Inducing Word Senses to Improve Web Search Result Clustering

Roberto Navigli and **Giuseppe Crisafulli**
Dipartimento di Informatica
Sapienza Università di Roma
navigli@di.uniroma1.it, crisafulli.giu@gmail.com

Abstract

In this paper, we present a novel approach to Web search result clustering based on the automatic discovery of word senses from raw text, a task referred to as Word Sense Induction (WSI). We first acquire the senses (i.e., meanings) of a query by means of a graph-based clustering algorithm that exploits cycles (triangles and squares) in the co-occurrence graph of the query. Then we cluster the search results based on their semantic similarity to the induced word senses. Our experiments, conducted on datasets of ambiguous queries, show that our approach improves search result clustering in terms of both clustering quality and degree of diversification.

1 Introduction

Over recent years increasingly huge amounts of text have been made available on the Web. Popular search engines such as Yahoo! and Google usually do a good job at retrieving a small number of relevant results from such an enormous collection of Web pages (i.e. retrieving with high precision, low recall). However, current search engines are still facing the lexical ambiguity issue (Furnas et al., 1987) – i.e. the linguistic property owing to which any particular word may convey different meanings. In a recent study (Sanderson, 2008) – conducted using WordNet (Miller et al., 1990) 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. Examples include: "buy B-52" (a cocktail? a bomber? a DJ workstation? tickets for a band?), "Alexander Smith quotes" (the novelist? the poet?), "beagle search" (dogs? the Linux search tool? the landing spacecraft?).

Ambiguity is often the consequence of the low number of query words entered on average by Web users (Kamvar and Baluja, 2006). While average query length is increasing – it is now estimated at around 3 words per query[1] – many search engines such as Google have already started to tackle the query ambiguity issue by reranking and diversifying their results, so as to prevent Web pages that are similar to each other from ranking too high on the list.

In the past few years, Web clustering engines (Carpineto et al., 2009) have been proposed as a solution to the lexical ambiguity issue 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. For instance, consider the following snippets returned for the *beagle search* query:

1. *Beagle* is a *search* tool that ransacks your...
2. ...the *beagle* disappearing in *search* of game...
3. *Beagle* indexes your files and *searches*...

While snippets 1 and 3 both concern the Linux search tool, they do not have any content word in

[1] http://www.hitwise.com/us/press-center/press-releases/google-searches-apr-09

Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 116–126,
MIT, Massachusetts, USA, 9-11 October 2010.

common except our query words. As a result, they will most likely be assigned to two different clusters.

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 referred to as Word Sense Induction (WSI). At the core of our approach is a graph-based algorithm that exploits cycles in the co-occurrence graph of the input query to detect the query's meanings. 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. A first, historical solution to query ambiguity is that of Web directories, that is taxonomies providing categories to which Web pages are manually assigned (e.g., the Open Directory Project – `http://dmoz.org`). 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 artists with the same surname classified as `Arts:Music:Bands and Artists`). While methods for the automatic classification of Web documents have been proposed (e.g., (Liu et al., 2005b; Xue et al., 2008)) and some problems have been effectively tackled (Bennett and Nguyen, 2009), these approaches are usually supervised and still suffer from relying on a predefined taxonomy of categories.

Semantic Information Retrieval (SIR). 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 Navigli (2009)). SIR is performed by indexing and/or searching concepts rather than terms, thus potentially coping with two linguistic phenomena: expressing a single meaning with different words (*synonymy*) and using the same word to express various different meanings (*polysemy*). Over the years, different methods for SIR have been proposed (Krovetz and Croft, 1992; Voorhees, 1993; Mandala et al., 1998; Gonzalo et al., 1999; Kim et al., 2004; Liu et al., 2005a, inter alia). However, contrasting results have been reported on the benefits of these techniques: it has been shown that WSD has to be very accurate to benefit Information Retrieval (Sanderson, 1994) – a result that was later debated (Gonzalo et al., 1999; Stokoe et al., 2003). Also, it has been reported that WSD has to be very precise on minority senses and uncommon terms, rather than on frequent words (Krovetz and Croft, 1992; Sanderson, 2000).

SIR relies on the existence of a reference dictionary to perform WSD (typically, WordNet) and thus suffers from its static nature 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).

