Clustering Web Search Results with Maximum Spanning Trees

Antonio Di Marco and Roberto Navigli

Dipartimento di Informatica,
Sapienza Università di Roma,
Via Salaria, 113 - 00198 Roma Italy
{dimarco,navigli}@di.uniroma1.it
http://lcl.uniroma1.it

Abstract. We present a novel method for clustering Web search results based on Word Sense Induction. First, we acquire the meanings of a query by means of a graph-based clustering algorithm that calculates the maximum spanning tree of the co-occurrence graph of the query. Then we cluster the search results based on their semantic similarity to the induced word senses. We show that our approach improves classical search result clustering methods in terms of both clustering quality and degree of diversification.

1 Introduction

The huge amount of text nowadays available on the Web makes language-related tasks, such as Information Retrieval, Information Extraction and Question Answering, increasingly difficult. Popular search engines such as Yahoo! and Google in general do a good job at finding a needle in a haystack, i.e., retrieving a small number of relevant results from such an enormous collection of Web pages. However, the current generation of search engines still lacks an effective way to address the issue of lexical ambiguity. In a recent study [34] – conducted using WordNet [25] and Wikipedia as sources of ambiguous words – it was reported that around 3% of Web queries and 23% of the most frequent queries are ambiguous. Such ambiguity is often due to the low average number of query words used by Web users [16]. While the average query length is increasing (now estimated at around 3 words per query) many search engines are addressing the query ambiguity issue by reranking and diversifying their results, so as to return Web pages that are not too similar to each other.

In recent years, Web clustering engines [7] have been proposed as a solution to the issue of lexical ambiguity in Web Information Retrieval. These systems group search results, by providing a cluster for each specific aspect (i.e., meaning) of the input query. Users can then select the cluster(s) and the pages therein that best answer their information needs. However, many Web clustering engines group search results on the basis of their lexical similarity, and therefore suffer from synonymy (same query expressed with different words) and polysemy (different user needs expressed with the same word).

In this paper we present a novel approach to Web search result clustering which is based on the automatic discovery of word senses from raw text – a task we refer to as

R. Pirrone and F. Sorbello (Eds.): AI*IA 2011, LNAI 6934, pp. 201–212, 2011.

Word Sense Induction (WSI). At the core of our approach is the identification of the user query's meaning using a graph-based algorithm which calculates a maximum spanning tree of the co-occurrence graph of the input query. Our experiments on two datasets of ambiguous queries show that our WSI approach boosts search result clustering in terms of both clustering quality and degree of diversification.

2 Related Work

Web directories such as the Open Directory Project are a first solution to query ambiguity. They provide taxonomies for the categorization of Web pages. Given a query, search results are organized by category. This approach has three main weaknesses: first, it is static, thus it needs manual updates to cover new pages; second, it covers only a small portion of the Web; third, it classifies Web pages based on coarse categories. This latter feature of Web directories makes it difficult to distinguish between instances of the same kind (e.g., pages about musicians with the same surname). While methods for the automatic classification of Web documents have been proposed and some problems have been tackled effectively [2], these approaches are usually supervised and still suffer from reliance on a predefined taxonomy of categories.

A different direction consists of associating explicit semantics (i.e., word senses or concepts) with queries and documents, that is, performing Word Sense Disambiguation (WSD, see [26] for a survey). SIR is performed by indexing and/or searching concepts rather than terms, thus potentially coping with both synonymy and polysemy. Over the years, different methods for SIR have been proposed [18,40,24,23, inter alia]. However, contrasting results have been reported on the benefits of these techniques: it was shown that WSD has to be very accurate to benefit Information Retrieval [33] – a result that was later debated [37].

SIR performs WSD using a reference knowledge resource (such as WordNet) and thus suffers from the static nature of the dictionary sense inventory and its inherent paucity of most proper nouns. This latter problem is particularly important for Web searches, as users tend to retrieve more information about named entities (e.g., singers, artists, cities) than concepts (e.g., abstract information about singers or artists).

