Extracting Collocations from Text Corpora

Dekang Lin
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
lindek@cs.umanitoba.ca

Abstract

A collocation is a habitual word combination. Collocational knowledge is essential for many tasks in natural language processing. We present a method for extracting collocations from text corpora. By comparison with the SUSANNE corpus, we show that both high precision and broad coverage can be achieved with our method. Finally, we describe an application of the automatically extracted collocations for computing word similarities.

1 Introduction

A collocation is a habitual word combination, such as "weather a storm", "file a lawsuit", and "the falling dollar". Many collocations are idiosyncratic in the sense that they are unpredictable by syntactic and semantic features. For example, "baggage" and "luggage" are synonyms. However, only "baggage" can be modified by "emotional", "historical", or "psychological".

It was argued in (Harris, 1968) that meanings of words are determined to a large extent by their collocational patterns. Collocational knowledge is essential for many natural language processing tasks. It provides a basis for choosing lexical items and is indispensable for generating collocationally restricted sentences (Smadja, 1993). It can also be used to better select a parse tree from the parse forest returned by a broad-coverage parser (Alshawi and Carter, 1994). (Collins, 1997) showed that the performance of statistical parsers can be improved by using lexicalized probabilities, which implicitly capture the collocational relationships between words. (Hindle, 1990) and (Hearst and Grefenstette, 1992) used word collocations as features to automatically discover similar nouns of a given noun.

Collocational knowledge is also of vital importance in second language acquisition. Due to their idiosyncratic nature, word collocations account for many mistakes made by second language learners (Leed and Nakhimovsky, 1979).

Despite the obvious importance of collocational knowledge, it is not usually available in manually compiled dictionaries. In this paper, we present a method for extracting collocations from text corpora. Our goal is to achieve broad coverage as well as high precision in collocation extraction. The broad coverage requirement poses new challenges compared with previous approaches. Although collocations are recurrent, a collocation does not necessarily occur many times in a moderately large corpus. For example, in a 22-million-word corpus containing Wall Street Journal and San Jose Mercury articles, the phrase "emotional baggage" occurred 3 times, "historical baggage" and "psychological baggage" occurred only once each. In order to achieve broad-coverage, a collocation needs to be extracted even if it occurs only a few times in the corpus.

In the remainder of this paper, we first review related work. We then describe the extraction steps which include the collection of dependency triples, automatic correction of the frequency counts of the extracted triples, and the filtering of the triples with mutual information. The resulting collocation database is compared the SUSANNE corpus (Sampson, 1995). Finally, we present an application of the extracted collocations for computing word similarities.

2 Related Work

(Choueka, 1988) presented a method for extracting consecutive word sequences of length 2 to 6. However, many collocations involve words that may be separated by other words, such as "file a lawsuit" or "file a class action lawsuit". (Church and Hanks, 1990) employed mutual information to extract pairs of words that tend to co-occur within a fixed-size window (normally 5 words). Although this overcomes the limitation of word adjacency, the

extracted pairs of words may not be directly related. For example, the words "doctor" and "hospital" often co-occur in a narrow window without being directly related:

> Doctors {arrive at, come from, come to, enter, go to, inspect, leave, sue, work at} hospitals.
>
> Hospitals {accuse, appoint, discipline, hire, include, pay, sue, tell, train} doctors.

As a result, "doctor" and "hospital" were one of the highest ranked collocations in (Church and Hanks, 1990). Xtract (Smadja, 1993) avoids this problem by taking the relative positions of co-occurring words into account. Co-occurring words with a narrower spread are given higher consideration. Smadja also generalized his method to extract collocations involving more than two words.

(Richardson, 1997) is concerned with extracting semantic relationships from machine readable dictionaries. The problem for assigning weights to extracted semantic relationships is very similar to that of ranking the extracted collocations. He proposed to use a fitted exponential curve, instead of observed frequency, to estimate the joint probabilities of events.

