Acquiring Collocations for Lexical Choice between Near-Synonyms

Diana Zaiu Inkpen and **Graeme Hirst**
Department of Computer Science
University of Toronto
{dianaz,gh}@cs.toronto.edu

Abstract

We extend a lexical knowledge-base of near-synonym differences with knowledge about their collocational behaviour. This type of knowledge is useful in the process of lexical choice between near-synonyms. We acquire collocations for the near-synonyms of interest from a corpus (only collocations with the appropriate sense and part-of-speech). For each word that collocates with a near-synonym we use a differential test to learn whether the word forms a less-preferred collocation or an anti-collocation with other near-synonyms in the same cluster. For this task we use a much larger corpus (the Web). We also look at associations (longer-distance co-occurrences) as a possible source of learning more about nuances that the near-synonyms may carry.

1 Introduction

Edmonds and Hirst (2002 to appear) developed a lexical choice process for natural language generation (NLG) or machine translation (MT) that can decide which near-synonyms are most appropriate in a particular situation. The lexical choice process has to choose between clusters of near-synonyms (to convey the basic meaning), and then to choose between the near-synonyms in each cluster. To group near-synonyms in clusters we trust lexicographers' judgment in dictionaries of synonym differences. For example *task, job, duty, assignment, chore, stint, hitch* all refer to a one-time piece of work, but which one to choose depends on the duration of the work, the commitment and the effort involved, etc.

In order to convey desired nuances of meaning and to avoid unwanted implications, knowledge about the differences among near-synonyms is necessary. I-Saurus, a prototype implementation of (Edmonds and Hirst, 2002 to appear), uses a small number of hand-built clusters of near-synonyms.

Our goal is to automatically acquire knowledge about distinctions among near-synonyms from a dictionary of synonym differences and from other sources such as free text, in order to build a new lexical resource, which can be used in lexical choice. Preliminary results on automatically acquiring a lexical knowledge-base of near-synonym differences were presented in (Inkpen and Hirst, 2001). We acquired denotational (implications, suggestions, denotations), attitudinal (favorable, neutral, or pejorative), and stylistic distinctions from *Choose the Right Word* (Hayakawa, 1994) (hereafter CTRW)[1]. We used an unsupervised decision-list algorithm to learn all the words used to express distinctions and then applied information extraction techniques.

Another type of knowledge that can help in the process of choosing between near-synonyms is collocational behaviour, because one must not choose a near-synonym that does not collocate well with the other word choices for the sentence. I-Saurus does not include such knowledge. The focus of the work we present in this paper is to add knowledge about collocational behaviour to our lexical knowledge-base of near-synonym differences. The lexical choice process implemented in I-Saurus gen-

[1]We are grateful to HarperCollins Publishers, Inc. for permission to use CTRW in this project.

erates all the possible sentences with a given meaning, and ranks them according to the degree to which they satisfy a set of preferences given as input (these are the denotational, attitudinal, and stylistic nuances mentioned above). We can refine the ranking so that it favors good collocations, and penalizes sentences containing words that do not collocate well.

We acquire collocates of all near-synonyms in CTRW from free text. We combine several statistical measures, unlike other researchers who rely on only one measure to rank collocations.

Then we acquire knowledge about less-preferred collocations and anti-collocations[2]. For example *daunting task* is a preferred collocation, while *daunting job* is less preferred (it should not be used in lexical choice unless there is no better alternative), and *daunting duty* is an anti-collocation (it must not be used in lexical choice). Like Church et al.(1991), we use the t-test and mutual information. Unlike them we use the Web as a corpus for this task, we distinguish three different types of collocations, and we apply sense disambiguation to collocations.

Collocations are defined in different ways by different researchers. For us collocations consist of consecutive words that appear together much more often than by chance. We also include words separated by a few non-content words (short-distance co-occurrence in the same sentence).

We are interested in collocations to be used in lexical choice. Therefore we need to extract **lexical collocations** (between open-class words), not **grammatical collocations** (which could contain closed-class words, for example *put on*). For now, we consider only two-word fixed collocations. In future work we will consider longer and more flexible collocations.

We are also interested in acquiring words that strongly associate with our near-synonyms, especially words that associate with only one of the near-synonyms in the cluster. Using these strong associations, we plan to learn about nuances of near-synonyms in order to validate and extend our lexical knowledge-base of near-synonym differences.

