Comparison of Collocation Extraction Measures for Document Indexing

Saša Petrović, Jan Šnajder, Bojana Dalbelo-Bašić, Mladen Kolar
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{*Sasa.Petrovic, Jan.Snajder, Bojana.Dalbelo, Mladen.Kolar*}*@fer.hr*

Abstract. *Automatic extraction of collocations from a corpus is a well-known problem in the field of natural language processing. It is typically carried out by employing some kind of a statistical measure that indicates whether or not two words occur together more often than by chance. As there is an aboundance of these measures proposed by various authors, we have compared some of them on a task of extracting collocations from a corpus of Croatian legal documents for the purpose of document indexing. We propose and evaluate extensions of these measures for collocations consisting of three words.*

Keywords. Corpus statistics, collocation extraction, statistical natural language processing, document indexing.

1. Introduction

There is no widely accepted definition of a collocation in the field of computational linguistics. Definitions range from identifying collocations with idioms, to saying that a collocation is just a set of words occuring together more often than by chance.

We set out to extract two types of collocations. The first type coincides with the definition of an *open compound* in [10]. An open compound is defined as an uninterrupted sequence of words that generally function as a single constituent in a sentence (i.e. *stock market*, *foreign exchange*, etc.). The second type of collocation we wanted to extract was less idiomatic and more compositional than an open compound, and it involved sequences of words often occuring together interrupted by a preposition or a conjunction, and describing similar concepts (e.g. *cure for cancer*, *guns and ammunition*, etc.).

There are many possible applications for collocation extraction [7]: finding multiple word combinations in text for indexing purposes in information retrieval, automatic language generation, word sense disambiguation in multilingual lexicography, improving text categorisation systems, etc.

The purpose of the whole process of extracting collocations was, in our case, improvement of the document indexing system CADIS [6]. We believe that the definition of a collocation we adopted here will be useful for indexing purposes. For the same reason, we include some types of trigrams that are not open compounds — that particular type of trigrams was found very useful for indexing performed by human experts. Focus of our work was to filter out non-collocations that could not otherwise be filtered out by POS tags and frequency alone. In order not to reduce the performance of an indexing system, we aim at high recall (near 100%). That is why we will use the F_1 measure only for comparison of association measures, but not for actually distinguishing collocations from non-collocations.

In the following section we give more insight into related work on this topic, after which, in Section 3, a formal approach to corpus preprocessing is described. Section 4 gives a brief introduction to the used measures and their possible extensions for trigrams. In Section 5 we describe our approach to evaluating measures in more detail, while Section 6 gives and discusses the results.

2. Related work

There are a lot of papers that deal with the problem of collocation extraction, but the lack of a widely accepted definition of a collocation leads to a great diversity in used measures and evaluation tehniques, depending on the purpose of collocation extraction. Smadja [10] uses collocation extraction for the purpose of language generation, so he seeks to capture longer collocations and especially idioms in order to improve his system. He uses a lot of statistical data (word frequencies, deviation, distances, strength, etc.) to accomplish the task. On the other hand, Goldman [5] uses his system FipsCo for terminology extraction, so he relies on a very powerful syntactic parser. Unlike both of them, Wu [14] sets out to extract collocations from a bilingual aligned corpus, and for this he uses a number of preprocessing steps in combination with the log-likelihood ratio and a word alignment algorithm.

There is also no agreed upon method for evaluating collocation extraction systems, so [10] employs the skills of a professional lexicographer, while on the other hand Thanopoulos [13] uses WordNet as a gold standard. Other authors like Evert [4] use a small sample of the entire set of candidates for comparison.

3. Corpus preprocessing

Collocations are extracted according to their ranking with respect to an association measure. These measures are based on raw frequencies of words and sequences of words (n-grams) in corpus, obtained as follows.

