Search with Synonyms: Problems and Solutions

Xing Wei, Fuchun Peng, Huishin Tseng, Yumao Lu, Xuerui Wang, Benoit Dumoulin
Yahoo! Labs
701 First Avenue, Sunnyvale, California, USA, 94089
xwei,fuchun,huihui,yumaol,xuerui,benoitd@yahoo-inc.com

Abstract

Search with synonyms is a challenging problem for Web search, as it can easily cause intent drifting. In this paper, we propose a practical solution to this issue, based on co-clicked query analysis, i.e., analyzing queries leading to clicking the same documents. Evaluation results on Web search queries show that synonyms obtained from this approach considerably outperform the thesaurus based synonyms, such as WordNet, in terms of keeping search intent.

1 Introduction

Synonym discovery has been an active topic in a variety of language processing tasks (Baroni and Bisi, 2004; Fellbaum, 1998; Lin, 1998; Pereira et al., 1993; Sanchez and Moreno, 2005; Turney, 2001). However, due to the difficulties of synonym judgment (either automatically or manually) and the uncertainty of applying synonyms to specific applications, it is still unclear how synonyms can help Web scale search task. Previous work in Information Retrieval (IR) has been focusing mainly on related words (Bai et al., 2005; Wei and Croft, 2006; Riezler et al., 2008). But Web scale data handling needs to be precise and thus synonyms are more appropriate than related words for introducing less noise and alleviating the efficiency concern of query expansion. In this paper, we explore both manually-built thesaurus and automatic synonym discovery, and apply a three-stage evaluation by separating synonym accuracy from relevance judgment and user experience impact.

The main difficulties of discovering synonyms for Web search are the following:

1. Synonym discovery is context sensitive. Although there are quite a few manually built thesauri available to provide high quality synonyms (Fellbaum, 1998), most of these synonyms have the same or nearly the same meaning only in some senses. If we simply replace them in search queries in all occurrences, it is very easy to trigger search intent drifting. Thus, Web search needs to understand different senses encountered in different contexts. For example, “baby” and “infant” are treated as synonyms in many thesauri, but “Santa Baby” has nothing to do with “infant”. “Santa Baby” is a song title, and the meaning of “baby” in this entity is different than the usual meaning of “infant”.

2. Context can not only limit the use of synonyms, but also broaden the traditional definition of synonyms. For instance, “dress” and “attire” sometimes have nearly the same meaning, even though they are not associated with the same entry in many thesauri; “free” and “download” are far from synonyms in traditional definition, but “free cd rewriter” may carry the same query intent as “download cd rewriter”.

3. There are many new synonyms developed from the Web over time. “Mp3” and “mpeg3” were not synonyms twenty years ago; “snp newspaper” and “snp online” carry the same query intent only after snponline.com was published. Manually editing synonym list is prohibitively expensive. Thus, we need an automatic synonym discovery system that can learn from huge amount of data and update the dictionary frequently.

In summary, synonym discovery for Web search is different from traditional thesaurus mining; it needs to be context sensitive and needs to be updated timely. To address these problems, we conduct context based synonym discovery from co-clicked queries, i.e., queries that share similar document click distribution. To show the effectiveness of our synonym discovery method on Web search, we use several metrics to demonstrate significant improvements: (1) synonym discovery accuracy that measures how well it keeps the same search intent; (2) relevance impact measured by Discounted Cumulative Gain (DCG) (Jarvelin and Kekalainen., 2002); and (3) user experience impact measured by online experiment.

The rest of the paper is organized as follows. In Section 2, we first discuss related work and differentiate our work from existing work. Then we present the details of our synonym discovery approach in Section 3. In Section 4 we show our query rewriting strategy to include synonyms in Web search. We conduct experiments on randomly sampled Web search queries and run the three-stage evaluation in Section 5 and analyze the results in Section 6. WordNet based synonym reformulation and a current commercial search engine are the baselines for the three-stage evaluation respectively. Finally we conclude the paper in Section 7.