3 Extracting Collocational Knowledge

Similar to (Alshawi and Carter, 1994) and (Grishman and Sterling, 1994), we use a parser to extract dependency triples from the text corpus. A dependency triple consists of a head, a dependency type and a modifier. For example, the triples extracted from the sentence "I have a brown dog" are:

```
(have V:subj:N I)
(have V:compl:N dog)
(dog N:jnab:A brown)
(dog N:det:D a)
```

The identifiers for the dependency types are explained in Table 1.

Our text corpus consists of 55-million-word Wall Street Journal and 45-million-word San Jose Mercury. Two steps are taken to reduce the number of errors in the parsed corpus. Firstly, only sentences with no more than 25 words are fed into the parser. Secondly, only complete parses are included in the parsed corpus. The 100 million word text corpus is parsed in about 72 hours on a Pentium 200 with 80MB memory. There are about 22 million words in the parse trees.

Table 1: Dependency types

Label	Relationship between:
N:det:D	a noun and its determiner
N:jnab:A	a noun and its adjectival modifier
N:nn:N	a noun and its nominal modifier
V:compl:N	a verb and its noun object
V:subj:N	a verb and its subject
V:jvab:A	a verb and its adverbial modifier

3.1 Automatic Correction of Parser Mistakes

In an effort to obtain a global parse, a parser often makes poor local decisions, such as choosing the wrong part of speech for lexically ambiguous words. This problem is especially acute when the parser uses a lexicon derived from general-purpose lexical resources, which tend to include many obscure word usages. Our lexicon is derived from the syntactic features in the WordNet (Miller, 1990). The words "job" and "class" can be verbs and "cancel" can be a noun in the WordNet.

Suppose a sentence contains "hold jobs". Since both "hold" and "job" can be used as nouns and verbs, the parser must consider all of the following possibilities:

1. the verb "hold" takes the noun "jobs" as its object;

2. the noun "hold" modifies another noun "jobs";

3. the noun "hold" is the subject of the verb "jobs".

Which one of the dependency relationships is chosen in the parse tree depends on which one of them fits better with the rest of the sentence.

Since the parser tends to generate correct dependency triples more often than incorrect ones, we can make automatic corrections to the frequency counts using a set of correction rules. A correction rule consists of threshold θ and a pair of dependency types (rel, rel') that may potentially be confused with each other. Examples of such pairs include (verb-object, noun-noun), (verb-object, subject-verb), and (noun-noun, subject-verb). If both (w_1, rel, w_2) and (w_1, rel', w_2) are found in the parsed corpus and the ratio between their frequency counts is greater than θ, the lower

frequency count is first added to the higher frequency count and then reset to 0. For example, there are 49 occurrences of verb-object relationship between "hold" and "job" and 1 occurrences of the noun-noun relationship between them. The frequency count of the former is increased to 50 and the frequency count of the latter is reduced to 0.

There do exist pairs of words that can be related via different types of relationships. For example, both the noun-noun modification and verb-object relationship are possible between "draft" and "accord". However, if both types of dependencies are plausible, the disparity between their frequencies is usually much smaller. In our parsed corpus, there are 6 occurrences of "draft an accord" and 4 occurrences of "a draft accord".

We found 699219 pairs of words in the parsed corpus between which there are more than one type of dependency relationships. We used 30 correction rules that modified the frequency counts of 62992 triples. We manually examined 200 randomly selected corrections and found that 95% of the corrections were indeed correct.

3.2 Weeding out coincidences

We now discuss the use of mutual information to separate collocations from dependency triples that occurred merely by coincidences.

A dependency triple (w_1, rel, w_2) can be regarded as the co-occurrence of three events:

  A: a randomly selected word is w_1;
  B: a randomly selected dependency type is rel;
  C: a randomly selected word is w_2.