In our first experiment, described in sections 2 and 3 (with results in section 4, and evaluation in section 5), we acquire knowledge about the collocational behaviour of the near-synonyms. In step 1 (section 2), we acquire potential collocations from the British National Corpus (BNC)[3], combining several measures. In section 3 we present: (step2) select collocations for the near-synonyms in CTRW; (step 3) filter out wrongly selected collocations using mutual information on the Web; (step 4) for each cluster we compose new collocations by combining the collocate of one near-synonym with the the other near-synonym, and we apply the differential t-test to classify them into preferred collocations, less-preferred collocations, and anti-collocations. Section 6 sketches our second experiment, involving word associations. The last two sections present related work, and conclusions and future work.

2 Extracting collocations from free text

For the first experiment we acquired collocations for near-synonyms from a corpus. We experimented with 100 million words from the Wall Street Journal (WSJ). Some of our near-synonyms appear very few times (10.64% appear fewer than 5 times) and 6.87% of them do not appear at all in WSJ (due to its business domain). Therefore we need a more general corpus. We used the 100 million word BNC. Only 2.61% of our near-synonyms do not occur; and only 2.63% occur between 1 and 5 times.

Many of the near-synonyms appear in more than one cluster, with different parts-of-speech. We experimented on extracting collocations from raw text, but we decided to use a part-of-speech tagged corpus because we need to extract only collocations relevant for each cluster of near-synonyms. The BNC is a good choice of corpus for us because it has been tagged (automatically by the CLAWS tagger).

We preprocessed the BNC by removing all words tagged as closed-class. To reduce computation time, we also removed words that are not useful for our purposes, such as proper names (tagged NP0). If we keep the proper names, they are likely to be among the highest-ranked collocations.

There are many statistical methods that can be used to identify collocations. Four general methods are presented by Manning and Schütze (1999). The first one, based on frequency of co-occurrence, does not consider the length of the corpus. Part-of-

[2]This term was introduced by Pearce (2001).

[3]http://www.hcu.ox.ac.uk/BNC/

speech filtering is needed to obtain useful collocations. The second method considers the means and variance of the distance between two words, and can compute flexible collocations (Smadja, 1993). The third method is hypothesis testing, which uses statistical tests to decide if the words occur together with probability higher than chance (it tests whether we can reject the null hypothesis that the two words occurred together by chance). The fourth method is (pointwise) mutual information, an information-theoretical measure.

We use Ted Pedersen's Bigram Statistics Package[4]. BSP is a suite of programs to aid in analyzing bigrams in a corpus (newer versions allow N-grams). The package can compute bigram frequencies and various statistics to measure the degree of association between two words: mutual information (MI), Dice, chi-square (χ^2), log-likelihood (LL), and Fisher's exact test.

The BSP tools count for each bigram in a corpus how many times it occurs, and how many times the first word occurs.

We briefly describe the methods we use in our experiments, for the two-word case. Each bigram *xy* can be viewed as having two features represented by the binary variables X and Y. The joint frequency distribution of X and Y is described in a contingency table. Table 1 shows an example for the bigram *daunting task*. n_{11} is the number of times the bigram *xy* occurs; n_{12} is the number of times x occurs in bigrams at the left of words other than y; n_{21} is the number of times y occurs in bigrams after words other that x; and n_{22} is the number of bigrams containing neither x nor y. In Table 1 the variable X denotes the presence or absence of *daunting* in the first position of a bigram, and Y denotes the presence or absence of *task* in the second position of a bigram. The marginal distributions of X and Y are the row and column totals obtained by summing the joint frequencies: $n_{+1} = n_{11} + n_{21}$, $n_{1+} = n_{11} + n_{12}$, and n_{++} is the total number of bigrams.

The BSP tool counts for each bigram in a corpus how many times it occurs, how many times the first word occurs at the left of any bigram (n_{+1}), and how many times the second words occurs at the right of any bigram (n_{1+}).

