Statistical Identification of Collocations in Large Corpora for Information Retrieval

Benjamin Lambert
Department of Computer Science
University of Illinois at Urbana-Champaign

May 23, 2004

Abstract

The linguistic phenomenon of collocation, the habitual juxtaposition of some words in natural language has been shown to benefit natural language processing tasks such as information retrieval. This paper examines the utility of several methods for collocation extraction for document retrieval, specifically for queries in question form.

1 Introduction

Natural language processing applications such as information retrieval and question answering often benefit from an input richer than just a stream of text. Often additional input, such as part-of-speech tags, lists of stop words, or parse trees provide information that facilitates the task at hand. One such type of input is collocation. Informally, collocations are words that tend to be used together. Collocations have been shown to improve performance on several natural language applications. Much of the comparative analysis of collocation extraction techniques has been against a gold standard set of human-judged collocations. This paper presents a comparison of several collocation extraction techniques evaluated by their utility for a specific information retrieval task.

Extracting true linguistic collocations from corpora for linguistic and lexicographical purposes is an interesting problem. However, when extracting collocations for the purpose of information retrieval it is not necessary that the extracted collocations be true collocations. For information retrieval, more important than true collocability is the performance benefit to the information retrieval. Thus, for information retrieval the results of collocation extraction techniques do not need to be consistent with human judgments of collocation. Instead, collocation extraction techniques can be compared indirectly by how their use affects information retrieval performance.

The potential benefits of collocational input can be seen in a simple information retrieval example. Suppose a user would like to retrieve documents from some large collection of documents (e.g. the Internet), about the Twenty Years War. One way for the retrieval engine to find relevant documents is to search for documents that contain the exact string Twenty Years War. Another way is to search for documents that contain each word in the query: “Twenty”, “Years”, and “War.” For this example, the latter query will not work well because each of the terms occur so frequently in irrelevant documents. If the retrieval engine was given the information that Twenty Years War is a collocation, it might have searched using the former technique.

One solution to this problem is to inform users that they must use quotation marks for queries like Twenty Years War. Although this solution might work for this example, where the search term is likely to be consistent throughout documents, it does not work when searching for information that is mentioned in varied ways. Names of people, for example, are often not mentioned in a consistent way, sometimes with middle names and titles sometime without. Although a search using quotes may be very precise, it will miss documents that do not contain the exact string.

Another solution may be to use the notion of collocation to automatically refine searches. Given a list of collocations the information retrieval engine might search for the exact phrase if it is given a query that contains a known collocation. Further, given a list of collocations, the retrieval engine can index the documents not only by words, but by known collocations. Indexing by collocation may make searching more efficient because there is no need to search for each word and combine the results, because the the compound search terms are indexed directly.

Many methods for automatically extracting collocations from corpora have been proposed and the relative performance of each technique has been measured by whether the collocations found are thought to be collocations by humans. This paper will instead compare how information retrieval performs using lists of collocations generated using several known techniques. Comparing collocation extraction based on human judgment is inherently difficult because often the human judges do not agree themselves.

Other applications on natural language that may benefit from collocational knowledge are sentence parsers and machine translations systems. Parsers may benefit by adding a constraint that known collocations must be in the same phrase. Machine translations systems may benefit as many collocations are idiomatic and so usually cannot be translated literally.

A naive approach to identifying collocations would be to simply count pairs of words that occur very frequently. Doing this would likely determine that the most common collocation in English is *is the*. Instead, more sophisticated statistical measures must be used to identify collocations. Several of these measures are detailed in this paper. Direct comparison among the algorithms is near impossible because no one can say whether or not a given expression is truly collocational. Thus, standard methods of evaluation in computer science, such as accuracy, precision, and recall, cannot be used. It seems that really the only way to judge whether or not the algorithm is working correctly, is to forego evaluating the algorithms ability to identify collocations, and instead directly judge how well it helps with the particular application.

The specific information retrieval task evaluated in this paper is document retrieval for short queries in sentence form. This type of information retrieval is usually the first step for applications such as question answering. The queries used are for very specific information, so there are generally very few relevant documents. Finding these few documents in a large corpus is very difficult. The use of collocation for this type of information retrieval, since the task is so hard, does not appear to help very much.

