Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy

Lushan Han[1], Tim Finin[1,2], Paul McNamee[2], Anupam Joshi[1] and Yelena Yesha[1]

[1] Computer Science and Electrical Engineering
University of Maryland, Baltimore County

[2] Human Language Technology Center of Excellence
Johns Hopkins University

29 December 2011

Abstract

Pointwise mutual information (PMI) is a widely used word similarity measure, but it lacks a clear explanation of how it works. We explore how PMI differs from distributional similarity, and we introduce a novel metric, PMI$_{max}$, that augments PMI with information about a word's number of senses. The coefficients of PMI$_{max}$ are determined empirically by maximizing a utility function based on the performance of automatic thesaurus generation. We show that it outperforms traditional PMI in the application of automatic thesaurus generation and in two word similarity benchmark tasks: human similarity ratings and TOEFL synonym questions. PMI$_{max}$ achieves a correlation coefficient comparable to the best knowledge-based approaches on the Miller-Charles similarity rating dataset.

This is a preprint of

> Lushan Han, Tim Finin, Paul McNamee, Anupam Joshi, and Yelena Yesha, Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy, IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, to appear.

This research was supported by MURI award FA9550-08-1- 0265 from the Air Force Ofce of Scientic Research, NSF award IIS-0326460, a gift from Microsoft, and the Human Language Technology Center of Excellence.

1 Introduction

Word similarity is a measure of how semantically similar a pair of words is, with synonyms having the highest value. It is widely used for applications in natural language processing (NLP), information retrieval, and artificial intelligence, including tasks like word sense disambiguation [1], malapropism detection [2], paraphrase recognition [3], image and document retrieval [4] and predicting hyperlink-following behavior [5]. There are two prevailing approaches to computing word similarity, based on either using of a thesaurus (e.g., WordNet [6]) or statistics from a large corpus. There are also hybrid approaches [7] combining the two methods. Many well-known word similarity measures have been based on WordNet [8, 9, 10] and most of semantic applications [1, 4, 2] rely on these taxonomy-based measures.

Organizing all words in a well-defined taxonomy and linking them together with different relations is a labor-intensive task that requires significant maintenance as new words and word senses are formed. Furthermore, existing WordNet-based similarity measures typically depend heavily on "IS-A" information, which is available for nouns but incomplete for verbs and completely lacking for adjectives and adverbs. Consequently, these metrics perform poorly (with accuracy no more than 25%) [11] in answering TOEFL synonym questions [12] where the goal is selecting which of four candidate choices is most like a synonym to a given word. In contrast, some corpus-based approaches achieve much higher accuracy on the task (above 80%) [13, 14].

We expect statistical word similarity to continue to play an important role in semantic acquisition from text [9] in the future. A common immediate application is automatic thesaurus generation, in which various statistical word similarity measures [15, 16, 17, 18, 19] have been proposed. These are based on the distributional hypothesis [20], which states that words occurring in the same contexts tend to have similar meanings. Thus, the meaning of a word can be represented by a context vector of accompanying words and their co-occurrences counts, modulated perhaps by weighting functions [18], measured either in document context, text window context, or grammatical dependency context. The context vectors can be further transformed to a space of reduced dimension by applying singular value decomposition (SVD), yielding the familiar latent semantic analysis (LSA) technique [12]. The similarity of two words is then computed as the similarity of their context vectors, and the most common metric is the cosine of the angle between the two vectors. We will refer this kind of word similarity as distributional similarity, following convention in the research community [21, 22, 2].

PMI has emerged as a popular statistical word similarity measure that is not based on the distributional hypothesis. Calculating PMI only requires simple statistics about two words: their marginal frequencies and their co-occurrence frequency in a corpus. In the ten years after PMI was introduced to statistical NLP by Church and Hanks [23] it was mainly used for measuring word association [24] and was not thought of as a word similarity measure. Along with other statistical association measures such as the t-test, χ^2-test, and likelihood ratio, PMI was commonly used for finding collocations [24]. PMI was also a popular weighting function used in computing distributional similarity measures [15, 17, 22]. Using PMI as a word similarity measure began with the work of Turney [25], who developed a technique he called PMI-IR that used page counts from a Web search engine to approximate frequency counts in computing PMI values for word pairs. This produced remarkably good performance in answering TOEFL synonym questions – an accuracy of 74% which outperformed LSA [12], and was the best result at that time.

Turney's result was surprising because finding synonyms was a typical task for distributional similarity measures, and PMI, a word association measure, performed even better. Terra and Clarke [13] redid the TOEFL experiments for a set of the most well-known word association measures including PMI, χ^2-test, likelihood ratio, and average mutual information. The experiments were based on a very large Web corpus, and document and text window contexts of various sizes were investigated. They found PMI performed the best overall, and obtained an accuracy of 81.25% with a window size of 16 to 32 words.

PMI-IR has subsequently been used as a word similarity measure in other applications with good results. Mihalcea [3] used PMI-IR, LSA, and six WordNet based similarity measures as a sub-module in computing text similarity and applying it to paraphrase recognition and found that PMI-IR slightly outperformed the others. In a task of predicting user click behavior, predicting the HTML hyperlinks that a user is most likely to select given an information goal, Kaur [5] also showed that PMI-IR performs better than LSA and six WordNet based measures.

Why PMI is effective as a word similarity measure is still not clear. Many researchers use Turney's good results of PMI-IR as empirical credence [13, 26, 3, 5, 27]. To explain the success of PMI, some propose the proximity hypothesis [26, 13] noting that similar words tend to occur near each other, which is quite different from the distributional hypothesis which assumes that similar words tend to occur in similar contexts. However, to our knowledge no further explanations have been provided in the literature.

In this paper, we offer an intuitive explanation for why PMI can be used as a word similarity measure and illustrate behavioral differences between first-order PMI similarity and second-order distributional similarity. We also provide new experiments and examples, allowing more insight into PMI similarity. Our main contribution is introducing a novel metric, PMI$_{max}$, that enhances PMI to take into account the fact that words have multiple senses. The new metric is derived from the assumption that more frequent content words have more senses. We show that PMI$_{max}$ significantly improves the performance of PMI in the application of automatic thesaurus generation and outperforms PMI on benchmark datasets including human similarity rating datasets and TOEFL synonym questions. PMI$_{max}$ also has the advantage of not requiring expensive resources, such as sense-annotated corpora.

The remainder of the paper proceeds as follows. In Section 2 we discuss PMI similarity and define the PMI$_{max}$ metric. In Section 3 we use experiments in automatic thesaurus generation to determine the coefficients of PMI$_{max}$, evaluate its performance, and examine its assumptions. Additional evaluation using benchmark datasets is presented in Section 4. Section 5 discusses potential applications of PMI$_{max}$ and our future work. We are particularly interested in exploiting behavioral differences between PMI similarity and distributional similarity in the application of semantic acquisition from text. Finally, we conclude the paper in Section 6.

2 Approach

We start this section by discussing why PMI can serve as a semantic similarity measure. Then, we point out a problem introduced by PMI's assumption that words only possess a single sense, and we propose a novel PMI metric to consider polysemy of words.

2.1 PMI as a Semantic Similarity Measure

Intuitively, the semantic similarity between two concepts[1] can be defined as how much commonality they share. Since there are different ways to define commonality, semantic similarity tends to be a fuzzy concept. Is *soldier* more similar to *astronomer* or to *gun*? If commonality is defined as purely involving *IS-A* relations in a taxonomy such as WordNet, then *soldier* would be more similar to *astronomer* because both are types of people. But if we base commonality on aptness to a domain, then *soldier* would be more similar to *gun*. People naturally do both types of reasoning, and to evaluate computational semantic similarity measures, the standard practice is to rely on subjective human judgments.

Many researchers think that semantic similarity ought to be based only on *IS-A* relations and that it represents a special case of semantic relatedness which also includes antonymy, meronymy and other

[1] A concept refers to a particular sense of a word and we use an italic word to signify it in this section.

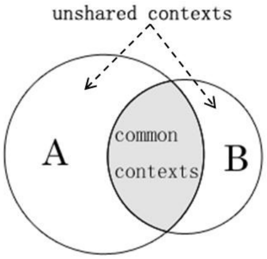

Figure 1: Common contexts between concepts A and B.

associations [2]. However, in the literature semantic similarity also refers to the notion of belonging to the same semantic domain or topic, and is interchangeable with semantic relatedness or semantic distance [24]. In this paper, we take the more relaxed view and use two indicators to assess the goodness of a statistical word similarity measure: (i) its ability to find synonyms, and (ii) how well it agrees with human similarity ratings.

We use Nida's example noted by Lin [17] to help describe why PMI can be a semantic similarity measure:

> A bottle of tezgüino is on the table.
> Everyone likes tezgüino.
> Tezgüino makes you drunk.
> We make tezgüino out of corn.

By observing the contexts in which the concept *tezgüino* is used, we can infer that *tezgüino* is a kind of alcoholic beverage made from corn, exemplifying the idea that a concept's meaning can be characterized by its contexts. By saying a concept has a context, we mean the concept is likely to appear in the context. For example, *tezgüino* is likely to appear in the context of "drunk", so *tezgüino* has the context "drunk". We may then define concept similarity as how much their contexts overlap, as illustrated in Figure 1. The larger the overlap becomes, the more similar the two concepts are, and vice versa.

Two concepts are more likely to co-occur in a common, shared context and less likely in an unshared one. In a shared context, both have an increased probability of appearing but in an unshared one, as in Figure 1, one is more likely but the other not. Generally, for two concepts with fixed sizes[2], the larger their context overlap is, the more co-occurrences result. In turn, the number of co-occurrences can be used to indicate the amount of common contexts between two concepts with fixed sizes.

The number of co-occurrences also depends on the sizes of the two concepts. Therefore, we need a normalized measure of co-occurrences to represent their similarity. PMI fits this role well. Equation 1 shows how to compute PMI for concepts in a sense-annotated text corpus, where f_{c_1} and f_{c_2} are the individual frequencies (counts) of the two concepts c_1 and c_2 in the corpus, and $f_d(c_1, c_2)$ is the co-occurrence frequency of c_1 and c_2 measured by the context window of d words and N is the total number of words in the corpus. In this paper, log always stands for natural logarithm.

$$\mathrm{PMI}(c_1, c_2) \approx \log(\frac{f_d(c_1, c_2) \cdot N}{f_{c_1} \cdot f_{c_2}}) \tag{1}$$

Traditionally, PMI is explained as the logarithmic ratio of the actual joint probability of two events to the expected joint probability if the two events were independent [23]. Here, we interpret it from a slightly different perspective and this interpretation is used in deriving our novel PMI metric in the next section. The term $f_{c_1} \cdot f_{c_2}$ can be interpreted as the number of all co-occurrence possibilities

[2] The size of a concept refers to the frequency count of the concept in a corpus.

or combinations between c_1 and c_2[3]. The term $f_d(c_1, c_2)$ gives the number of co-occurrences actually fulfilled. Thus, the ratio $\frac{f_d(c_1,c_2)}{f_{c_1} \cdot f_{c_2}}$ measures the extent to which two concepts tend to co-occur. By analogy to the correlation in statistics which measures the degree that two random variables tend to co-increase/decrease, PMI measures the likelihood that two concepts tend to co-occur versus occurring alone. In this sense, we say that PMI computes the correlation between the concepts c_1 and c_2. We also refer to the semantic similarity that PMI represents as correlation similarity.

