Extracting Nested Collocations

Katerina T. Frantzi and Sophia Ananiadou
Dept. of Computing
Manchester Metropolitan University
Manchester, M1 5GD, U.K.
{K.Frantzi,S.Ananiadou}@doc.mmu.ac.uk

Abstract

This paper provides an approach to the semi-automatic extraction of collocations from corpora using statistics. The growing availability of large textual corpora, and the increasing number of applications of collocation extraction, has given rise to various approaches on the topic. In this paper, we address the problem of *nested collocations*; that is, those being part of longer collocations. Most approaches till now, treated substrings of collocations as collocations, only if they appeared frequently enough by themselves in the corpus. These techniques left a lot of collocations unextracted. In this paper, we propose an algorithm for a semi-automatic extraction of nested uninterrupted and interrupted collocations, paying particular attention to nested collocation.

1 Introduction

The increased interest in collocation extraction comes from the fact that they can be used for many NLP applications such as machine translation, machine aids for translation, dictionary construction, and second language learning, to name a few.

Recently, large scale textual corpora give the potential of working with the real data, either for grammar inferring, or for enriching the lexicon. These corpus-based approaches have also been used for the extraction of collocations.

In this paper we are concerned with *nested collocations*. Collocations that are substrings of other longer ones. Regarding this type of collocation, the approaches till now could be divided into two groups: those that do not refer to *substrings of collocations* as a particular problem, (Church and Hanks, 1990; Kim and Cho, 1993; Nagao and Mori, 1994), and those that do (Kita et al., 1994; Smadja, 1993; Ikehara et al., 1995; Kjellmer, 1994). However, even the latter, deal with only part of the problem: they try not to extract the unwanted substrings of collocations. In favour of this, they leave a large number of nested collocations unextracted.

In section 2 collocations are briefly discussed and the problem is determined. In section 3 our approach to the problem, the algorithm and an example are given. In section 4 the experiments are discussed and the method is compared with that proposed by (Kita et al., 1994). In section 5 there are comments on related work and finally Section 6 contains the conclusions and the future work.

2 Collocations - The Problem

Collocations are pervasive in language: "letters" are "delivered", "tea" is "strong" and not "powerful", we "run programs", and so on. Linguists have long been interested in collocations and the definitions are numerous and varied. Some researchers include multi-element compounds as examples of collocations; some admit only collocations consisting of pairs of words, while others admit only collocations consisting of a maximum of five or six words; some emphasize syntagmatic aspects, others semantic aspects. The common points regarding collocations appear to be, as (Smadja, 1993) suggests[1]: they are arbitrary (it is not clear why to "fall through" means to "fail"), they are domain-dependent ("interest rate", "stock market"), they are recurrent and cohesive lexical clusters: the presence of one of the collocates strongly suggests the rest of the collocation ("United" could imply "States" or "Kingdom").

[1] He classifies collocations into predicative relations, rigid noun phrases and phrasal templates.

It is not the goal of this paper to provide yet another definition of collocation. We adopt as a working definition the one by (Sinclair, 1991)

> Collocation is the occurrence of two or more words within a short space of each other in a text.

Let us recall that *collocations are domain-dependent.* Sublanguages have remarkably high incidences of collocation (Ananiadou and McNaught, 1995). (Frawley, 1988) neatly sums up the nature of sublanguage, showing the key contribution of collocation:

sublanguage is strongly lexically based

sublanguage texts focus on content

lexical selection is syntactified in sublanguages

collocation plays a major role in sublanguage

sublanguages demonstrate elaborate lexical cohesion.

The particular structures found in sublanguage texts reflect very closely the structuring of a sublanguage's associated conceptual domain. It is the particular syntactified combinations of words that reveal this structure. Since we work with sublanguages we can use "small" corpora as opposed as if we were working with a general language corpus. In the Brown Corpus for example, which consists of one million words, there are only 2 occurrences of "reading material", 2 of "cups of coffee", 5 of "for good" and 7 of "as always", (Kjellmer, 1994).

We extract uninterrupted and interrupted collocations. The interrupted are phrasal templates only and not predicative relations. We focus on the problem of the extraction of those collocations we call *nested collocations.* These collocations are at the same time substrings of other longer collocations. To make this clear, consider the following strings: "New York Stock Exchange", "York Stock", "New York" and "Stock Exchange". Assume that the first string, being a collocation, is extracted by some method able to extract collocations of length two or more. Are the other three extracted as well? "New York" and "Stock Exchange" *should* be extracted, while "York Stock" should not. Though the examples here are from domain-specific lexical collocations, grammatical ones can be nested as well: "put down as", "put down for", "put down to" and "put down".