2 What is collocation?

Collocation is defined in the Oxford English Dictionary as “The habitual juxtaposition or association, in the sentences of a language, of a particular word with other particular words; a group of words so associated.” Informally, collocations are groups of words that tend to be used together, either by necessity or by convention. In computer science, collocation is sometimes instead used to refer to non-compositional phrases or other related linguistic phenomena, but collocation is the most general of these. Collocation does not impose any constraints on the types of words or their meanings.

Non-compositional phrases are a similar to collocations. Non-compositional phrases are often described as phrases which are semantically different from what would be the expected combination of the constituents meanings. Non-compositional phrases are a subset of collocations because collocations need not be non-compositional. There are degrees of compositionality in phrases, for example, *white wine* (which is really yellow) and *kick the bucket* (which has nothing to do with buckets). Most phrases do have some degree of compositionality, sometimes because of analogy as in *kick the bucket*. Indeed it makes sense that most phrases should have some degree of compositionality as there were composed at some point, otherwise they would not exist in the language. Non-compositional phrases are often also characterized for their atomicity.

Multi-word expressions are another a type of collocation. Collocation subsumes multiword expressions because it does not require the words to be an expression. An expression is often a manifestation of something, whereas collocation only requires that the words be collocated. Terminology is a type of multiword expression.

For the purposes of information retrieval, it may be the case that multi-word expressions are more useful than collocations in general. This is because often one searches for something, not for some idiom. It is not clear whether expressions can be automatically differentiated from general collocations. Natural language processing applications that require a deeper analysis of the language, such as speech generation and language understanding, should model all types of collocation.

Non-compositional phrases and multi-word expressions are large classes of collocations that are often referred to in computer science literature. There are many specific types of collocations as well. Perhaps the simplest is proper nouns. To precisely identify most people, both first and last names must be specified (given name and family name), that is the two names must be collocated.

Idiomatic expressions are examples of non-compositional phrases. Idioms are often used instead of a more explicit description, so they are a more general and applicable to more situations. Since idioms may be used in many contexts, the words in the idiom tend to be collocated.

Phrasal verbs are verb-particle constructions for which the precise meaning can only be known by reading both the verb and its particle. For example, to take off (as an airplane), which without the word off would have a very different meaning. Phrasal verbs are a type of collocation for which the words need not be consecutive, for example to look (something) up.

A very subtle type of collocation, often called institutionalized expressions, is defined by the non-substitutable nature of the constituent words. An example of this is the phrase strong tea, which would unlikely be referred to as powerful tea. As the notion of collocation is very general, this type of collocation may be very common.

3 Related Research

Many attempts to automatically extract collocations from text have been made, going back to as early as 1970s with Karen Sparck Jones. The amount of research on this topic is so great, that only a few of the most recent attempts are outlined in this section. Most of the research on computational collocability could be classified as: quantifying compositionality, identification, determining semantics, and applications (such as terminology extraction of machine translation).

It is very difficult to evaluate methods of collocation identification. Bannard et. al. use a very objective approach, by having human subjects perform the same identifications [BBL]. The specific identification they have both humans and computers perform is to rank the level of compositionality in verb-particle constructions. For example, given the verb-particle construction *lift out*, they ask: was the something lifted? Is the something now out? Their results were mixed, but the fact that there was much disagreement among the human judges is very telling of the difficultly at evaluation the identification of collocations.

Similarly Baldwin et. al., attempt to automatically classify multi-word expressions into three categories: non-decomposable, idiosyncratically decomposable, and simple decomposable [BBTW]. Respectively, each has increasing compositionality. Their goal is to empirically determine under which of these three categorizations each MWE in a corpus falls. They use three semantic similarity measures to compare the each MWE to its constituent words. Each measure is based on the location of the two constituent words in the WordNet tree. They also introduce the technique of checking whether a MWE is hyperonymous of its constituent words. Identification of the MWE is taken as granted, as they use WordNet defined MWEs. Distinct boundaries for the three defined classes of MWE are not drawn.

Other research focuses instead entirely on identifying collocations. This is often done by computing statistics on counts of bigrams or other word patters in text. Some research relies on a dependency parse input. This allows the collocation identification to only occur over words that are know, up to the accuracy of the parser, to be related in some grammatical respect. This seems to increase precision, but injure recall. Dekang Lin calculate the mutual information of a dependency parse triple consisting of the two words and the relation between them [Lin99].