Because a concept has the largest context overlap with itself, a concept has the largest chance to co-occur with itself. In other words, a concept has the strongest correlation with itself (auto-correlation). Auto-correlation is closely related to word burstiness, a phenomenon that words tend to appear in bursts. If the "one sense per discourse" hypothesis [28] is applied, word burstiness would be essentially the same as concept burstiness. Thus word burstiness is a reflection of the auto-correlation of concepts.

It is interesting that synonym correlation derives from the auto-correlation of concepts. If identical concepts happen within a distance as short as a few sentences, writers often prefer using synonyms to avoid excessive lexical repetition. The probability of substituting synonyms depends on the nature of the concept as well as the writer's literary style. In our observations, synonym correlation often has a value very close to auto-correlation for verb, adjective and adverb concepts, but a value a little bit lower for nominal concepts. One reason may be the ability to use a pronoun as an alternative to a synonym in English. Although verbs, adjectives and adverbs also have pronominal forms, they are less powerful than pronouns and used less frequently.

PMI similarity is related to distributional similarity in that both are context-based similarity measures. However, they use contexts differently which results in different behaviors. First, the PMI similarity between two concepts is determined by how much their contexts overlap, while distributional similarity depends on the extent that two concepts have similar context distributions. For example, *garage* has a very high PMI similarity with *car* because *garage* rarely occurs in contexts which are not subsumed by the contexts of *car*. However, distributional similarity typically does not consider *garage* to be very similar to *car* because the context distributions of *garage* and *car* vary considerably. While one might expect all words related by a PART-OF relation to have high PMI similarity, this is not the case. *Table* is not PMI-similar to *leg*, because there are many other contexts related to *leg* but not *table*, and vice versa.

Second, for two given concepts, distributional similarity obtains collective contextual information for each concept and computes how similar their context vectors are. For distributional similarity it does not require the concepts to co-occur in the same contexts to be similar. In contrast, PMI similarity emphasizes the propensity for two concepts to co-occur in the exactly same contexts. For example, the distributional similar concepts for *car* may include not only *automobile*, *truck*, and *train*, but also *boat*, *ship*, *carriage* and *chariot*. This shows the ability of distributional similarity to find "indirect" similarity. However, a problem with distributional similarity is that it cannot distinguish them and separate them into three categories: "land vehicle", "water vehicle", and "archaic vehicle". On the other hand, the PMI-similar concepts for *car* do not contain *boat* or *carriage* because they do not co-occur with *car* in the same contexts frequently enough.

These differences suggest that PMI similarity should be a valuable complement to distributional similarity. Although the idea that PMI can complement distributional similarity is not new, current techniques [29, 27] have used them as separate features in statistical models and do not exploit how that they differ. In Section 5 we will show through examples that we can support interesting applications by exploiting their behavioral differences.

[3] The requirement for multiplication rather than addition can be easily understood by an example: if one individual frequency increased twofold, all co-occurrence possibilities would be doubled rather than increased by the individual frequency

2.2 Augmenting PMI to Account for Polysemy

Since it is very expensive to produce a large sense-tagged corpus, statistical semantic similarity measures are often computed based on words rather than word senses [2]. However, when PMI is applied to measure correlation between words[4], it has a problem because it assumes that words only have a single sense. Consider "make" and "earn" as an example. "Make" has many senses, only one of which is synonymous with "earn", and so it is inappropriate to divide by the whole frequency of "make" in computing the PMI correlation similarity between "make" and "earn", since only a fraction of "make" occurrences have the same meaning of "earn".

PMI has a well-known problem that it tends to over-emphasize the association of low frequency words [30, 31, 32]. We conjecture that the fact that more frequent content words tend to have more senses, as shown in Figure 4, is an important cause for PMI's frequency bias. More frequent words are disadvantaged in producing high PMI value because they tend to have more unrelated or less related senses and thereby bear an extra burden by including the frequencies of these senses in their marginal counts. We should distinguish this cause from the one described by Dunning [33] that the normality assumption breaks down on rare words, which can be ruled out by using a minimum frequency threshold.

Although it can be difficult to determine which sense of a word is being used, this does not prevent us from making a more accurate assumption than the "single sense" assumption. We will demonstrate a significant improvement over traditional PMI by merely assuming that more frequent content words have a greater number of senses.

We start by modeling the number of the senses of an open-class word (i.e., noun, verb, adjective, or adverb) as a power function of the log frequency of the word with a horizontal translation q. More specifically,

$$y_w = a(\log(f_w) + q)^p \tag{2}$$

where f_w and y_w are the frequency (count) of the word w and its number of senses respectively; a, p and q are three coefficients needed to resolve. The displacement q is necessary because f_w is not a normalized measure of the word w and varies on the size of the selected corpus. We require $(\log(f_w) + q > 0)$ to only take the strict monotone increasing part of the power function. The power function assumption is based on observing the graphs in Figure 4, which show the dependence between a word's log frequency in a large corpus (e.g., two billion words) and its number of senses obtained from WordNet. This relationship, though simple, is better modeled using a power function rather than a linear one.

We next estimate a word pair's PMI value between their closest senses using two assumptions. Given a word pair, it is hard to know the proportions at which the closest senses are engaged in their own words. Since it can be either a major or minor sense, we simply assume the average proportion $\frac{1}{y_w}$. Consequently, the frequency of a word used as the sense most correlated with a sense in the other word is estimated as $\frac{f_w}{y_w}$.

The co-occurrence frequency between a word pair w_1 and w_2, represented by $f_d(w_1, w_2)$, is also larger than the co-occurrence frequency between the two particular senses of w_1 and w_2. To estimate the co-occurrence frequency between the two senses we assume that w_1 and w_2 have strongest correlation only on the two particular senses and otherwise normal correlation. Normal correlation is the expected correlation between common English words and we denote it using PMI value k[5]. Therefore, the co-occurrence frequency between the two particular senses of w_1 and w_2 can be estimated by subtracting the co-occurrence frequency contributed by other combinations of senses, denoted by x, from the total

[4] The same formula as in Equation 1 is used, except that senses are replaced by words

[5] More explanations will be given in Section 3.4 after the k is empirically learned and exhibited in Table 5.

co-occurrence frequency between the two words. More specifically,

$$f_d(w_1, w_2) - x$$

where x is computed using the definition of PMI by solving Equation 3.

$$\log(\frac{x \cdot N}{f_{w_1} \cdot f_{w_2} - \frac{f_{w_1}}{y_{w_1}} \cdot \frac{f_{w_2}}{y_{w_2}}}) = k \tag{3}$$

Equation 3 amounts to asking the question – with a correlation degree of k, how many co-occurrences are expected among the remaining co-occurrence possibilities resulted by excluding the possibilities between the two senses of interest from the total possibilities.

Finally, the modified PMI, called PMI$_{max}$, between the two words w_1 and w_2 is given in Equation 4.

$$\begin{aligned} &\mathrm{PMI}_{max}(w_1, w_2) \\ &= \log(\frac{(f_d(w_1,w_2)-x)\cdot N}{\frac{f_{w_1}}{y_{w_1}} \cdot \frac{f_{w_2}}{y_{w_2}}}) \\ &= \log(\frac{(f_d(w_1,w_2)-\frac{e^k}{N}\cdot(f_{w_1}\cdot f_{w_2}-\frac{f_{w_1}}{y_{w_1}}\cdot\frac{f_{w_2}}{y_{w_2}}))\cdot N}{\frac{f_{w_1}}{y_{w_1}} \cdot \frac{f_{w_2}}{y_{w_2}}}) \end{aligned} \tag{4}$$

PMI$_{max}$ estimates the maximum correlation between two words, i.e., the correlation between their closest senses. In circumstances where we cannot know the particular senses used, it is reasonable to take the maximum similarity among all possible sense pairs as a measure of word similarity. For example, when sense information is unavailable, the shortest path assumption is often taken to compute word similarity in the WordNet-based measures. While the assumptions made in deriving the PMI$_{max}$ may appear naive, we will demonstrate their effectiveness in later sections. More sophisticated models may lead to better results and we plan to explore this in future work.

3 Experiments

In this section, we determine values for the coefficients used in PMI$_{max}$ by selecting ones that maximize performance in automatic thesaurus generation, where the core task is to find synonyms for a target word. Synonyms can be synonymous to the target word in their major or minor senses, and synonyms with more senses tend to be more difficult to find because they have relatively less semantic overlap with the target. Because more frequent words tend to have more senses, the synonyms under-weighted by PMI are often those with high frequency. We hope PMI$_{max}$ can fix or alleviate this problem and find synonyms without a frequency bias.

Since word similarity is usually measured within the same part of speech (POS) (e.g., in [8, 9, 10]), we learn different coefficient sets for nouns, verbs, adjectives and adverbs. We further examine the learned relations between the frequency of a word with a particular POS and its number of senses within the POS (i.e., Equation 2) using knowledge from WordNet.

This section also serves as an evaluation of PMI$_{max}$ on the task of automatic thesaurus generation. We start by describing our corpus, evaluation methodology, and the gold standard. Next we present and analyze the performance of basic PMI, PMI$_{max}$, and a state-of-the-art distributional similarity measure.

3.1 Corpus Selection

To learn coefficients, we prefer a balanced collection of carefully written and edited text. Since PMI is sensitive to noise and sparse data, a large corpus is required. The British National Corpus (BNC) is one of the most widely used corpora in text mining with more than 100 million words. Though large enough for distributional similarity based approaches, it is not of sufficient size for PMI.

Given these considerations, we selected the Project Gutenberg [34] English eBooks as our corpus. It comprises more than 27,000 free eBooks, many of which are well-known. A disadvantage of the collection is its age – most of the texts were written more than 80 years ago. Consequently, many new terms (e.g., "software", "Linux") are absent. We processed the texts to remove copyright statements and excluded books that are repetitions of similar entries (e.g., successive editions of the CIA World Factbook) or are themselves a thesaurus (e.g., Roget's Thesaurus, 1911). We further removed unknown thesaurus-like data in the corpus with a classifier using a simple threshold based on the percent of punctuation characters. This simple approach is effective in identifying thesaurus-like data, which typically consists of sequences of single words separated by commas or semi-colons. Last, we performed POS tagging and lemmatization on the entire corpus using the Stanford POS tagger [35]. Our final corpus contains roughly two billion words, almost 20 times as large as the BNC corpus.

3.2 Evaluation Methodology

Various methods for evaluating automatically generated thesauri have been used in previous research. Some evaluate their results directly using subjective human judgments [15, 36] and others indirectly by measuring the impact on task performance [37].

Some direct and objective evaluation methodologies have also been proposed. The simplest is to directly compare the automatically and manually generated thesauri [16]. However, a problem arises in that current methods do not directly supply synonyms but rather lists of candidate words ranked by their similarity to the target. To enable automatic comparison, Lin [17] proposed first defining word similarity measures for hand crafted thesauri and then transforming them into the same format as the automatically generated thesaurus – a vector of words weighted by their similarities to the target. Finally cosine similarity is computed between the vectors from machine generated and hand crafted thesauri. However, it is unclear if such transformation is a good idea since it adds to the gold standard a large number of words that are not synonyms but related words, a deviation from the original goal of automatic thesaurus generation.