A third approach to query ambiguity is search result clustering. Given a query, a flat list of text snippets returned from one or more commonly-available search engines is clustered using some notion of textual similarity. At the root of the clustering approach lies van Rijsbergen's [32] cluster hypothesis: "closely associated documents tend to be relevant to the same requests", whereas documents concerning different meanings of the input query are expected to belong to different clusters. Approaches to search result clustering can be classified as data-centric or description-centric [7]. The former focus more on the problem of data clustering than on presenting the results to the user. A pioneering example is Scatter/Gather [13], which divides the dataset into a small number of clusters and, after the selection of a group, performs clustering again and proceeds iteratively. Developments of this approach have been proposed which improve on cluster quality and retrieval performance [17]. Other data-centric approaches use agglomerative hierarchical clustering (e.g., LASSI [42]), rough sets [28] or exploit link information [47]. Description-centric approaches are, instead, more focused on the description

to produce for each cluster of search results. Among the most popular and successful approaches are those based on suffix trees [44]. Other methods in the literature are based on formal concept analysis [8], singular value decomposition [30], spectral clustering [11] and graph connectivity measures [14].

Diversification is another research topic dealing with the issue of query ambiguity. Its aim is to reorder top search results using criteria that maximize their diversity. Similarity functions have been used to measure the diversity among documents and between document and query [5]. Other techniques include the use of conditional probabilities to determine which document is most different from higher-ranking ones [9] or use affinity ranking [46], based on topic variance and coverage. More recently, an algorithm called Essential Pages [38] has been proposed to reduce information redundancy and return Web pages that maximize coverage with respect to the input query.

In our work we perform WSI to dynamically acquire an inventory of senses of the input query. Instead of clustering on the basis of the surface similarity of Web snippets, we use our induced word senses to group snippets. This framework was proposed for the first time in [27], where an effective graph algorithm based on triangles and squares was presented. In this paper we use the same framework to introduce maximum spanning trees for WSI-driven search result clustering. Very little further work on this topic has been done: vector-based WSI was successfully shown to improve bag-of-words ad-hoc Information Retrieval [36] and experimental studies [10] have provided interesting, though preliminary, insights into the use of WSI for Web search result clustering. More recently the use of hidden topics has been proposed to identify query meanings [29]. However, topics – estimated from a universal dataset – are query-independent and thus their number needs to be found beforehand. In contrast, we cluster snippets according to a dynamic and finer-grained notion of sense.

3 Approach

Web search result clustering is typically performed in three steps:

1. Given a query q, a search engine (e.g., Yahoo!) is used to retrieve a list of results $R = (r_1, \ldots, r_n)$;
2. A clustering $\mathcal{C} = (C_0, C_1, \ldots, C_m)$ of the results in R is obtained by means of a clustering algorithm;
3. The clusters in $\mathcal{C}$ are optionally labeled with an appropriate algorithm (e.g., see [6] and [43]) for visualization purposes.

In this paper we aim at improving step 2 by means of a graph-based Word Sense Induction algorithm: given a query q, we first use a text corpus to automatically induce the word senses of q (Section 3.1); then we cluster the Web results using the previously-acquired word senses (Section 3.2).

3.1 Word Sense Induction

Word Sense Induction is a task aimed at dynamically identifying the set of senses denoted by a word. These methods acquire word senses from text by grouping word occurrences exploiting the idea that a given word, when used in a specific sense, tends to

co-occur with the same neighbouring words [15]. Several approaches to WSI have been proposed in the literature (see [26] for a survey), including context-vector clustering [35], word clustering [22] and co-occurrence graphs [41,27].

A successful approach to WSI is HyperLex [39], a graph algorithm based on the identification of hubs in co-occurrence graphs. However, HyperLex has to cope with a high number of parameters to be tuned [1]. To overcome this issue we propose a simple yet effective graph algorithm for WSI, which we describe hereafter. The algorithm consists of two steps: graph construction and identification of word senses.

