql: A language for continuous queries over streams and relations. In *Database Programming Languages*, pages 1–19. Springer, 2004.

2. Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.
3. Marek Grác. *Rapid Development of Language Resources*. PhD thesis, Dissertation, Masaryk University in Brno, 2013.
4. Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý, Vít Suchomel, et al. The tenten corpus family. In *Proc. Int. Conf. on Corpus Linguistics*, 2013.
5. Pavel Rychlý. A lexicographer-friendly association score. *Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN*, pages 6–9, 2008.