(Smadja, 1993; Kita et al., 1994; Ikehara et al., 1995), mention about substrings of collocations. Smadja's Xtract produces only the biggest possible n-grams. Ikehara et al., exclude the substrings of the retrieved collocations.

A more precise approach to the problem is provided by (Kita et al., 1994). They extract a substring of a collocation if it appears a significant amount of times by itself. The following example illustrates the problem and their approach: consider the strings a="in spite" and b="in spite of", with $n(a)$ and $n(b)$ their numbers of occurrencies in the corpus respectively. It will always be $n(a) \geq n(b)$, so whenever b is identified as a collocation, a is too. However, a should not be extracted as a collocation. So, they modify the measure of *frequency of occurrence* to become

$$K(a) = (|a| - 1)(n(a) - n(b)) \qquad (1)$$

where
a is a word sequence
$|a|$ is the length of a
$n(a)$ is the number of occurrencies of a in the corpus.
b is every word sequence that contains a
$n(b)$ is the number of occurrencies of b

As a result they do not extract the sub-strings of longer collocations unless they appear a *significant* amount of times by themselves in the corpus.

The problem is not solved. Table 2 gives the extracted by Cost-Criteria n-grams containing "Wall Street". The corpus consists of 40,000 words of market reports. Only those n-grams of frequency 3 or more are considered. It can be seen that "Wall Street" is not extracted as a collocation, though it has a frequency of occurrence of 38.

Table 1: n-grams extracted by Cost-Criteria containing "Wall Street"

C-C	F	Candidate Collocations
114	19	Staff Reporter of The Wall Street Journal
6	3	Wall Street analysts
0	19	Reporter of The Wall Street Journal
0	19	Staff Reporter of The Wall Street
-72	20	of The Wall Street Journal
-108	22	The Wall Street Journal
-108	26	Wall Street Journal
-152	19	Reporter of The Wall Street
-152	38	Wall Street
-228	20	of The Wall Street
-230	23	The Wall Street

3 Our approach - The Algorithm

We call the extracted strings *candidate collocations* rather than *collocations*, since what we accept as collocations depends on the application. It is the human judge that will give the final decision. This is the reason we consider the method as semi-automatic.

Let us consider the string "New York Stock Exchange". Within this string, that has already been extracted as a candidate collocation, there are two substrings that should be extracted, and one that should not. The issue is how to distinguish when a substring of a candidate collocation *is* a candidate collocation, and when it is not. Kita et al. assume that the substring is a candidate collocation if it appears by itself (with a relatively high frequency). To this we add that:

> the substring appears in more than one candidate collocations, even if it does not appear by itself.

"Wall Street", for example, appears 30 times in 6 longer candidate collocations, and 8 times by itself. If we considered only the number of times it appears by itself, it would get a low value as a candidate collocation. We have to consider the number of times it appears within longer candidate collocations. A second factor is the number of these longer collocations. The greater this number is, the better the string is distributed, and the greater its value as a candidate collocation. We make the above conditions more specific and give the measure for a string being a candidate collocation. The measure is called *C-value* and the factors involved are the string's frequency of occurrence in the corpus, its frequency of occurrence in longer candidate collocations, the number of these longer candidate collocations and its length. Regarding its length, we consider longer collocations to be "more important" than shorter appearing with the same frequency. More specifically, if $|a|$ is the length[2] of the string a, its *C-value* is analogous to $|a|-1$. The -1 is given since the shortest collocations are of length 2, and we want them to be "of importance" 2-1=1.
More specifically:

1. If a has the same frequency with a longer candidate collocation that contains a, it is assigned *C-value(a)=0* i.e. is not a collocation. It is straightforward that in this case a appears in one only longer candidate collocation.

2. If $n(a)$ is the number of times a appears, and a is not a substring of an already extracted candidate collocation, then a is assigned

$$C\text{-}value(a) = (|a| - 1)n(a) \qquad (2)$$

3. If a appears as a substring in one or more collocations (not with the same frequency), then it is assigned

$$C\text{-}value(a) = (|a| - 1)(n(a) - \frac{t(a)}{c(a)}) \qquad (3)$$

where $t(a)$ is the total frequency of a in longer candidate collocations and $c(a)$ the number of those candidate collocations. This is the most complicate case.