We chose a more intuitive and straightforward evaluation methodology. We use the recall levels in six different top n lists to show how high the synonyms of the head word in an entry in the "gold standard" occur in the automatically generated candidate list. The selected values for n are 10, 25, 50, 100, 200, and 500. A key application for automated thesaurus induction is to assist lexicographers in identifying synonyms, and we feel that a list of 500 candidates is the largest set that would be practical. We considered using Roget's Thesaurus or WordNet as the gold standard but elected not to use either. Roget's categories are organized by topic and include many related words that are not synonyms. WordNet, in contrast, has very fine-grained sense definitions for its synonyms, so relatively few synonyms can be harvested without exploiting hypernyms, hyponyms and co-hyponyms. Moreover, we wanted a gold standard thesaurus that is contemporaneous with our collection.

We chose Putnam's Word Book [38] as our gold standard. It contains more than 10,000 entries, each consisting of a head word, its part of speech, all its synonyms, and sometimes a few antonyms. The different senses (very coarse in general) of the head word are separated by semi-colons, and within each sense the synonyms are separated by commas, as in the following example.

> quicken, v. revive, resuscitate, animate; excite, stimulate, incite; accelerate, expedite, hasten, advance, facilitate, further

	All		Unique Sense		1st Synonym	
	Entries	Pairs	Entries	Pairs	Entries	Pairs
noun	2286	10994	1103	3762	886	886
verb	1187	6866	623	2515	559	559
adj.	1015	5944	579	2143	454	454
adv.	39	109	32	76	27	27

Table 1: The number of entries and synonym pairs for each POS category under our three scenarios.

While the coverage of Putnam's Word Book synonyms is not complete, it is extensive, so that our recall metric should be close to the true recall. On the other hand, measuring true precision is difficult since the top n lists can contain proper synonyms which are not included by Putnam. The different top n lists will give us a rough idea about how "precision" varies with the recall. In addition, we supply another measure – the average rank over all the synonyms contained by a top n list. We give an example below to show how to compute the recall measures and average ranks. The top 50 candidates computed using basic PMI for the verb "quicken" are:

> exhilarate, invigorate, energize, regenerate, alert, pulse, slacken, ***accelerate***, deaden, begrudge, husk, recreate, cluck, constrict, ***stimulate***, intensify, career, stagnate, lag, throb, toughen, whinny, enliven, ***resuscitate***, retard, broaden, rejuvenate, rebuff, lather, sharpen, plummet, pulsate, nerve, dull, miscalculate, weld, sicken, infuse, shrill, blunt, heighten, distance, deepen, neigh, near, kindle, rouge, freshen, amplify, hearten

Only three synonyms for the verb "quicken" in the gold standard appear among the top 50 candidates: "accelerate", "stimulate", and "resuscitate" with ranks 8, 15, and 24. Thus, the recall values for top 10, top 25 and top 50 lists are 1/12, 3/12 and 3/12, respectively (12 is the total number of synonyms for the verb "quicken" in our gold standard). The average ranks for top 10, top 25 and top 50 lists are 8, 15.67 and 15.67, respectively.

We initially process Putnam's Word Book to filter out entries whose head words have a frequency less than $10,000$ in the Gutenberg corpus and further eliminate words with frequency less than 700 in the synonym lists of the remaining head words. In our experiment, we observed that many synonyms have PMI values slightly above 7.0 (see Table 10). Using the thresholds $10,000$ and 700 enables them to co-occur at least four or five times, which is typically required to ensure PMI, a statistical approach, works reasonably [23]. In addition, multi-words terms and antonyms were removed.

Words can have multiple senses and many synonyms are not completely synonymous but overlap in particular senses. We would like to evaluate PMI's performance in finding synonyms with different degrees of overlap. To enable this kind of evaluation, we tested using three scenarios: (i) all entries; (ii) entries with a unique sense; and (iii) entries with a unique sense and only using the first synonym[6]. Our rationale is that the single sense words should have a greater semantic overlap with their synonyms and moreover the largest semantic overlap with their first synonyms. Although we require the head words to have a unique sense, their synonyms may still be polysemous. Table 1 shows the number of entries and synonym pairs in three scenarios with different part of speech tags.

3.3 Performance of Basic PMI

In our basic PMI algorithm, word co-occurrences[7] are counted in a moving window of a fixed size that scans the entire corpus. To select the optimal window size for the basic PMI metric, we experimented with 14 sizes, starting at ±5 and ending at ±70 with a step of five words. Windows were not allowed

[6] We exclude the entries whose first synonyms have a frequency less than 700

[7] Our word co-occurrence matrix is based on a predefined vocabulary of more than 22,000 common English words and its final dimensions are 26,000 × 26,000 when words are POS tagged.

		T10	T25	T50	T100	T200	T500	Avg
all entries	noun	0.22	0.36	0.47	0.58	0.68	0.80	0.52
	verb	0.23	0.36	0.46	0.57	0.69	0.83	0.52
	adj.	0.26	0.40	0.52	0.63	0.74	0.86	0.57
	adv.	0.29	0.50	0.62	0.75	0.84	0.96	0.66
unique sense	noun	0.27	0.43	0.55	0.67	0.76	0.86	0.59
	verb	0.30	0.45	0.56	0.67	0.78	0.90	0.61
	adj.	0.32	0.46	0.59	0.70	0.80	0.91	0.63
	adv.	0.32	0.55	0.68	0.77	0.87	0.96	0.69
1st synonym	noun	0.32	0.50	0.63	0.75	0.83	0.91	0.66
	verb	0.38	0.55	0.66	0.76	0.85	0.93	0.69
	adj.	0.36	0.55	0.68	0.78	0.86	0.95	0.70
	adv.	0.44	0.59	0.78	0.85	0.93	0.96	0.76

Table 2: Recall for basic PMI using a ±40 words window

		T10	T25	T50	T100	T200	T500	Avg
all entries	noun	0.29	0.44	0.56	0.66	0.75	0.85	0.59
	verb	0.33	0.47	0.58	0.68	0.77	0.88	0.62
	adj.	0.38	0.53	0.64	0.73	0.82	0.90	0.67
	adv.	0.49	0.67	0.77	0.84	0.89	0.98	0.77
unique sense	noun	0.36	0.53	0.64	0.74	0.82	0.90	0.67
	verb	0.43	0.57	0.69	0.77	0.85	0.93	0.71
	adj.	0.45	0.61	0.71	0.80	0.88	0.94	0.73
	adv.	0.53	0.73	0.83	0.87	0.92	0.98	0.81
1st synonym	noun	0.45	0.65	0.75	0.85	0.89	0.94	0.76
	verb	0.57	0.71	0.82	0.87	0.91	0.97	0.81
	adj.	0.58	0.74	0.84	0.90	0.96	0.98	0.83
	adv.	0.63	0.93	0.93	0.93	0.96	1.00	0.90

Table 3: Recall for PMI$_{max}$ using ±40 words window

to cross a paragraph boundary and we used a stop-word list consisting of only the three articles "a", "an" and "the".

We found that performance initially improves as the window size increases. However, the performance enters a plateau when the window size reaches ±40, a window that corresponds to about four lines of text in a typically formatted book. We note that our optimal window size is slightly larger than the 16-32 size obtained by Terra and Clarke [13]. Data sparseness may account for the difference because Terra and Clarke used a much larger corpus (53 billion words) and therefore data sparseness was less severe in their case. The six recall levels of basic PMI and their average for the context window of ±40 words are shown in Table 2. The average ranks are presented in Figure 2 for the four POS categories. Six markers on each line correspond to six top n lists. The x and y coordinates of a marker are the recall level and the average rank for the corresponding list. The average ranks for unique sense scenario are omitted due to space limitations. Note that in populating the ranked candidate list, we rule out the words whose frequency is under 700. This is consistent with the preprocessing we did for the gold standard.

The table and figures show that the "first synonym" category has better performance than the "unique sense" one, which is again better than the "all entries" category. We conclude that the synonyms with more semantic overlap have stronger correlation as measured by basic PMI. Among all

		T10	T25	T50	T100	T200	T500	Avg
all entries	noun	0.38	0.49	0.57	0.66	0.74	0.83	0.61
	verb	0.34	0.45	0.53	0.62	0.72	0.84	0.58
	adj.	0.37	0.48	0.58	0.66	0.75	0.86	0.62
	adv.	0.53	0.66	0.72	0.78	0.83	0.94	0.74
unique sense	noun	0.47	0.59	0.67	0.75	0.81	0.88	0.70
	verb	0.44	0.56	0.64	0.73	0.81	0.90	0.68
	adj.	0.45	0.57	0.66	0.74	0.82	0.91	0.69
	adv.	0.55	0.68	0.73	0.77	0.84	0.93	0.75
1st synonym	noun	0.62	0.74	0.80	0.84	0.88	0.91	0.80
	verb	0.58	0.70	0.77	0.83	0.90	0.95	0.79
	adj.	0.62	0.72	0.81	0.86	0.91	0.96	0.81
	adv.	0.67	0.78	0.81	0.89	0.93	0.96	0.84

Table 4: Recall for Distributional Similarity – PPMIC

POS tags, adverbs have the best performance, followed by adjectives and verbs, with nouns exhibiting the worst performance.

3.4 PMI$_{max}$ Coefficient Tuning and Performance

A number of coefficients must be determined for PMI$_{max}$. We find their optimal values by maximizing a utility function based on the performance of automatic thesaurus generation. *The utility function is defined as the average of the recall levels in the six different top n lists.* The intuitive basis behind this is to improve recall in the six lists while giving emphasis to smaller lists since each list subsumes all its smaller-sized lists. Increasing recall in a fixed top n list typically results in the improvement of precision in the list. Therefore, this utility function measures precision as well. To see why, suppose our gold standard supplies the complete set of synonyms for a target word. Then the ratio between the recall and the precision in a fixed top n list is a constant. Though not complete, our gold standard can be thought as supplying a random sample of synonyms. Thus, the recall in a top n list can be used to estimate the true recall and thereby the constant ratio property should hold.