Graph construction. Given a target query q, we build a co-occurrence graph $G_q = (V, E)$ such that V is a set of context words related to q and E is the set of undirected edges, each denoting a co-occurrence between pairs of words in V. To determine the set of co-occurring words V, we use the Google Web1T corpus [4], a large collection of n-grams ($n = 1, \ldots, 5$) – i.e., windows of n consecutive tokens – occurring in one terabyte of Web documents. First, for each content word w we collect the total number $c(w)$ of its occurrences and the number of times $c(w, w')$ that words w and w' occur together in any 5-gram (we include inflected forms in the count); second, we use the Dice coefficient to determine the strength of co-occurrence between w and w':

$$Dice(w, w') = \frac{2c(w, w')}{c(w) + c(w')}. \quad (1)$$

The graph $G_q = (V, E)$ is built as follows:

- Our initial vertex set $V^{(0)}$ contains all the content words from the snippet results of query q (excluding stopwords); then, we add to $V^{(0)}$ the highest-ranking words co-occurring with q in the Web1T corpus, i.e., those words w for which $c(q, w) \geq \delta$ and $Dice(q, w) \geq \delta'$ (the thresholds are established experimentally, see Section 4.1). We set $V := V^{(0)}$ and $E := \emptyset$.
- Given a pair of words $\{w, w'\} \in V \times V$, if $max\{\frac{c(w,w')}{c(w)}, \frac{c(w,w')}{c(w')}\} \geq \sigma$, we add edge $\{w, w'\}$ to E with weight $Dice(w, w')$.
- Finally, we remove disconnected vertices.

Identification of word senses. Given the co-occurrence graph G_q for query q, we perform the following steps:

1. Eliminate from G_q all nodes whose degree is 1.
2. Calculate the maximum spanning tree (MST) T_{G_q} of the graph.
3. Work on T_{G_q} by iteratively eliminating the minimum-weight edge $e \in T_{G_q}$ such that its endpoints each have degree ≥ 2 until we obtain N connected components (i.e., word clusters) or there are no more edges to eliminate.

We provide an example of co-occurrence graph for the query *beagle* in Figure 1(a). The maximum spanning tree (step 2 above) is shown in bold, whereas the result of step 3 above, i.e., the final meaning components or word senses, is shown in Figure 1(b).

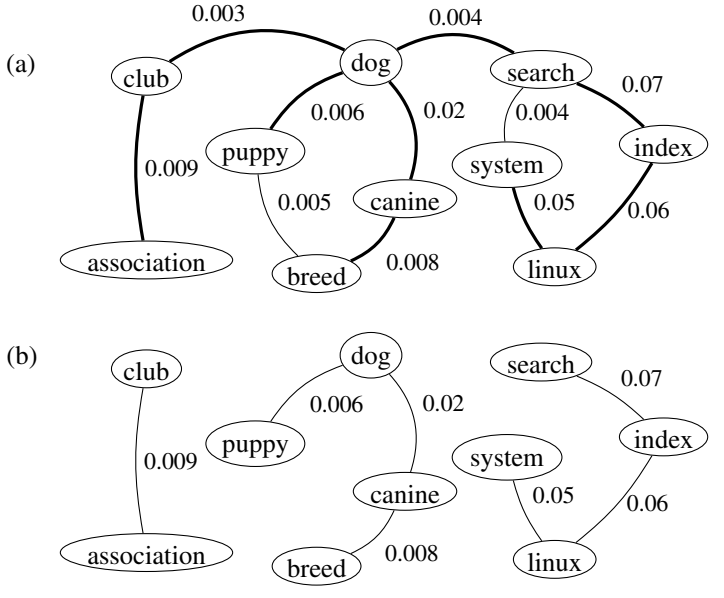

Fig. 1. The *MST* example ($N = 3$): (a) co-occurrence graph G and maximum spanning tree T_G (edges in bold); (b) the word senses induced after edge removal.

3.2 Clustering Web Results

We submit our input query q to a search engine, which returns a list of relevant search results $R = (r_1, \ldots, r_n)$. We transform the text snippet corresponding to each result r_i into a bag of words b_i. To this end, we apply tokenization, stopwords and target word removal, and lemmatization[1]). For instance, given the snippet:

"the *beagle* is a breed of medium-sized dog",

we produce the following bag of words:

{ *breed*, *medium*, *size*, *dog* }.