If the form of the collocation is known for some specific domain, such as medical terminology, collocation identification is much easier. Daille expands on this by transforming collocations once they have been found [Dai]. Terms are first identified by scanning corpus documents for phrases matching preset regular expressions. Once candidate phrase are identified, transformational rules (defined by the patterns that many French terms conform to) are applied. If any transformed terms match terms in the corpus, then the two phrases are deemed to be synonymous. Adjectives that appear in many distinct synonymous terms are more likely to be relational. Terms generated by the transformations but not present in the corpus, which contain a relational adjectives are more likely to be actual terms than those composed from non-relational adjectives. Results appear to be very good except for the term generation, which suffers from prefixes being wrongly dropped (creating terms like enzyme action which should be enzyme reaction).

Tomokiyo and Hurst also attempt to find domain specific collocations, using newsgroup post from groups with distinct topics [TH]. They define two metrics, phraseness and informativeness, to identify keyphrases. They define phraseness as the point-wise KL divergence between an N-gram model and a unigram model of the documents in one set (i.e. documents with a common topic). Informativeness is the point-wise KL-divergence of assuming that a phrase was drawn from the background model (document not in the current set) and assuming it came from the foreground model. They combine the two metrics into one metric by computing the KL divergence of assuming an n-gram model drawn from the foreground model with the assumption of a unigram, background model. As does much other research, this research suffers from the inability to quantify performance. Subjective evaluation was impressive though.

Whereas much research attempts to fine-tune, or apply collocation techniques to specific domains the research is constrained by both the domain and the language. Dias attempts to forgo all these assumptions and find arbitrary collocations [Dia]. Instead, candidate collocations are "positional n-grams" which are sets of n words each in conjunction with its POS and distance in words from the first word of the n-gram (an example positional 2-gram: (word1, NN, 0), (word3, NN, +2)). Dias calculates various cohesiveness measures. To account for the fact that any subset of a collocation is itself a collocation, the algorithm dynamically changes the size of candidate collocations.

There is a great amount of research on better ways to find collocation (using a semantic tagger for example [PRA$^+$]) or ways collocations can be applied to other tasks (such as lexicography [Vil]). The research described in this paper is among the research that attempts to use collocation for a specific task.

4 Extracting Collocations from Corpora

Model

The model for statistical identification of collocations is very simple. The corpus D is a set of n documents d_i. Each document d_i consists of a set of sentences each denoted s_i^j. Each sentence is a list of words, where word k is denoted as $s_i^j[k]$. Each distinct token in the corpus documents is a word w_i. Queries on the corpus are defined to be sentences which may or may not be contained in any document in the corpus (i.e. they are lists of words).

The corpus is modeled as a set of documents, so the order of the documents does not matter. Each documents is a set of sentences so the order of the sentences also does not matter in this model. The presence of a collocation may depend on the context within the document or the context within the corpus (e.g. temporal dependence), so this is a simplifying assumption. The order of the words in each sentence does, of course, matter. Each token in the sentences is mapped to one and only one word w, so in this model differently spelled words are different words (i.e. chair != chairs).

For this model it is also assumed that there is a known function $pos(s_I^j[k])$, that maps each token of each sentence to its part of speech in that sentence. For example, given sentence s_j^i = "The man spoke." $pos(s_j^i[2]) =$ singular noun. A list of the parts of speech in the range of this function is available from the Cognitive Computation Group web site (http://l2r.cs.uiuc.edu/ cogcomp/). It is assumed that this function is known because the accuracy of state of the art automatic part of speech taggers is very high [**?**].

The broadest definition of collocation in this model is simply a set C of n words w_i, that are used *together* in the sentences of the corpus D more frequently than would be expected. This broad definition does not impose any restrictions on the order or position of the words. To simplify identification of collocations, background knowledge of the language can be used. In collocation extraction literature, the three most used language-specific constraints on the words in a collocation are on the number of words, the order of the words, and the parts of speech of the words.

Specific constraints can be formalized for the types of collocations that have a consistent syntactic structure. For example, in English, proper names consist of consecutive words. In English, the verb component of a phrasal verbs come before the particle. Idioms are not necessarily syntactically consistent. Other constraints that are not necessarily justified linguistically can be placed on the form of a collocation to increase the performance of automatic collocation extraction. Empirical evidence justifies constraining collocations to occur within a window size of five [Sma94].

