Automated Thesaurus Generation

Jay Liu
jayliu@cs.utexas.edu

December 12, 2008

Abstract

In this paper, we explore how to generate groups of closely related words that imitate entries in a manually created thesaurus. This is done using words from a number of different sources. The quality of the generated list as compared to the thesaurus is judged using a variety of metrics.

1 Introduction

Most of the words in the English language may have different meanings depending on the contexts in which they are used. The goal of word sense disambiguation in natural language processing is to clearly identify these different usages of the same word to improve understanding of the language. Once a word has all its meanings identified, it is helpful to associate it with other words that share one or more of its senses to create a group. Such a group of words with similar meanings are herein referred to as a cluster. A word may belong to multiple clusters, since it may share different meanings with multiple sets of words. For example, the word "mouse" could refer to an animal, and therefore is related to words such as "cat" and "dog", while in the context of computing, it most likely imply a different meaning, and hence may relate with the words "printer" and "keyboard" [Dorow and Widdows, 2003].

A word cluster has multiple interesting applications. One such application is in information retrieval [Chen et al., 1995, Park and Choi, 1996]. In such a system, given a search term, additional terms with similar meanings can be used to augment the search result. It can also be used for concept discovery, since such a generated cluster may surface additional relationships between words that were previously unidentified.

Such organization of words into groups is also the basic idea behind a language thesaurus, such as *Roget's Thesaurus* for the English language. A thesaurus contains manual clustering of words that share similar meanings or are used in similar contexts. Thesaurus is not only a useful resource to help understand related words and phrases, but it can also catalog correlation between words across multiple domains that were previously not obvious. An inclusive thesaurus touches on many subjects, and the creation of such work requires knowledge across many domains. Doing so manually requires domain experts and therefore can be expensive. An automated generation process should allow a thesaurus to be kept accurate and up to date with less human intervention.

Roget's Thesaurus is one of the most popular English thesauri in use. Its content is commonly accepted and trusted. This paper attempts to use current word clustering techniques and observe how well their results mimic the entries of a real thesaurus. The results suggest that an acceptable degree of similarities can be achieved. However, there exist practical limitations that require further work in this field..

2 Algorithm and Data

2.1 Thesaurus Structure

To find clusters that mimic *Roget's Thesaurus*, it is useful to understand its structure. *Roget* contains a hierarchical view of concepts, from broad ones to specific ones. The list of included concepts certainly do not contain all the possible concepts that may exist. Rather, it reflects a careful selection process that chooses the most interesting concepts for a typical reader for inclusion. [Kilgarriff and Yallop, 2000] explores this deliberate organization and its implication for automated thesaurus generation. In particular, it points out the "looseness" of the words in these concepts, referring to many words that share remote similarity but are nevertheless grouped closely together in the thesaurus. Such looseness, also observed by [Tokunaga et al., 1995], creates a challenge in automated generation, since the performance of such systems typically rely on the tightness of the clusters. A quick glance at [Roget, 1991] gives one such example: the words "stand" and "obtain" are grouped under the "existence" concept with "be".

For the experiments, the target thesaurus is [Roget, 1991] (hereafter referred to as ROGET). It is a revised version of the 1911 edition of the *Roget's Thesaurus*. It has been released to the public domain and is freely available through *Project Gutenburg*. [Kennedy and Szpakowicz, 2008] evaluated the relevance of using this version as opposed to the more recent 1987 edition. They were found to contain some differences in content, but after comparison test with *WordNet*, they could both be used effectively.

Additionally, [Kennedy and Szpakowicz, 2008] cataloged its eight levels of hierarchy. The hierarchy names are shown in Table 1. The smallest category, "semicolon", numbers 43,196. Since ROGET has a total of 98,924 words, it works out to be 2.3 words per such group. The next larger group, "paragraph", averages 9.7 words.