As a result of the above processing steps, we obtain a list of bags of words $B = (b_1, \ldots, b_n)$. Now, our aim is to cluster our Web results R, i.e., the corresponding bags of words B. To this end, rather than considering the interrelationships between them (as is done in traditional search result clustering), we intersect each bag of words $b_i \in B$ with the sense clusters $\{S_1, \ldots, S_m\}$ acquired as a result of our Word Sense Induction algorithm (cf. Section 3.1). The sense cluster with the largest intersection with b_i is selected as the most likely meaning of r_i. Formally:

$$Sense(r_i) = \begin{cases} \underset{j=1,\ldots,m}{\operatorname{argmax}} |b_i \cap S_j| \text{ if } \max\limits_{j} |b_i \cap S_j| > 0 \\ 0 \qquad\qquad\qquad\qquad \text{else} \end{cases} \tag{2}$$

[1] We use the WordNet lemmatizer.

where 0 denotes that no sense is assigned to result r_i, as the intersection is empty for all senses S_j. Otherwise the function returns the index of the sense having the largest overlap with b_i – the bag of words associated with the search result r_i. As a result of sense assignment for each $r_i \in R$, we obtain a clustering $\mathcal{C} = (C_0, C_1, \ldots, C_m)$ such that:

$$C_j = \{r_i \in R : Sense(r_i) = j\}, \tag{3}$$

that is, C_j contains the search results classified with the j-th sense of query q (C_0 includes unassigned results). Finally, we sort the clusters in our clustering $\mathcal{C}$ based on their "quality". For each cluster $C_j \in \mathcal{C} \setminus \{C_0\}$, we determine its similarity with the corresponding meaning S_j by calculating the following formula:

$$avgsim(C_j, S_j) = \frac{\sum_{r_i \in C_j} sim(r_i, S_j)}{|C_j|}. \tag{4}$$

The formula determines the average similarity between the search results in cluster C_j and the corresponding sense cluster S_j. The similarity between a search result r_i and S_j is determined as the normalized overlap between its bag of words b_i and S_j:

$$sim(r_i, S_j) = sim(b_i, S_j) = \frac{|b_i \cap S_j|}{|b_i|}. \tag{5}$$

Finally, we rank the elements r_i within each cluster C_j by their similarity $sim(r_i, S_j)$. We note that the ranking and optimality of clusters can be improved with more sophisticated techniques [12,19,20,21, inter alia]. However, this is outside the scope of this paper.

4 Experiments

4.1 Experimental Setup

Test Sets. We conducted our experiments on two datasets:

- AMBIENT (AMBIguous ENTries), a dataset which contains 44 ambiguous queries[2]. The sense inventory for the senses (i.e., subtopics)[3] of queries is given by Wikipedia disambiguation pages. For instance, given the *beagle* query, its disambiguation page in Wikipedia provides the senses of dog, Mars lander, computer search service, etc. The most appropriate Wikipedia senses were associated with the top 100 Web results of each query returned by the Yahoo! search engine (overall, 4400 search results were sense tagged).
- MORESQUE (MORE Sense-tagged QUEry results), a new dataset of 114 ambiguous queries which we developed as a complement to AMBIENT following the guidelines provided by the authors of the latter. In fact, our aim was to study the behaviour of Web search algorithms on queries of different lengths, ranging from

[2] `http://credo.fub.it/ambient`

[3] In the following, we use the terms *subtopic* and *word sense* interchangeably.

1 to 4 words. However, the AMBIENT dataset is composed mostly of single-word queries. MORESQUE provides dozens of queries of length 2, 3 and 4, together with the 100 top results from Yahoo! for each query annotated as in the AMBIENT dataset (overall, we tagged 11,400 snippets).

Parameters. Our graph-based algorithm has just one parameter: the maximum number N of clusters. We experimentally set this value to 5 using a small development set of queries and snippets. We used the same development set to learn the three parameters (δ, δ' and σ) needed for the construction of the co-occurrence graphs.