In English, many noun-noun and adjective-noun compounds form collocations. Initial experiments described in this paper only account for noun-noun and adjective-noun compounds. Formally these are pairs of words $s_i^j[k]$ and $s_i^j[k+1]$ where either $pos(s_i^j[k]) = adjective$ or $pos(s_i^j[k]) = noun$ and $pos(s_i^j[k+1]) = noun$.

Other constraints that can be used include limiting words to not contain numeric symbols. Although one would not want to remove stop words if looking for idiomatic expressions or phrasal verbs, stop words could be removed for finding other types of collocations.

Statistical Identification

The definition of collocation is rather subjective, so an algorithm must instead approximate human judgment. Contemporary algorithms use statistics of the word occurrences to identify collocations. Statistically, one would expect that the words of a collocation would appear in the same sentence with probability greater than the probability that the words randomly appear in the same sentence. That is the events of each words occurring in a sentence are statistically dependent. Specifically, $P(w_1, w_2 \in s) > P(w_1 \in s) \cdot P(w_2 \in s)$. A statistical model such as this assumes that words not in collocation are statistically independent. This is profoundly not true, as the words in any sentence are intricately related.

Most methods for extracting collocations from corpora count all occurrences of collocations subject to some constraints, then compute some statistics on the frequencies of candidate collocations and frequencies of their constituent words. Candidate collocations are then deemed to be collocations if some statistic crosses some threshold. For example, using the assumption that collocations are only two words, the algorithm might classify words w_a, w_b as a collocation if the variance of the random variable $D = (x - y)$ where $s_i^j[x] = w_a$ and $s_i^j[y] = w_b$ is less than some threshold theta.

It is not obvious what value to use for the threshold. Especially since many people would probably dispute collocations on the borderline between collocational and non-collocational. Some of the studies which attempt to identify true collocation, have studied the performance of varying thresholds. Because the experiments described in this paper are not to the ends of discovering true collocation, but rather for application to information retrieval the thresholds are chosen empirically without scientific justification.

Variance

One statistic that is consistent with the notion of collocations that have words always in the same order, is the variance of their distances in sentences. Given two words, they are more likely to be collocations if most of the time when they appear in the same sentence they are the same distance from each other. The example given in the introduction was a search for documents about J.S. Bach. One expects that in almost all sentences that contain the strings *J.S.* and *Bach*, they will be adjacent and in the same order. Thus one expects the variance among the distances between the two strings to be very low. Manning and Shutze attribute the use of variance for collocation discovery to Frank Smadja.

For a given pair of words w_a, w_b, the variance is computed to be:

$$\frac{\sum (x-y)^2}{n-1} \text{where} s_i^j[x] = w_a \text{and} s_i^j[y] = w_b$$

and n is the number of occurrences of w_a, w_b in the same sentence. Generally there would be another constraint on the distance within the sentence, such as $|x - y| \leq 5$. For the experiments described in this paper, the constraint used was $|x - y| \leq 1$.

Point-wise mutual information

Point-wise mutual information is the information theoretic amount of information that two events provide. If two uncommon events occur frequently together, then their mutual occurrence has a high information content. The information theoretic definition of mutual information is between distributions. Point-wise mutual information only computes the mutual information between two events.

The events that the point-wise mutual information are computed for is the event that word w_i is present in an arbitrary sentence. The desired statistic is the point-wise mutual information between two words in a candidate collocation. The point-wise mutual information between the words w_i and w_j in some candidate collocation is defined to be:

$$MI(w_i, w_j) = \frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)}$$

T-Test

Another way to test if two words are statistically collocated is to build a model of the corpus, where the words are distributed stochastically by their frequencies in the corpus over a normal distribution. This is called the null hypothesis. Then the sampled probability of each candidate collocation can be compared to the null hypothesis. If the candidate collocation is significantly more probable than the estimation by the null hypothesis, then it is deemed a collocation.

The score of the t-test is computed as:

$$\frac{P(w_1, w_2) - P(w_1) \cdot P(w_2)}{\sqrt{\frac{P(w_1,w_2)}{N}}}$$

where N is the number of words in the corpus. This score can be interpreted as the difference between the probabilities under the real distribution and probability under the null hypothesis divided by a factor to normalize the variance. The denominator should really be $\sqrt{\frac{s^2}{N}}$, but since all the probabilities considered here are small, $P(w_1, w_2)$ is a good approximation of s^2. [MS99]

Chi-Squared Test

The χ^2 test is similar to the t-test in that it compares the observed data to the null hypothesis. The χ^2 test instead assumes that the words in the null hypothesis are distributed over a χ^2 distribution, which may be more characteristic of natural written language [MS99].