Search Result Clustering. A more popular approach to query ambiguity is that of search result clustering. Typically, given a query, the system starts from a flat list of text snippets returned from one or more commonly-available search engines and clusters them on the basis of some notion of textual similarity. At the root of the clustering approach lies van Rijsbergen's (1979) 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 (Carpineto et al., 2009). The former focus more on the problem of data clustering than on presenting the results to the user. A pioneering example is Scatter/Gather (Cutting et al., 1992), 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 (Ke et al., 2009). Other data-centric approaches use agglomerative hierarchical clustering (e.g., LASSI (Yoelle Maarek and Pelleg, 2000)), rough sets (Ngo and Nguyen, 2005) or exploit link information (Zhang et al., 2008).

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 (Zamir et al., 1997; Zamir and Etzioni, 1998), including later developments (Crabtree et al., 2005; Bernardini et al., 2009). Other methods in the literature are based on formal concept analysis (Carpineto and Romano, 2004), singular value decomposition (Osinski and Weiss, 2005), spectral clustering (Cheng et al., 2005), spectral geometry (Liu et al., 2008), link analysis (Gelgi et al., 2007), and graph connectivity measures (Di Giacomo et al., 2007). Search result clustering has also been viewed as a supervised salient phrase ranking task (Zeng et al., 2004).

Diversification. Another recent research topic dealing with the query ambiguity issue is diversification, which aims to rerank top search results based on criteria that maximize their diversity. One of the first examples of diversification algorithms is based on the use of similarity functions to measure the diversity among documents and between document and query (Carbonell and Goldstein, 1998). Other techniques use conditional probabilities to determine which document is most different from higher-ranking ones (Chen and Karger, 2006) or use affinity ranking (Zhang et al., 2005), based on topic variance and coverage. More recently, an algorithm called Essential Pages (Swaminathan et al., 2009) has been proposed to reduce information redundancy and return Web pages that maximize coverage with respect to the input query.

Word Sense Induction (WSI). In contrast to the above approaches, we perform WSI to dynamically acquire an inventory of senses of the input query. Instead of performing clustering on the basis of the surface similarity of Web snippets, we use our induced word senses to group snippets. Very little work on this topic exists: vector-based WSI was successfully shown to improve bag-of-words ad-hoc Information Retrieval (Schütze and Pedersen, 1995) and preliminary studies (Udani et al., 2005; Chen et al., 2008) have provided interesting insights into the use of WSI for Web search result clustering. A more recent attempt at automatically identifying query meanings is based on the use of hidden topics (Nguyen et al., 2009). However, in this approach topics – estimated from a universal dataset – are query-independent and thus their number needs to be established beforehand. In contrast, we aim to cluster snippets based on a dynamic and finer-grained notion of sense.

3 Approach

Web search result clustering is usually performed in three main 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 Zamir and Etzioni (1998) and Carmel et al. (2009)) for visualization purposes.

Our key idea is to improve step 2 by means of a Word Sense Induction algorithm: given a query q, we first dynamically induce, from a text corpus, the set of word senses of q (Section 3.1); next, we cluster the Web results on the basis of the word senses previously induced (Section 3.2).

3.1 Word Sense Induction

Word Sense Induction algorithms are unsupervised techniques aimed at automatically identifying the set of senses denoted by a word. These methods induce word senses from text by clustering word occurrences based on the idea that a given word – used in a specific sense – tends to co-occur with the same neighbouring words (Harris, 1954). Several approaches to WSI have been proposed in the literature (see Navigli (2009) for a survey), ranging from clustering based on context vectors (e.g., Schütze (1998)) to word clustering (e.g., Lin (1998)) and co-occurrence graphs (e.g., Widdows and Dorow (2002)).

Successful approaches such as HyperLex (Véronis, 2004) – a graph algorithm based on the identification of hubs in co-occurrence graphs – have to cope with a high number of parameters to be tuned (Agirre et al., 2006). To deal with this issue we propose two variants of a simple, yet effective, graph-based algorithm for WSI, that we

describe hereafter. The algorithm consists of two steps: graph construction and identification of word senses.

3.1.1 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 (Brants and Franz, 2006), 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 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 rationale behind Dice is that dividing by the sum of total counts of the two words drastically decreases the ranking of words that tend to co-occur frequently with many other words (e.g., *new*, *old*, *nice*, etc.).

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 $Dice(q, w) \geq \delta$ (the threshold δ is established experimentally, see Section 4.1). We set $V := V^{(0)}$ and $E := \emptyset$.