Hierarchy Name	Score
Thesaurus	0
Class	2
Section	4
Subsection	6
Head group	8
Head	10
Part of Speech	12
Paragraph	14
Semicolon	–

Table 1: ROGET hierarchy and score

2.2 Feature Triplet

[Lin, 1998] proposed a feature triplet $||w, r, w'||$ to express the existence of a relationship r between w and w'. Two words w_1 and w_2 are similar if they both share the syntactic relationship r with respect to a third word w'. Using this representation, a number of systems were created to discover word clusters from documents. A system called "clustering by committee", CBC, which tracks similarity between such triplets, was introduced in [Pantel and Lin, 2002].

Such triplets can be obtained by analyzing a grammar dependency tree, shown in Figure 1. In such a tree, the leaves are the words in a sentence, and the non-leaf nodes contain the relations between the children. MINIPAR[1] is a language parser that takes unannotated sentences as inputs, and outputs such dependency trees. These trees are then parsed to create the desired triplets.

Since the quality of the features depends on the syntactic parsing results, it is also interesting to experiment using previously manually annotated text. The Stanford Parser[2] is able to create dependency trees from annotated text. It generates a type of dependency tree called "Stanford

[1] Available at `http://www.cs.ualberta.ca/~lindek/minipar.htm`

[2] Available at `http://nlp.stanford.edu/software/lex-parser.shtml`. The output choice of "Collapsed dependencies with propagation of conjunct dependencies" is chosen, because it collapses prepositions, as well as creates additional dependencies by breaking down conjunctions.

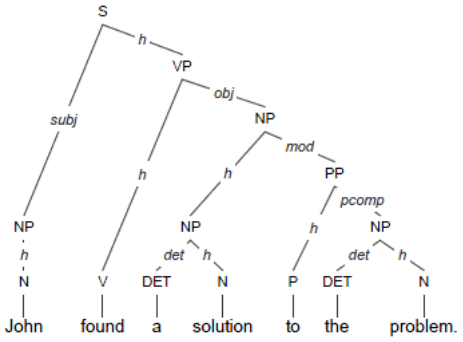

Figure 1: Dependency tree from [Pantel, 2003].

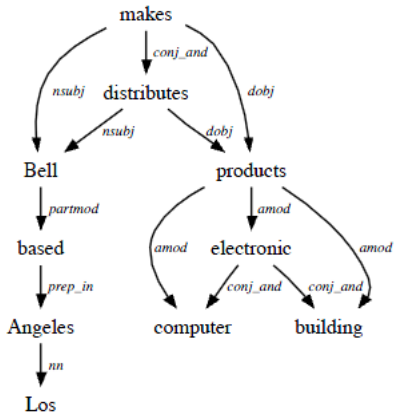

Figure 2: Stanford dependency tree from [de Marneffe and Manning, 2008].

dependency tree", an example of which is shown in Figure 2. While it looks different from Figure 1, feature triplets can just as easily be extracted.

In a triplet, the word w is to be clustered, and a concatenation of $r + w'$ is considered to be an identifying feature of the element. An imaginary example is $||elephant, (subject\ of), moves||$. We aim to group "elephant" with other words that are subjects of "moves". Trivially, the number of triplets can be doubled by performing an inverse to create $||w', r^{-1}, w||$. This would create grouping with "moves" for which "elephant" is a subject, so we create a triplet $||moves, (subject\ of)^{-1}, elephant||$.

2.3 Corpus

One source of text comes from Wikipedia[3]. Wikipedia is chosen as a corpus due to its overwhelming size, which would hopefully provide the diversity of language needed for the thesaurus. It contains a large number of articles in a wide variety of subjects, with greatly varied writing style. Due to the tremendously large size[4], only a very small subset of it is used: the first fifty-two thousand lines of its article texts (hereafter called WIKI). Many of the sentences were found to be very long, some well over 100 words. They impaired parse quality. To resolve this issue, very long sentences are either broken down into shorter phrases or discarded.