[4]http://www.d.umn.edu/∼tpederse/code.html

	y	$\neg y$	
x	$n_{11} = 66$	$n_{12} = 54$	$n_{1+} = 120$
$\neg x$	$n_{21} = 4628$	$n_{22} = 15808937$	$n_{2+} = 15813565$
	$n_{+1} = 4694$	$n_{+2} = 15808991$	$n_{++} = 15813685$

Table 1: Contingency table for *daunting task* (x = *daunting*, y = *task*).

Mutual information, $I(x;y)$, compares the probability of observing words x and word y together (the joint probability) with the probabilities of observing x and y independently (the probability of occurring together by chance) (Church and Hanks, 1991).

$$I(x;y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

The probabilities can be approximated by: $P(x) = n_{+1}/n_{++}$, $P(y) = n_{1+}/n_{++}$, $P(x,y) = n_{11}/n_{++}$. Therefore:

$$I(x;y) = \log_2 \frac{n_{++}n_{11}}{n_{+1}n_{1+}}$$

The **Dice** coefficient is related to mutual information and it is calculated as:

$$Dice(x,y) = \frac{2P(x,y)}{P(x)+P(y)} = \frac{2n_{11}}{n_{+1}+n_{1+}}$$

The next methods fall under hypothesis testing methods. **Pearson's Chi-square** and **Log-likelihood ratios** measure the divergence of observed (n_{ij}) and expected (m_{ij}) sample counts ($i = 1,2,\ j = 1,2$). The expected values are for the model that assumes independence (assumes that the null hypothesis is true). For each cell in the contingency table, the expected counts are: $m_{ij} = \frac{n_{i+}n_{+j}}{n_{++}}$. The measures are calculated as (Pedersen, 1996):

$$\chi^2 = \Sigma_{i,j} \frac{(n_{ij} - m_{ij})^2}{m_{ij}}$$

$$LL = 2\, \Sigma_{i,j} \frac{\log_2 n_{ij}^2}{m_{ij}}$$

Log-likelihood ratios (Dunning, 1993) are more appropriate for sparse data than chi-square.

Fisher's exact test is a significance test that is considered to be more appropriate for sparse and skewed samples of data than statistics such as the log-likelihood ratio or Pearson's Chi-Square test

(Pedersen, 1996). Fisher's exact test is computed by fixing the marginal totals of a contingency table and then determining the probability of each of the possible tables that could result in those marginal totals. Therefore it is computationally expensive. The formula is:

$$P = \frac{n_{1+}!n_{2+}!n_{+1}!n_{+2}!}{n_{++}!n_{11}!n_{12}!n_{21}!n_{22}!}$$

Because these five measures rank collocations in different ways (as the results in the Appendix will show), and have different advantages and drawbacks, we decided to combine them in choosing collocations. We choose as potential collocations for each near-synonym a collocation that is selected by at least two of the measures. For each measure we need to choose a threshold T, and consider as selected collocations only the T highest-ranked bigrams (where T can differ for each measure). By choosing higher thresholds we increase the precision (reduce the chance of accepting wrong collocations). By choosing lower thresholds we get better recall. If we opt for low recall we may not get many collocations for some of the near-synonyms. Because there is no principled way of choosing these thresholds, we prefer to choose lower thresholds (the first 200,000 collocations selected by each measure, except Fisher's measure for which we take all 435,000 collocations ranked 1) and to filter out later (in step 2) the bigrams that are not true collocations, using mutual information on the Web.

3 Differential collocations

For each cluster of near-synonyms, we now have the words that occur in preferred collocations with each near-synonym. We need to check whether these words collocate with the other near-synonyms in the same cluster. For example, if *daunting task* is a preferred collocation, we check whether *daunting* collocates with the other near-synonyms of *task*.

We use the Web as a corpus for differential collocations. We don't use the BNC corpus to rank less-preferred and anti-collocations, because their absence in BNC may be due to chance. We can assume that the Web (the portion retrieved by search engines) is big enough that a negative result can be trusted.

We use an interface to AltaVista search engine to count how often a collocation is found. (See Table 2 for an example.[5]) A low number of co-occurrences indicates a less-preferred collocation. But we also need to consider how frequent the two words in the collocation are. We use the differential t-test to find collocations that best distinguish between two near-synonyms (Church et al., 1991), but we use the Web as a corpus. Here we don't have part-of-speech tags but this is not a problem because in the previous step we selected collocations with the right part-of-speech for the near-synonym. We approximate the number of occurrences of a word on the Web with the number of documents containing the word.