Systems. We compared MST against the best systems reported by [3] (cf. Section 2):

- **Lingo** [30]**:** a Web clustering engine implemented in the Carrot^2 open-source framework[4] that clusters the most frequent phrases extracted using suffix arrays.
- **Suffix Tree Clustering (STC)** [43]**:** the original Web search clustering approach based on suffix trees.
- **KeySRC** [3]**:** a state-of-the-art Web clustering engine built on top of STC with part-of-speech pruning and dynamic selection of the cut-off level of the clustering dendrogram.
- **Essential Pages (EP)** [38]**:** a recent diversification algorithm that selects fundamental pages which maximize the amount of information covered for a given query.
- **Yahoo!:** the original search results returned by the Yahoo! search engine.

The first three of the above are Web search result clustering approaches, whereas the last two produce lists of possibly diversified results (cf. Section 2).

4.2 Experiment 1: Clustering Quality

Measure. While assessing the quality of clustering is a notably hard problem, given a gold standard $\mathcal{G}$ we can calculate the **Rand index** (RI) of a clustering $\mathcal{C}$, a common quality measure in the literature, determined as follows [31]:

$$\mathrm{RI}(\mathcal{C}) = \frac{a}{\binom{|\mathcal{W}|}{2}} \tag{6}$$

where $\mathcal{W}$ is the union set of all the snippets in $\mathcal{C}$ and a is the number of snippet pairs put into the same (or different) cluster in both $\mathcal{C}$ and $\mathcal{G}$. For the gold standard $\mathcal{G}$ we use the clustering induced by the sense annotations provided in our datasets for each snippet. Similarly to what was done in Section 3.2, untagged results are grouped together in a special cluster of $\mathcal{G}$.

Results. The results of all systems on the AMBIENT and MORESQUE datasets according to the average Rand index are shown in Table 1[5]. In accordance with previous results in the literature, KeySRC performed generally better than the other search result

[4] `http://project.carrot2.org`

[5] For reference systems we used the implementations of [3] and [30].

Table 1. Results by Rand index (percentages)

System	AMBIENT	MORESQUE	All
MST	81.53	86.67	85.24
Lingo	62.75	52.68	55.49
STC	61.48	51.52	54.29
KeySRC	66.49	55.82	58.78

clustering systems, especially on smaller queries. Our Word Sense Induction system, MST, outperformed all other systems by a large margin, thus showing a higher clustering quality. Interestingly, all clustering systems perform more poorly on longer queries (i.e., on the MORESQUE dataset) whereas our WSI system overturns this trend performing better with longer queries.

4.3 Experiment 2: Diversification

Measure. We performed a second experiment to assess the ability of our clustering algorithms to diversify the top results returned by a search engine. For each query q, one natural way of measuring a system's performance is to calculate the **subtopic recall-at-K** [45] given by the number of different subtopics retrieved for q in the top K results returned:

$$\text{S-recall@K} = \frac{|\bigcup_{i=1}^{K} subtopics(r_i)|}{M} \tag{7}$$

where $subtopics(r_i)$ is the set of subtopics manually assigned to the search result r_i and M is the number of subtopics for query q (note that in our experiments M is the number of subtopics occurring in the 100 results retrieved for q, so S-recall@100 = 1). However, this measure is only suitable for systems returning ranked lists (such as Yahoo! and EP). Given a clustering $\mathcal{C} = (C_0, C_1, \ldots, C_m)$, we flatten it to a list as follows: we add to the initially empty list the first element of each cluster C_j ($j = 1, \ldots, m$); then we iterate the process by selecting the second element of each cluster C_j such that $|C_j| \geq 2$, and so on. The remaining elements returned by the search engine, but not included in any cluster of $\mathcal{C} \setminus \{C_0\}$, are appended to the bottom of the list in their original order. Note that the elements are selected from each cluster according to their internal ranking (e.g., for our algorithms we use Formula 5 introduced in Section 3.2).

Results. We compared the output of our system with the original snippet list returned by Yahoo! and the output of the EP diversification algorithm (cf. Section 4.1).

The S-recall@K (with $K = 3, 5, 10, 15, 20$) calculated on AMBIENT + MORESQUE is reported in Table 2. MST performs best, with a subtopic recall greater than all other systems. We observe that KeySRC and EP perform worse than Yahoo! with low values of K and generally better with higher values of K.