Another source of text (BOOK) is from [Tolstoy, 2005], a large fiction novel. This differs from WIKI in that most of the words revolve around a single topic, hence restricted to a narrow domain. The writing style and word usage are, however, more consistent.

The WSJ corpus from Penn Treebank is used as the annotated corpus sample. The focus of WSJ is news reporting.

Lastly, all three corpuses were combined to create a fourth corpus called COMBINED.

Table 2 summarizes the sizes of the three resulting triplet corpuses. The sizes of triplets include all the generated triplets plus their inverses. While ideally all values should be high, we desire each word to be associated with large number of features.

There are some words that are not usable by the algorithm due to their low frequency counts. Low frequency is defined as a word that appears fewer than two times, or has fewer than two distinct features. The number of unique usable words that are extracted from all the triplets range from 1 to 3%.

For comparison purpose, ROGET contains 59,768 unique entries [Kennedy and Szpakowicz, 2008]. Since the algorithm only generates clus-

[3]Available from `http://download.wikimedia.org/`. The full version without revision history is used.

[4]It yielded 48 million sentences.

	BOOK	WSJ	WIKI	COMBINED
Input triplets	202,152	984,752	3,277,316	4,464,220
Unique words	7,803	21,420	82,050	90,565
Usable unique words	5,721	18,405	58,224	66,443
Usable word per triplet	2.8%	1.9%	1.8%	1.5%
Unique features	34,162	180,525	340,879	531,046

Table 2: Corpus size details. ROGET has 25,273 unique words.

ters of single words and not phrases, it can only hope to match those in ROGET, of which there are 25,273. It should already be obvious that coverage will be an issue, as also noted by [Curran and Moens, 2002].

2.4 Clustering algorithm

The generation of the thesaurus will be attempted using an implementation[5] based on the CBC algorithm described in [Pantel, 2003]. The implementation accepts a collection of feature triplets. In brief, CBC first aggregates all the features $r + w'$ for the element w. Then, iteratively, elements that share similar features are clustered together, provided that elements in each cluster meets a minimum similarity threshold. Clusters that have high similarity and are dissimilar to existing clusters are promoted to "committee" status. The feature values of a committee are the centroids of the feature values of its constituents. Finally, all elements are compared against these committees, and are attached to the committee if a similarity threshold is exceeded.

The mutual information score, which is used to calculate how relevant a feature is for an element, is given by:

$$mi_{w_i f_j} = \log \frac{C(w_i f_j)/N}{(\sum_i C(w_i f_j)/N)(\sum_j C(w_i f_j)/N)},$$

where $C()$ is the count function. $C(w_i f_j)$ is the number of times feature f_j appears in element w_i, and N is the number of input triplets.

[5]Source code available from `http://www.cs.utexas.edu/~jayliu/projects/clustering/`.

The similarity score between two elements, used to judge how similar any two elements are based on their feature mutual information scores, is calculated using the cosine similarity measure:

$$sim_{ij} = \frac{MI_i \cdot MI_j}{||MI_i||||MI_j||}$$

where MI_i is a vector containing all the mutual information scores for the element i. Other similarity measurements could also have been used. Some of them are attempted by [Curran and Moens, 2002].

There are a number of parameters that can be tweaked to significantly change the characteristics of the output, such as cluster size and committee selection threshold. Some of the most notables regarding clusters used in the experiments are:

- θ **Mutual similarity** Describes inter-cluster similarity. A low threshold inhibits new cluster creation. High threshold risks cluster duplication.

- σ **Internal similarity** This is intra-cluster similarity. High value creates a closely-knit group, but may discourage gathering of more remotely similar elements.

- μ **Membership threshold** How similar an element must be to the cluster to be included. A low threshold allows more elements to be included in a cluster, allowing for "looseness", but may invite dissimilar items.

3 Experimental Evaluation

3.1 Evaluation Metrics

Three quality indicators are used. The first one is the F-score as a broad measure of the clustering algorithm. It is a combination of pair precision (F_{PP}) and recall (F_R).