The ***t*-test** can also be used in the hypothesis testing method to rank collocations. It looks at the mean and variance of a sample of measurements, where the null hypothesis is that the sample was drawn from a normal distribution with mean μ. It measures the difference between observed ($\bar{x}$) and expected means, scaled by the variance of the data (s^2), which in turn is scaled by the sample size (N).

$$t = \frac{\bar{x} - \mu}{\sqrt{\frac{s^2}{N}}}$$

We are interested in the **Differential *t*-test**, which can be used for hypothesis testing of differences. It compares the means of two normal populations:

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{s_1^2}{N} + \frac{s_2^2}{N}}}$$

Here the null hypothesis is that the average difference is $\mu = 0$.Therefore $\bar{x} - \mu = \mu = \bar{x_1} - \bar{x_2}$. In the denominator we add the variances of the two populations.

If the collocations of interest are xw and yw (or similarly wx and wy), then we have the approximations $\bar{x_1} = s_1^2 = P(x,w)$ and $\bar{x_2} = s_2^2 = P(y,w)$; therefore:

$$t = \frac{P(x,w) - P(y,w)}{\sqrt{\frac{P(x,w)+P(y,w)}{n_{++}}}} = \frac{n_{xw} - n_{yw}}{\sqrt{n_{xw} + n_{yw}}}$$

If w is a word that collocates with one of the near-synonyms in a cluster, and x is each of the near-

[5]The search was done on 13 March 2002.

synonyms, we can approximate the mutual information relative to *w*:

$$\frac{P(w,x)}{P(x)} = \frac{n_{wx}}{n_x}$$

where $P(w)$ was dropped because it is the same for various *x* (we cannot compute if we keep it, because we don't know the total number of bigrams on the Web).

We use this measure to eliminate collocations wrongly selected in step 1. We eliminate those with mutual information lower that a threshold. We describe the way we chose this threshold (T_{mi}) in section 5.

We are careful not to consider collocations of a near-synonym with a wrong part-of-speech (our collocations are tagged). But there is also the case when a near-synonym has more than one major sense. In this case we are likely to retrieve collocations for senses other than the one required in the cluster. For example, for the cluster *job*, *task*, *duty*, etc., the collocation *import/N duty/N* is likely to be for a different sense of *duty* (the customs sense). Our way of dealing with this is to disambiguate the sense used in each collocations (we assume one sense per collocation), by using a simple Lesk-style method (Lesk, 1986). For each collocation, we retrieve instances in the corpus, and collect the content words surrounding the collocations. This set of words is then intersected with the context of the near-synonym in CTRW (that is the whole entry). If the intersection is not empty, it is likely that the collocation and the entry use the near-synonym in the same sense. If the intersection is empty, we don't keep the collocation.

In step 3, we group the collocations of each near-synonym with a given collocate in three classes, based on the *t*-test values of pairwise collocations. We compute the *t*-test between each collocation and the collocation with maximum frequency, and the *t*-test between each collocation and the collocation with minimum frequency (see Table 2 for an example). Then, we need to determine a set of thresholds that classify the collocations in the three groups: preferred collocations, less preferred collocations, and anti-collocations. The procedure we use in this step is detailed in section 5.

x	**Hits**	**MI**	***t* max**	***t* min**
task	63573	0.011662	-	252.07
job	485	0.000022	249.19	22.02
assignment	297	0.000120	250.30	17.23
chore	96	0.151899	251.50	9.80
duty	23	0.000022	251.93	4.80
stint	0	0	252.07	-
hitch	0	0	252.07	-

Table 2: The second column shows the number of hits for the collocation *daunting x*, where *x* is one of the near-synonyms in the first column. The third column shows the mutual information, the fourth column, the differential *t*-test between the collocation with maximum frequency (*daunting task*) and *daunting x*, and the last column, the *t*-test between *daunting x* and the collocation with minimum frequency (*daunting hitch*).

4 Results

We obtained 15,813,685 bigrams. From these, 1,350,398 were distinct and occurred at least 4 times.

We present some of the top-ranked collocations for each measure in the Appendix. We present the rank given by each measure (1 is the highest), the value of the measure, the frequency of the collocation, and the frequencies of the words in the collocation.