2 Related Works

Automatically discovering synonyms from large corpora and dictionaries has been popular topics in natural language processing (Sanchez and Moreno, 2005; Senellart and Blondel, 2003; Turney, 2001; Blondel and Senellart, 2002; van der Plas and Tiedemann, 2006), and hence, there has been a fair amount of work in calculating word similarity (Porzel and Malaka, 2004; Richardson et al., 1998; Strube and Ponzetto, 2006; Bollegala et al., 2007) for the purpose of discovering synonyms, such as information gain on ontology (Resnik, 1995) and distributional similarity (Lin, 1998; Lin et al., 2003). However, the definition of synonym is application dependent and most of the work has been applied to a specific task (Turney, 2001) or restricted in one domain (Baroni and Bisi, 2004). Synonyms extracted using these traditional approaches cannot be easily adopted in Web search where keeping search intent is critical.

Our work is also related to semantic matching in IR: manual techniques such as using handcrafted thesauri and automatic techniques such as query expansion and clustering all attempts to provide a solution, with varying degrees of success (Jones, 1971; van Rijsbergen, 1979; Deerwester et al., 1990; Liu and Croft, 2004; Bai et al., 2005; Wei and Croft, 2006; Cao et al., 2007). These works focus mainly on adding in loosely semantically related words to expand literal term matching. But related words may be too coarse for Web search considering the massive data available.

3 Synonym Discovery based on Co-clicked Queries

In this section, we discuss our approach to synonym discovery based on co-clicked queries in Web search in detail.

3.1 Co-clicked Query Clustering

Clustering has been extensively studied in many applications, including query clustering (Wen et al., 2002). One of the most successful techniques for clustering is based on distributional clustering (Lin, 1998; Pereira et al., 1993). We adopt a similar approach to our co-clicked query clustering. Each query is associated with a set of clicked documents, which in turn associated with the number of views and clicks. We then compute the distance between a pair of queries by calculating the Jensen-Shannon(JS) divergence (Lin, 1991) between their clicked URL distributions. We start with that every query is a separate cluster, and merge clusters greedily. After clusters are generated, pairs of queries within the same cluster can be considered as co-clicked/related queries with a similarity score computed from their JS divergence.

$$Sim(q_k|q_l) = D_{JS}(q_k||q_l) \quad (1)$$

3.2 Query Pair Alignment

To make sure that words are replacement for each other in the co-clicked queries, we align words in the co-clicked query pairs that have the same length (number of terms), and have the same terms for all positions except one. This is a simplification for complicated aligning processes. Previous work on machine translation (Brown et al., 1993) can be used when complete alignment is needed for modeling. However, as we have tremendous amount of co-clicked query data, our restricted version of alignment is sufficient to obtain a reasonable number of synonyms. In addition, this restricted approach eliminates much noise introduced in those complicated aligning processes.

3.2.1 Synonym Discovery from Co-clicked Query Pair

Synonyms discovered from co-clicked queries have two aspects of word meaning: (1) general meaning in language and (2) specific meaning in the query. These two aspects are related. For example, if two words are more likely to carry the same meaning in general, then they are more likely to carry the same meaning in specific queries; on the other hand, if two words often carry the same meaning in a variety of specific queries, then we tend to believe that the two words are synonyms in general language. However, neither of these two aspects can cover the other. Synonyms in general language may not be used to replace each other in a specific query. For example, "sea" and "ocean" have nearly the same meaning in language, but in the specific query "sea boss boat", "sea" and "ocean" cannot be treated as synonyms because "sea boss" is a brand; also, in the specific query "women's wedding attire", "dress" can be viewed as a synonym to "attire", but in general language, these two words are not synonyms. Therefore, whether two words are synonyms or not for a specific query is a synthesis judgment based on both of general meaning and specific context.

We develop a three-step process for synonym discovery based on co-clicked queries, considering the above two aspects.

Step 1: Get all synonym candidates for word w_i in general meaning.