Given that the two datasets complement each other in terms of query lengths (with AMBIENT having queries of length $\leq$ 2 and MORESQUE with many queries of length $\geq$ 3), we studied the S-recall@K trend for the two datasets. The results are

Table 2. S-recall@K on all queries (percentages)

System	K=3	K=5	K=10	K=15	K=20
MST	54.7	65.6	79.2	86.7	90.7
Yahoo!	49.2	60.0	72.9	78.5	82.7
EP	40.6	53.2	68.6	77.2	83.3
KeySRC	44.3	55.8	72.0	79.1	83.2

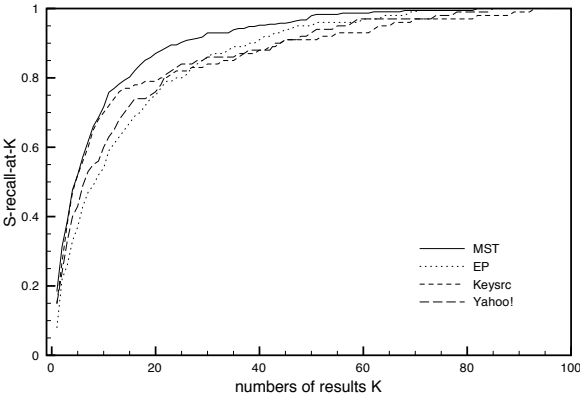

Fig. 2. Results by S-recall@K on AMBIENT

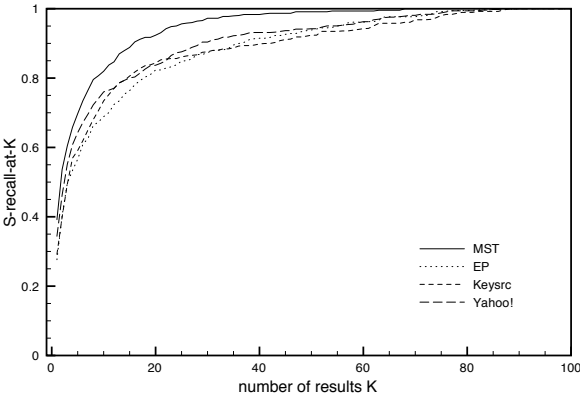

Fig. 3. S-recall@K on MORESQUE

shown in Figures 2 and 3. While KeySRC does not show large differences in the presence of short and long ambiguous queries, our graph-based algorithm does. For instance, as soon as $K = 3$ the MST algorithm obtains S-recall values of 38.72% and 60.46% on AMBIENT and MORESQUE, respectively. The difference decreases as K increases, but is still significant when $K = 15$. We hypothesize that, because they are less

ambiguous, longer queries are easier to diversify with the aid of WSI. However, we note that even with low values of K MST obtains higher S-recall than the other systems (with KeySRC competing on AMBIENT when $K \leq 10$).

5 Conclusions

We have presented a new approach to Web search result clustering. Key to our approach is the idea of inducing senses for the target query automatically by means of a simple, yet effective algorithm based on the maximum spanning tree of the cooccurrence graph. The results of a Web search engine are then mapped to the query senses and clustered accordingly.

The paper provides two contributions. First we corroborate our previous finding [27] that WSI greatly improves the quality of search result clustering as well as the diversification of the snippets returned as a flat list. We provide a clear indication on the usefulness of a loose notion of sense to cope with ambiguous queries. This is in contrast to research on Semantic Information Retrieval, which has obtained contradictory and often inconclusive results. The main advantage of WSI lies in its dynamic production of word senses that cover both concepts (e.g., *beagle* as a breed of dog) and instances (e.g., *beagle* as a specific instance of a space lander). In contrast, static dictionaries such as WordNet – typically used in Word Sense Disambiguation – by their very nature encode mainly concepts. Second, we propose a simple graph algorithm that induces the senses of our queries. Our algorithm has only a single parameter for the sense induction step.

Given the lack of ambiguous query datasets available [34], we hope our new dataset will be useful in future comparative experiments. Moreover, its requirement of a Web corpus of n-grams is not an onerous one, as such corpora are available for several languages and can be produced for any language of interest.