We selected collocations for all 914 clusters in CTRW (5419 near-synonyms in total). An example of collocations extracted for the near-synonym *task* is:

```
daunting/A task/N
 -- MI   24887      10.8556
 -- LL    5998     907.96
 -- X2   16341  122196.8257
 -- Dice  2766       0.0274
repetitive/A task/N
 -- MI   64110       6.7756
 -- X2  330563     430.4004
```

where the numbers are, in order, the rank given by the measure and the value of the measure.

We filtered out the collocations using MI on the Web (step 2), and then we applied the differential *t*-test (step 3). Table 2 shows the values of MI between *daunting x* and *x*, where *x* is one of the near-synonyms of *task*. It also shows *t*-test values between (some) pairs of collocations. Table 3

Near-synonyms	*daunting*	*particular*	*tough*
task	$\surd$	$\surd$	$\surd$
job	?	$\surd$	$\surd$
assignment	$*$	$\surd$	$\surd$
chore	$*$	?	$*$
duty	$*$	$\surd$	$*$
stint	$*$	$*$	$*$
hitch	$*$	$*$	$*$

Table 3: Example of results for collocations.

presents an example of results for differential collocations, where $\surd$ marks preferred collocations, ? marks less-preferred collocations, and $*$ marks anti-collocations.

Before proceeding with step 3, we filtered out the collocations in which the near-synonym is used in a different sense, using the Lesk method explained above. For example, *suspended/V duty/N* is kept while *customs/N duty/N* and *import/N duty/N* are rejected. The disambiguation part of our system was run only for a subset of CTRW, because we have yet to evaluate it. The other parts of our system were run for the whole CTRW. Their evaluation is described in the next section.

5 Evaluation

Our evaluation has two purposes: to get a quantitative measure of the quality of our results, and to choose thresholds in a principled way.

As described in the previous sections, in step 1 we selected potential collocations from BNC (the ones selected by at least two of the five measures). Then, we selected collocations for each of the near-synonyms in CTRW (step 2). We need to evaluate the MI filter (step 3), which filters out the bigrams that are not true collocations, based on their mutual information computed on the Web. We also need to evaluate step 4, the three way classification based on the differential t-test on the Web.

For evaluation purposes we selected three clusters from CTRW, with a total of 24 near-synonyms. For these, we obtained 916 collocations from BNC according to the method described in section 2.

We had two human judges reviewing these collocations to determine which of them are true collocations and which are not. We presented the collocations to the judges in random order, and each collocation was presented twice. The first judge was consistent (judged a collocation in the same way both times it appeared) in 90.4% of the cases. The second judge was consistent in 88% of the cases. The agreement between the two judges was 67.5% (computed in a strict way, that is we considered agreement only when the two judges had the same opinion including the cases when they were not consistent). The consistency and agreement figures show how difficult the task is for humans.

We used the data annotated by the two judges to build a standard solution, so we can evaluate the results of our MI filter. In the standard solution a bigram was considered a true collocation if both judges considered it so. We used the standard solution to evaluate the results of the filtering, for various values of the threshold T_{mi}. That is, if a bigram had the value of MI on the Web lower than a threshold T_{mi}, it was filtered out. We choose the value of T_{mi} so that the accuracy of our filtering program is the highest. By accuracy we mean the number of true collocations (as given by the standard solution) identified by our program over the total number of bigrams we used in the evaluation. The best accuracy was 70.7% for $T_{mi} = 0.0017$. We used this value of the threshold when running our programs for all CTRW.

As a result of this first part of the evaluation, we can say that after filtering collocations based on MI on the Web, approximately 70.7% of the remaining bigrams are true collocation. This value is not absolute, because we used a sample of the data for the evaluation. The 70.7% accuracy is much better than a baseline (approximately 50% for random choice). Table 4 summarizes our evaluation results.