- For each word $w \in V^{(0)}$, we select the highest ranking words co-occurring with w in Web1T, that is those words w' for which $Dice(w, w') \geq \delta$. We add each of these words to V (note that some w' might already be in $V^{(0)}$) and the corresponding edge $\{w, w'\}$ to E with weight $Dice(w, w')$. Finally, we remove disconnected vertices.

3.1.2 Identification of word senses

The main idea behind our approach is that edges in the co-occurrence graph participating in cycles are likely to connect vertices (i.e., words) belonging to the same meaning component. Specifically, we focus on cycles of length 3 and 4, called respectively triangles and squares in graph theory.

For each edge e, we calculate the ratio of triangles in which e participates:

$$\textit{Tri}(e) = \frac{\text{\# triangles } e \text{ participates in}}{\text{\# triangles } e \text{ could participate in}} \quad (2)$$

where the numerator is the number of cycles of length 3 in which $e = \{w, w'\}$ participates, and the denominator is the total number of neighbours of w and w'. Similarly, we define a measure $Sqr(e)$ of the ratio of squares (i.e., cycles of length 4) an edge e participates in to the number of possible squares e could potentially participate in:

$$\textit{Sqr}(e) = \frac{\text{\# squares } e \text{ participates in}}{\text{\# squares } e \text{ could participate in}} \quad (3)$$

where the numerator is the number of squares containing e and the denominator is the number of possible distinct pairs of neighbours of w and w'. If no triangle (or square) exists for e, the value of the corresponding function is set to 0.

In order to disconnect the graph and determine the meaning components, we remove all the edges whose *Tri* (or *Sqr*) value is below a threshold σ. The resulting connected components represent the word senses induced for the query q. Notice that the number of senses is dynamically chosen based on the co-occurrence graph and the algorithm's thresholds.

Our triangular measure is the edge counterpart of the clustering coefficient (or curvature) for vertices, previously used to perform WSI (Widdows and Dorow, 2002). However, it is our hunch that measuring the ratio of squares an edge participates in provides a stronger clue of how important that edge is within a meaning component. In Section 4, we will corroborate this idea with our experiments.

3.1.3 An example

As an example, let $q =$ *beagle*. Two steps are performed:

1. **Graph construction.** We build the co-occurrence graph $G_{beagle} = (V, E)$, an excerpt of which is shown in Figure 1(a).

2. **Identification of word senses.** We calculate the *Sqr* values of each edge in the graph. The edges e whose $Sqr(e) < \sigma$ are removed (we assume $\sigma = 0.25$). For instance, $Sqr(\{$ *dog, breed* $\}) = \frac{1}{2}$, as the edge participates in the square *dog – breed – puppy – canine – dog*, but it could also have participated in the potential square *dog – breed – puppy – search – dog*. In fact, the other neighbours of *dog* are *canine*, *puppy* and *search*, and the other neighbour of *breed* is *puppy*, thus the square can only be closed by connecting *puppy* to either *canine* or *search*. In our example, the only edges whose *Sqr* is below σ are: { *dog, puppy* }, { *dog, search* } and { *linux, mission* } (they participate in no square). We remove these edges and select the resulting connected components as the senses of the query *beagle* (shown in Figure 1(b)). Note that, if we selected triangles as our pruning measure, we should also remove the following edges { *search, index* }, { *index, linux* }, { *linux, system* } and { *system, search* }. In fact, these edges do not participate in any triangle (while they do participate in a square). As a result, we would miss the computer science sense of the query.

3.2 Clustering of Web results

Given our query q, we submit it to a search engine, which returns a list of relevant search results $R = (r_1, \ldots, r_n)$. We process each result r_i by considering the corresponding text snippet and transforming it to a bag of words b_i (we apply tokenization, stopwords and target word removal, and lemmatization[2]). 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, 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

[2]We use the WordNet lemmatizer.