We use a constrained brute force search to maximize the utility function. The PMI$_{max}$ coefficients (a, p, q and k) form a four dimensional search space that is computationally expensive to search. By averaging the number of senses extracted from WordNet over nouns, verbs, adjectives and adverbs with frequency around the minimum threshold (i.e., 700) respectively, we found that their mean values all fall into the range between one and two. So we make a simplifying assumption that words with frequency of 700 in our corpus have just one sense, reducing the search space to three dimensions. With this assumption, we can solve the coefficients a to be $\frac{1}{(\log(700)+q)^p}$. Then Equation 2 can be updated to

$$y_w = \frac{(\log(f_w) + q)^p}{(\log(700) + q)^p} \tag{5}$$

In exploring the three dimensional space, we let p be in the range [0..10] stepped by 0.5, which comprises 21 choices. To avoid searching in a continuous real range, we sample evenly spaced points. We choose the range [0..10] because we expect p to be positive and not a very high power. Similarly, we let q be in the range [-6..10] stepped by 1, yielding 17 choices. We set the left boundary as -6 because it is the smallest number that keeps ($\log(700) + q$) positive and the right boundary to 10 because we don't expect the displacement to be large due to the large corpus that we use. We let e^k be in the range [0..100] stepped by 10, which has 11 values. This spans a range from a very weak correlation to

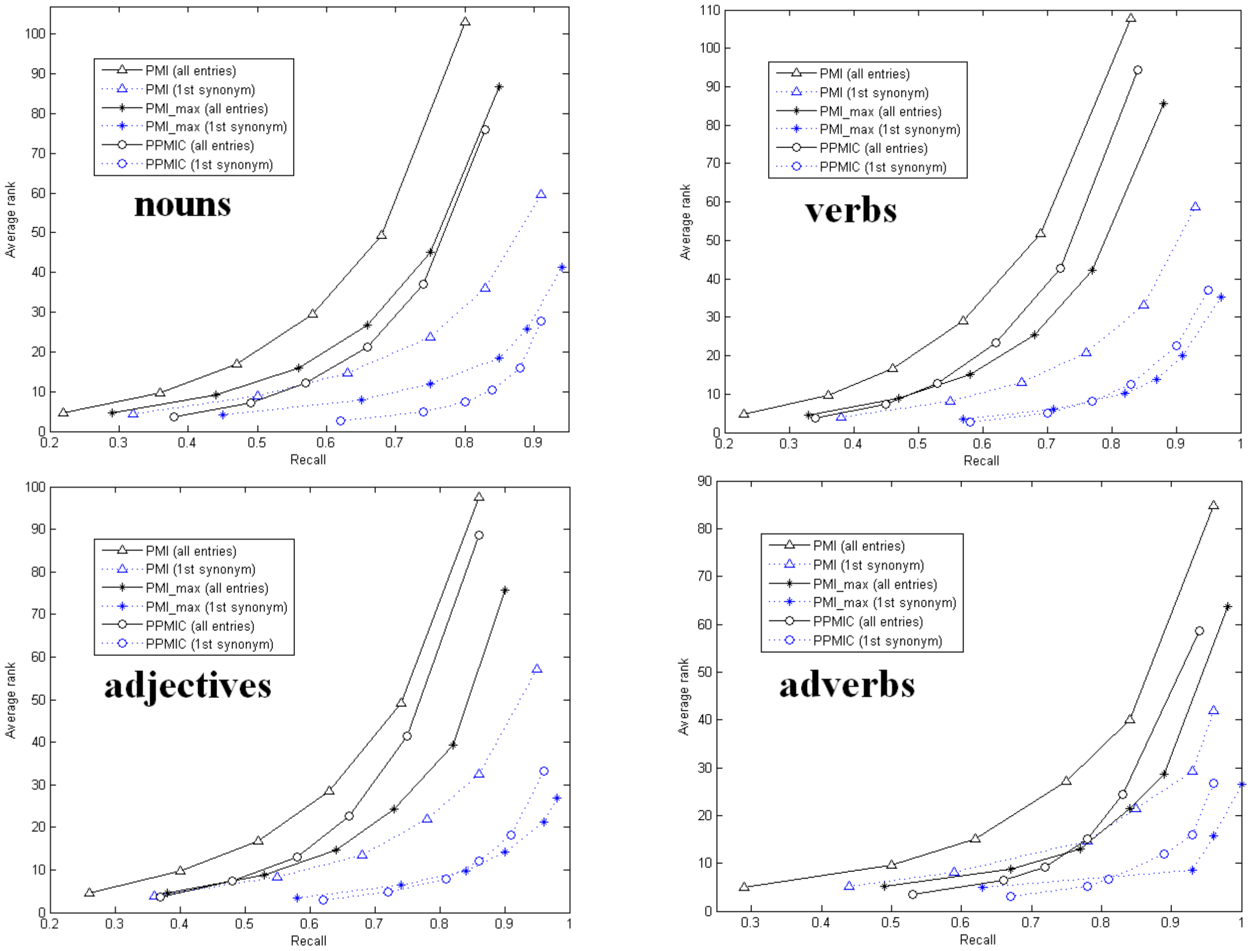

Figure 2: Average ranks for nouns, verbs, adjectives and adverbs. The six marks on each line correspond to six top n lists and their x and y coordinates give the recall and the average rank.

fairly strong correlation under our setting of context window of ±40 words. A total of 3927 candidate solutions are included in our search space.

In order to avoid overfitting, we randomly partitioned the Putnam entries under each POS, except adverbs[8], into two equal size datasets, resulting in three pairs of datasets for noun, verb and adjective and one single dataset for adverb. We then carried out 3927 experiments on each dataset separately, iterating and testing all the entries, and computing recall values for the six top n lists averaging over all entries, and finally calculating the utility function. All computations were based on the words co-occurrence matrix generated using the moving window of ±40 words. The optimal coefficients are shown in Table 5. For nouns and verbs, the optimal coefficients on the two datasets are the same. In the case of adjectives, although the optimal coefficients are different, their curves generated by Equation 5 are close, which is illustrated in the adjective graph in Figure 4. The strong agreement on the pairs of datasets is not accidental. By sorting the 3927 solutions using their utility values, we find that the higher a solution appears, the closer its curve is to the optimal solution's curve. In other words, the curves tend to converge to the optimal one as their utilities increase. Note that two combinations of p and q, though may vary dramatically in individual p and q values, can yield close curves, as shown in the adjective graph in Figure 4.

With our experimental settings, PMI values as the Ks in Table 5 indicate a correlation which is

[8] We only had 39 adverbial entries and 109 synonym pairs in total, so we did not apply two-fold cross-validation to it.

	p	**q**	e^k	**k**
noun (1,2)	1.5	-4	30	3.4
verb (1,2)	1.5	-5	40	3.7
adj. (1)	2	-4	70	4.2
adj. (2)	5	4	70	4.2
adv.	3	0	40	3.7

Table 5: Optimal coefficients on three partitioned datasets and one intact dataset

close to "normal correlation" for their particular POS. If we compute PMI values for all possible word pairs (within the same POS) satisfying our filtering criteria and use them to draw a histogram, we observe a familiar bell distribution. "Normal correlation" is the center of the distribution. Note that according to the Equation 4, these Ks impose a lower bound on what PMI$_{max}$ can compute. In many applications these lower bounds would not cause a problem because people are typically interested in correlations which are stronger than "normal correlation". For example, as illustrated in Table 10 for human similarity ratings, noun pairs holding PMI value around 3.4 would be judged as having no or very low similarity.

The performance of automatic thesaurus generation using two-fold cross-validation is shown in Table 3 and Figure 2. Although the coefficients are learned from the "all entries" scenario, the same coefficients are applied to generate the results for the "unique sense" and "first synonym" scenarios. As Table 3 shows, the recall values enjoy significant improvements over basic PMI for all of the scenarios, POS tags, and top n lists. Some recall values, for example, verbs in top 10 list as "first synonym", received a 50% increase. The improvements on all the recall values are statistically significant ($p <$ 0.001 two-tailed paired t-test). Regarding average rank, the comparison should be based on the same recall level instead of the same top n list. This is because a list with larger recall may contain more synonyms with bigger ranks and therefore draw down the average rank. Figure 2 clearly shows that, at the same recall level, average ranks for PMI$_{max}$ get across-the-board improvements upon basic PMI.

Compare PMI$_{max}$'s top 50 candidates for the verb "quicken" to those shown for PMI in Section 3.2.

> slacken, ***stimulate***, invigorate, regenerate, throb, ***accelerate***, intensify, exhilarate, kindle, deepen, deaden, sharpen, retard, enliven, near, awaken, ***revive***, thrill, heighten, overtake, pulse, broaden, stir, lag, sicken, infuse, expand, slow, brighten, dilate, strengthen, dull, purify, refresh, ***hasten***, begrudge, spur, trot, career, nerve, freshen, hurry, blunt, sanctify, warm, cluck, inspire, lengthen, speed, impart

For this example, four synonyms for the word "quicken" appear in the top 50 candidate list and are marked as bold. Their ranks are 2, 6, 17 and 35, so the recall values for top 10, 25 and 50 lists are 2/12, 3/12 and 4/12 respectively. The average ranks for top 10, 25 and 50 lists are 4, 8.3 and 15. These numbers all improve upon the numbers using basic PMI. High frequency verbs like "spur", "hurry", and "speed", which are not shown by basic PMI, also enter the ranking.

Figure 3 shows additional examples of synonym lists produced by PMI and PMI$_{max}$, displaying a word and its top 50 most similar words for each syntactic category. Words presented in both lists are marked as italic. The examples show that synonyms are generally ranked at the top places by both methods. However, antonyms can have rankings as high as synonyms because antonyms (e.g., "twist" and "straighten") have many contexts in common and a single negation may reverse their meaning to one another. Among all POS examples, the noun "car" exhibits the worse performance. Nevertheless, synonyms and similar concepts for "car", such as "automobile", "limousine" and "truck" still show up in the top 50 lists of both PMI and PMI$_{max}$. The very top places of "chauffeur", "garage" and "headlight" suggests that both measures highly rank a special kind of relation: nouns that are formed

(PMI) **twist_VB**: *contort*, *squirm*, *writhe*, *knot*, *twine*, *twirl*, *wriggle*, *warp*, card, *coil*, *wrench*, *loop*, *dislocate*, *braid*, interlock, untangle, snake, *distort*, dent, wiggle, *grimace*, unwind, mildew, slump, *stem*, flex, *splinter*, *crook*, thud, *stunt*, groove, *tangle*, *claw*, *mat*, hunch, lunge, *char*, *curl*, unhook, joint, clamp, blotch, crochet, constrict, rotate, sprain, lacquer, vein, *fork*, *protrude*

(PMI$_{max}$) **twist_VB**: *writhe*, *knot*, *squirm*, *contort*, *twine*, *wriggle*, *twirl*, *warp*, *coil*, *wrench*, *distort*, *curl*, *crook*, *stem*, round, *tangle*, *braid*, *grimace*, wind, spin, fasten, *loop*, jerk, *stunt*, *dislocate*, weave, curve, double, clutch, *protrude*, tie, screw, tear, *splinter*, bend, *mat*, hook, tug, crumple, *claw*, wrinkle, grip, roll, tighten, dangle, *char*, straighten, *fork*, pull, strangle

(PMI) **car_NN**: *garage*, *trolley*, *headlight*, *chauffeur*, *limousine*, *motorist*, *motor*, *siding*, *locomotive*, *caboose*, *subway*, *freight*, *automobile*, *conductor*, motorcycle, *driveway*, *axle*, *radiator*, *brake*, *throttle*, speeding, *uptown*, *curb*, *auto*, skid, *balloon*, *truck*, refrigerator, *driver*, downtown, parachute, *gasoline*, *steering*, spin, mileage, *passenger*, *racing*, *train*, *engine*, purr, suitcase, chute, tractor, *taxi*, *railroad*, traction, goggles, *elevator*, toot, standstill