Next, we proceeded with evaluating the differential t-test three-way classifier. For each cluster, for each collocation, new collocations were formed from the collocate and all the near-synonyms in the cluster. In order to learn the classifier, and to evaluate its results, we had the two judges manually classify a sample data into preferred collocations, less-preferred collocations, and anti-collocations. We used 2838 collocations obtained for the same three clusters from 401 collocations (out of the initial 916) that remained after filtering. We built a standard solution for this task, based on the classifications of both judges. When the judges agreed, the class was

Step	Baseline	Our system
Filter (MI on the Web)	50%	70.7%
Dif. t-test classifier	71.4%	84.1%

Table 4: Accuracy of our main steps.

clear. When they did not agree, we designed simple rules, such as: when one judge chose the class preferred collocation, and the other judge chose the class anti-collocation, the class in the solution was less-preferred collocation. The agreement between judges was 80%; therefore we are confident that the quality of our standard solution is high. We used this standard solution as training data to learn a decision tree[6] for our three-way classifier. The features in the decision tree are the t-test between each collocation and the collocation from the same group that has maximum frequency on the Web, and the t-test between the current collocation and the collocation that has minimum frequency (as presented in Table 2). We could have set aside a part of the training data as a test set. Instead, we did 10-fold cross validation to quantify the accuracy on unseen data. The accuracy on the test set was 84.1% (compared with a baseline that chooses the most frequent class, anti-collocations, and achieves an accuracy of 71.4%). We also experimented with including MI as a feature in the decision tree, and with manually choosing thresholds (without a decision tree) for the three-way classification, but the accuracy was lower than 84.1%.

The three-way classifier can fix some of the mistakes of the MI filter. If a wrong collocation remained after the MI filter, the classifier can classify it in the anti-collocations class.

We can conclude that the collocational knowledge we acquired has acceptable quality.

6 Word Association

We performed a second experiment, where we looked for long distance co-occurrences (words that co-occur in a window of size K). We call these associations, and they include the lexical collocations we extracted in section 2.

We use BSP with the option of looking for bigrams in a window larger than 2. For example if the window size is 3, and the text is *vaccine/N cure/V available/A*, the extracted bigrams are *vaccine/N cure/V*, *cure/V available/A*, and *vaccine/N available/A*. We would like to choose a large (4–15) window size; the only problem is the increase in computation time. We look for associations of a word in the paragraph, not only in the sentence. Because we look for bigrams, we may get associations that occur to the left or to the right of the word. This is an indication of strong association.

We obtained associations similar to those presented by Church et al.(1991) for the near-synonyms *ship* and *boat*. Church et al. suggest that a lexicographer looking at these associations can infer that a *boat* is generally smaller than a *ship*, because they are found in *rivers* and *lakes*, while the *ships* are found in *seas*. Also, *boats* are used for small jobs (e.g., *fishing*, *police*, *pleasure*), whereas *ships* are used for serious business (e.g., *cargo*, *war*). Our intention is to use the associations to automatically infer this kind of knowledge and to validate acquired knowledge.

For our purpose we need only very strong associations, and we don't want words that associate with all near-synonyms in a cluster. Therefore we test for anti-associations using the same method we used in section 3, with the difference that the query asked to AltaVista is: x NEAR y (where x and y are the words of interest).

Words that don't associate with a near-synonym but associate with all the other near-synonyms in a cluster can tell us something about its nuances of meaning. For example *terrible slip* is an anti-association, while *terrible* associates with *mistake*, *blunder*, *error*. This is an indication that *slip* is a minor error.

Table 5 presents some preliminary results we obtained with $K = 4$ (on half the BNC and then on the Web), for the differential associations of *boat* (where $\surd$ marks preferred associations, ? marks less-preferred associations, and $*$ marks anti-associations). We used the same thresholds as for our experiment with collocations.

7 Related work

There has been a lot of work done in extracting collocations for different applications. We have already

[6]We used C4.5, http://www.cse.unsw.edu.au/~quinlan

Near-synonyms	*fishing*	*club*	*rowing*
boat	$\surd$	$\surd$	$\surd$
vessel	$\surd$	$*$	$*$
craft	?	?	?
ship	$*$	?	?

Table 5: Example of results for associations.

mentioned some of the most important contributors.

Like Church et al.(1991), we use the t-test and mutual information, but unlike them we use the Web as a corpus for this task (and a modified form of mutual information), and we distinguish three types of collocations (preferred, less-preferred, and anti-collocations).

We are concerned with extracting collocations for use in lexical choice. There is a lot of work on using collocations in NLG (but not in the lexical choice sub-component). There are two typical approaches: the use of phrasal templates in the form of canned phrases, and the use of automatically extracted collocations for unification-based generation (McKeown and Radev, 2000).