The statistic computed here is:

$$\sum_{i,j} \frac{(E_{i,j} - O_{i,j})^2}{E_{i,j}}$$

where the i, j ranges over the rows and columns of the co-occurence table. For the case where the table dimensions are 2 by 2, the equation simplifies to:

$$\frac{N(O_{11}O_{22} - O_{12}O_{21})^2}{(O_{11} + O_{12})(O_{11} + O_{21})(O_{12} + O_{22})(O_{21} + O_{22})}$$

This equation was taken from Manning and Shutze [MS99].

Threshold

It is easier to choose how many of the candidate collocations to use when using a statistical significance test, because the computed values correspond to the probability that the event was sample from the same distribution. The actual probability can then be looked up in a table. A threshold can be set which corresponds to some low probability that the sample was sampled from the distribution. For example, if collocations are desired to have probability less than 5% that they were sampled from the null hypothesis distribution, the corresponding χ^2 score can be found to be X, then only collocations with χ^2 score greater than X are determined to be collocations.

For measures such as variance and mutual information there is no such direct method of choosing which collocations to use. These two methods rank collocations by their variance and mutual information respectively, but they do not make any decision about how many of the top ranked collocations to use. Several values were tried for the experiments described in this paper, and show the choice is not very meaningful for this task.

5 Effect on Information Retrieval

The use of collocations has been found to increase the performance of information retrieval. Specifically, they are found to increase the ability of search algorithms to find relevant documents given a query. The set of documents used in this research is the AQUAINT corpus from the TREC 2002 conference. The corpus consists of over one million articles from the NY Times news service, AP wire news briefs, and XIE. Queries over the documents are short sequences of words in sentence format. An example of one of the queries is *Where does the vice president live when in office?* Because the queries are very specific, very few of the documents in the corpus are relevant to each query, from as few as a single document and usually fewer than ten of the one million documents. The performance of the information retrieval engine is measured by conducting 408 queries.

To find relevant documents the LEMUR toolkit for language modeling and information retrieval is used. LEMUR indexes a corpus, then given a query, finds documents containing the words in the query. The documents are then ranked by statistic such as TFIDF or Okapi. The experiments described in this paper use simple Okapi (BM25) to rank the relevance of documents. The retrieval engine returns one thousand documents for each query. Each graph plots the average (over the 408 queries) interpolated precision versus recall.

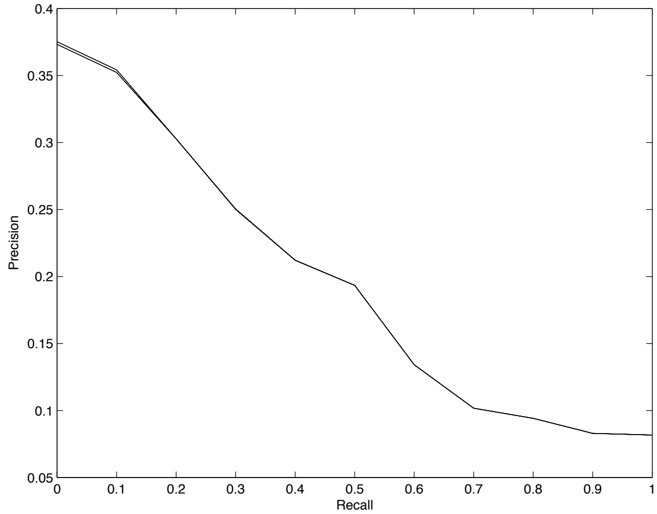
Each collocation extraction technique was first used to identify a set of collocations in the corpus. For each collocation technique, the original corpus is modified to reflect the known collocations. In some cases this means replacing the collocated words with a single token identifying the collocation as a unit to be treated as though it were a single word. The same process is performed on the queries. Once the corpus is modified it is indexed and the queries are evaluated.

One issue is that when collocated terms are replaced by a single token some information is lost. This is because the search algorithm is not cognizant of the components of a collocation and the new collocations appear to be words entirely unrelated to the original words. To mitigate the loss of information, known collocations can be used in addition to the original words rather than replace them. For each of the collocation extraction techniques, both strategies were tested. In general, retaining the original terms was advantageous, but not by very much. Because it is not very difficult to include those search terms in the indexing and query, one should generally include them.