(PMI$_{max}$) **car_NN**: *chauffeur*, *garage*, *motor*, *trolley*, *locomotive*, *conductor*, *automobile*, *limousine*, *freight*, *headlight*, *train*, *driver*, *brake*, *siding*, *passenger*, *engine*, *balloon*, *railroad*, *curb*, *axle*, wheel, *truck*, *motorist*, *auto*, *driveway*, *subway*, platform, track, *caboose*, speed, tire, compartment, *radiator*, baggage, depot, rail, *steering*, *elevator*, seat, shaft, station, *racing*, vehicle, *taxi*, *gasoline*, *throttle*, occupant, ambulance, *uptown*, road

(PMI) **ridiculous_JJ**: *nonsensical*, *laughable*, *puerile*, *absurd*, *preposterous*, *contemptible*, *sublime*, *ludicrous*, *grotesque*, *farcical*, *bizarre*, *melodramatic*, *insipid*, snobbish, *pedantic*, *comic*, *impertinent*, *incongruous*, *indecent*, *comical*, *odious*, panicky, *wasteful*, grandiose, *idiotic*, inane, *conceited*, *disgusting*, *satirical*, sobering, outlandish, narrow-minded, *despicable*, unreliable, stilted, messy, good-humored, *paltry*, irreverent, *extravagant*, regrettable, *degrading*, *humiliating*, humdrum, *pompous*, *frivolous*, exasperating, antiquated, *silly*, dowdy

(PMI$_{max}$) **ridiculous_JJ**: *absurd*, *sublime*, *contemptible*, *preposterous*, *grotesque*, *ludicrous*, *laughable*, *puerile*, *comic*, *nonsensical*, *odious*, *impertinent*, foolish, *silly*, *extravagant*, *comical*, *bizarre*, *incongruous*, childish, *insipid*, *disgusting*, *indecent*, *conceited*, vulgar, monstrous, fantastic, *idiotic*, *frivolous*, pathetic, *pedantic*, *satirical*, stupid, *paltry*, *melodramatic*, humorous, awkward, sentimental, *humiliating*, exaggerated, trivial, *farcical*, *pompous*, amused, *wasteful*, *despicable*, serious, *degrading*, senseless, funny, tragic

(PMI) **commonly_RB**: *supposedly*, *incorrectly*, *customarily*, *erroneously*, *technically*, *conventionally*, *ordinarily*, chemically, psychologically, traditionally, *infrequently*, credibly, sexually, predominantly, *popularly*, *improperly*, currently, *rarely*, *locally*, *seldom*, *usually*, mistakenly, *sparingly*, *generally*, legitimately, *frequently*, experimentally, *universally*, preferably, capriciously, *relatively*, hugely, rationally, *variously*, *exclusively*, vertically, negligently, *annually*, *habitually*, philosophically, fourthly, correspondingly, extensively, rigorously, *sometimes*, *necessarily*, *chiefly*, *invariably*, primarily, symmetrically

(PMI$_{max}$) **commonly_RB**: *generally*, *usually*, *frequently*, *seldom*, *rarely*, *sometimes*, *ordinarily*, often, *erroneously*, most, *technically*, *supposedly*, more, *universally*, *chiefly*, *incorrectly*, *exclusively*, especially, less, *necessarily*, *infrequently*, namely, *popularly*, *relatively*, *annually*, occasionally, hence, *invariably*, therefore, also, *customarily*, either, properly, likewise, *habitually*, widely, largely, formerly, very, *improperly*, comparatively, *variously*, *conventionally*, particularly, *sparingly*, however, *locally*, strictly, principally, equally

Figure 3: Four pairs of examples, showing the top 50 most similar words by PMI and PMI_max

specifically for use in relation to the target noun. In the definitions of these nouns, the target noun is typically used. For example, “chauffeur” is defined in WordNet as “a man paid to drive a privately owned car”. The advantage of these nouns in computing PMI similarity comes from the fact that they seldom have their own contexts which are not subsumed by the contexts of the target noun. PMI similarity is context-based, which means that the similarity is not solely relied on “IS-A” relation but an overall effect of all kinds of relations embodied by the contexts in a corpus.

The examples show that PMI and PMI$_{max}$ capture almost the same kind of semantic relations and many of the words in their top 50 lists are the same. Their key difference is how they rank low frequency and high frequency words. PMI is predisposed towards low frequency words and PMI$_{max}$ alleviates this bias. For example, the topmost candidates for “twist”, “ridiculous” and “commonly” produced by PMI are “contort”, “nonsensical” and “supposedly” respectively, which are less common than the corresponding words generated by PMI$_{max}$, “writhe”, “absurd” and “generally”, although they have close meanings. The overall adjustment made by PMI$_{max}$ is that low frequency words move down the list and high frequency words move up with the constraint that more similar words are still ranked higher. Low frequency words at the top of the PMI list are often good synonyms. Though moved down, they typically remain in the PMI$_{max}$ list. Low frequency words which are more lowly ranked tend to be not very similar to the target; in the PMI$_{max}$ list, they are replaced with higher frequency words.

In the example of “twist”, high frequency words (e.g., “round”, “wind”, “spin”, “fasten”, “jerk”, “weave”, “curve”, “double” and “bend”) move in the list to replace low frequency words including noisy words (e.g., “blotch” and “lacquer”), less similar words (e.g., “groove” and “lunge”), and less commonly used similar words (e.g., “flex” and “crochet”). In the example of “car”, two important similar words “vehicle” and “wheel” appear in the PMI$_{max}$ list after the adjustment.

Because PMI$_{max}$ estimates semantic similarity between the closest senses of two words, it has an advantage over PMI in discovering polysemous synonyms or similar words. As examples, “double” has a sense of *bend* and PMI$_{max}$ find it similar to “twist”; “platform” can mean vehicle carrying weapons and PMI$_{max}$ find it similar to “car”; “pathetic” has a sense of *inspiring scornful pity* and PMI$_{max}$ find it similar to “ridiculous”.

The frequency counts of the verb “double”, noun “platform” and adjective “pathetic” in our Gutenberg corpus are 32385, 60114 and 28202 and their assumed senses (according to Equation 5 and Table 5) are 6.5, 4.5 and 6.0 respectively. They are ranked at 106th , 64th and 108th places in their PMI lists. They are able to move up in the PMI$_{max}$ lists because they have more assumed senses than many words in the PMI lists. Here we zoom in on a concrete example that compares a moving-out word “blotch” to a moving-in word “double”. “Blotch” has a frequency count 1149 and co-occurs with “twist” 18 times, producing a PMI value 6.35. “Double” has a lower PMI value, 5.98, and co-occurs with “twist” 349 times. However, since “blotch” only has 1.5 assumed senses its PMI$_{max}$ value is 8.71, which is smaller than the PMI$_{max}$ value between “double” and “twist”, 9.75.

The PMI$_{max}$ list for the adverb “commonly” has some words that are often seen in a stop words list, such as “more”, “less”, “hence”, “also”, “either”, “very” and “however”. These words have very high frequencies but few senses. PMI$_{max}$ erroneously judges them similar to “commonly” because their high frequency mistakenly suggests many senses.

3.5 Comparison to Distributional Similarity

To demonstrate the efficacy of PMI$_{max}$ in automatic thesaurus generation, we compare it with a state-of-the-art distributional similarity measure proposed by Bullinaria and Levy [14]. Their method achieved the best performance after a series of work on distributional similarity from their group [39, 40, 41]. The method is named Positive PMI components and Cosine distances (PPMIC) because it uses positive pointwise mutual information to weight the components in the context vectors and standard cosine to measure similarity between vectors. Bullinaria and Levy demonstrated that PPMIC was remarkably effective on a range of semantic and syntactic tasks, achieving, for example, an accuracy of 85% on TOEFL synonym test using the BNC corpus. Using PMI as the weighting function, and cosine as the similarity function is a popular choice for measuring distributional similarity [30]. What makes PPMIC different is a minimal context window size (±1 word window) and the use of a high dimension context vector that does not remove of low frequency components. Bullinaria and Levy found that these uncommon settings are essential to make PPMIC work extremely well, though they are not generally good choices for other similarity measures.

Our PPMIC implementation differs from Bullinaria and Levy’s in using lemmatized and POS-tagged words (noun, verb, adjectives and adverbs) rather than unprocessed words as vector dimension. This variation is simply due to convenience of reusing what we already have. We tested our implementation of PPMIC on TOEFL synonym test and obtained a score of 80%. The slightly lower performance may result from the variation we made or the use of the outdated Gutenberg corpus to answer questions about modern English. Nevertheless, 80% is a very good score on TOEFL synonym

	noun	verb	adj.	adv.
PMI	0.133	0.148	0.161	0.155
PMI$_{max}$	0.173	0.219	0.242	0.224
PPMIC	0.283	0.255	0.276	0.374

Table 6: Top-1 Precision of PMI, PMI$_{max}$ and PPMIC

	noun	verb	adj.	adv.
PMI	0.120	0.160	0.163	0.103
PMI$_{max}$	0.168	0.256	0.261	0.179
PPMIC	0.433	0.442	0.436	0.487

Table 7: MAP values of PMI, PMI$_{max}$ and PPMIC

test. As an example, Bullinaria and Levy's previous best result, before inventing PPMIC, was 75% [41].

The performance of PPMIC in the automatic thesaurus generation is shown in Table 4 and Figure 2. As we did for PMI and PMI$_{max}$, we exclude words with frequency less than 700 or with different POS from the target word in the candidate list. Unlike PMI and PMI$_{max}$, PPMIC has very good performance on nouns, which is even slightly better than verbs and adjectives. Unsurprisingly, PPMIC outperforms PMI on almost all the recall values and average ranks. The improvements on the recall values are statistically significant ($p < 0.001$ two-tailed paired t-test). When comparing with PMI$_{max}$, PPMIC has obvious advantage on nouns but just competing performance on other POS categories. Although PPMIC leads PMI$_{max}$ on recall values for the small top n lists, such as top-10, PMI$_{max}$ can generally catch up quickly and outrun PPMIC for the subsequent longer lists. The same trend can also be observed on average ranks depicted in Figure 2. When considering all the recall values in Table 3 and Table4, PMI$_{max}$ has a significantly better performance than PPMIC ($p < 0.01$ two-tailed paired t-test). Compared with PMI$_{max}$, PPMIC seems to be able to rank a portion of synonyms very highly but it fails to give high scores to the remaining synonyms.

3.6 Mean Average Precision Evaluation

Mean Average Precision (MAP) is a common measure used to evaluate systems in information retrieval (IR) tasks. Automatic thesaurus generation can be evaluated as an IR task if we make an analogy between an IR query and the need to identify synonyms for a target word (in this analogy correct synonyms are the relevant documents). We compare PMI, PMI$_{max}$ and PPMIC using MAP in Table 6. PMI$_{max}$ is significantly better than PMI ($p < 0.01$ two-tailed paired t-test). PPMIC numerically outperforms PMI$_{max}$ but the improvements are not statistically significant ($p > 0.05$ two-tailed paired t-test). A higher average precision does not necessarily entail a lower average rank. For example, suppose system A ranks three synonyms of a target word at 1st, 9th and 10th places and system B ranks them at 3rd, 5th and 6th places. A's average precision of 0.51 is higher than B's 0.41, but B's average rank is 4.67, which is smaller than A's, 6.67.