In this step, we would like to get all synonym candidates for a word. This step corresponds to Aspect (1) to catch the general meaning of words in language. We consider all the co-clicked queries with the word and sum over them, as in Eq. 2

$$P(w_j|w_i) = \frac{\sum_k sim_k(w_i \rightarrow w_j)}{\sum_{w_j}\sum_k sim(w_i \rightarrow w_j)} \quad (2)$$

where $sim_k(w_i \rightarrow w_j)$ represents the similarity score (see Section 3.1) of a query q_k that aligns w_i to w_j. So intuitively, we aggregate scores of all query pairs that align w_i to w_j, and normalize it to a probability over the vocabulary.

Step 2: Get synonyms for word w_i in query q_k.

In this step, we would like to get synonyms for a word in a specific query. We define the probability of reformulating w_i with w_j for query q_k as the similarity score shown in Eq. 3.

$$P\left(w_j|w_i, q_k\right) = sim_k(w_i \rightarrow w_j) \quad (3)$$

Step 3: Combine the above two steps.

Now we have two sets of estimates for the synonym probability, which is used to reformulate w_i with w_j. One set of values are based on general language information and another set of values are based on specific queries. We apply three combination approaches to integrate the two sets of values for a final decision of synonym discovery: (1) two independent thresholds for each probability, (2) linear combination with a coefficient, and (3) linear combination in log scale as in Eq. 4, with λ as a mixture coefficient.

$$\begin{aligned} P_{q_k}(w_j|w_i) \propto \lambda \log P(w_j|w_i) \\ +(1-\lambda)\log P(w_j|w_i, q_k) \end{aligned} \quad (4)$$

In experiments we found that there is no significant difference with the results from different combination methods by finely tuned parameter setting.

3.2.2 Concept based Synonyms

The simple word alignment strategy we used can only get the synonym mapping from single

term to single term. But there are a lot of phrase-to-phrase, term-to-phrase, or phrase-to-term synonym mappings in language, such as “babe in arms” to “infant”, and “nyc” to ”new york city”. We perform query segmentation on queries to identify concept units from queries based on an unsupervised segmentation model (Tan and Peng, 2008). Each unit is a single word or several consecutive words that represent a meaningful concept.

4 Synonym Handling in Web Search

The automatic synonym discovery methods described in Section 3 generate synonym pairs for each query. A simple and straightforward way to use the synonym pairs would be “equalizing” them in search, just like the “OR” function in most commercial search engines.

Another method would be to re-train the whole ranking system using the synonym feature, but it is expensive and requires a large size training set. We consider this to be future work.

Besides general equalization in all cases, we also apply a restriction, specially, on whether or not to allow synonyms to participate in document selection. For the consideration of efficiency, most Web search engines has a document selection step to pre-select a subset of documents for full ranking. For the general equalization, the synonym pair is treated as the same even in the document selection round; in a conservative variation, we only use the original word for document selection but use the synonyms in the second phase finer ranking.

5 Experiments

In this section, we present the experimental results for our approaches with some in-depth discussion.

5.1 Evaluation Metrics

We have several metrics to evaluate the synonym discovery system for Web search queries. They corresponds to the three stages during the system development. Each of them measures a different aspect.

Stage 1: accuracy. Because we are more interested in the application of reformulating Web search queries, our guideline to the editorial judgment focuses on the query intent change and context-based synonyms. For example, “transporters” and “movers” are good synonyms in the context of “boat” because “boat transporters” and “boat movers” keep the same search intent, but “ocean” is not a good synonym to “sea” in the query of “sea boss boats” because “sea boss” is a brand name and “ocean boss” does not refer to the same brand. Results are measured with accuracy by the number of discovered synonyms (which reflects coverage).

Stage 2: relevance. To evaluate the effectiveness of our semantic features we use DCG, a widely-used metric for measuring Web search relevance.

Stage 3: user experience. In addition to the search relevance, we also evaluate the practical user experience after logging all the user search behaviors during a two-week online experiment.

Web CTR: the Web click through rate (Sherman and Deighton, 2001; Lee et al., 2005) is defined as

$$CTR = \frac{\text{number of clicks}}{\text{total page views}},$$

where a page view (PV) is one result page that a search engine returns for a query.