Mutual information compares the observed number of co-occurrences with the number predicted by a default model which invariably makes independence assumptions. In (Alshawi and Carter, 1994), the mutual information of a triple is defined as:

$$\log \frac{P(A, B, C)}{P(A) \times P(B) \times P(C)}.$$

This definition assumes that when a dependency triple is not a collocation, the three events A, B, and C are independent of one another. This, however, is not the case since the part of speech of the two words in the triple is determined by the type of the dependency relation. For example, if rel is `N:det:D`, then w_1 must be a noun and w_2 must be a determiner.

We make a more reasonable independence assumption: A and C are assumed to be conditionally independent given B. The Bayesian Networks (Pearl, 1988) that represents the independence assumptions in (Alshawi and Carter, 1994) and the independence assumptions made here are shown in Figure 1(a) and 1(b) respectively.
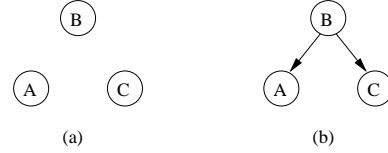

Figure 1: Default probabilistic models of a dependency triple

Under our assumption, the mutual information of (w_1, rel, w_2) is calculated as:

$$\log \frac{P(A, B, C)}{P(B) \times P(A|B) \times P(C|B)}$$

The probabilities in the above formula can be estimated by the frequencies of the dependency triples. However, it is a well-known problem that the probabilities of observed rare events are over-estimated and the probabilities of unobserved rare events are under-estimated. We therefore adjusted the frequency count of (w_1, rel, w_2) with a constant c:

$$
\begin{aligned}
P(A, B, C) &= \frac{\|w_1, rel, w_2\| - c}{\|*, *, *\|}, \\
P(B) &= \frac{\|*, rel, *\|}{\|*, *, *\|}, \\
P(A|B) &= \frac{\|w_1, rel, *\|}{\|*, rel, *\|}, \\
P(C|B) &= \frac{\|*, rel, w_2\|}{\|*, rel, *\|},
\end{aligned}
$$

where $\|w_1, rel, w_2\|$ denotes frequency count of (w_1, rel, w_2) in the parsed corpus. If a wild card $*$ is used, the value is summed over all the possible words or relation types. For example, $\|*, rel, *\|$ denotes the total frequency counts of dependency triples where the relation type is rel. Table 2 shows the top 15 objects of "drink", ranked according to the mutual information measure, with or without adjusting $\|w_1, rel, w_2\|$. Clearly, after the adjustment, many of the previously highly-ranked triples that occurred only once were demoted.

Table 2: Top 15 objects of "drink"

Without adjustments			With adjustments (c=0.95)		
	F	I		F	I
hard liquor	2	11.4	tap water	3	7.7
tap water	3	11.1	herbal tea	3	7.7
seawater	1	11.0	hard liquor	2	7.5
herbal tea	3	11.0	scotch	4	7.0
decaf	1	10.9	milkshake	2	6.8
mixed drink	1	10.6	beer	38	6.6
nectar	1	10.4	slivovitz	2	6.6
milkshake	2	10.4	malathion	2	6.6
slivovitz	2	10.4	vodka	5	6.4
malathion	2	10.3	gin	2	6.2
eggnog	1	10.3	coffee	20	6.1
chocolate milk	1	10.3	alcoholic beverage	3	6.1
malt liquor	1	9.9	champagne	7	6.1
Diet Coke	1	9.9	alcohol	18	6.0
iced tea	1	9.8	iodine	2	6.0

F: frequency; I: mutual information.

4 Evaluation

In this paper, we will show how a term bank can be used to evaluate coverage of a term extraction program. Coverage has been very difficult to measure. The classic references on term extraction, such as (Church and Hanks, 1990) and (Choueka, 1988), haven't been able to say very much about coverage, since term banks have only recently become available.

In (Alshawi and Carter, 1994), the collocations and their associated scores were evaluated indirectly by their use in parse tree selection. The merits of different measures for association strength are judged by the differences they make in the precision and the recall of the parser outputs. In (Smadja, 1993), the third stage of Xtract, in which syntactic tags are assigned to the extracted word combinations, is evaluated by a lexicographer.