PPMIC's higher MAP value results from its much better performance in placing synonyms at the very top ranks. The top-1 precision of PMI, PMI$_{max}$ and PPMIC are supplied in Table 7. PPMIC has excellent precision, considering our gold standard only provides a subset of synonyms. However, Tables 6 and 7 again imply that PPMIC's performance degrades faster than that of PMI$_{max}$ in finding more synonyms. Typically lexicographers desire high recall, therefore, the higher MAP scores for PPMIC do not necessarily make it a more compelling approach for lexicography than PMI$_{max}$.

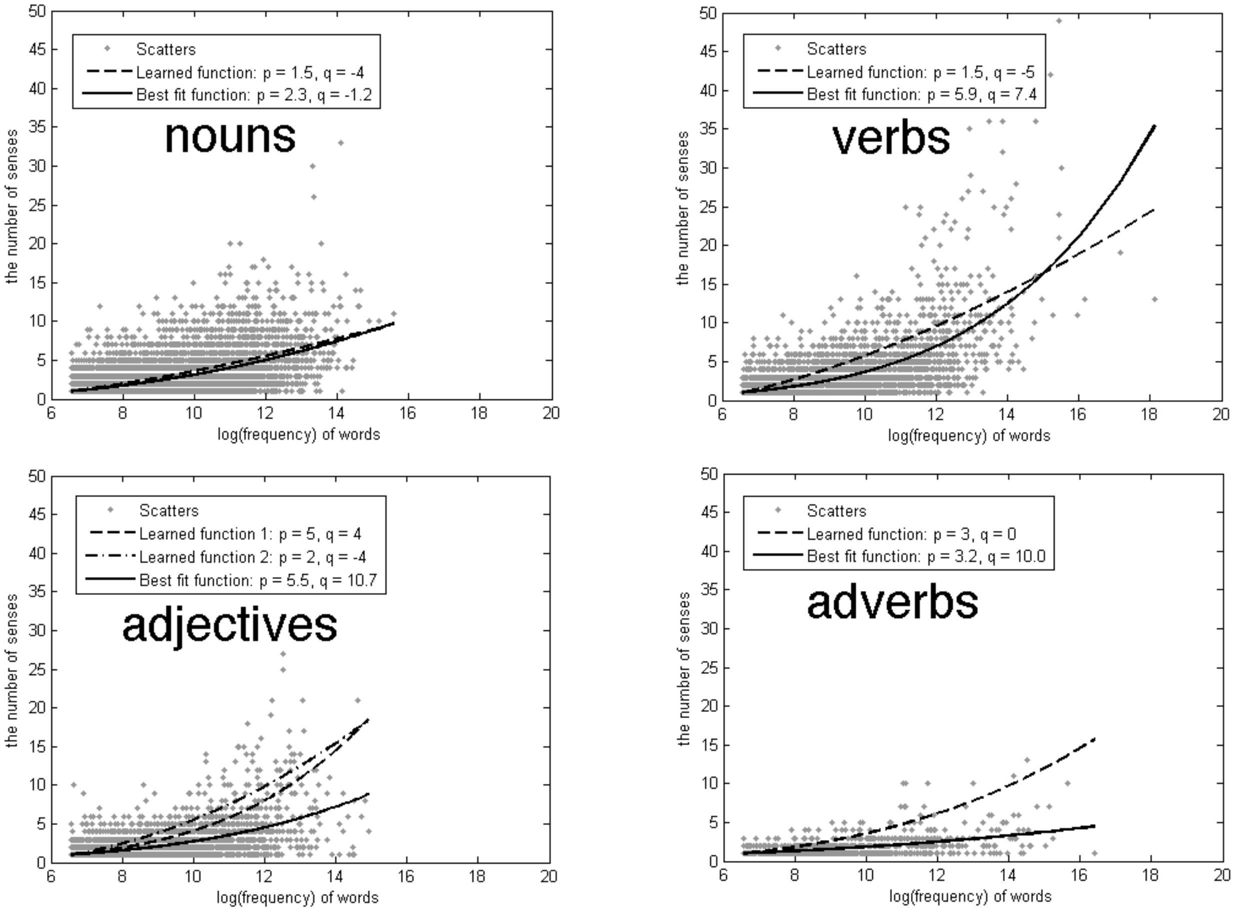

Figure 4: The frequency and the number of senses for nouns, verbs, adjectives and adverbs.

3.7 Examining the Assumptions in PMI$_{max}$

The key assumption made for PMI$_{max}$ is found in Equation 5. In Section 3.4 we determined coefficients by maximizing the performance of PMI$_{max}$ for automatic thesaurus generation. Note that the function is learned using an automated statistical approach. We will examine this function by comparing it with knowledge from human judgments extracted from WordNet.

Since we learned a different combination of coefficients for each POS category, we examine them separately. For each POS-tagged word with frequency above 700 in our corpus, we get its number of senses within its POS from WordNet and group them into noun, verb, adjective, and adverb categories, resulting in 8467 nouns, 3705 verbs, 3763 adjectives and 1095 adverbs. We generated four scatter plots using the natural logarithm of the frequency of a POS-tagged word as x-axis and its number of senses as y-axis. Matlab's non-linear least squares problem solver (LSQCURVEFIT) was used to find coefficients p, q that best fit Equation 5 to the scatter plots, producing the results in Figure 4. For nouns, the automatically learned function is almost identical to the best-fit function and for verbs they are quite close. For adjectives and adverbs, both the learned functions have steeper slope than the corresponding best fit functions. The noun class probably enjoys the best match because the assumptions used in deriving PMI$_{max}$ work best for nouns and worst for adjectives and adverbs.

The closeness between the learned functions and the best-fit functions suggests that fitting Equation 5 could be an alternative way to determine the coefficients of PMI$_{max}$. To see how this approach performs, we also give its recall values in the six top n lists for the "all entries" scenario in Table 8.

		T10	T25	T50	T100	T200	T500	Avg
all entries	noun	0.29	0.44	0.56	0.66	0.75	0.85	0.59
	verb	0.33	0.46	0.57	0.67	0.77	0.88	0.61
	adj.	0.37	0.52	0.63	0.73	0.81	0.89	0.66
	adv.	0.43	0.64	0.71	0.83	0.86	0.96	0.74

Table 8: Recall for PMI_{max} using coefficients in best-fit functions

Nouns and verbs have almost the same performance as in Table 3. Even adjectives and adverbs have close performance to their counterparts in Table 3. It shows that for adjectives and adverbs, most of the performance gain is achieved by changing the horizontal line $y = 1$ (i.e., the "single sense" assumption) to the places of the best-fit functions.

In deriving PMI_{max} in Section 2.2, we implicitly assumed that different senses of a word are not similar. However, it has been argued that WordNet's senses are too fine-grained [42], that is, different WordNet senses of the same word can be similar or highly correlated. According to this, the learned functions should have a lower slope than the best-fit functions, which is inconsistent with our results.

This is probably because our frequency-sense model is too simple and something is not very accurately modeled. Another possibility is that some other factors, which are irrelevant to senses, also contribute to the frequency bias of PMI. To counteract that part of bias, PMI_{max} could yield a steeper slope than what word polysemy requires. Other causes may possibly exist, such as different sets of words used in the automatic thesaurus generation experiment and in creating the plots, or missing senses in WordNet.

3.8 Low Frequency Words Evaluation

We learned coefficients for PMI_{max} using the occurrence frequency thresholds $10,000$ and 700 for target words and their synonyms. Now we would like to check if PMI_{max} learned in this way could produce consistently better results than PMI for low frequency target words. To this end, we extracted Putnam entries whose head words have a frequency between 700 and 2,000, obtaining 238, 103 and 155 entries and 598, 282 and 449 synonym pairs for nouns, verbs and adjectives, respectively. The recall values at six top n lists ("all entries" scenario) for PMI, PMI_{max} and PPMIC are shown in Table 9. Interestingly, PPMIC's performance for low frequency words is even better than its performance for high frequency words, as compared to Table 4. More data typically leads to more accurate results in statistical approaches. PPMIC's unusual result implies that its performance is also affected by the degree of word polysemy. The case of low frequency words is easier for PPMIC because they only have a few senses.

On the contrary, PMI and PMI_{max} have degraded performances compared to Tables 2 and 3 due to insufficient data for them to work reasonably. In our experiment, the expected co-occurrence between two strong synonyms (PMI value 8.0) with the frequencies 700 and 1,000 is only *one*. Thus, many low frequency synonyms lack even a single co-occurrence with their targets while many noisy, low frequency words are ranked at top places by chance. PMI_{max} produces consistently better results than PMI ($p < 0.001$ two-tailed paired t-test) but the improvement rate is generally lowered, especially for large top n lists. We see three reasons for this: some low frequency synonyms cannot be found regardless of the length of the ranked list because they have no co-occurrences with the target, PMI_{max} ranks noisy words downward but does not remove them in the relatively long lists, and the coefficients are not optimized for low frequency target words.

		T10	T25	T50	T100	T200	T500	Avg
PMI	noun	0.26	0.36	0.45	0.58	0.66	0.78	0.52
	verb	0.19	0.28	0.41	0.53	0.62	0.75	0.46
	adj.	0.23	0.35	0.42	0.57	0.74	0.87	0.53
PMI$_{max}$	noun	0.32	0.44	0.53	0.63	0.70	0.80	0.57
	verb	0.30	0.40	0.46	0.55	0.64	0.75	0.52
	adj.	0.31	0.42	0.54	0.65	0.74	0.87	0.59
PPMIC	noun	0.48	0.61	0.67	0.74	0.81	0.89	0.70
	verb	0.42	0.53	0.62	0.66	0.72	0.84	0.63
	adj.	0.43	0.56	0.66	0.74	0.81	0.88	0.68

Table 9: Recall for Low Frequency Words

4 Benchmark Evaluation

In the preceding section, we demonstrated the effectiveness of PMI$_{max}$ for automatic thesaurus generation. In this section, we will evaluate PMI$_{max}$, tuned for the automatic thesaurus generation task, on common benchmark datasets including the Miller-Charles (MC) dataset [46] and the TOEFL synonym test [25], allowing us to compare our results with previous systems. In these datasets, there are words which occur less often in the Gutenberg corpus than our frequency cutoff of 700. For these words, we assume a single sense and apply Equation 4 to compute PMI$_{max}$.

4.1 Human Similarity Ratings

The widely used MC dataset [8, 10, 9, 43, 22, 44, 45] consists of 30 pairs of nouns rated by a group of 38 human subjects. Each subject rates "similarity of meaning" for each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The mean of the individual ratings for each pair is used as its semantic similarity. The MC dataset was taken from Rubenstein-Goodenough's [47] original data for 65 word pairs by selecting ten pairs of high, intermediate, and low levels of similarity respectively. Although the MC experiment was performed 25 years after Rubenstein-Goodenough's, the Pearson correlation coefficient between the ratings is 0.97. Four years later, the same experiment was replicated again by Resnik [8] with ten subjects. Their mean ratings also had a very high correlation coefficient of 0.96. Resnik also showed that the individual ratings in his experiment have an average correlation coefficient 0.8848 with the MC mean ratings.