Abandon rate: the percentage of queries that are abandoned by user neither clicking a result nor issuing a query refinement.

5.2 Data

A period of Web search query log with clicked URLs are used to generate co-clicked query set. After word alignment that extracts the co-clicked query pairs with same number of units and with only one different unit, we obtain 12.1M unsegmented query pairs and 11.9M segmented query pairs.

Since we run a three-stage evaluation, there are three independent evaluation set respectively:

1. accuracy test set. For the evaluation of synonym discovery accuracy, we randomly sampled 42K queries from two weeks of query log, and

evaluate the effectiveness of our synonym discovery model with these queries. To test the synonym discovery model built on the segmented data, we segment the queries before using them as evaluation set.

2. relevance test set. To evaluate the relevance impact by the synonym discovery approach, we run experiments on another two weeks of query log and randomly sampled 1000 queries from the affected queries (queries that have differences in the top 5 results after synonym handling).

3. user experience test set. The user experience test is conducted online with a commercial search engine.

5.3 Results of Synonym Discovery Accuracy

Here we present the results of WordNet thesaurus based query synonym discovery, co-clicked based term-to-term query synonym discovery, and co-click concept based query synonym discovery.

5.3.1 Thesaurus-based Synonym Replacement

The WordNet thesaurus-based synonym replacement is a baseline here. For any word that has synonyms in the thesaurus, thesaurus-based synonym replacement will rewrite the word with synonyms from the thesaurus.

Although thesaurus often provides clean information, synonym replacement based on thesaurus does not consider query context and introduces too many errors and noise. Our experiments show that only **46%** of the discovered synonyms are correct synonyms in query. The accuracy is too low to be used for Web search queries.

5.3.2 Co-clicked Query-based Context Synonym Discovery

Here we present the results from our approach based on co-clicked query data (in this section the queries are all original queries without segmentation). Figure 1 shows the accuracy of synonyms by the number of discovered synonyms. By applying different thresholds as cut-off lines to Eq. 4, we get different numbers of synonyms from the same test set. As we can see, loosening the threshold can give us more synonym pairs, but it could hurt the accuracy.

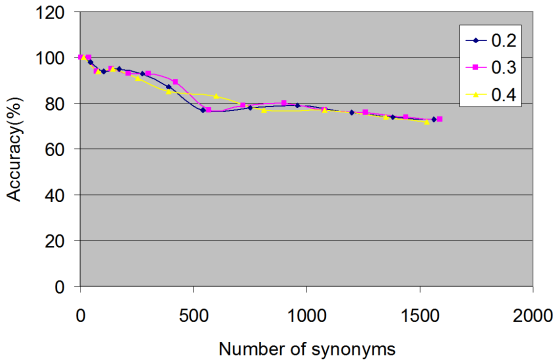

Figure 1: Accuracy versus number of synonyms with term based synonym discovery

Figure 1 demonstrates how accuracy changes with the number of synonyms. Y-axis represents the percentage of correctly discovered synonyms, and X-axis represents the number of discovered synonyms, including both of correct ones and wrong ones. The three different lines represents three different parameter settings of mixture weights (λ in Eq. 4, which is 0.2, 0.3, or 0.4 in the figure). The figure shows accuracy drops by increasing the number of synonyms. More synonym pairs lead to lower accuracy.

From Figure 1 we can see: Firstly, three curves with different thresholds almost overlap, which means the effectiveness of synonym discovery is not very sensitive to the mixture weight. Secondly, accuracy is monotonically decreasing as more synonyms are detected. By getting more synonyms, the accuracy decreases from **100%** to less than **80%** (we are not interested in accuracies lower than 80% due to the high precision requirement of Web search tasks, so the graph contains only high-accuracy results). This trend also confirms the effectiveness of our approach (the accuracy for a random approach would be a constant).

5.3.3 Concept based Context Synonym Discovery

We present results from our model based on segmented co-clicked query data in this section.