In this section, we evaluated the following types of collocations:

$R = \{$subject-verb, verb-object, adjective-noun, noun-noun$\}$

with the SUSANNE corpus (Sampson, 1995), which contains parse trees of 64 of the 500 texts in the Brown Corpus of American English. The texts are evenly distributed over the following four Brown genre categories:

A   press reportage;
G   belles letters, biography, memoirs, etc.;
J   "learned" (technical and scholarly prose);
N   adventure and Western fiction.

We first converted constituency parse trees in the SUSANNE corpus into dependency trees. We then extracted dependency triples that belong to R and occurred more than once within the same category. For each such recurring triple (w_1, rel, w_2), we retrieved all the extracted collocations $(w_1, rel_1, w_2), \ldots, (w_1, rel_k, w_2)$. The triple (w_1, rel_i, w_2) is classified as **correct** if $rel_i = rel$. When $rel_i \neq rel$, we classify (w_1, rel_i, w_2) as **incorrect** if it is caused by parser errors; otherwise, we classify it as **additional**. For example, SUSANNE corpus contains two triples in which [N frame] is the prenominal modifier of [N building]. The extracted collocations include an incorrect triple in which [V frame] takes [N building] as the object. For another example, the SUSANNE corpus contains two triples in which [N court] is the prenominal modifier of [N order]. The extracted collocations include the same triple, together with an additional triple where [NP court] is the subject of [V order].

We define coverage[1] to be the percentage of the recurring dependency triples in the SUSANNE corpus that are found in the extracted colloca-

---

[1]We do not use the term "recall", because the highest possible value is not 100% due to humans' ability to generate novel sentences.

tions: coverage = correct/recurring; and precision as correct/(correct+incorrect+additional).

Table 3 shows the result for each genre in SU-SANNE corpus. The "recurring" row contains the number of distinct dependency triples that occurred more than once in SUSANNE. Since the parsed corpus contains only of the news paper articles, the coverage for genre A is much higher than G, N and especially J.

Table 3: Evaluation with SUSANNE corpus

	A	G	J	N
recurring	548	268	592	256
correct	358	147	164	139
incorrect	5	2	4	5
additional	0	1	4	0
coverage	65.3%	54.9%	27.7%	54.2%
precision	98.6%	98.6%	97.6%	96.4%

5 Application: Word Similarity

We can view each collocation that a word participates in as a feature of the word. For example, if (avert, V:compl:N, duty) is a collocation, we say that "duty" has the feature obj-of(avert) and "avert" have the feature obj(duty). Other words that also have the feature obj-of(avert) include "default", "crisis", "eye", "panic", "strike", "war", etc.

From the extracted collocations we retrieve all the features of a word. Table 4 shows a subset of the features of "duty" and "sanction". Each row corresponds to a feature. A 'x' in the "duty" or "sanction" column means that the word has that feature. The feature "subj-of(include)" is possessed by nouns which were used as subjects of "include" in the parsed corpus. The feature "adj(fiduciary)" is possessed by nouns that were modified by "fiduciary" in the parsed corpus.

Table 4: Features of "duty" and "sanction"

Feature	duty	sanction	$-\log P(f)$
f_1: subj-of(include)	x	x	3.15
f_2: obj-of(assume)	x		5.43
f_3: obj-of(avert)	x	x	5.88
f_4: obj-of(ease)		x	4.99
f_5: obj-of(impose)	x	x	4.97
f_6: adj(fiduciary)	x		7.76
f_7: adj(punitive)	x	x	7.10
f_8: adj(economic)		x	3.70

The similarity between the words can be computed according to their features. The similarity measure we adopted is based on a proposal in (Lin, 1997), where the similarity between two objects is defined to be the amount of information contained in the commonality between the objects divided by the amount of information in the descriptions of the objects. Let $F(w)$ denote the set of features possessed by w. The similarity between two words are defined as follows:

$$\text{sim}(w_1, w_2) = \frac{2 \times I(F(w_1) \cap F(w_2))}{I(F(w_1)) + I(F(w_2))}$$

where $I(S)$ is the amount of information contained in a set S of features. Assuming that features are independent of one another, $I(S) = -\sum_{f \in S} \log P(f)$.