3.1. Obtaining n-grams

Let W be a set of words and P be a set of punctuation symbols, and $W \cap P = \varnothing$. We represent the corpus C as a sequence of tokens, i.e. words and punctuation symbols, of finite length k:

$$C = (t_1, t_2, \ldots, t_k) \in (W \cup P)^k.$$

Let $W^+ = \bigcup_{n=1}^{\infty} W^n$ be the set of all word sequences. An *n-gram* is a sequence of words $(w_1, w_2, \ldots, w_n) \in W^+$. From now on, as a shorthand, we write $w_1 w_2 \cdots w_n$ instead of $(w_1, w_2 \ldots, w_n)$. Each occurence of an n-gram can be represented by a tuple $(w_1 \cdots w_n, i) \in W^+ \times \mathbb{N}$, where $i \in \mathbb{N}$ is the position of the n-gram in C. Let S be the set of all n-gram occurences in corpus C, defined as follows:

$$S = \Big\{ \; (w_1 \cdots w_n, i) \in W^+ \times \mathbb{N} : \\ (i \leq k - n + 1) \; \wedge \\ (1 \leq j \leq n)(w_j = t_{i+j-1}) \; \Big\}.$$

Note that n-grams from S do not cross sentence boundaries set by the punctation symbols from P. There are exceptions to this rule: when a word and a punctuation following it form an abbreviation, then the punctuation is ignored. We preprocess the corpus C to reflect this before obtaining n-grams.

3.2. Lemmatisation

Words of an n-gram occur in sentences in inflected forms, resulting in various forms of a single n-gram. In order to conflate these forms to a single n-gram, each word has to be *lemmatised*, i.e. a lemma for a given inflected form has to be found. In this work we restrict ourselves to ambiguous lemmatisation by not taking into account the context of the word. Let $lm : W \to \wp(W)$ be the lemmatisation function mapping each word into a set of ambiguous lemmas, where $\wp$ is the powerset operator. If a word $w \in W$ cannot be lemmatised for any reason, then $lm(w) = w$.

Another linguistic information obtained by lemmatisation is the word's part-of-speech (POS). In this work we only consider the following four: nouns (N), adjectives (A), verbs (V) and stop-words (X). Here stop-words include prepositions and conjunctions. Let $POS = \{N, A, V, X\}$ be the set of corresponding POS tags. Let function $pos : W \to \wp(POS)$ associate to each word a set of ambiguous POS tags. If word $w \in W$ cannot be lemmatised, then POS is unknown and we set $pos(w) = POS$. Let $POS^+ = \bigcup_{n=1}^{\infty} POS^n$ be the set all POS tag sequences, called *POS patterns*.

3.3. Counting and POS filtering

Let $f : W^+ \to \mathbb{N}_0$ be a function associating to each n-gram its frequency in the corpus C. It is defined as follows:

$$f(w_1 \cdots w_n) = \Big| \Big\{ \; (w'_1 \cdots w'_n, i) \in S : \\ (1 \leq j \leq n)(lm(w_j) \cap lm(w'_j) \neq \varnothing) \Big\} \Big|.$$

Due to lemmatisation, the obtained frequency is insensitive to n-gram inflection.

Only n-grams of the appropriate POS patterns will be considered collocation candidates. Let $POS_f \subseteq POS^+$ be the set of allowable POS patterns defining the *POS filter*. An n-gram $w_1 w_2 \cdots w_n$ is said to pass the POS filter iff:

$$POS_f \cap \prod_{j=1}^{n} pos(w_j) \neq \varnothing,$$

where Π denotes the Cartesian product.

4. Association measures

4.1. Definitions for digrams

Association measures (AMs) are used to indicate the strength of association of two words. We will now describe four commonly used measures along with some of their properties.

Pointwise mutual information[1] (PMI) [2] is a measure that comes from the field of information theory, and it measures the amount of information we have about the occurence of one word if we are provided with information about occurence of the other word. It is given by the formula:

$$I(x, y) = \log_2 \frac{P(xy)}{P(x)P(y)}, \tag{1}$$

where x and y are words and $P(x)$, $P(y)$, $P(xy)$ are probabilities of occurence of words x, y, and digram xy, respectively. Those probabilities are approximated by relative frequencies of the words or digrams in the corpus.

The *Dice coefficient* is defined as:

$$DICE(x, y) = \frac{2f(xy)}{f(x)f(y)}, \tag{2}$$

where $f(x)$, $f(y)$, $f(xy)$ are frequencies of words x, y and digram xy, respectivley. The Dice coefficient is sometimes considered superior to information theoretic measures, especially in translating using a bilingual aligned corpus [7].

[1] The definition of mutual information we used here is more common in corpus linguistic than in information theory, where the definition of average mutual information is more commonly used.