Original Query	New Query with Synonyms	Intent
Examples of thesaurus-based based synonym replacement		
basement window wells drainage	basement window wells drain	same
billabong boardshorts sale	billabong boardshorts sales event	
bigger stronger faster documentary	larger stronger faster documentary	
yahoo	hayseed	different
maryland judiciary case search	maryland judiciary pillowcase search	
free cell phone number lookup	free cell earpiece number lookup	
Examples of term-to-term synonym discovery		
airlines jobs	airlines careers	same
area code finder	area code search	
acai berry	acai fruit	
acai berry	acai juice	different
ace	hardware	
crest toothpaste coupon	crest whitestrips coupon	
Examples of concept based synonym discovery		
ae	american_eagle outfitters	same
apartments_for_rent	apartment_rentals	
arizona time_zone	arizona time	
cortrust bank credit_card	cortrust bank mastercard	different
david_beckham	beckham	
dodge_caliber	dodge	

Table 1: Examples of query synonym discovery: the first section is thesaurus based, second section is co-clicked data based term-to-term synonym discovery, and the last section is concept based synonym discovery.

The modeling part is the same as the one for Section 5.3.2, and the only difference is that the data were segmented. We have shown in Section 5.3.2 that the mixture weight is not an crucial factor within a reasonable range, so we present only the result with one mixture weight in Figure 2. As in Section 5.3.2, the figure shows that the accuracy of synonym discovery is sensitive to the threshold. It confirms that our model is effective and setting threshold to Eq. 4 is a feasible and sound way to discover not only single term synonyms but also phrase synonyms.

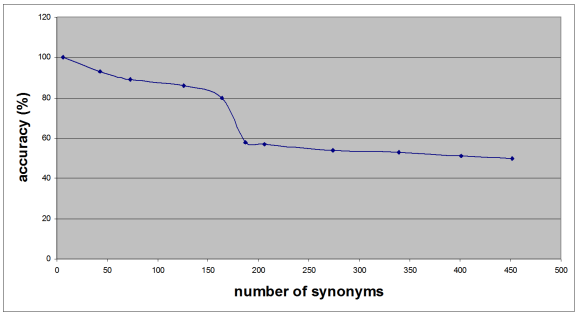

Figure 2: Accuracy versus number of synonyms with concept based synonym discovery

Table 1 shows some anecdotal examples of query synonyms with the thesaurus-based synonym replacement, context sensitive synonym discovery, and concept based context sensitive synonym discovery. In contrast, the upper part of each section shows positive examples (query intents remain the same after synonym replacement) and the lower part shows negative examples (query intents change after synonym replacement).

5.4 Results of Relevance Impact

We run relevance test on 1000 randomly sampled affected queries. With the automatic synonym discovery approach we apply our synonym handling method described in Section 4. Results of DCG improvements by different thresholds and synonym handling settings are presented in Table 2. Thresholds are selected empirically from the accuracy test in Section 5.3 (we run a small size relevance test on the accuracy test set and set the range of thresholds based on that). Note that in our relevance experiments we use term-to-term synonym pairs only. For the relevance impact of concept-based synonym discovery, we would like to study it in our future work.

From Table 2 we can see that the automatic synonym discovery approach we presented significantly improves search relevance on various settings, which confirms the effectiveness of our synonym discovery for Web search queries. We conjecture that avoiding synonym in document selection is of help. This is because precision is more important to Web search than recall for the huge amount of data available on the Web.

Relevance impact with synonym handling			
		doc-selection	
threshold1	threshold2	participation	DCG
0.8	0.02	no	**+1.7%**
0.8	0.02	yes	**+1.3%**
0.8	0.05	no	**+1.8%**
0.8	0.05	yes	**+1.4%**

Table 2: Relevance impact with synonym handling by different parameter settings. "Threshold1" is the threshold for context-based similarity score–Eq. 3; "threshold2" is the threshold for general case similarity score–Eq. 2; "doc-selection participation" refers to whether or not let synonym handling participate in document selection. All improvements are statistically significant by Wilcox significance test.