The probability $P(f)$ of a feature f is estimated by the percentage of words that have feature f among the set of words that have the same part of speech. For example, there are 32868 distinct nouns in the parsed corpus, 1405 of which were used as the subject of "include". The probability of subj-of(include) is $\frac{1405}{32868}$. The probability of the feature adj(fiduciary) is $\frac{14}{32868}$ because only 14 (unique) nouns were modified by "fiduciary". The amount of information in the feature adj(fiduciary) is larger than the amount of information in subj-of(include). This agrees with our intuition that saying a word can be modified by "fiduciary" is more informative than saying that the word can be the subject of "include".

The column titled $-\log P(f)$ in Table 4 shows the amount of information contained in each feature. If the features in Table 4 were all the features of "duty" and "sanction", the similarity between duty and sanction would be:
$$\frac{2 \times I(\{f_1, f_3, f_5, f_7\})}{I(\{f_1, f_2, f_3, f_5, f_6, f_7\}) + I(\{f_1, f_3, f_4, f_5, f_7, f_8\})}.$$

The top-60 most similar words to "duty" identified by our program are as follows:

responsibility 0.13, position 0.10, sanction 0.10, tariff 0.09, obligation 0.09, fee 0.09, post 0.08, job 0.08, role 0.08, tax 0.08, penalty 0.08, condition 0.07, function 0.07, assignment 0.07, power 0.07, expense 0.07, task 0.07, deadline 0.07, training 0.07, work 0.07, standard 0.06, ban 0.06, restriction 0.06, authority 0.06, commitment 0.06, award 0.06, liability 0.06, requirement 0.06, staff 0.06, mem-

bership 0.06, limit 0.06, pledge 0.06, right 0.05, chore 0.05, mission 0.05, care 0.05, title 0.05, capability 0.05, patrol 0.05, fine 0.05, faith 0.05, seat 0.05, levy 0.05, violation 0.05, load 0.05, salary 0.05, attitude 0.05, bonus 0.05, schedule 0.05, instruction 0.05, rank 0.05, purpose 0.05, personnel 0.04, worth 0.04, jurisdiction 0.04, presidency 0.04, exercise 0.04

The word "duty" has three senses in the WordNet: (a) responsibility, (b) work, task, (c) tariff, all of which are included in the above list.

Two words are a pair of respective nearest neighbors (RNNs) if each is the other's most similar word. Our program found 622 pairs of RNNs among 5230 nouns that occurred at least 50 times in the parsed corpus. Table 5 shows one in every 10 RNNs. The list of RNNs looks strikingly good. Only a few are questionable. Some of the pairs may look peculiar at first glance. Detailed examination may actually reveal that they are quite reasonable. For example, the 221 ranked pair is "captive" and "westerner". It is very unlikely that any manually created thesaurus will consider them as near-synonyms. We examined all 274 occurrences of "westerner" in a 45-million-word San Jose Mercury corpus and found that 55% of them refer to westerners in captivity.

6 Conclusion and Future Work

We presented a method for extracting word collocations from text corpus using a broad coverage parser. By taking advantage of the fact that our parser produces correct parses more often than incorrect ones, we were able to automatically correct some of the parser mistakes. We also proposed a more realistic probabilistic model for calculating mutual information. The comparison with the SUSANNE corpus shows that both high precision and broad coverage can be achieved with our method. Finally, we presented an application of the extracted collocations for computing word similarities. Our results clearly showed that semantic similarity between words can be captured by the syntactic collocation patterns of the words.

In this paper, a collocation is defined to be a dependency relationship between two words that occurs significantly more frequently than by chance. One possible way to extend our work to deal with multi-word collocations is to adopt a similar strategy for dealing with N-grams in (Smadja, 1993).