Variance

As we do not have a principled approach to deciding a threshold on the number of collocations to use, when using the variance statistic, the first experiment was to find an appropriate value. Initially the value of 10,000, 50,000, 100,000, and 500,000 were used. For all four thresholds, the performance was nearly identical. This appears to be because collocations that appear in the queries either have very high variance or very low variance. For each of the thresholds, the collocations used are nearly identical. Going from 10,000 to 50,000 increased the number of collocations found in the queries by two. Many more collocations were found in the corpus itself, but they did not help because they are not in the queries.

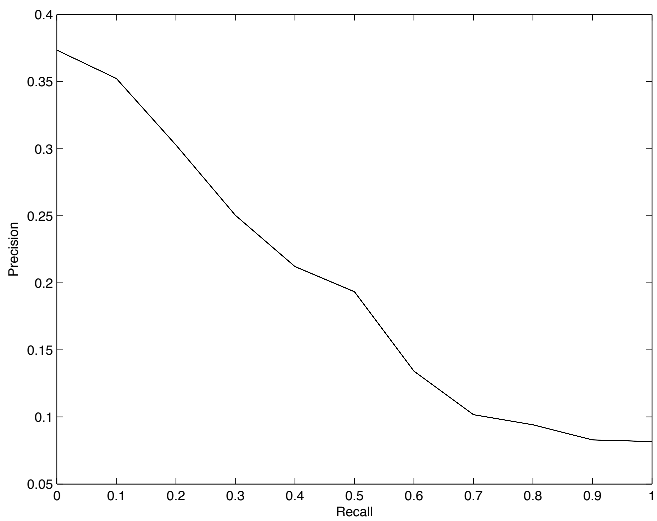
There are two curves plotted on the graph, one for collocation replacement the other for collocation addition. Collocation addition did slightly better, because there was no information loss, but the gains are incredibly small. This may indicate that addition is better than replacement, but only tenuously.



Point-wise mutual information

As for variance, the initial numbers of collocations used were 10,000, 50,000, 100,000, and 500,000. Again, the performance in all four situations was nearly identical. Again, this seems to be due to the fact that for each case, nearly all of the collocations found in the queries were in the top 10,000. Like variance, this identification and use of collocations helped only marginally.

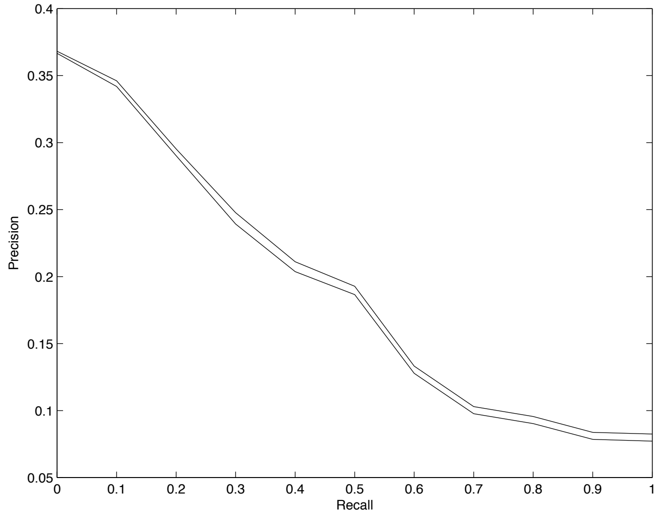
There are in fact two curves plotted on this graph, one for collocation replacement the other for collocation addition. Though the numbers were not identical, they were so close that difference is not perceivable at this scale.