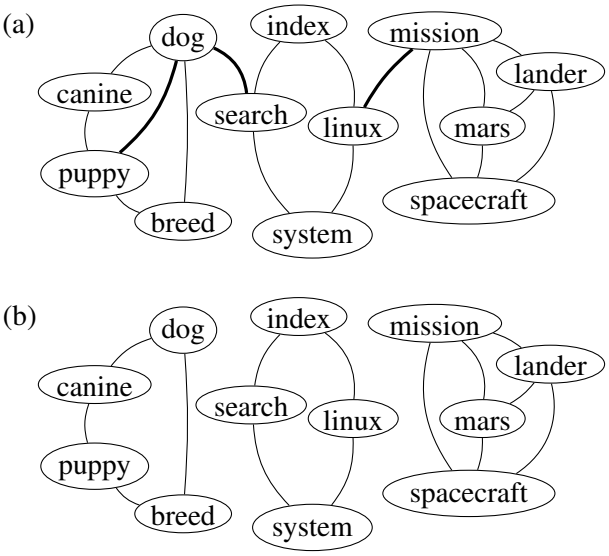

Figure 1: The *beagle* example: (a) graph construction, "weak" edges (according to *Sqr*) drawn in bold, (b) the word senses induced after edge removal.

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:

$$Sense(r_i) = \begin{cases} \underset{j=1,\ldots,m}{\operatorname{argmax}} |b_i \cap S_j| & \text{if } \max_j |b_i \cap S_j| > 0 \\ 0 & \text{else} \end{cases} \quad (4)$$

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:

$$C_j = \{r_i \in R : Sense(r_i) = j\}, \quad (5)$$

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:

$$avgsim(C_j, S_j) = \frac{\sum_{r_i \in C_j} sim(r_i, S_j)}{|C_j|}. \quad (6)$$

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:

$$sim(r_i, S_j) = sim(b_i, S_j) = \frac{|b_i \cap S_j|}{|b_i|}. \quad (7)$$

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 (Crabtree et al., 2005; Kurland, 2008; Kurland and Domshlak, 2008; Lee et al., 2008, 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 recently released dataset which contains 44 ambiguous queries[3]. The sense inventory for the meanings (i.e., subtopics)[4] of queries is given by Wikipedia disambiguation pages. For instance, given the *beagle* query, its disambiguation page in Wikipedia provides the meanings of dog, Mars lander, computer search service, beer brand, etc. The top 100 Web results of each query returned by the Yahoo! search engine were tagged with the most appropriate query senses according to Wikipedia (amounting to 4400 sense-annotated search results). To our knowledge, this is currently the largest dataset of ambiguous queries available on-line. Other datasets, such as those from the TREC competitions, are not focused on distiguishing the subtopics of a query.

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

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

dataset	queries	queries by length				avg. polys.
		1	2	3	4	
AMBIENT	44	35	6	3	0	17.9
MORESQUE	114	0	47	36	31	6.7

Table 1: Statistics on the datasets of ambiguous queries.

- 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 its authors. In fact, our aim was to study the behaviour of Web search algorithms on queries of different lengths, ranging from 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). We decided to carry on using Yahoo! mainly for homogeneity reasons.

We report the statistics on the composition of the two datasets in Table 1. Given that the snippets could possibly be annotated with more than one Wikipedia subtopic, we also determined the average number of subtopics per snippet. This amounted to 1.01 for AMBIENT and 1.04 for MORESQUE for snippets with at least one subtopic annotation (51% and 53% of the respective datasets). We can thus conclude that multiple subtopic annotations are infrequent.

Parameters. Our graph-based algorithms have two parameters: the Dice threshold δ for graph construction (Section 3.1.1) and the threshold σ for edge removal (Section 3.1.2). The best parameters, used throughout our experiments, were $(\delta = 0.00033, \sigma = 0.45)$ with triangles and $(\delta = 0.00033, \sigma = 0.33)$ with squares. The parameter values were obtained as a result of tuning on a small in-house development dataset. The dataset was built by automatically identifying monosemous words and creating pseudowords following the scheme proposed by Schütze (1998).

Systems. We compared Triangles and Squares against the best systems reported by Bernardini et al. (2009, cf. Section 2):

- **Lingo** (Osinski and Weiss, 2005): a Web clustering engine implemented in the Carrot[2] open-source framework[5] that clusters the most frequent phrases extracted using suffix arrays.
- **Suffix Tree Clustering (STC)** (Zamir and Etzioni, 1998): the original Web search clustering approach based on suffix trees.
- **KeySRC** (Bernardini et al., 2009): 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)** (Swaminathan et al., 2009): 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 (Rand, 1971; Manning et al., 2008):

$$\text{RI}(\mathcal{C}) = \frac{\sum_{(w,w')\in\mathcal{W}\times\mathcal{W},w\neq w'} \delta(w,w')}{|\{(w,w') \in \mathcal{W}\times\mathcal{W} : w \neq w'\}|} \quad (8)$$

where $\mathcal{W}$ is the union set of all the words in $\mathcal{C}$ and $\delta(w,w') = 1$ if any two words w and w' are in the same cluster both in $\mathcal{C}$ and in the gold standard $\mathcal{G}$ or they are in two different clusters in both $\mathcal{C}$ and $\mathcal{G}$, otherwise $\delta(w,w') = 0$. In other words, we calculate the percentage of word pairs that are in the same configuration 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 (i.e., each cluster contains the snippets manually associated with a particular Wikipedia subtopic). Similarly to what was done in Section 3.2, untagged results are grouped together in a special cluster of $\mathcal{G}$.