Table 5: Respective Nearest Neighbors

Rank	Respective Nearest Neighbors	Similarity
1	earnings profit	0.50
11	revenue sale	0.39
21	acquisition merger	0.34
31	attorney lawyer	0.32
41	data information	0.30
51	amount number	0.27
61	downturn slump	0.26
71	there way	0.24
81	fear worry	0.23
91	jacket shirt	0.22
101	film movie	0.21
111	felony misdemeanor	0.21
121	importance significance	0.20
131	reaction response	0.19
141	heroin marijuana	0.19
151	championship tournament	0.18
161	consequence implication	0.18
171	rape robbery	0.17
181	dinner lunch	0.17
191	turmoil upheaval	0.17
201	biggest largest	0.17
211	blaze fire	0.16
221	captive westerner	0.16
231	imprisonment probation	0.16
241	apparel clothing	0.15
251	comment elaboration	0.15
261	disadvantage drawback	0.15
271	infringement negligence	0.15
281	angler fishermen	0.14
291	emission pollution	0.14
301	granite marble	0.14
311	gourmet vegetarian	0.14
321	publicist stockbroker	0.14
331	maternity outpatient	0.13
341	artillery warplanes	0.13
351	psychiatrist psychologist	0.13
361	blunder fiasco	0.13
371	door window	0.13
381	counseling therapy	0.12
391	austerity stimulus	0.12
401	ours yours	0.12
411	procurement zoning	0.12
421	neither none	0.12
431	briefcase wallet	0.11
441	audition rite	0.11
451	nylon silk	0.11
461	columnist commentator	0.11
471	avalanche raft	0.11
481	herb olive	0.11
491	distance length	0.10
501	interruption pause	0.10
511	ocean sea	0.10
521	flying watching	0.10
531	ladder spectrum	0.09
541	lotto poker	0.09
551	camping skiing	0.09
561	lip mouth	0.09
571	mounting reducing	0.09
581	pill tablet	0.08
591	choir troupe	0.08
601	conservatism nationalism	0.08
611	bone flesh	0.07
621	powder spray	0.06

We can start with 2-word collocations. The extracted collocations with reasonably high frequency are treated as "words". We then extract an extended set of triples that involve such "words". The same algorithm for 2-word collocation can be used to collect and filter the extended set of triples.

Acknowledgment

References

Hiyan Alshawi and David Carter. 1994. Training and scaling preference functions for disambiguation. *Computational Linguistics*, 20(4):635–648, December.

Y. Choueka. 1988. Looking for needles in a haystack or locating interesting collocational expressions in large textual databases. In *Proceedings of the RIAO Conference on User-Oriented Content-Based Text and Image Handling*, Cambridge, MA, March 21-24.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, March.

M. J. Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain, July.

Ralph Grishman and John Sterling. 1994. Generalizing automatically generated selectional patterns. In *Proceedings of COLING-94*, pages 742–747, Kyoto, Japan.

Zelig S. Harris. 1968. *Mathematical Structures of Language*. Wiley, New York.

Marti A. Hearst and Gregory Grefenstette. 1992. A method for refining automatically-discovered lexical relations. In Carl Weir, editor, *Statistically-Based Natural Language Programming Techniques*, number W-92-01 in Technical Report. AAAI Press.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *Proceedings of ACL-90*, pages 268–275, Pittsburg, Pennsylvania, June.

R. L. Leed and A. D. Nakhimovsky. 1979. Lexical functions and language learning. *Slavic and East European Journal*, 23(1).

Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of ACL/EACL-97*, pages 64–71, Madrid, Spain, July.

George A. Miller. 1990. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, California.

Stephen D. Richardson. 1997. *Determining Similarity and Inferring Relations in a Lexical Knowledge Base*. Ph.D. thesis, The City University of New York.

Geoffrey R. Sampson. 1995. *English for the Computer*. Oxford University Press.

Frank Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–178.