T-Test

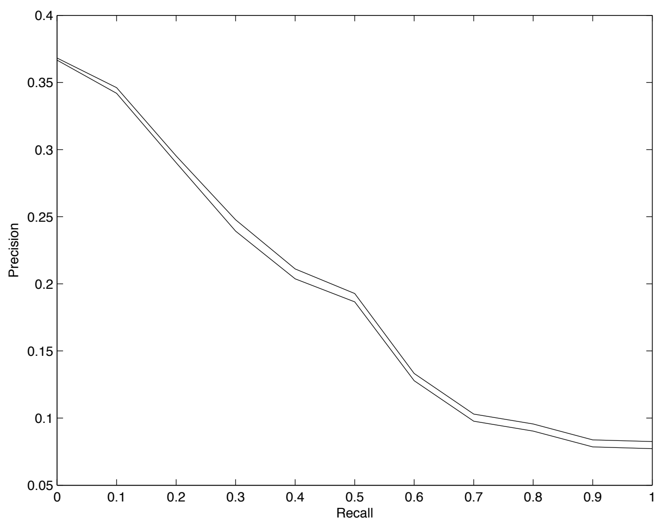
Though the t-test offers a better justified threshold selection, results were not any better. The threshold selection for the t-test is to first choose a probability threshold, where it is required that collocations not belong to the null hypothesis distribution with some probability. The t-test score for that probability can then be looked up in a table. Several probabilities were tried, but all yielded nearly identical results. The graph shown is for probability 0.005 which corresponds to t-test score 2.576.

The difference between collocation replacement and collocation addition is slightly more pronounced in this graph, but again is on a very small scale.



Chi-Squared Test

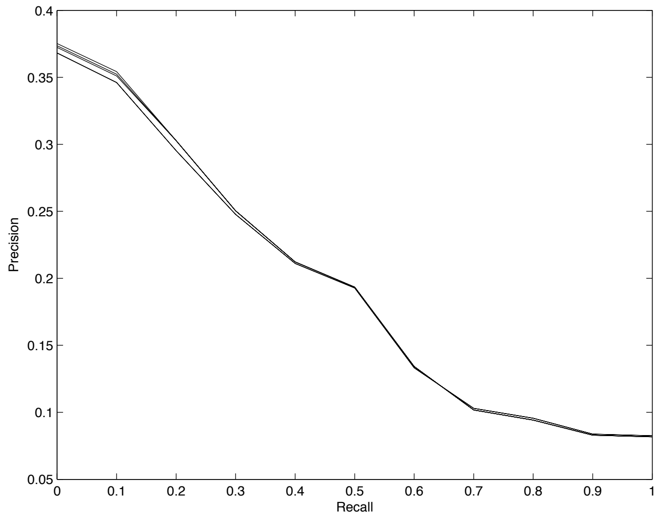
Perhaps not surprisingly, the χ^2 test did not fare much better than the other tests. As for the t-test the difference between collocation replacement and substitution is slightly more visible, but not particularly significant.



Comparison Among Methods

The following graph plots the performance for each of the collocation identification techniques with the best parameters for that technique and the performance of the task without any use of collocation. This graph best illustrates how limited the performance gains are. Gains in precision are almost entirely limited to when recall is below 0.4. And any gains in precision here are at most one percent. Overall average precision often only increased as little as 0.5%.

Perhaps it is the comparison among the number of collocations to use for variance and mutual information metrics that is most telling about the performance for this task. Because very large increases in the number of collocations available has a minimal affect on the queries, it may be that there are just too few collocations available in the queries to help. This may mean that this particular task is not one that cannot easily benefit from the knowledge of collocations. The poverty of query text may be the prohibitive factor. Likewise, the fact that so few documents are relevant to each query makes this a difficult endeavor to begin with.



6 Conclusions

The experiments described in this paper show that using automatic extraction of collocations does not aid very much document selection for tasks similar to question answering. Using collocations as indexable tokens, although useful for some natural language processing tasks is not particularly beneficial to this task. Within the framework of question answering, collocations may be very useful for passage or answer selection.

References

[BBL] Colin Bannard, Timothy Baldwin, and Alex Lascarides. A statistical approach to the semantics of verb-particles. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[BBTW] Timothy Baldwin, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. An empirical model of multiword expression decomposability. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[Dai] Béatrice Daille. Conceptual structuring through term variations. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[Dia] Gael Dias. Multiword unit hybrid extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[Lin99] Dekang Lin. Automatic identification of non-compositional phrases. In *In Proceedings of ACL-99*, 1999.

[MS99] Christopher Manning and Hinrich Shutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[PRA+] Scott S. L. Piao, Paul Rayson, Dawn Archer, Andrew Wilson, and Tony McEnery. Extracting multiword expressions with a semantic tagger. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[Sma94] Frank Z. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1), 1994.

[TH] Takashi Tomokiyo and Matthew Hurst. Language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.

[Vil] Aline Villavicencio. Verb-particle constructions and lexical resources. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions*.