[5] http://project.carrot2.org

System	AMBIENT	MORESQUE	All
Squares	72.59	65.41	67.28
Triangles	66.13	64.47	64.93
Lingo	62.75	52.68	55.49
STC	61.48	51.52	54.29
KeySRC	66.49	55.82	58.78

Table 2: Results by Rand index (percentages).

Results. The results of all systems on the AMBIENT and MORESQUE datasets according to the average Rand index are shown in Table 2[6]. In accordance with previous results in the literature, KeySRC performed generally better than the other search result clustering systems, especially on smaller queries. Our Word Sense Induction systems, Squares and Triangles, outperformed all other systems by a large margin, thus showing a higher clustering quality (with the exception of KeySRC performing better than Triangles on AMBIENT). Interestingly, all clustering systems perform more poorly on longer queries (i.e., on the MORESQUE dataset), however our WSI systems, and especially Triangles, are more robust across query lengths. Compared to Triangles, the Squares algorithm performs better, confirming our hunch that Squares is a more solid graph pattern.

4.3 Experiment 2: Diversification

Measure. Search result clustering can also be used to diversify the top results returned by a search engine. Thus, for each query q, one natural way of measuring a system's performance is to calculate the **subtopic recall-at-**K (Zhai et al., 2003) given by the number of different subtopics retrieved for q in the top K results returned:

$$\text{S-recall@K} = \frac{|\bigcup_{i=1}^{K} subtopics(r_i)|}{M} \quad (9)$$

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

[6] For reference systems we used the implementations of Bernardini et al. (2009) and Osinski and Weiss (2005).

System	K=3	K=5	K=10	K=15	K=20
Squares	51.9	63.4	75.8	83.3	87.4
Triangles	50.8	62.4	75.2	82.7	86.6
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

Table 3: S-recall@K on all queries (percentages).

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 7 introduced in Section 3.2).

Results. For the sake of clarity and to save space, we selected the best systems from our previous experiment, namely Squares, Triangles and KeySRC, and compared their output 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 3. Squares and Triangles show the highest degree of diversification, with a subtopic recall greater than all other systems, and with Squares consistently performing better than Triangles. It is interesting to observe that KeySRC performs 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 ≥ 3), we studied the S-recall@K trend for the two datasets. The results are 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 algorithms do. For instance, as soon as $K = 3$ the Squares algorithm obtains S-recall values of 37% and 57.5% on AMBIENT and MORESQUE, respectively. The

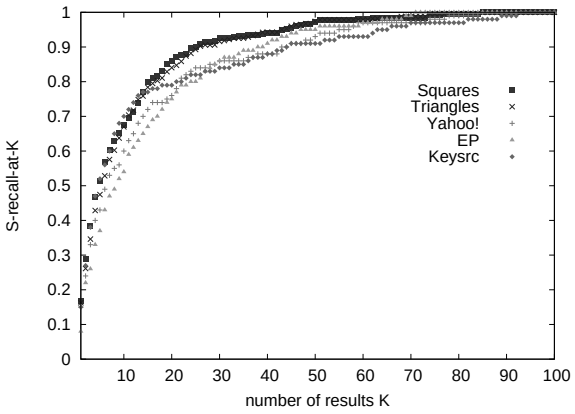

Figure 2: Results by S-recall@K on AMBIENT.