The importance of the number of occurrences of a string in a longer string is illustrated with the denominator of the fraction in Equation 3. The bigger the number of strings a substring appears in, the smaller the fraction $\frac{total\ frequency\ of\ occurrence}{number\ of\ occurrences}$, the bigger the *C-value* of the string.

The algorithm for the extraction of the candidate collocations follows:

extract the n-grams
decide on the lowest frequency of collocations
remove the n-grams below this frequency
for all n-grams a of maximum length
    calculate their *C-value*= $(n-1)n(a)$
    for all substrings b
        revise $t(b)$
        revise $c(b)$
for all smaller n-grams a in descending order
    if (total frequency of a)=(frequency of a in a longer string)
        a is NOT a collocation
    else
        if a appears for the first time
            C-value= $(n-1)n(a)$
        else
            C-value= $(n-1)(n(a) - \frac{t(a)}{c(a)})$
        for all substrings b
            revise $t(b)$
            revise $c(b)$

The above algorithm computes the *C-value* of each string in an incremental way. That is, for each string a, we keep a tuple $\langle n(a), t(a), c(a)\rangle$ and we revise the $t(a)$ and $c(a)$ values. For each n-gram b, every time it is found in a longer extracted

[2] We use the same notation with (Kita et al., 1994).

n-gram a, the values $t(b)$ and $c(b)$ are revised:
$t(b) = t(b) + (n(a) - t(a))$
$c(b) = c(b) + 1$.
In the initial stage, $n(a)$ is set to the frequency of a appearing on its own, and $t(a)$ and $c(a)$ are set to 0.

Table 2: n-grams extracted by C-value containing "Wall Street"

C-V	F	Candidate Collocations
114	19	Staff Reporter of The Wall Street Journal
37.34	26	Wall Street Journal
36	22	The Wall Street Journal
33	38	Wall Street
31.34	23	The Wall Street
6	3	Wall Street analysts
4	20	of The Wall Street Journal
0	19	Reporter of The Wall Street
0	19	Reporter of The Wall Street Journal
0	19	Staff Reporter of The Wall Street
0	20	of The Wall Street

An example:
Let us calculate the *C-value* for the string "Wall Street". Table 2 shows all the strings that appear more that twice, and that contain "Wall Street".
1. The analysis starts from the longest string, the 7-gram "Staff Reporter of The Wall Street Journal". Its *C-value* is calculated from Equation 2. For each substrings contained in the 7-gram, the number 19 (the frequency of the 7-gram) is kept, as its (till now) frequency of occurrence in longer strings. For each of them, the fact that they have been already found in a longer string is kept as well. Therefore, t("Wall Street")=19 and c("Wall Street")=1.
2. We continue with the two 6-grams. Both of them, "Reporter of The Wall Street Journal" and "Staff Reporter of The Wall Street" get *C-value=0* since they appear with the same frequency as the 7-gram that contains them. Therefore, they do not form candidate collocations and they do not change the t("Wall Street") and the c("Wall Street") values.
3. For the 5-grams, there is one appearing with a frequency bigger than that of the 7-gram it is contained in, "of The Wall Street Journal". This gets its *C-value* from Equation 3. Its substrings increase their frequency of occurrence (as substrings) by 20 -19=1 (20 is the frequency of the 5-gram and 19 the frequency it appeared in longer candidate collocations), and the number of occurrence as substring by 1. Therefore, t("Wall Street")=19+1=20 and c("Wall Street")=1+1=2. The other 5-gram is not a candidate collocations (it gets *C-value=0*).
4. For the 4-grams, the "The Wall Street Journal" occurs in two longer n-grams and therefore gets its *C-value* from Equation 3. From this string, t("Wall Street")=20+2=22 and c("Wall Street")=2+1=3. The "of The Wall Street" is not accepted as a candidate collocations since it appears with the same frequency as the "of The Wall Street Journal".
5. "Wall Street analysts" appears for the first time so it gets its *C-value* from Equation 2. "Wall Street Journal" and "The Wall Street" appearing in longer extracted n-grams get their values from Equation 3. They make t("Wall Street")=22+3+4+1=30 and c("Wall Street")=3+1+1+1=6.
6. Finally, we evaluate the *C-value* for "Wall Street" from Equation 3. We find *C-value*("Wall Street")=33.