Statistical NLG systems (such as Nitrogen (Langkilde and Knight, 1998)) make good use of the most frequent words and their collocations. But such a system cannot choose a less-frequent synonym that may be more appropriate for conveying desired nuances of meaning, if the synonym is not a frequent word.

Finally, there is work related to ours from the point of view of the synonymy relation.

Turney (2001) used mutual information to detect the best answer to questions about synonyms from Test of English as a Foreign Language (TOEFL) and English as a Second Language (ESL). Given a problem word (with or without context), and four alternative words, the question is to choose the alternative most similar in meaning with the problem word. His work is based on the assumption that two synonyms are likely to occur in the same document (on the Web). This can be true if the author needs to avoid repeating the same word, but not true when the synonym is of secondary importance in a text. The alternative that has the highest PMI-IR (pointwise mutual information for information retrieval) with the problem word is selected as the answer. We used the same measure in section 3 — the mutual information between a collocation and a collocate that has the potential to discriminate between near-synonyms. Both works use the Web as a corpus, and a search engine to estimate the mutual information scores.

Pearce (2001) improves the quality of retrieved collocations by using synonyms from WordNet (Pearce, 2001). A pair of words is considered a collocation if one of the words significantly prefers only one (or several) of the synonyms of the other word. For example, *emotional baggage* is a good collocation because *baggage* and *luggage* are in the same synset and $*$*emotional luggage* is not a collocation. As in our work, three types of collocations are distinguished: words that collocate well; words that tend to not occur together, but if they do the reading is acceptable; and words that must not be used together because the reading will be unnatural (anti-collocations). In a similar manner with (Pearce, 2001), in section 3, we don't record collocations in our lexical knowledge-base if they don't help discriminate between near-synonyms. A difference is that we use more than frequency counts to classify collocations (we use a combination of t-test and MI).

Our evaluation was partly inspired by Evert and Krenn (2001). They collect collocations of the form noun-adjective and verb-prepositional phrase. They build a solution using two human judges, and use the solution to decide what is the best threshold for taking the N highest-ranked pairs as true collocations. In their experiment MI behaves worse that other measures (LL, t-test), but in our experiment MI on the Web achieves good results.

8 Conclusions and Future Work

We presented an unsupervised method to acquire knowledge about the collocational behaviour of near-synonyms.

Our future work includes improving the way we combine the five measures for ranking collocations, maybe by giving more weight to the collocations selected by the log-likelihood ratio. We also plan to experiment more with disambiguating the senses of the words in a collocation.

Our long-term goal is to acquire knowledge about near-synonyms from corpora and other sources, by bootstrapping with our initial lexical knowledge-base of near-synonym differences. This includes validating the knowledge already asserted and learning more distinctions.

Acknowledgments

We thank Gerald Penn, Olga Vechtomova, and three anonymous reviewers for their helpful comments on previous drafts of this paper. We thank Eric Joanis and Tristan Miller for helping with the judging task. Our work is financially supported by the Natural Sciences and Engineering Research Council of Canada and the University of Toronto.

Appendix

The first 10 collocations selected by each measure are presented below. Note that some of the measures rank many collocations equally at rank 1: MI 358 collocations; LL one collocation; χ^2 828 collocations; Dice 828 collocations; and Fisher 435,000 collocations (when the measure is computed with a precision of 10 digits — higher precision is recommended, but the computation time becomes a problem). The rest of the columns are: the rank assigned by the measure, the value of the measure, the frequency of the collocation in BNC, the frequency of the first word in the first position in bigrams, and the frequency of the second word in the second position in bigrams.