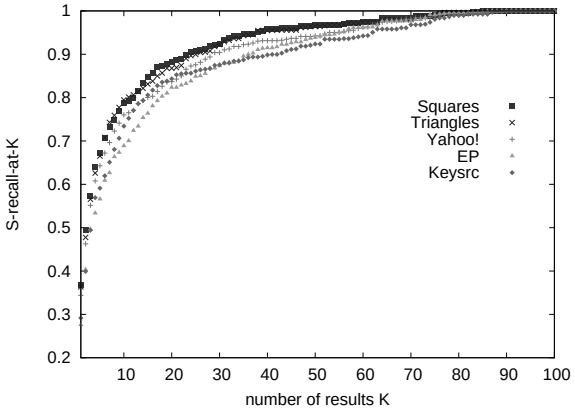

Figure 3: S-recall@K on MORESQUE.

difference decreases as K increases, but is still significant when K = 10. 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, Squares and Triangles obtain higher S-recall than the other systems (with KeySRC competing on AMBIENT when $K \leq$ 15). Finally, we observe that – with low values of K – the Squares algorithm performs significantly better than Triangles on shorter queries, and only slightly better on longer ones.

5 Discussion

Results. Our results show that our graph-based algorithms are able to consistently produce clusters of better quality than all other systems tested in our experiments. The results on S-recall@K show that our approach can also be used effectively as a diversification technique, performing better than a very

recent proposal such as Essential Pages. The latter outperforms Yahoo! and KeySRC when $K \geq 30$ on AMBIENT, whereas on MORESQUE it performs generally worse until higher values of K are reached. If we analyze the entire dataset of 158 queries by length, EP works best after examining at least 20 results on 1- and 2-word ambiguous queries, whereas on longer queries a larger number of documents (≥ 30) needs to be analyzed before surpassing Yahoo! performance.

The above considerations might not seem intuitive at first glance, as the average polysemy of longer queries is lower (17.9 on AMBIENT vs. 6.7 on MORESQUE according to our gold standard). However, we note that while the kind of ambiguity of 1-word queries is generally coarser (e.g., *beagle* as dog vs. lander vs. search tool), with longer queries we often encounter much finer sense distinctions (e.g., *Across the Universe* as song by The Beatles vs. a 2007 film based on the song vs. a Star Trek novel vs. a rock album by Trip Shakespeare, etc.). Word Sense Induction is able to deal better with this latter kind of ambiguity as discriminative words become part of the meanings acquired.

Performance issues. Inducing word senses from the query graph comes at a higher computational cost than other non-semantic clustering techniques. Indeed, the most time-consuming phase of our approach is the construction of the query graph, which requires intensive querying of our database of co-occurrences calculated from the Web1T corpus. While graphs can be precomputed or cached, previously unseen queries will still require the construction of new graphs. Instead, triangles and squares, as well as the resulting connected components, can be calculated on the fly.

6 Conclusions

In this paper we have presented a novel approach to Web search result clustering. Our key idea is to induce senses for the target query automatically by means of a graph-based algorithm focused on the notion of cycles. The results of a Web search engine are then mapped to the query senses and clustered accordingly.

The paper provides three novel contributions. First, we show that WSI boosts the quality of search result clustering and improves 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 two simple, yet effective, graph algorithms to induce the senses of our queries. The best performing approach is based on squares (cycles of length 4), a novel graph pattern in WSI. Third, we contribute a new dataset of 114 ambiguous queries and 11,400 sense-annotated snippets which complements an existing dataset of ambiguous queries[7]. Given the lack of ambiguous query datasets available (Sanderson, 2008), we hope our new dataset will be useful in future comparative experiments. Finally, we note that our approach needed very little tuning. Moreover, its requirement of a Web corpus of n-grams is not a stringent one, as such corpora are available for several languages and can be produced for any language of interest.

As regards future work, we intend to combine our clustering algorithm with a cluster labeling algorithm. We also aim to implement a number of Word Sense Induction algorithms and compare them in the same evaluation framework with more Web search and Web clustering engines. Finally, it should be possible to use precisely the same approach presented in this paper for document clustering, by grouping the contexts in which the target query occurs – and we will also experiment on this in the future.

Acknowledgments

We thank Google for providing the Web1T corpus for research purposes. We also thank Massimiliano D'Amico for producing the output of KeySRC and EP, and Stanislaw Osinski and Dawid Weiss for their

[7]The MORESQUE dataset is available at the following URL: `http://lcl.uniroma1.it/moresque`

help with Lingo and STC. Additional thanks go to Jim McManus, Senja Pollak and the anonymous reviewers for their useful comments.

References

Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Evaluating and optimizing the parameters of an unsupervised graph-based WSD algorithm. In *Proc. of TextGraphs '06*, pages 89–96, New York, USA.