Next two measures emerge from the field of statistics. They deal with hypotesis testing, i.e. with acceptance or rejection of the *null-hypotesis* (in our case the *null-hypotesis* being "words x and y occur together by chance"). First of these measures is the chi-square test, defined as:

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \tag{3}$$

where O_{ij} and E_{ij} are observed and expected frequencies in a contingency table [7].

The log-likelihood ratio (LL) [9] (entropy version) is defined as:

$$G^2 = \sum_{i,j} O_{ij} \log \frac{O_{ij}}{E_{ij}}. \tag{4}$$

4.2. Extending the measures for trigrams

All existing AMs are defined for the association between two words, which, obviously, makes them inadequate for extracting collocations consisting of three words. Therefore, we need to extend the existing measures. An overview of the existing extensions of PMI is given in [12]. We tested the following formulæ for PMI:

$$I_a(x, y, z) = \frac{P(xyz)}{P(x)P(y)P(z)}, \tag{5}$$

$$I_b(x, y, z) = \frac{I(xy, z) + I(x, yz)}{2}, \tag{6}$$

$$I_c(x, y, z) = \frac{I(x, y) + I(y, z) + I(x, z)}{3}. \tag{7}$$

Formula (5) is the natural extension of PMI for n-gram of any size n [8], formula (6) is due to Boulis [1], and formula (7) is proposed by Tadic [12].

Along with extending the Dice coefficient in the same way as PMI was extended in formulas (6) and (7), we also tested the natural extension of the Dice coefficient for trigrams [8]:

$$DICE(x, y, z) = \frac{3f(xyz)}{f(x) + f(y) + f(z)}. \tag{8}$$

We also propose a heuristics for trigrams based on the assumption that for different types of collocations one should use different

AMs. It basically consists of combining POS information with AMs as follows:

$$H(x,y,z) = \begin{cases} 2I(x,z) & \text{if } X \in pos(y), \\ I_a(x,y,z) & \text{otherwise.} \end{cases}$$

If the second word in the trigram is a stopword (i.e. POS tag is X), we only compute the strength of association between the other two words, otherwise we compute the strength of association among all three words.

5. Evaluating AMs

5.1. Corpus preprocessing

The corpus we used for obtaining n-grams and their frequencies and testing of AMs consists of 7008 Croatian legal documents from the Croatian National Corpus [3]. It contains over 1 million words, 167 911 lemmas, 1 816 121 digrams, and 4 656 013 trigrams.

For lemmatising Croatian, we used a morphological lexicon constructed by rule-based automatic acquisition [11]. The so obtained lexicon is not perfectly accurate, thus prone to lemmatisation and POS tagging errors. The POS filters used for digrams are AN and NN, while for trigrams the following filters were used: ANN, AAN, NAN, NNN, NXN. Note that, as said in 3.2., the words not found in the dictionary are given all possible POS tags.

5.2. Our approach to evaluation

Comparison of AMs is usually done by having an expert evaluate n-best candidates for each measure, and manually assign each n-gram a label indicating whether it is a collocation or not. This is a time expensive procedure, and it can be very tiresome for the human expert. When we take into account also the size of our corpus and the number of measures we want to compare, it becomes clear that such a comparison is impossible.

Therefore, we adopted the approach used by Evert [4] and extracted a small random sample of positive and negative examples (i.e. collocations and non-collocations), which we used to compute the precision and recall among n-best candidates for each measure. The positive examples were extracted by having a human expert read randomly setected documents and extract obvious collocations from them (e.g. *martial art*, *organized crime*, etc.). In other words, we extracted the positive examples before applying the POS and frequency filters, rather than after like in [4]. This was done so we could also compare the effect POS and frequency filtering have on the recall, i.e. how much mistakes are due to lemmatisation and how many collocations will be lost by applying the frequency filter. The negative examples were extracted by having a human expert isolate the obvious non-collocations from a list of collocations that passed a certain POS filter (e.g. *different schedule*, *every person*, etc.). This means they were extracted after applying the POS filter because if we did that before the filtering, we would get a lot of negative examples that do not pass the POS filter, resulting in an unrealistically high precision (which would be due to a good filter, not a good measure). The random sample for digrams consists of 229 collocations (considered positive examples) and 229 non-collocations (considered negative examples), and for trigrams it consists of 100 collocations and 100 non-collocations. This, of course, does not reflect the true state of the whole population, as there are naturally more negative than positive examples. But, it does give us a solid basis to compare our measures on, as one would normally expect that the relative performance of measures is independent of the test sample.