The pair precision represents the total number of matching pairs out of all clustered pairs, and is calculated from:

$$F_{PP} = \frac{matched\ pairs}{all\ clustered\ pairs}.$$

A matching pair is a pair of words that both appear in the same "Class" hierarchy in ROGET. Recall, on the other hand, is the ratio of the number of unique elements that were involved in a match, over the total number of usable unique elements from the input:

$$F_R = \frac{matched\ elements}{usable\ elements}.$$

And finally, the F-score is expressed as:

$$F = \frac{2F_{PP}F_R}{F_{PP} + F_R}.$$

These measures were chosen because they reflect a trade-off relationship. Each two words in a cluster form a pair. Therefore, an extremely large cluster may contain numerous pairs. The precision measurement should gauge the correctness of those pairs. This prevents the algorithm from grouping all words into one extremely large cluster. To encourage clustering, the recall measure indicates how many of the words that were fed into the system were successfully clustered with another word. This gives a clear idea of how successful the algorithm was able to make use of the data. Ideally, both scores should be high. However, they reach a limit where a trade-off must be made to either include more words to help recall at the risk of lower precision, or focus on correctness of the cluster by only grouping words whose relationships are extremely certain. But since few words may have such strong relationships, the recall will suffer.

The second quality indicator is a scoring system. For each of the matched pairs, we perform a more precise calculation. Evaluation of the quality of the match relative to ROGET is based on the distance measurement of [Jarmasz and Szpakowicz, 2003]. It is inversely proportional to the smallest distance between two words in the ROGET concept hierarchy tree. The scoring is shown in Table 1. The only difference is that we do not distinguish between semicolon groups; semicolon groups usually consist of only one or two words, so we do not attempt such granularity. Words that are in the same paragraph have a distance of 0, and will be scored 14. Words that are completely dissimilar have a maximum distance of 14, and will be scored 0. Any decreases in distance is rewarded in 2 point increments. We present score as a percentage:

$$score\ \% = \frac{total\ \text{ROGET}\ score\ achieved}{14 \cdot number\ of\ pairs}.$$

The third indicator we are interested in is the coverage of ROGET. This is easily obtainable by:

$$coverage = \frac{uniquely\ matched\ words}{sizeof(\text{ROGET})}.$$

This is useful to show how much data is required to find all the words that appear in ROGET. A related concept called efficiency is used to determine the ratio of success to total input:

$$efficiency = \frac{uniquely\ matched\ words}{number\ of\ input\ features}.$$

3.2 Methodology

The feature triplets are passed to the clustering algorithm as input. The adjustable parameters that control the characteristics of the generated clusters are then set. The clustering algorithm produces a list of clusters, each of which consists of a delimited list of words.

For each cluster, the constituent elements are mutually paired, and the similarity score based on their distance in the ROGET hierarchy is stored. Elements from different clusters are not scored. The score is kept cumulatively.

The experiment chiefly investigates the effects of θ, σ, and μ. There are, however, many other parameters to the algorithm. The following compromise values were chosen to be fixed. From experimentation, they help balance between cluster growth and cluster quality.

- An element must appear at least twice to be considered for analysis.
- Each element must have at least two different features.
- The minimum cluster size is chosen to be 3. I.e. there must at least be 3 similar elements before a cluster is created.
- Minimum mutual information score is set to $\log 100$. A feature in an element should appear 100 times as often as they would if they were independent.
- Minimum similarity score is set to 0.1. Elements with lower similarity scores are not considered to be related.

3.3 Results

Table 3 records the highest F-scores achieved by the corpuses, along with their thesaurus score percentages. The experiments were run until they suggest stable F-scores. Recall can always be increased by maximizing cluster creation and removing minimum membership threshold. Different corpus achieved their high score using different values of θ, σ, and μ. In general, a less strict set of parameters works better for a larger corpus. There is a strong downward trend of all quality measures as corpus size increases, even though the quantity of matches are increasing. There is no noticeable effect from using input features that were created from different parsers.