Paul N. Bennett and Nam Nguyen. 2009. Refined experts: improving classification in large taxonomies. In *Proc. of SIGIR '09*, pages 11–18, Boston, MA, USA.

Andrea Bernardini, Claudio Carpineto, and Massimiliano D'Amico. 2009. Full-subtopic retrieval with keyphrase-based search results clustering. In *Proc. of WI '09*, pages 206–213, Milan, Italy.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram, ver. 1, ldc2006t13. In *LDC*, PA, USA.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proc. of SIGIR '98*, pages 335–336, Melbourne, Australia.

David Carmel, Haggai Roitman, and Naama Zwerdling. 2009. Enhancing cluster labeling using Wikipedia. In *Proc. of SIGIR '09*, pages 139–146, MA, USA.

Claudio Carpineto and Giovanni Romano. 2004. Exploiting the potential of concept lattices for information retrieval with CREDO. *Journal of Universal Computer Science*, 10(8):985–1013.

Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. 2009. A survey of web clustering engines. *ACM Computing Surveys*, 41(3):1–38.

Harr Chen and David R. Karger. 2006. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proc. of SIGIR '06*, pages 429–436, Seattle, WA, USA.

Jiyang Chen, Osmar R. Zaïane, and Randy Goebel. 2008. An unsupervised approach to cluster web search results based on word sense communities. In *Proc. of WI-IAT 2008*, pages 725–729, Sydney, Australia.

David Cheng, Santosh Vempala, Ravi Kannan, and Grant Wang. 2005. A divide-and-merge methodology for clustering. In *Proc. of PODS '05*, pages 196–205, New York, NY, USA.

Daniel Crabtree, Xiaoying Gao, and Peter Andreae. 2005. Improving web clustering by cluster selection. In *Proc. of WI '05*, pages 172–178, Compiègne, France.

Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. of SIGIR '92*, pages 318–329, Copenhagen, Denmark.

Emilio Di Giacomo, Walter Didimo, Luca Grilli, and Giuseppe Liotta. 2007. Graph visualization techniques for web clustering engines. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):294–304.

George W. Furnas, Thomas K. Landauer, Louis Gomez, and Susan Dumais. 1987. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Fatih Gelgi, Hasan Davulcu, and Srinivas Vadrevu. 2007. Term ranking for clustering web search results. In *Proc. of WebDB '07*, Beijing, China.

Julio Gonzalo, Anselmo Penas, and Felisa Verdejo. 1999. Lexical ambiguity and Information Retrieval revisited. In *Proc. of EMNLP/VLC 1999*, pages 195–202, College Park, MD, USA.

Zellig Harris. 1954. Distributional structure. *Word*, 10:146–162.

Maryam Kamvar and Shumeet Baluja. 2006. A large scale study of wireless search behavior: Google mobile search. In *Proc. of CHI '06*, pages 701–709, New York, NY, USA.

Weimao Ke, Cassidy R. Sugimoto, and Javed Mostafa. 2009. Dynamicity vs. effectiveness: studying online clustering for scatter/gather. In *Proc. of SIGIR '09*, pages 19–26, MA, USA.

Sang-Bum Kim, Hee-Cheol Seo, and Hae-Chang Rim. 2004. Information Retrieval using word senses: root sense tagging approach. In *Proc. of SIGIR '04*, pages 258–265, Sheffield, UK.

Robert Krovetz and William B. Croft. 1992. Lexical ambiguity and Information Retrieval. *ACM Transactions on Information Systems*, 10(2):115–141.

Oren Kurland and Carmel Domshlak. 2008. A rank-aggregation approach to searching for optimal query-specific clusters. In *Proc. of SIGIR '08*, pages 547–554, Singapore.

Oren Kurland. 2008. The opposite of smoothing: a language model approach to ranking query-specific document clusters. In *Proc. of SIGIR '08*, pages 171–178, Singapore.

Kyung Soon Lee, W. Bruce Croft, and James Allan. 2008. A cluster-based resampling method for pseudo-relevance feedback. In *Proc. of SIGIR '08*, pages 235–242, Singapore.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of the* 17^{th} *COLING*, pages 768–774, Montreal, Canada.

Shuang Liu, Clement Yu, and Weiyi Meng. 2005a. Word Sense Disambiguation in queries. In *Proc. of CIKM '05*, pages 525–532, Bremen, Germany.

Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma. 2005b. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, 7(1):36–43.