Due to the absence of the word "woodland" in earlier versions of WordNet, only 28 pairs are actually adopted by most researchers. Our PMI implementation has a similar problem in that "implement" has very low frequency (117) in the Gutenberg corpus so that there are no co-occurrences for "implement" and "crane". Since their PMI value is undefined, only 29 pairs can be used for our experiment. For these, the correlation coefficient between the MC and PMI ratings is 0.794, and for MC and PMI$_{max}$ is 0.852. However, in order to compare with other people's results, it is necessary to have an equal setting. Fortunately, the published papers of most previous researchers included lists of ratings for each pair, allowing the comparison of our work over 27 pairs as shown in Table 10.

The six previous measures in Table 10 give the best results that we can find in the literature on the MC dataset. Among them, Resnik [8], Jiang and Conrath [10], Lin [9], and Li et al [43] are WordNet-based approaches while CODC [44] and WebSVM [45] are Web-based, making use of page counts and snippets. PMI$_{max}$ is the first corpus-based approach that enters the same performance level as other best approaches with a score 0.856, a 7.5% improvement over basic PMI. Although even basic PMI has a decent performance 0.796, PPMIC only obtains a correlation 0.654 which does not match its performance in automatic thesaurus generation or TOEFL synonym test. However this is not new. In

Word Pair	Miller-Charles	Resnik [8]	J & C [10]	Lin [9]	Li et al [43]	CODC [44]	WebSVM [45]	PMI	PMI$_{max}$	PPMIC
car-automobile	3.92	8.0411	30	1	1	0.4229	0.98	7.570	10.498	0.392
gem-jewel	3.84	14.929	30	1	1	0.353	0.686	7.985	10.778	0.447
journey-voyage	3.84	6.7537	27.497	0.89	0.779	0.2666	0.996	5.336	8.567	0.477
boy-lad	3.76	8.424	25.839	0.85	0.778	0.2828	0.974	5.581	9.168	0.527
coast-shore	3.7	10.808	28.702	0.93	0.779	0.2923	0.945	6.606	10.039	0.511
asylum-madhouse	3.61	15.666	28.138	0.97	0.779	0.1845	0.773	8.016	9.614	0.102
magician-wizard	3.5	13.666	30	1	0.999	0.2076	1	8.008	10.242	0.295
midday-noon	3.42	12.393	30	1	1	0.2994	0.819	6.417	9.041	0.301
furnace-stove	3.11	1.7135	17.792	0.18	0.585	0.1982	0.889	6.935	9.585	0.310
food-fruit	3.08	5.0076	23.775	0.24	0.17	0.2355	0.998	5.970	9.388	0.337
bird-cock	3.05	9.3139	26.303	0.83	0.779	0.2295	0.593	6.567	9.607	0.287
bird-crane	2.97	9.3139	24.452	0.67	0.472	0	0.879	7.119	9.853	0.245
implement-tool	2.95	6.0787	29.311	0.8	0.778	0.2506	0.684	8.689	10.128	0.070
brother-monk	2.82	2.9683	19.969	0.16	0.779	0.1956	0.377	4.824	7.980	0.250
brother-lad	1.66	2.9355	20.326	0.2	0.355	0.1811	0.344	4.421	7.613	0.256
car-journey	1.16	0	17.649	0	0	0.2049	0.286	4.969	8.201	0.124
monk-oracle	1.1	2.9683	18.611	0.14	0.168	0	0.328	4.639	7.038	0.151
food-rooster	0.89	1.0105	17.657	0.04	0	0	0.06	4.721	7.019	0.064
coast-hill	0.87	6.2344	25.461	0.58	0.366	0	0.874	5.198	8.545	0.362
forest-graveyard	0.84	0	14.52	0	0.132	0	0.547	4.859	7.337	0.209
monk-slave	0.55	2.9683	20.887	0.18	0.35	0	0.375	3.798	6.026	0.238
coast-forest	0.42	0	15.538	0.16	0.17	0.1686	0.405	5.100	8.352	0.296
lad-wizard	0.42	2.9683	20.717	0.2	0.355	0	0.22	4.275	6.521	0.124
chord-smile	0.13	2.3544	17.535	0.2	0	0	0	4.436	7.012	0.178
glass-magician	0.11	1.0105	17.098	0.06	0	0	0.18	4.894	7.632	0.075
noon-string	0.08	0	12.987	0	0	0	0.018	3.757	5.742	0.081
rooster-voyage	0.08	0	12.506	0	0	0	0.017	3.679	4.919	0.044
Correlation	**1**	**0.791**	**0.836**	**0.834**	**0.883**	**0.837**	**0.847**	**0.796**	**0.856**	**0.654**

Table 10: Comparisons of PMI$_{max}$ with PMI, PPMIC and previous measures on the MC dataset

an intensive study of distributional similarity, Weeds [22] applied ten different distributional measures, based on the BNC corpus, to the MC dataset with a top correlation coefficient of 0.62. The value of distributional similarity appears to vary significantly with the target words selected. Although it can generate a properly ranked candidate list, the absolute similarity value is not consistent across different target words selected.

We compare PMI$_{max}$ with basic PMI on two other standard datasets: Rubenstein-Goodenough (RG) [47] and WordSim353 [48]. Among 65 word pairs in the RG dataset we removed five with undefined PMI and PMI$_{max}$ values, and performed the experiment on the rest. WordSim353 contains 353 word pairs, each of which is scored by 13 to 16 human subjects on a scale from 0 to 10. We removed proper nouns, such as "FBI" and "Arafat", because they are not supported by our current implementation of PMI and PMI$_{max}$. We also imposed a minimum frequency threshold of 200 to WordSim353 because it contains many modern words such as "seafood" and "Internet" which occur infrequently in our corpus. Choosing 200 as the threshold is a compromise between keeping most of the word pairs available and making PMI and PMI$_{max}$ perform reliably. After all the preprocessing, 289 word pairs remains. The comparison results of PMI, PMI$_{max}$ and PPMIC on the two datasets along with the MC dataset are shown in Table 11. Both the Pearson correlation and Spearman rank correlation coefficients are investigated.

While PMI$_{max}$'s Pearson correlation on the RG dataset is lower than its on the MC dataset, its Spearman rank correlation on the RG dataset is higher. WordSim353 is a more challenging dataset than MC and RG. Our results, though lower than those on MC and RG, are still very good [48]. The improvements of PMI$_{max}$ over basic PMI using either Pearson correlation or Spearman rank correlation

	MC(27)	RG(60)	WS353(289)
PMI (Pearson)	0.796	0.791	0.570
PMI$_{max}$ (Pearson)	0.856	0.818	0.625
PPMIC (Pearson)	0.654	0.707	0.381
PMI (Spearman)	0.784	0.790	0.635
PMI$_{max}$ (Spearman)	0.839	0.844	0.666
PPMIC (Spearman)	0.703	0.705	0.353

Table 11: Comparing PMI$_{max}$ with PMI on three datasets

are all statistically significant ($p < 0.05$ with two-tailed paired t-test). Both PMI and PMI$_{max}$ have a consistently higher performance than PPMIC on the three datasets.

4.2 TOEFL Synonym Questions

Eighty synonym questions were taken from the Test of English as a Foreign Language (TOEFL). Each is a multiple choice synonym judgment, where the task is to select from four candidates the one having the closest meaning to the question word. Accuracy, which is the percentage of correctly answered questions, is used to evaluate performance. The average score of foreign students applying to US colleges from non-English speaking countries was 64.5% [12].

This TOEFL synonym dataset is widely used as a benchmark for comparing the performance of computational similarity measures [12, 25, 41, 13, 49, 21, 14]. Currently, the best results from corpus-based approaches are achieved by LSA [49], PPMIC [14] and PMI-IR [13], with scores of 92%, 85% and 81.25%, respectively.

Since the words used in our implementations of PMI, PMI$_{max}$ and PPMIC are POS tagged, we first assign a POS tag to each TOEFL synonym question by choosing the common POS of the question and candidate words. The results on the TOEFL synonym test for PMI, PMI$_{max}$, and PPMIC are 72.5%, 76.25% and 80%, respectively. Although the Gutenberg corpus has two billion words, it is still small compared with Web collections used by PMI-IR. Thus, unlike PMI-IR, data sparseness is still a problem limiting the performance of PMI$_{max}$. For example, there are three "no answer" questions[9] for which the question word has no co-occurrence with any of four candidate words. It is known that TOEFL synonym questions contain some infrequently used words [14]. In addition, some TOEFL words common in modern English, such as "highlight" and "outstanding", were uncommon at the time of Gutenberg corpus. 76.25% is an encouraging result because it demonstrates that PMI$_{max}$ need not rely on search engines to be effective in a difficult semantic task like the TOEFL synonym questions.

5 Discussion and Future Work

PMI$_{max}$ can be used in various applications that require word similarity measures. However, we are more interested in combining PMI$_{max}$ with distributional similarity in the area of semantic acquisition from text because this direction is not yet explored. We start our discussion by looking at an example, the top 50 most similar words for the noun "car" generated by PPMIC.

> train, automobile, boat, wagon, carriage, engine, vehicle, motor, truck, coach, cab, wheel, ship, machine, cart, locomotive, chariot, canoe, vessel, craft, horse, bus, auto, driver, sleigh, gun, launch, taxi, buggy, barge, yacht, ambulance, passenger, freight, box, round, plane, trolley, station, team, street, track, window, rider, chair, mule, elevator, bicycle, door, shaft

[9] They are treated as incorrectly answered questions.

The example shows that, as a state-of-the-art distributional similarity, PPMIC has an amazing ability to find the concepts that are functionality or utility similar to “car”. These concepts, such as “train”, “boat”, “carriage”, “vehicle”, are typically neighboring concepts of “car” in a taxonomy structure such as WordNet. In contrast, PMI$_{max}$, as illustrated in the “car” example in Figure 3, can only find synonymous concepts and “siblings” concepts (e.g. “train” and “truck”) but miss the “cousin” concepts (e.g. “boat” and “carriage”). This is because PMI similarity measures the degree that two concepts tend to co-occur over the *exactly same* contexts. “boat” and “carriage” are missing because only a small proportion of contexts of “car” relates it to watercraft or archaic vehicles. Seemingly not as powerful as distributional similarity, PMI similarity provides a very useful complement to it.

There are many potential applications. Below we suggest two directions related to automatic thesaurus or ontology generation. First, we could devise an improved approach for high-precision automatic thesaurus generation by taking intersection of the candidate lists generated by PMI$_{max}$ and a state-of-the-art distributional similarity, such as PPMIC. For example, the intersection of two top-50 candidate lists generated by PMI$_{max}$ and PPMIC for “car” includes:

> train, automobile, engine, vehicle, motor, truck, wheel, locomotive, auto, driver, taxi, ambulance, passenger, freight, trolley, station, track, elevator, shaft

Many inappropriate distributional similar terms like “boat” and correlational similar terms like “chauffeur” are filtered out by the intersection. This makes synonyms, for example “auto”, have higher ranks in the resulting candidate list than either of the parent lists.

