Extracting Nested Collocations

Katerina FRANTZI†, Sophia ANANIADOU†, Junichi TSUJII‡

†NTT Communication Science Laboratories*
{katerina,sophia}@nttkb.ntt.jp

‡University of Tokyo
tsujii@is.s.u-tokyo.ac.jp

1 Introduction

Large scale textual corpora give the potential of working with the real data, (Ananiadou et al., 1992), either for grammar inferring, or for enriching the lexicon. Corpus-based approaches have also been used for the extraction of collocations. Much research has been going on in this area recently, inspired mainly by statistical and probability based algorithms found in information retrieval.

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

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 can be divided into two groups: those that do not refer to *substrings of collocations* (Church and Hanks, 1990; Kim and Cho, 1993; Nagao and Mori, 1994) and those that do (Smadja, 1993; Ikehara et al., 1995; Kita et al., 1994; Kjellmer, 1994). 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.

2 Collocations - The Problem

Collocation is pervasive in language: 'letters' are 'delivered', 'soup' is 'eaten' and not 'drunk', 'tea' is 'strong' and not 'powerful', we 'run' programs, and so on. The definitions are numerous and varied.

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".

Collocations are domain-dependent, recurrent, cohesive lexical clusters. Sublanguages[1] have remarkably high incidences of collocation (Frawley, 1988; Ananiadou and McNaught, 1995).

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. We will come back to this point later.

In this paper, we examine one type of collocations, what we call *nested collocations*. These collocations are at the same time substrings of other longer collocations. Consider the following strings: "New York Stock Exchange", "York Exchange", "New York" and "Stock Exchange". Assume that the first is extracted as a collocation. Are the rest three extracted as well? "New York" and "Stock Exchange" *should* be extracted, while "York Stock" should not.

3 Our approach - The Algorithm

Let us consider again the string "New York Stock Exchange"[2]. Within this string, that has already been extracted as a candidate collocation[3] there are two substrings that should be extracted as well and one that

*Authors' permanent address: Manchester Metropolitan University, {K.Frantzi,S.Ananiadou}@doc.mmu.ac.uk

[1] Sublanguage utterances are often characterised as 'deviant' with respect to general language.

[2] The examples here are from domain-specific lexical collocations, but grammatical ones can be nested as well: "put down as", "put down for", "put down to" and "put down".

[3] We call the extracted strings *candidate collocations* rather than *collocations*, since the final decision depends on the application.

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). We alter this as: the substring could be a candidate collocation if it appears in more than one longer candidate collocations, even if it does not appear "many" times by itself.
"Wall Street", for example, appears 35 times in 9 longer candidate collocations, and only 4 times by itself. If we only considered the amount of times it appeared by itself, it would be among the least probable candidate collocations. We have to consider the number of times it appears within other longer candidate collocations. The number of these longer collocations is important too. The greater this number, the more independent the string is regarding its context We put the above conditions together into one measure that gives the value for a string being a candidate collocation. We call the measure *C-value* (from Collocation-value). The factors involved are the string's total frequency of occurrence in the corpus, its frequency of occurrence in longer (already extracted) candidate collocations, the number of these longer candidate collocations and the string's length. Regarding its length, we consider longer collocations to be more important. Consider two strings, one of length 2 and one of length 4, both appearing 20 times in the corpus. The probability of 4 words appearing together in the corpus is much smaller than that of 2 words, making the string of length 4 a "stronger" collocation than that of the 2. More specifically, if $|a|$ is the length[4] 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 take them to be of importance $2 - 1 = 1$. We evaluate the *C-value* of a string a with the following formula:

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

unless a appears for the first time, so it is assigned

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

It is straightforward that if a has the same frequency with one longer candidate collocation that contains a, it is assigned $C\text{-}value(a) = 0$ i.e. is not a collocation.
In the above equations
$|a|$ is the length of the string, as mentioned above,
$n(a)$ is the total frequency of occurrence of the string on the corpus,
$t(a)$ is the frequency of occurrence of a in longer (already extracted) candidate collocations.
$c(a)$ the number of those candidate collocations.
Let us give the algorithm:

find the n-grams
decide on the lowest frequency of collocations
remove n-grams below this frequency
for all n-grams a of maximum length
    calculate their $C\text{-}value = (n-1)n(a)$
    for all substrings b
        find the frequency of occurrence of b
        find the number of occurrences of 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
            $C\text{-}value = (n-1)n(a)$
        else
            $C\text{-}value = (n-1)(n(a) - \frac{t(a)}{c(a)})$
        for all substrings b
            find the frequency of occurrence of b
            find the number of occurrences of b

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

The above algorithm computes the *C-value* of each string in an incremental way. That is, for each string a, we keep a triplet $\langle n(a), t(a), c(a)\rangle$ and we revise these values incrementally. 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.

The operation *find* in the algorithm, revises these values for a substring b in a recursive way for a longer string a. Therefore, it computes the new values $t'(b)$ and $c'(b)$ by:
$t'(b) = t(b) + (n(a) - t(a))$
$c'(b) = c(b) + 1$.

C-value	Frequency	Candidate Collocations
114	19	Staff Reporter of The Wall Street Journal
43.2	27	Wall Street Journal
39	23	The Wall Street Journal
35.1111	39	Wall Street
10	2	A Wall Street Journal News Roundup
6	2	Wall Street Journal reporter
6	3	Wall Street analysts
6	3	Wall Street sources

Table 1: Candidate collocations by *C-value* including "Wall Street"

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 length up to 10. 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 2 or more were kept since we adopt Kjellmer's approach on the frequencies of collocations.

4 Related Work

(Choueka et al., 1983) proposed an algorithm that could analyse uninterrupted collocations of an arbitrary length. Being based though on the observed frequency, it extracts the substrings of the extracted collocations, without checking whether these form collocations or not.

(Church and Hanks, 1990) proposed the association ratio, a measure based on mutual information, 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, 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, and 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 (more that 100 occurrencies).

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.

(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, as mentioned before, (Kita et al., 1994) propose a measure called Cost–Criteria that extracts a substring of a collocation only if it appears a significant amount of times by itself.

5 Conclusions and Future Work

As collocation identification finds many applications (in general language and in sublanguages), 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 focused on one type of collocation for which little work has been done: the *nested collocations*. We showed that these types of collocation exist, and proposed an algorithm for their extraction.

In future, we plan to extend our algorithm to include interrupted collocations. We are also going to apply it for term extraction: consider the following terms from the medical sublanguage, "hay fever conjunctivitis", its substrings "hay fever" and "fever conjunctivitis", and the collocation "hay fever affects". If "hay fever conjunctivitis" and "hay fever affects" were extracted (as a term and a domain-specific collocation respectively), by an automatic term and collocation recognition algorithm, and "hay fever" does not appear a significant amount of times by itself in the medical corpus, the latter would not be extracted. Even if linguistic knowledge is used (eg. in terms of morphological rules, or a part-of-speech tagger), it would not differentiate between "hay fever" and "fever conjunctivitis" (both consist of nouns). What makes the difference is that "hay fever" appears in both the 3-word extracted word-sequences, while "fever conjunctivitis" does not.

References

Ananiadou, S.; Tsujii, J.; Arad, I. and Sekine S. 1992. Linguistic Knowledge Acquisition from Corpora. In *Proc. of FGNLP*, pages 61–81.

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.

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.

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.

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.