Acknowledgments. We thank Google for providing the Web1T corpus for research purposes; Massimiliano D'Amico for producing the output of KeySRC and EP; Stanislaw Osinski and Dawid Weiss for their help with Lingo and STC; Jim McManus for his useful comments on the original manuscript. The second author gratefully acknowledges the support of the ERC Starting Grant MultiJEDI No. 259234.

References

1. Agirre, E., Martínez, D., de Lacalle, O.L., Soroa, A.: Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In: Proc. of TextGraphs 2006, New York, USA, pp. 89–96 (2006)
2. Bennett, P.N., Nguyen, N.: Refined experts: improving classification in large taxonomies. In: Proc. of SIGIR 2009, Boston, MA, USA, pp. 11–18 (2009)
3. Bernardini, A., Carpineto, C., D'Amico, M.: Full-subtopic retrieval with keyphrase-based search results clustering. In: Proc. of WI 2009, Milan, Italy, pp. 206–213 (2009)
4. Brants, T., Franz, A.: Web 1t 5-gram, ver. 1, ldc2006t13. In: LDC, PA, USA (2006)
5. Carbonell, J., Goldstein, J.: The use of mmr, diversity-based reranking for reordering documents and producing summaries. In: Proc. of SIGIR 1998, Melbourne, Australia, pp. 335–336 (1998)

6. Carmel, D., Roitman, H., Zwerdling, N.: Enhancing cluster labeling using Wikipedia. In: Proc. of SIGIR 2009, MA, USA, pp. 139–146 (2009)
7. Carpineto, C., Osiński, S., Romano, G., Weiss, D.: A survey of web clustering engines. ACM Computing Surveys 41(3), 1–38 (2009)
8. Carpineto, C., Romano, G.: Exploiting the potential of concept lattices for information retrieval with CREDO. Journal of Universal Computer Science 10(8), 985–1013 (2004)
9. Chen, H., Karger, D.R.: Less is more: probabilistic models for retrieving fewer relevant documents. In: Proc. of SIGIR 2006, Seattle, WA, USA, pp. 429–436 (2006)
10. Chen, J., Zaïane, O.R., Goebel, R.: An unsupervised approach to cluster web search results based on word sense communities. In: Proc. of WI-IAT 2008, Sydney, Australia, pp. 725–729 (2008)
11. Cheng, D., Vempala, S., Kannan, R., Wang, G.: A divide-and-merge methodology for clustering. In: Proc. of PODS 2005, New York, NY, USA, pp. 196–205 (2005)
12. Crabtree, D., Gao, X., Andreae, P.: Improving web clustering by cluster selection. In: Proc. of WI 2005, Compiègne, France, pp. 172–178 (2005)
13. Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proc. of SIGIR 1992, Copenhagen, Denmark, pp. 318–329 (1992)
14. Di Giacomo, E., Didimo, W., Grilli, L., Liotta, G.: Graph visualization techniques for web clustering engines. IEEE Transactions on Visualization and Computer Graphics 13(2), 294–304 (2007)
15. Harris, Z.: Distributional structure. Word 10, 146–162 (1954)
16. Kamvar, M., Baluja, S.: A large scale study of wireless search behavior: Google mobile search. In: Proc. of CHI 2006, New York, NY, USA, pp. 701–709 (2006)
17. Ke, W., Sugimoto, C.R., Mostafa, J.: Dynamicity vs. effectiveness: studying online clustering for scatter/gather. In: Proc. of SIGIR 2009, MA, USA, pp. 19–26 (2009)
18. Krovetz, R., Croft, W.B.: Lexical ambiguity and Information Retrieval. ACM Transactions on Information Systems 10(2), 115–141 (1992)
19. Kurland, O.: The opposite of smoothing: a language model approach to ranking query-specific document clusters. In: Proc. of SIGIR 2008, Singapore, pp. 171–178 (2008)
20. Kurland, O., Domshlak, C.: A rank-aggregation approach to searching for optimal query-specific clusters. In: Proc. of SIGIR 2008, Singapore, pp. 547–554 (2008)
21. Lee, K.S., Croft, W.B., Allan, J.: A cluster-based resampling method for pseudo-relevance feedback. In: Proc. of SIGIR 2008, Singapore, pp. 235–242 (2008)
22. Lin, D.: Automatic retrieval and clustering of similar words. In: Proc. of the 17th COLING, Montreal, Canada, pp. 768–774 (1998)
23. Liu, S., Yu, C., Meng, W.: Word Sense Disambiguation in queries. In: Proc. of CIKM 2005, Bremen, Germany, pp. 525–532 (2005)
24. Mandala, R., Tokunaga, T., Tanaka, H.: The use of WordNet in Information Retrieval. In: Proc. of the COLING-ACL Workshop on Usage of Wordnet in Natural Language Processing, Montreal, Canada, pp. 31–37 (1998)
25. Miller, G.A., Beckwith, R.T., Fellbaum, C.D., Gross, D., Miller, K.: WordNet: an online lexical database. International Journal of Lexicography 3(4), 235–244 (1990)
26. Navigli, R.: Word Sense Disambiguation: a survey. ACM Computing Surveys 41(2), 1–69 (2009)
27. Navigli, R., Crisafulli, G.: Inducing word senses to improve web search result clustering. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP), Boston, USA, pp. 116–126 (2010)
28. Ngo, C.L., Nguyen, H.S.: A method of web search result clustering based on rough sets. In: Proc. of WI 2005, Compiègne, France, pp. 673–679 (2005)