5.5 Results of User Experience Impact

In addition to the relevance impact, we also evaluated the practical user experience impact by CTR and abandon rate (defined in Section 5.1) through a two-week online run. Results show that the synonym discovery method presented in this paper improves Web CTR by **2%**, and decreases abandon rate by **11.4%**. All changes are statistically significant, which indicates synonyms are indeed beneficial to user experience.

6 Discussion and Error Analysis

From Table 1, we can see that our approach can catch not only traditional synonyms, which are the synonyms that can be found in manually-built thesaurus, but also context-based synonyms, which may not be treated as synonyms in a standard dictionary or thesaurus. There are a variety of synonyms our approach discovered:

1. Synonyms that are not considered as synonyms in traditional thesaurus, such as "berry" and "fruit" in the context of "acai". "acai berry" and "acai fruit" refer to the same fruit.

2. Synonyms that have different part-of-speech features than the corresponding original words, such as "finder" and "search". Users searching "area code finder" and users searching "area code search" are looking for the same content. In the context of Web search queries, part-of-speech is not an important factor as most queries are not grammatically perfect.

3. Synonyms that show up in recent concepts, such as "webmail" and "email" in the context of "cox". The new concept of "webmail" or "email" has not been added to many thesauri yet.

4. Synonyms not limited by length, such as "crossword puzzles" and "crossword", "homes for sale" and "real estate". The segmenter helps our system discover synonyms in various lengths.

With these many variations, the synonyms discovered by our approach are not the "synonyms" in the traditional meaning. They are context sensitive, Web data oriented and search effective synonyms. These synonyms are discovered by the statistical model we presented and based on Web search queries and clicked data.

However, the click data themselves contain a huge amount of noise. Although they can reflect the users' intents in some big picture, in many specific cases synonyms discovered from co-clicked data are biased by the click noise. In our application—Web search query reformulation with synonyms, accuracy is the most important thing and thus we are interested in error analysis. The errors that our model makes in synonym discovery are mainly caused by the following reasons:

(1) There are some concepts well accepted such as "cnn" means "news" and "amtrak" means "train". And users searching "news" tend to click CNN Web site; users searching "train" tend to click Amtrak Web site. With our model, "cnn" and "news", "amtrak" and "train" are discovered to be synonyms, which may hurt the search of "news" or "train" in general meaning.

(2) Same clicks by different intents. Although clicking on same documents generally indicates same search intent, different intents could result in same or similar clicks, too. For example, the queries of "antique style wedding rings" and "antique style engagement rings" carry different intents, but very usually, these two different intents lead to the clicks on the same Web site. "Booster seats" and "car seats", "brighton handbags" and "brighton shoes" are other two examples in the same case. For these examples, clicking on Web URLs are not precise enough to reflect the subtle difference of language concepts.

(3) Bias from dominant user intents. Most people searching "apartment" are looking for an apartment to rent. So "apartment for rent" and "apartment" have similar clicked URLs. But these two are not synonyms in language. In these cases, popular user intents dominate and bias the meaning of language, which causes problems. "Airline baggage restrictions" and "airline travel restrictions" is another example.

(4) Antonyms. Many context-based synonym discovery methods suffer from the antonym problem, because antonyms can have very similar contexts. In our model, the problem has been reduced by integrating clicked-URLs. But still, there are some examples, such as "spyware" and "antispyware", resulting in similar clicks. To learn how to "protect a Web site", a user often needs to learn what are the main methods to "attack a Web site", and these different-intent pairs lead to the same clicks because different intents do not have to mean different interests in many specific cases.

Although these problems are not common, but when they happen, they cause a bad user search experience. We believe a solution to these problems might need more advanced linguistic analysis.

7 Conclusions

In this paper, we have developed a synonym discovery approach based on co-clicked query data, and improved search relevance and user experience significantly based on the approach.

For future work, we are investigating more synonym handling methods to further improve the synonym discovery accuracy, and to handle the discovered synonyms in more ways than just the query side.