Various values of θ, σ, and μ were used to observe their effects on the F-scores. The tests were conducted using BOOK, and the results are shown in Figure 3. All three have noticeable impacts on the F-scores. Larger θ results in lower pair precision. Larger σ results in higher pair precision, but loses recall at a faster rate. A larger μ decreases both pair precision and recall.

Figure 4 shows how many words in ROGET were discovered from our clusters, and how efficiently we used the input features to achieve that coverage. Quite intuitively, having a larger corpus allowed us to discover more words. However, discovery of additional words required many more features, as shown by the decrease in efficiency.

3.4 Discussion

The clustering algorithm was able to achieve a maximum F-score of 40%. From [Jarmasz and Szpakowicz, 2003, Kennedy and Szpakowicz, 2008], it had been shown that F-scores calculated from ROGET is comparable to those calculated from *WordNet*. While these values are not directly comparable to the *WordNet* performance from [Pantel, 2003] due to implementation differences and measurement criteria, there, a peak score of around 40% was achieved with strict cluster parameters. In both cases, clusters with looser constraints performed better than those with more restricted parameters.

It is important to note that these scores only reflect the correspondence of the result with ROGET. Those clusters that do not match may still be correct clusters. The example pair "keyboard" and "printer" score 0 because "keyboard" does not exist in ROGET[6]. The organizational nature of ROGET is such that it may selectively include or exclude word concepts in a somewhat subjective fashion. Automated clustering, having no such limitation, would attempt to produce all matches. Such deliberate organization impedes automated clustering from perfectly recreating a thesaurus by itself. However, after having discovered all the appropriate clus-

[6] [Merriam-Webster Online, 2008] shows that the word "keyboard" appeared in 1819, long before the 1911 thesaurus. "Printer" appears under the head "Printing". "Keyboard" is included in the most recent edition of *Roget*, but only associated with "piano", and still far from "printer" [Kipfer, 2005].

θ	σ	μ	Clusters	Score	Pair Match	Matched Words	F_{PP}	F_R	F
BOOK with 5721 usable words									
0.6	0.2	0.1	237	17.0%	23248/43051	1773	54.0%	31.0%	39.4%
WSJ with 18,405 usable words									
0.9	0.05	0.05	1417	8.6%	49225/157831	3775	31.2%	20.5%	24.74%
WIKI with 58,224 usable words									
0.8	0.2	0.05	2847	4.9%	212936/1191003	8962	17.9%	15.4%	16.56%
COMBINED corpus with 66,443 usable words									
0.9	0.05	0.05	6077	2.6%	98061/849099	9521	11.5%	14.3%	12.7%

Table 3: Best F-score results

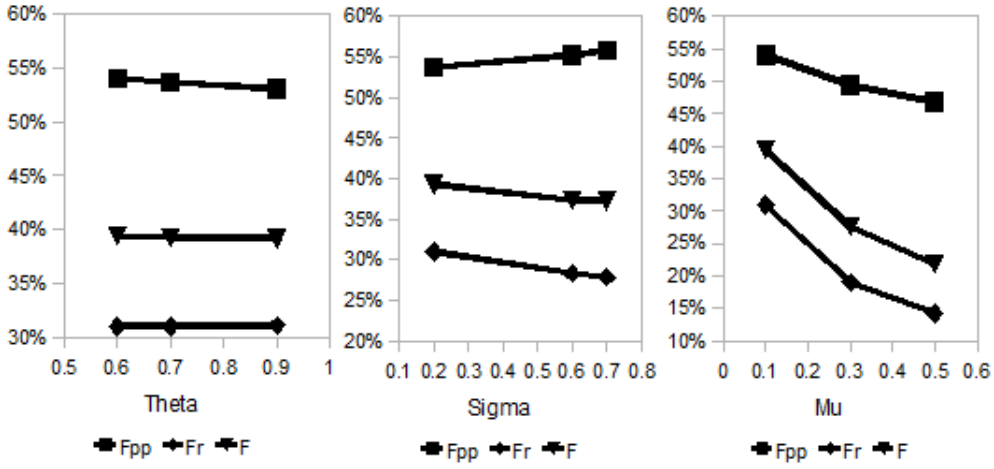

Figure 3: Effect of θ, σ, and μ on F-score.