The second application is more interesting and challenging. We know that distributional similarity can find neighboring concepts of the target word in a taxonomy. A subsequent question on the course is how we could classify these concepts into different clusters corresponding to the “sibling”, “parent” and “cousin” sets in a taxonomy. This is an largely unsolved problem in ontology learning from text that the combination of PMI$_{max}$ and distributional similarity can help address.

For example, of the 50 most distributional similar words of “car”, which are most likely to be classified together with “boat”? A simple approach is to take the intersection of two top-50 candidate lists generated by PPMIC and PMI$_{max}$ for “car” and “boat” respectively[10]. The results for “boat” and several other examples obtained this way are shown in Figure 5. The words that can be classified with “boat” include its “sibling” concepts (conveyances on water) and “parent” concepts (vessel, craft) but no “cousin” concepts (conveyances on land). Similarly, the intersection list for “carriage” includes archaic vehicles and rejects modern ones. Although in the example of “chariot”, “car” appears in the list, it is due to a different sense of “car”[11]. This shows that polysemy of words can add even more complexity to this problem. Regarding to identifying the “parent” set, common words can be good cues. For example, “boat” and “ship” have the common word “vessel” whereas “carriage”, “chariot”, and “truck” share the words “vehicle” and “wheel”. This suggests that “boat” and “ship” may have parent “vessel” while “carriage”, “chariot”, and “truck” may have parent “vehicle” or “wheel”. This also implies that in the time of Gutenberg corpus, about eighty years ago, “vehicle” cannot be used to describe “boat” or “ship”. We confirm this hypothesis by checking our gold standard, Putnam’s Word Book, in which “vehicle” is shown as a synonym for “car”, “cart”, “wagon” and other conveyances on land but not for “boat” and “ship”, for which “vessel” is used instead. As another example, the word “horse” is only associated with “carriage” and “chariot”, which implies that they are historical vehicles powered by a “horse”.

[10] The target word is also included in its candidate list and the words in the intersection are ranked by their orders in the PPMIC list.

[11] The original meaning of car is similar to “chariot”, however this sense is missing in WordNet.

boat: ship, canoe, vessel, craft, launch, barge, passenger

ship: boat, vessel, passenger

carriage: train, vehicle, coach, cab, wheel, cart, horse, driver, passenger, station, street, window, door

chariot: car, vehicle, coach, wheel, horse, driver, team

truck: car, train, automobile, wagon, engine, vehicle, motor, wheel, cart, locomotive, auto, driver, ambulance, freight, trolley

Figure 5: Examples for classifying distributional similar words for "car"

6 Conclusion

In this paper, we described the characteristics of PMI as a measure of semantic similarity for words, and directly compared it with distributional similarity. We developed a new metric, PMI$_{max}$, by augmenting traditional PMI to take into account the number senses that a word has. We experimentally showed that PMI$_{max}$ outperforms PMI in automatic thesaurus generation and on benchmark datasets for human similarity ratings and TOEFL synonym questions. PMI$_{max}$ also gives, among corpus-based approaches, the highest correlation coefficient of 0.856 for the Miller-Charles dataset based on 27 pairs.

Our experiments have demonstrated that PMI need not rely on Web search engine data or an information retrieval index to be effective in a range of semantic tasks. Compared with distributional similarity, PMI is a lightweight measure, though it requires a larger corpus to be effective. With the vast amount of data available today, data sparseness, becomes a much less severe issue than twenty years ago when Church and Hanks popularized the use of PMI in computational linguistics. We anticipate that PMI and PMI$_{max}$ will play an important role in lexical semantic applications in the future.

References

[1] P. Resnik, "Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language," *Journal of Aritificial Intelligence Research*, vol. 11, pp. 95–130, 1999.

[2] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.

[3] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *Proc. 21st National Conf. on Artificial Intelligence*, Boston MA, 2006, pp. 775–780.

[4] G. Varelas, E. Voutsakis, P. Raftopoulou, E. Petrakis, and E. Milios, "Semantic similarity methods in wordnet and their application to information retrieval on the web," in *Proc. ACM Workshop on Web Information and Data Management*, Bremen, Germany, 2005.

[5] I. Kaur and A. J. Hornof, "A comparison of LSA, wordnet and PMI-IR for predicting user click behavior," in *Proc. Human Factors in Computing Systems Conf.* ACM Press, 2005, pp. 51–60.

[6] G. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, p. 41, 1995.

[7] S. Mohammad and G. Hirst, "Distributional measures of concept-distance: A task oriented evaluation," in *Proc. Conf. on Empirical Methods in Natural Language Processing*, 2006, pp. 35–43.

[8] P. Resnik, "Using information content to evaluate semantic similarity," in *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, 1995.

[9] D. Lin, "An information-theoretic definition of similarity," in *Proc. Int. Conf. on Machine Learning*, 1998.

[10] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. Int. Conf. on Research in Computational Linguistics*, 1997.

[11] M. Jarmasz and S. Szpakowicz, "Roget's thesaurus and semantic similarity," in *Proc. Int. Conf. on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, 2003, pp. 212–219.

[12] T. Landauer and S. Dumais, "A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge," in *Psychological Review, 104*, 1997, pp. 211–240.

[13] E. Terra and C. L. A. Clarke, "Frequency estimates for statistical word similarity measures," in *Proc. Human Language Technology and North American Chapter of the ACL Conf.*, 2003, pp. 244–251.

[14] J. Bullinaria and J. Levy, "Extracting semantic representations from word cooccurrence statistics: A computational study," *Behavior Research Methods*, vol. 39, no. 3, pp. 510–526, 2007.

[15] D. Hindle, "Noun classification from predicate-argument structures," in *Proc. Annual Meeting of the ACL*, Pittsburg PA, 1990, pp. 268–275.

[16] G. Grefenstette, *Explorations in Automatic Thesaurus Discovery.* Boston, USA: Kluwer Academic Publishers, 1994.

[17] D. Lin, "Automatic retrieval and clustering of similar words," in *Proc. 17th Int. Conf. on Computational Linguistics*, 1998, pp. 768–774.

[18] J. R. Curran and M. Moens, "Improvements in automatic thesaurus extraction," in *Proc. Workshop on Unsupervised Lexical Acquisition*, Philadelphia, PA, USA, 2002, pp. 59–66.

[19] D. Yang and D. M. Powers, "Automatic thesaurus construction," in *Proc. 31st Australasian Conf. on Computer Science*, vol. 74, 2008, pp. 147–156.

[20] Z. Harris, *Mathematical Structures of Language.* New York, USA: Wiley, 1968.

[21] S. Pado and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.

[22] J. E. Weeds, "Measures and applications of lexical distributional similarity," Ph.D. dissertation, University of Sussex, 2003.

[23] K. Church and P. Hanks, "Word association norms, mutual information and lexicography," in *Proc. 27th Annual Conf. of the ACL*, 1989, pp. 76–83.

[24] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing.* Cambridge, US: MIT Press, 1999.

[25] P. Turney, “Mining the web for synonyms: PMI-IR versus LSA on TOEFL,” in *Proc. 12th European Conf. on Machine Learning*, 2001, pp. 491–502.

[26] D. Higgins, “Which statistics reflect semantics? rethinking synonymy and word similarity,” in *Proc. Int. Conf. on Linguistic Evidence*, Tübingen, Germany, 2004, pp. 265–284.

[27] N. Kaji and M. Kitsuregawa, “Using hidden markov random fields to combine distributional and pattern-based word clustering,” in *Proc. of the 22nd Int. Conf. on Computational Linguistics*, 2008, pp. 401–408.

[28] W. Gale, K. Church, and D. Yarowsky, “One sense per discourse,” in *Proc. DARPA Speech and Natural Language Workshop*, Harriman, NY, 1992, pp. 233–237.

[29] P. Turney, M. Littman, J. Bigham, and V. Shnayder, “Combining independent modules to solve multiple-choice synonym and analogy problems,” in *Proc. RANLP-2003*, 2003, pp. 482–489.

[30] P. Pantel and D. Lin, “Discovering word senses from text,” in *Proc. ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002, pp. 613–619.

[31] J. Weeds and D. Weir, “Finding and evaluating sets of nearest neighbours,” in *Proc. 2nd Int. Conf. on Corpus Linguistics*, Lancaster, UK, 2003.

[32] H. Wu and M. Zhou, “Synonymous collocation extraction using translation information,” in *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, 2003, pp. 120–127.

[33] T. Dunning, “Accurate methods for the statistics of surprise and coincidence,” *Computational Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.

[34] M. Hart, “Project gutenberg electronic books,” http://www.gutenberg.org/wiki/Main_Page, 1997.

[35] K. Toutanova, D. Klein, C. Manning, W. Morgan, A. Rafferty, and M. Galley, “Stanford log-linear part-of-speech tagger,” http://nlp.stanford.edu/software/tagger.shtml, 2000.

[36] F. Smadja, “Retrieving collocations from text: Xtract,” *Computational Linguistics*, vol. 19, no. 1, pp. 143–178, 1993.

[37] I. Dagan, S. Marcus, and S. Markovitch, “Contextual word similarity and estimation from sparse data,” in *Proc. ACL-93*, Columbus, Ohio, 1993, pp. 164–171.

[38] L. A. Flemming, “Putnam’s Word Book,” http://www.gutenberg.org/files/13188/13188-8.txt, 1913.

[39] M. Patel, J. A. Bullinaria, and J. P. Levy, “Extracting semantic representations from large text corpora,” in *Fourth Neural Computation and Psychology Workshop: Connectionist Representations.* London: Springer, 1997, pp. 199–212.

[40] J. P. Levy, J. A. Bullinaria, and M. Patel, “Explorations in the derivation of semantic representations from word co-occurrence statistics,” *South Pacific Journal of Psychology*, vol. 10, pp. 99–111, 1998.

[41] J. P. Levy and J. A. Bullinaria, “Learning lexical properties from word usage patterns: Which context words should be used?” in *Sixth Neural Computation and Psychology Workshop: Connectionist Models of Learning, Development and Evolution.* London: Springer, 2001, pp. 273–282.

[42] R. Navigli, “Meaningful clustering of senses helps boost word sense disambiguation performance,” in *Proc. 21th Int. Conf. on Computational Linguistics*, Sydney, Australia., 2006, pp. 105–112.

[43] Y. Li, Z. Bandar, and D. McLean, “An approach for measuring semantic similarity between words using multiple information sources,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 871–882, 2003.

[44] H. Chen, M. Lin, and Y. Wei, “Novel association measures using web search with double checking,” in *Proc. COLING/ACL 2006*, 2006, pp. 1009–1016.

[45] D. Bollegala, Y. Matsuo, and M. Ishizuka, “Measuring semantic similarity between words using web search engines,” in *Proc. WWW*, 2007.

[46] G. Miller and W. Charles, “Contextual correlates of semantic similarity,” *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.

[47] H. Rubenstein and J. Goodenough, “Contextual correlates of synonymy,” *CACM*, vol. 8, no. 10, pp. 627–633, 1965.

[48] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, “Placing search in context: The concept revisited,” *ACM Transactions on Information Systems*, vol. 20, no. 1, pp. 116–131, 2002.

[49] R. Rapp, “Word sense discovery based on sense descriptor dissimilarity,” in *Proc. 9th Machine Translation Summit*, 2003, pp. 315–322.