4 Experiments - Comparison

The corpus used for the experiments is quite small (40,000 words) and consists of material from the Wall Street Journal newswire. For these experiments we used n-grams of maximum length 10. Longer n-grams appear once only (because of the size of the corpus). The maximum length of the n-grams to be extracted is variable and depends on the size of the corpus and the application.

From the extracted n-grams, those with a frequency of 3 or more were kept (other approaches get rid of n-grams of such low frequencies (Smadja, 1993)). These n-grams were forwarded into the implementation of our algorithm as well as our implementation of the algorithm by (Kita et al., 1994).

The Cost-Criteria algorithm needs a second threshold (besides the one for the frequency of the n-grams): for every n-gram a, $K(a)$ is evaluated, and only those n-grams with this value greater than the preset threshold will take part to the rest of the algorithm. We set this threshold to 3 again for the same reason as above (the gain we would get for precision if we had set a higher threshold would be lost on recall).

Table 3 shows the candidate collocations with the higher values, extracted with *C-value*. A lot of candidate collocations extracted may seem unimportant. This is because the algorithm extracts the word sequences that are frequent. Which of these candidate collocations we should keep depends on the application. Brill's part-of-speech tagger, (Brill, 1992), was used to remove the n-grams that had an article as their last word.

Table 3: Extracted candidate collocation with *C-value* in descending order.

C-V	F	Candidate Collocations
184	92	WALL STREET JOURNAL
114	19	Staff Reporter of The Wall Street Journal
87.6	93	United States
79.6	44	the United States
53.2	59	the United
49.5	20	<number> to <money> from
44.75	25	to <money> from
44	48	said it
41.17	44	the company
37.34	26	Wall Street Journal
36	6	<number> to <money> from <money> a year
36	22	The Wall Street Journal
33	38	Wall Street
31.34	23	The Wall Street
27.8	31	a year
27	3	There were <number> selling days in the period this year
27	3	x There were <number> selling days in the period this
27	30	to be
24	27	will be
24	10	at the end of
23.34	27	the company's
21.34	27	compared with
21	10	<COMMENT> Paragraphing Error </COMMENT>
20	10	<money> a share
20	5	priced at <number> to yield
19.67	23	White House
18.5	23	the market
18	18	Total cars
18	6	in the United States
18	9	Tan Sri Khoo
18	9	The United States
18	21	National Bank
17	17	has been
17	17	said Mr
17	17	said that
17	11	the end of
17	21	of its
16.4	19	fourth quarter
16	16	Diamond Shamrock
16	4	<number> <COMMENT> Paragraphing Error <COMMENT>
16	8	as well as
16	19	that it
15.5	20	more than
15	15	had been
15	15	it is
15	3	<money> at the end of <number>
15	3	In a Securities and Exchange Commission
15	3	a <number> for <number> stock split
15	3	sales rose <number> to <money> from
15	5	that the United States
15	18	because of

Among the extracted n-grams we can see the domain-specific candidate collocations, such as "Staff Reporter of the Wall Street", "National Bank" etc., and those that appear within other collocations rather than by themselves, "Wall Street Journal","Wall Street" etc.
There are, however, problems:

1. We did not calculate the precision or recall of the *C-value* algorithm. These calculations depend on the definition of collocation and they are domain dependent. (Kjellmer mentions 19 categories of collocation (Kjellmer, 1994)).

2. As it can be seen from Table 3, one string appearing both in small and capital letters is treated as two different strings. The problem can be partially solved if we use a canonical form. However, if we want to apply the algorithm for the extraction of domain-specific collocations, case is pertinent.

3. ".", in strings like "e.t.c.", "et al." etc., is taken as a sentence boundary even when it is not.

4. How to filter out the extracted n-grams that are not relevant to the application (for the candidate collocations) we are interested in, is another problem. Actually, for some of the extracted n-grams ("to be", "has been", "said that", etc.), we cannot think of any application that these n-grams would be useful. And though some of them could be filtered out by a part-of-speech tagger, we cannot say this for all the types of the "unwanted" extracted n-grams.

5 Related Work

Besides the work by Kita et al. mentioned earlier, there are other interesting approaches to the extraction of collocations.

(Choueka et al., 1983) proposed a method, based on the observed frequency of sequencies of words, to extract uninterrupted collocations, but the results are dependent on the size of the corpus.