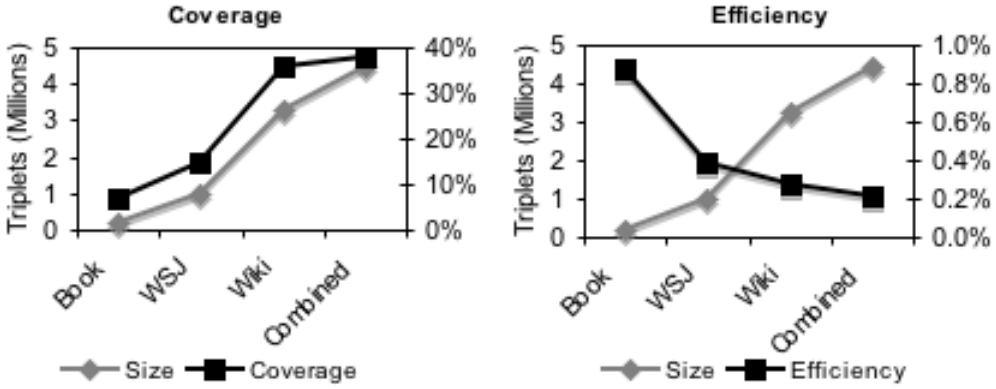

Figure 4: Coverage and efficiency for corpuses of different sizes.

ters, it is possible for the human effort to be limited to the selection process rather than the creation process. This would significantly reduce the amount of manual work. What this shows is that after the automated generation of clusters, there is still great value in the manual organization of these clusters.

The results on coverage and efficiency show a diminishing rate of return given more data. To achieve very high recall rate, we would need an extremely large amount of data, and may cause the precision score to suffer as well. This shows that the collection of data needs to be a more deliberate process. Ideal data are those that contain a large number of uniques words used under all known contexts. One example that comes to mind is in fact a dictionary, which is of course itself a manually edited work.

4 Related Work

The idea of generating a thesaurus automatically through word clustering techniques had been previously explored, notably by [Grefenstette, 1993]. The general technique is to observe a word's neighbors, and words that have similar neighbors tend to have similar meaning. This work furthers some of the ideas introduced there, and creates quantitative measurements for accuracy.

CBC had been evaluated, and compared to other clustering algorithms, using *WordNet* as the gold standard in [Pantel, 2003, Cicurel et al., 2006], as opposed to *Roget*. In addition to CBC, another system that uses the feature triplet idea was devised by [Dorow and Widdows, 2003] by graphically representing $||w, r, w'||$ as vertex-edge-vertex in a graph and creating clusters through the use of random walks.

There are other approaches to the clustering problem as well. For example, [Park and Choi, 1996] uses a Bayesian network to specifically help surface low frequency terms. The motivation is related to the "looseness" of typical thesaurus clusters. Words in such clusters may not appear in high frequency relative to each other, but each such occurrence should be surfaced.

5 Future Work

One important improvement that can be explored is to distinguish between synonyms and antonyms. One feature of ROGET is the hierarchy of "Head group". This groups concepts with opposite meanings, as such "existence" versus "inexistence". This would very much improve the usability of a thesaurus. One possible solution is to match morphological features, such as prefixes and suffices that may indicate negation. Another approach by [Lin et al., 2003] is to explore nearby words that imply negative relationships.

One of the drawbacks of this system is that it does not preserve phrases. [Grefenstette, 1993] also discusses generating clusters of phrases by looking further around the words for common features. A significant portion of ROGET is composed of short phrasal expressions. While our measurement ignores phrases, any additional attempt to preserve and analyse word groups would help improve the overall coverage.