6. Results

6.1. Digrams

The results for digrams are shown in Fig. 1. They were obtained after applying POS filter and frequency filter with a threshold of 3, meaning that a digram has to appear at least 3 times in the corpus to pass the frequency filter. Out of all digrams in the corpus, 49.5% passed the POS filter, 31.6% passed the frequency filter, and 14.1% passed both filters. We use both of filters because the maximum recall for all digrams that pass these filters is 95%, and we decided to tolerate a loss of about 5% of collocations. The loss of 5% of collocations is due to POS tagging errors (e.g. a NN collocation with one of the nouns incorrectly tagged as a verb does not pass the POS filter), and to

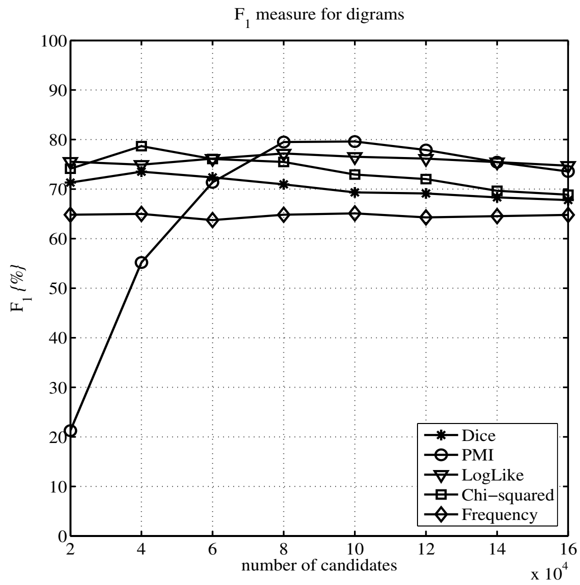

Figure 1: F_1 measure for digrams

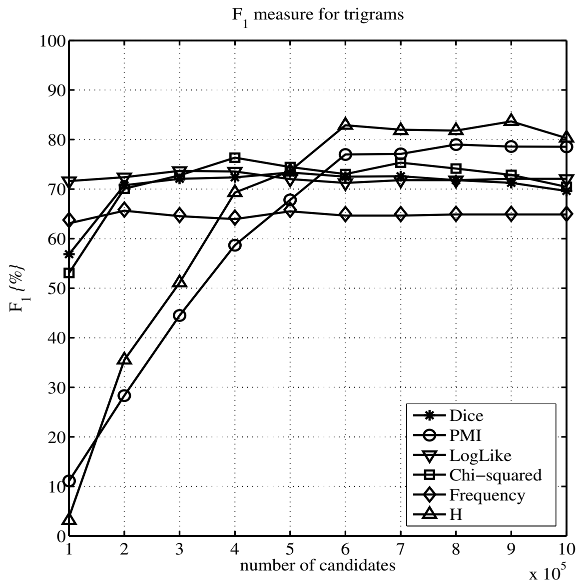

Figure 2: F_1 measure for trigrams

the fact that there are collocations appearing less than 3 times in the corpus. From Fig. 1 it is obvious that all of the tested measures perform better than sorting by raw frequency (which justifies the use of AMs) and that PMI performs the best, followed by chi-square and LL, while the Dice coefficient performs worst.

6.2. Trigrams

We have tested the formulæ for PMI and Dice coefficient given in Section 4, as well as the extensions of chi-square and LL measures. These extensions are obtained from (6) and (7) by replacing PMI with chi-square and LL, respectively. We will, however, omit the results of all but the best extension of each measure (for each measure, the maximum F_1 score of the best extension outperforms the maximum F_1 score of other extensions by 2-5%).

Out of all trigrams in the corpus, 32.4% passed the POS filter, 19.5% passed the frequency filter (with the threshold of 3), and only 6.1% of all trigrams passed both filters. Maximum recall for trigrams that passed both filters was only 93%, which is unacceptable in our opinion. Therefore, unlike with digrams, we decided to use only POS filtering thereby achieving a very good recall of 99%.