Some of the collocations ranked 1 by MI:

source-level/A debugger/N	1	21.9147	4	4	4
prosciutto/N crudo/N	1	21.9147	4	4	4
rumpy/A pumpy/A	1	21.9147	4	4	4
thrushes/N blackbirds/N	1	21.9147	4	4	4
clickity/N clickity/N	1	21.9147	4	4	4
bldsc/N microfilming/V	1	21.9147	4	4	4
chi-square/A variate/N	1	21.9147	4	4	4
long-period/A comets/N	1	21.9147	4	4	4
tranquillizers/N sedatives/N	1	21.9147	4	4	4
one-page/A synopsis/N	1	21.9147	4	4	4

First 10 collocations selected by LL:

prime/A minister/N	1	123548	9464	11223	18825
see/V p./N	2	83195	8693	78213	10640
read/V studio/N	3	67537	5020	14172	5895
ref/N no/N	4	62486	3630	3651	4806
video-taped/A report/N	5	52952	3765	3765	15886
secretary/N state/N	6	51277	5016	10187	25912
date/N award/N	7	48794	3627	8826	5614
hon./A friend/N	8	47821	4094	10345	10566
soviet/A union/N	9	44797	3894	8876	12538
report/N follows/V	10	44785	3776	16463	6056

Some of the collocations ranked 1 by χ^2:

lymphokine/V activated/A	1	15813684	5	5	5
config/N sys/N	1	15813684	4	4	4
levator/N depressor/N	1	15813684	5	5	5
nobile/N officium/N	1	15813684	11	11	11
line-printer/N dot-matrix/A	1	15813684	4	4	4
dermatitis/N herpetiformis/N	1	15813684	9	9	9
self-induced/A vomiting/N	1	15813684	5	5	5
horoscopic/A astrology/N	1	15813684	5	5	5
mumbo/N jumbo/N	1	15813684	12	12	12
long-period/A comets/N	1	15813684	4	4	4

Some of the collocations ranked 1 by Dice:

clarinets/N bassoons/N	1	1.00	5	5	5
email/N footy/N	1	1.00	4	4	4
tweet/V tweet/V	1	1.00	5	5	5
garage/parking/N vehicular/A	1	1.00	4	4	4
growing/N coca/N	1	1.00	5	5	5
movers/N seconders/N	1	1.00	5	5	5
elliptic/A integrals/N	1	1.00	8	8	8
viscose/N rayon/N	1	1.00	15	15	15
cause-effect/A inversions/N	1	1.00	5	5	5
first-come/A first-served/A	1	1.00	6	6	6

Some of the collocations ranked 1 by Fisher:

roman/A artefacts/N	1	1.00	4	3148	108
qualitative/A identity/N	1	1.00	16	336	1932
literacy/N education/N	1	1.00	9	252	20350
disability/N pension/N	1	1.00	6	470	2555
units/N transfused/V	1	1.00	5	2452	12
extension/N exceed/V	1	1.00	9	1177	212
smashed/V smithereens/N	1	1.00	5	194	9
climbing/N frames/N	1	1.00	5	171	275
inclination/N go/V	1	1.00	10	53	51663
trading/N connections/N	1	1.00	6	2162	736

References

Kenneth Church and Patrick Hanks. 1991. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29.

Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. 1991. Using statistics in lexical analysis. In Uri Zernik, editor, *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, pages 115–164. Lawrence Erlbaum.

Ted Dunning. 1993. Accurate methods for statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Philip Edmonds and Graeme Hirst. 2002 (to appear). Near-synonymy and lexical choice. *Computational Linguistics*, 28(2).

Stefan Evert and Brigitte Krenn. 2001. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the of the Association for Computational Linguistics (ACL'2001)*, Toulouse, France.

S. I. Hayakawa. 1994. *Choose the Right Word*. Harper-Collins Publishers.

Diana Zaiu Inkpen and Graeme Hirst. 2001. Building a lexical knowledge-base of near-synonym differences. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'2001)*, Pittsburgh.

Irene Langkilde and Kevin Knight. 1998. The practical value of N-grams in generation. In *Proceedings of the International Natural Language Generation Workshop*, Niagara-on-the-Lake, Ontario.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of SIGDOC Conference*, Toronto.

Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.

Kathleen McKeown and Dragomir Radev. 2000. Collocations. In R. Dale, H. Moisl, and H. Somers, editors, *Handbook of Natural Language Processing*. Marcel Dekker.

Darren Pearce. 2001. Synonymy in collocation extraction. In *Proceedings of the Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh.

Ted Pedersen. 1996. Fishing for exactness. In *Proceedings of the South-Central SAS Users Group Conference (SCSUG-96)*, Austin, Texas.

Frank Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177.

Peter Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, pages 491–502, Freiburg, Germany.