Ying Liu, Wenyuan Li, Yongjing Lin, and Liping Jing. 2008. Spectral geometry for simultaneously clustering and ranking query search results. In *Proc. of SIGIR '08*, pages 539–546, Singapore.

Rila Mandala, Takenobu Tokunaga, and Hozumi Tanaka. 1998. The use of WordNet in Information Retrieval. In *Proc. of the COLING-ACL workshop on Usage of Wordnet in Natural Language Processing*, pages 31–37, Montreal, Canada.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

George A. Miller, Richard T. Beckwith, Christiane D. Fellbaum, Derek Gross, and Katherine Miller. 1990. WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

Roberto Navigli. 2009. Word Sense Disambiguation: a survey. *ACM Computing Surveys*, 41(2):1–69.

Chi Lang Ngo and Hung Son Nguyen. 2005. A method of web search result clustering based on rough sets. In *Proc. of WI '05*, pages 673–679, Compiègne, France.

Cam-Tu Nguyen, Xuan-Hieu Phan, Susumu Horiguchi, Thu-Trang Nguyen, and Quang-Thuy Ha. 2009. Web search clustering and labeling with hidden topics. *ACM Transactions on Asian Language Information Processing*, 8(3):1–40.

Stanislaw Osinski and Dawid Weiss. 2005. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Mark Sanderson. 1994. Word Sense Disambiguation and Information Retrieval. In *Proc. of SIGIR '94*, pages 142–151, Dublin, Ireland.

Mark Sanderson. 2000. Retrieving with good sense. *Information Retrieval*, 2(1):49–69.

Mark Sanderson. 2008. Ambiguous queries: test collections need more sense. In *Proc. of SIGIR '08*, pages 499–506, Singapore.

Hinrich Schütze and Jan Pedersen. 1995. Information Retrieval based on word senses. In *Proceedings of SDAIR'95*, pages 161–175, Las Vegas, Nevada, USA.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Christopher Stokoe, Michael J. Oakes, and John I. Tait. 2003. Word Sense Disambiguation in Information Retrieval revisited. In *Proc. of SIGIR '03*, pages 159–166, Canada.

Ashwin Swaminathan, Cherian V. Mathew, and Darko Kirovski. 2009. Essential pages. In *Proc. of WI '09*, pages 173–182, Milan, Italy.

Goldee Udani, Shachi Dave, Anthony Davis, and Tim Sibley. 2005. Noun sense induction using web search results. In *Proc. of SIGIR '05*, pages 657–658, Salvador, Brazil.

Cornelis Joost van Rijsbergen. 1979. *Information Retrieval*. Butterworths, second edition.

Jean Véronis. 2004. HyperLex: lexical cartography for Information Retrieval. *Computer Speech and Language*, 18(3):223–252.

Ellen M. Voorhees. 1993. Using WordNet to disambiguate word senses for text retrieval. In *Proc. of SIGIR '93*, pages 171–180, Pittsburgh, PA, USA.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proc. of the* 19^{th} *COLING*, pages 1–7, Taipei, Taiwan.

Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu. 2008. Deep classification in large-scale text hierarchies. In *Proc. of SIGIR '08*, pages 619–626, Singapore.

Israel Ben-Shaul Yoelle Maarek, Ron Fagin and Dan Pelleg. 2000. Ephemeral document clustering for web applications. *IBM Research Report RJ 10186*.

Oren Zamir and Oren Etzioni. 1998. Web document clustering: a feasibility demonstration. In *Proc. of SIGIR '98*, pages 46–54, Melbourne, Australia.

Oren Zamir, Oren Etzioni, Omid Madani, and Richard M. Karp. 1997. Fast and intuitive clustering of web documents. In *Proc. of KDD '97*, pages 287–290, Newport Beach, California.

Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. 2004. Learning to cluster web search results. In *Proc. of SIGIR '04*, pages 210–217, Sheffield, UK.

ChengXiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proc. of SIGIR '03*, pages 10–17, Toronto, Canada.

Benyu Zhang, Hua Li, Yi Liu, Lei Ji, Wensi Xi, Weiguo Fan, Zheng Chen, and Wei-Ying Ma. 2005. Improving web search results using affinity graph. In *Proc. of SIGIR '05*, pages 504–511, Salvador, Brazil.

Xiaodan Zhang, Xiaohua Hu, and Xiaohua Zhou. 2008. A comparative evaluation of different link types on enhancing document clustering. In *Proc. of SIGIR '08*, pages 555–562, Singapore.