Many different similarity scoring systems can be used. [Pantel, 2003] and [Curran and Moens, 2002] provide thorough overviews of some of the more popular clustering algorithms and element and relative feature similarity algorithms to replace cosine similarity and mutual information. Each may be used to particularly emphasize certain aspects of the data.

In addition, the clustering algorithm has a large number of parameters that can be adjusted. In this specific implementation, there are 15 parameters that affect clustering. While here we discussed the most significant 3, changing any of the other ones will impact the results as well. For example, the minimum occurrence count for an element to be considered useful is currently 2, but for a word to be analyzed more strictly, a higher occurrence count would provide better information about the usage of the word. This

would, however, require larger corpuses.

6 Conclusion

We have shown how the creation of a thesaurus can be attempted using word clustering techniques. A clustering algorithm was applied to text from sources of differing nature, and the quality of the resulting clusters were judged against a manually created thesaurus. Various quantitative scoring systems were used to provide better representation of the result quality. From this, we believe that such automated system could mimic a manual work given enough quality data. Quality data means a large diversity in word usage.

References

Hsinchun Chen, Tak Yim, and David Fye. Automatic thesaurus generation for an electronic community system. *Journal of the American Society For Information Science*, 1995.

Laurent Cicurel, Stephan Bloehdorn, and Philipp Cimiano. Clustering of polysemic words. In *Proceedings of the 30th Annual Conference of the Gesellschaft fr Klassifikation e.V.*, 2006.

James R. Curran and Marc Moens. Improvements in automatic thesaurus extraction. In *Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pages 59–66, 2002.

Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual. `http://nlp.stanford.edu/software/dependencies_manual.pdf`, 2008.

Beate Dorow and Dominic Widdows. Discovering corpus-specific word senses. In *Proceedings of EACL*, pages 79–82, 2003.

Gregory Grefenstette. Automatic thesaurus generation from raw text using knowledge-poor techniques. In *Making Sense of Words. Ninth Annual Conference of the UW Centre for the New OED and Text Research*, 1993.

Mario Jarmasz and Stan Szpakowicz. Roget's thesaurus and semantic similarity. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 212–219, 2003.

Alistair Kennedy and Stan Szpakowicz. Evaluating Roget's thesauri. In *Proceedings of ACL-08: HLT*, pages 416–424, Columbus, Ohio, June 2008.

Adam Kilgarriff and Colin Yallop. What's in a thesaurus? In *Proceedings of Second International Conference on Language Resources and Evaluation*, pages 1371–1379, 2000.

Barbara Ann Kipfer, editor. *Roget's 21st Century Thesaurus, 3rd Edition.* Philip Lief Group, 2005.

Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774, 1998.

Dekang Lin, Shaojun Zhao, Lijuan Qin, and Ming Zhou. Identifying synonyms among distributionally similar words. In *Proceedings of IJCAI-03*, pages 1492–1493, 2003.

Merriam-Webster Online. Merriam-Webster online dictionary. `http://www.merriam-webster.com/dictionary/`, 2008.

Patrick Pantel. *Clustering by Committee.* PhD thesis, University of Alberta, 2003.

Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619, 2002.

Young C. Park and Key-Sun Choi. Automatic thesaurus construction using bayesian networks. *Information Processing and Management*, 32(5):543–553, 1996. ISSN 0306-4573.

Peter Mark Roget. Roget's Thesaurus, 1911 Edition with Revisions by MICRA, Inc. `http://www.gutenberg.org/etext/22`, 1991.

Takenobu Tokunaga, Makoto Iwayama, and Hozumi Tanaka. Automatic thesaurus construction based on grammatical relations. In *IJCAI*, pages 1308–1313, 1995.

Leo Tolstoy. Anna Karenina, Translated by Constance Garnett. `http://www.gutenberg.org/etext/1399`, 2005.