(Church and Hanks, 1990), proposed the association ratio, a measure based on mutual information (Fano, 1961), to estimate word association norms. They identify pairs of words that appear together more often than by chance. The collocations they identify could also be due to semantic reasons. They allow gaps between the words and therefore extract interrupted word sequences. Since they only deal with collocations of length two (though mutual information can be extended for an arbitrary number of events, (Fano, 1961; McElliece, 1977)), they do not consider nested collocations.

(Kim and Cho, 1993), proposed mutual information to calculate the degree of word association

of compound words. They extend the measure for three words in a different way than that defined by (Fano, 1961), and no mention is given to how their formulas would be extended for word-sequences of length more that three. They do not consider nested collocations.

(Smadja, 1993), extracts uninterrupted as well as interrupted collocations (predicative relations, rigid noun phrases and phrasal templates). The system performs very well under two conditions: the corpus must be large, and the collocations we are interested in extracting, must have high frequencies.

(Nagao and Mori, 1994), extract collocations using the following rule: longer collocations and frequent collocations are more important. An improvement to this algorithm is that of (Ikehara et al., 1995). They proposed an algorithm for the extraction of uninterrupted as well as interrupted collocations from Japanese corpora. The extraction involves the following conditions: longer collocations have priority, more frequent collocations have priority, substrings are extracted only if found in other places by themselves.

Finally, the *Dictionary of English Collocations*, (Kjellmer, 1994), includes n-grams appearing even only once. For each of them its exclusive frequency (number of occurrences the n-gram appeared by itself), its inclusive frequency (number of times it appeared in total) and its relative frequency (the ratio of its actual frequency to its expected frequency), is given.

6 Conclusions and Future Work

As collocation identification (either in general language or in sublanguages) finds many applications, the need to automate, as much as possible, that process increases. Automation is helped by the recent availability of large scale textual corpora.

In this paper we dealt with the extraction of uninterrupted and interrupted collocations focusing on those we call *nested collocations* (those being substrings of other collocations). A method for their extraction was proposed.

In future, we plan to extend our algorithm to include predicative relations. We are going to incorporate linguistic knowledge to improve the results. Finally, this algorithm will be applied for term extraction.

7 Acknowledgements

We thank our anonymous reviewers for their comments.

References

Ananiadou, S.; McNaught, J. 1995. Terms are not alone: term choice and choice terms. In *Journal of Aslib Proceedings*, vol.47,no.2:47–60.

Brill, E. 1992. A simple rule-based part of speech tagger. In *Proc. of the Third Conference of Applied Natural Language Processing, ACL*, pages 152–155.

Choueka, Y., Klein, T. and Neuwitz, E. 1983. Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus. In *Journal of Literary and Linguistic Computing*, 4:34–38.

Church, K.W. and Hanks, P. 1990. Word Association Norms, Mutual Information, and Lexicography. In *Computational Linguistics*, 16:22–29.

Frawley, W. 1988. Relational models and metascience. In Evens, M. (ed.) *Relational models of the lexicon*, Cambridge:Cambridge University Press, 335–372.

Nagao, M. and Mori, S. 1994. A new Method of N–gram Statistics for Large Number of n and Automatic Extraction of Words and Phrases from Large Text Data of Japanese. In *Proc. of COLING*, pages 611–615.

Fano, R.M. 1961. In *Transmission of information: a statistical theory of communications*, M.I.T. Press, New York.

Ikehara, S.; Shirai, S. and Kawaoka, T. 1995. Automatic Extraction of Collocations from Very Large Japanese Corpora using N-gram Statistics. In *Transactions of Information Processing Society of Japan*, 11:2584–2596. (in Japanese).

Kim, P.K. and Cho, Y.K. 1993. Indexing Compound Words from Korean Texts using Mutual Information. In *Proc. of NLPRS*, pages 85–92.

Kita, K.; Kato, Y.; Omoto, T. and Yano, Y. 1994. A Comparative Study of Automatic Extraction of Collocations from Corpora: Mutual Information vs. Cost Criteria. In *Journal of Natural Language Processing*, 1:21–33.

Kjellmer, G. 1994. *A Dictionary of English Collocations*, Clarendon Press, Oxford.

McEliece, R.J. 1977. *The Theory of Information and Coding*, Addison Wesley, London.

Sinclair, J. 1991. In J. Sinclair and R. Carter, editors, *Corpus,Concordance, Collocation*. Oxford University Press, Oxford, England.

Smadja, F. 1993. Retrieving Collocations from Text: Xtract. In *Computational Linguistics*, 19:143–177.