References

Bai, J., D. Song, P. Bruza, J.Y. Nie, and G. Cao. 2005. Query Expansion using Term Relationships in Language Models for Information Retrieval. In *Proceedings of the ACM 14th Conference on Information and Knowledge Management*.

Baroni, M. and S. Bisi. 2004. Using Cooccurrence Statistics and the Web to Discover Synonyms in a Technical Language. In *LREC*.

Blondel, V. and P. Senellart. 2002. Automatic Extraction of Synonyms in a Dictionary. In *Proc. of the SIAM Workshop on Text Mining*.

Bollegala, D., Y. Matsuo, and M. Ishizuka. 2007. Measuring Semantic Similarity between Words using Web Search Engines. In *Proceedings of the 16th international conference on World Wide Web (WWW)*.

Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263.

Cao, G., J.Y. Nie, and J. Bai. 2007. Using Markov Chains to Exploit Word Relationships in Information Retrieval. In *Proceedings of the 8th Conference on Large-Scale Semantic Access to Content*.

Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, Mass.

Jarvelin, K. and J. Kekalainen. 2002. Cumulated Gain-Based Evaluation Evaluation of IR Techniques. *ACM TOIS*, 20:422–446.

Jones, K. S., 1971. *Automatic Keyword Classification for Information Retrieval*. London: Butterworths.

Lee, Uichin, Zhenyu Liu, and Junghoo Cho. 2005. Automatic Identification of User Goals in Web Search. In *In the World-Wide Web Conference (WWW)*.

Lin, D., S. Zhao, L. Qin, and M. Zhou. 2003. Identifying Synonyms among Distributionally Similar Words. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*.

Lin, J. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Lin, D. 1998. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING/ACL-98*, pages 768–774.

Liu, X. and B. Croft. 2004. Cluster-based Retrieval using Language Models. In *Proceedings of SIGIR*.

Pereira, F., N. Tishby, and L. Lee. 1993. Distributional Clustering of English Words. In *Proceedings of ACL*, pages 183 – 190.

Porzel, R. and R. Malaka. 2004. A Task-based Approach for Ontology Evaluation. In *ECAI Workshop on Ontology Learning and Population*.

Resnik, P. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *Proceedings of IJCAI-95*, pages 448 – 453.

Richardson, S., W. Dolan, and L. Vanderwende. 1998. MindNet: Acquiring and Structuring Semantic Information from Text. In *36th Annual meeting of the Association for Computational Linguistics*.

Riezler, Stefan, Yi Liu, and Alexander Vasserman. 2008. Translating Queries into Snippets for Improved Query Expansion. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING'08)*.

Sanchez, D. and A. Moreno. 2005. Automatic Discovery of Synonyms and Lexicalizations from the Web. In *Proceedings of the 8th Catalan Conference on Artificial Intelligence*.

Senellart, P. and V. D. Blondel. 2003. Automatic Discovery of Similar Words. In Berry, M., editor, *A Comprehensive Survey of Text Mining*. Springer-Verlag, New York.

Sherman, L. and J. Deighton. 2001. Banner advertising: Measuring effectiveness and optimizing placement. *Journal of Interactive Marketing*, 15(2):60–64.

Strube, M. and S. P. Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of AAAI*.

Tan, B. and F. Peng. 2008. Unsupervised Query Segmentation using Generative Language Models and Wikipedia. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 347–356.

Turney, P. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning*.

van der Plas, Lonneke and Jorg Tiedemann. 2006. Finding Synonyms using Automatic Word Alignment and Measures of Distributional Similarity. In *Proceedings of the COLING/ACL 2006*, pages 866–873.

van Rijsbergen, C.J., 1979. *Information Retrieval*. London: Butterworths.

Wei, X. and W. B. Croft. 2006. LDA-based Document Models for Ad-hoc Retrieval. In *Proceedings of SIGIR*, pages 178–185.

Wen, J.R., J.Y. Nie, and H.J. Zhang. 2002. Query Clustering Using User Logs. *ACM Transactions on Information Systems*, 20(1):59–81.