29. Nguyen, C.-T., Phan, X.-H., Horiguchi, S., Nguyen, T.-T., Ha, Q.-T.: Web search clustering and labeling with hidden topics. ACM Transactions on Asian Language Information Processing 8(3), 1–40 (2009)
30. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems 20(3), 48–54 (2005)
31. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846–850 (1971)
32. van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworths (1979)
33. Sanderson, M.: Word Sense Disambiguation and Information Retrieval. In: Proc. of SIGIR 1994, Dublin, Ireland, pp. 142–151 (1994)
34. Sanderson, M.: Ambiguous queries: test collections need more sense. In: Proc. of SIGIR 2008, Singapore, pp. 499–506 (2008)
35. Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–124 (1998)
36. Schütze, H., Pedersen, J.: Information Retrieval based on word senses. In: Proceedings of SDAIR 1995, Las Vegas, Nevada, USA, pp. 161–175 (1995)
37. Stokoe, C., Oakes, M.J., Tait, J.I.: Word Sense Disambiguation in Information Retrieval revisited. In: Proc. of SIGIR 2003, Canada, pp. 159–166 (2003)
38. Swaminathan, A., Mathew, C.V., Kirovski, D.: Essential pages. In: Proc. of WI 2009, Milan, Italy, pp. 173–182 (2009)
39. Véronis, J.: HyperLex: lexical cartography for Information Retrieval. Computer Speech and Language 18(3), 223–252 (2004)
40. Voorhees, E.M.: Using WordNet to disambiguate word senses for text retrieval. In: Proc. of SIGIR 1993, Pittsburgh, PA, USA, pp. 171–180 (1993)
41. Widdows, D., Dorow, B.: A graph model for unsupervised lexical acquisition. In: Proc. of the 19th COLING, Taipei, Taiwan, pp. 1–7 (2002)
42. Maarek, Y., Ron Fagin, I.B.S., Pelleg, D.: Ephemeral document clustering for web applications. IBM Research Report RJ 10186 (2000)
43. Zamir, O., Etzioni, O.: Web document clustering: a feasibility demonstration. In: Proc. of SIGIR 1998, Melbourne, Australia, pp. 46–54 (1998)
44. Zamir, O., Etzioni, O., Madani, O., Karp, R.M.: Fast and intuitive clustering of web documents. In: Proc. of KDD 1997, Newport Beach, California, pp. 287–290 (1997)
45. Zhai, C., Cohen, W.W., Lafferty, J.: Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In: Proc. of SIGIR 2003, Toronto, Canada, pp. 10–17 (2003)
46. Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., Ma, W.-Y.: Improving web search results using affinity graph. In: Proc. of SIGIR 2005, Salvador, Brazil, pp. 504–511 (2005)
47. Zhang, X., Hu, X., Zhou, X.: A comparative evaluation of different link types on enhancing document clustering. In: Proc. of SIGIR 2008, Singapore, pp. 555–562 (2008)