The results for trigrams are shown in Fig. 2. PMI outperformed the other three widely used measures, but the heuristics we proposed gave even better results. This confirms our intuition that different AMs should be used for extracting different types of collocations. It is also interesting to note that the best extension of LL, Dice and chi-square showed to be the one derived from (6), indicating that when extracting collocations consisting of three words, one should compute the mean between strength of association of initial/final digram with ending/starting single word.

6.3. Finding relevant collocations for indexing purposes

For indexing purposes we cannot simply take the n-best candidates of the best measure as collocations, because that would lead to problems if we wished to extend our corpus (a bigger corpus obviously contains more collocations than a smaller one). On the other hand, using a threshold of an AM for distinguishing collocations from non-collocations is insensitive to corpus size. This is due to the fact that a threshold of an AM tells us how strong two (or more) words need to be associated to be considered a collocation. Obviously, that does not depend on how many other, “stronger” collocations are there in the corpus.

For example, after finding PMI to be the best choice for extracting collocations consisting of two words, we computed recall and precision for each threshold of PMI ranging from 0 to 20 (with a step of 1) and then decided that a threshold of 4 (determined by the maximum recall with the best precision) will be used to indicate if digram is a collocation or not. For trigrams, we used a threshold of 5.

7. Conclusion

In this paper we compared four widely used AMs for extracting collocations consisting of two words in a corpus of Croatian legal documents. The results showed that PMI outperforms LL, chi-square and the Dice coefficient.

There are very few measures mentioned in the literature for extracting collocations consisting of three words. We therefore proposed extensions of the chi-square and LL measures in the same manner PMI and Dice were extended. Surprisingly, LL and the Dice coefficient performed similarly, while PMI again outperformed the other three tested measures. Also, we proposed a heuristics based on the assumption that for different types of collocations we should use different AMs. That heuristics gave very good results, outperforming all of the tested measures.

For the actual use of collocation extraction in document indexing, one needs to find an optimal threshold of a chosen AM, and we outlined how to determine such a threshold.

For future work, we plan to experiment with other AMs and extend them for tetragrams.

References

[1] Boulis C. Clustering of Cepstrum Coefficients Using Pairwise Mutual Information. Tehnical Report EE516, University of Washington, Seattle, 2002.

[2] Church K, Hanks P. Word association norms, mutual information, and lexicography. Computational linguistics, 16(1): 22–29; 1990.

[3] Croatian national corpus. http://www.hnk.ffzg.hr [01/16/2005]

[4] Evert S, Krenn B. Using small random samples for the manual evaluation of statistical evaluation measures. Computer speech and language, 19: 450–466; 2005.

[5] Goldman J-P, Wehrli E. FipsCo: A syntax-based system for terminology extraction. Grammar and Natural Language Processing Conference. Universite du Quebec at Montreal, 2001.

[6] Kolar M, Vukmirović I, Dalbelo Bašić B, Šnajder J. Computer-aided document indexing system. Journal of Computing and Information Technology, 13(4): 299–305; 2005.

[7] Manning C, Schütze H. Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA; 1999.

[8] McInnes B T. Extending the log-likelihood measure to improve collocation identification. Master thesis, University of Minnesota; 2004.

[9] Oakes M P. Statistics for corpus linguistics. Edinbourgh University Press, 1998.

[10] Smadja F, McKeown K. Automatically extracting and representing collocations for language generation. In Proc. of the 28th Annual Meeting of the ACL; 1990. p. 252–259.

[11] Šnajder J. Rule-based automatic acquisition of large-coverage morphological lexicons for information retrieval. Tech. Report, MZOS 2003-082, ZEMRIS, FER, University of Zagreb, 2005.

[12] Tadić M, Šojat K. Finding multiword term candidates in Croatian. In Proc. of IESL2003 Workshop, Borovets, Bulgaria; 2003. p. 102–107.

[13] Thanopoulos A, Fakotakis N, Kokkinakis G. Comparative evaluation of collocation extraction metrics. In Proc. of the LREC 2002 Conference; 2002. p. 609–613.

[14] Wu C-C, Chang J S. Bilingual collocation extraction based on syntactic and statistical analyses. Computational Linguistics and Chinese Language Processing, 9(1): 1–20; 2004.