Finding synonyms and other semantically-similar terms from coselection data

Glyn Caon[1], Mark Truran[2] and Helen Ashman[1]

[1] Computer and Information Science
University of South Australia
Mawson Lakes, SA 5095, Australia

[2] School of Computer Science
University of Teesside
Middlesbrough, UK

caogy001@mymail.unisa.edu.au

m.a.truran@tees.ac.uk

helen.ashman@unisa.edu.au

Abstract

Clickthrough data has been proposed for numerous uses, and this paper describes how a special form of clickthough data, coselection data, can form non-ambiguous clusters that can then be used to detect semantic similarity between query terms. This semantic similarity assessment can be applied to distinct terms in the same language, giving rise to synonyms, or in different languages, indicating possible translations. It can determine alternative names or descriptors for items which do not occur in traditional thesauri, such as phrases, proper nouns or technical terms. The semantic similarity is calculated without any use of external reference materials and without any analysis of content.

Keywords: Synonym discovery, implicit association, implicit relevance feedback, web search analysis.

1 Introduction

Ashman et al. (2007) conjectured that investigating the semantic similarity (or overlap) between clusters can give rise to a number of useful tools, such as synonym and translation discovery. In this paper we seek to prove this conjecture with an experiment performed over a significant collection of real user data.

Detecting semantic similarity between terms is a challenging task, often requiring the use of external reference materials such as dictionaries, parallel corpora and thesauri. In this paper, we discuss a method for discovering semantic similarity between query terms sent by users to search engines. The underlying principle of the method was shown by Ashman et al. (2011) to be feasible. Since then, further investigation has shown that the method can indeed detect semantic similarity, with scope bounded only by the queries made by the users themselves. In the data used for the current experiment, 'synonyms'[1] have been found for proper nouns, including product names, technical terms and multi-word phrases, and even in one case for a cross-language term, i.e. a translation.

1.1 Coselections

This method is based on the implicit relevance judgements that users make when selecting from the results of searches. As such, the semantic similarity detected is a byproduct of organic user search. This coselection data is a special form of clickthrough data and is an implied semantic association between a search term and a URL. Coselection data occurs when the user has made more than one selection from the results, so that there is not only the semantic association between the search term and each of the selected URLs, but there is additionally an implied semantic association between any two of the selected URLs. That is, the user is assumed to have had a single purpose in mind when making selections from the result set, and will not be distracted by irrelevant or ambiguous results (Ashman et al. (2011) evaluated this assumption and found it reliable).

Coselections thus form a similarity measure between URLs, which does not rely on a transitive association between URLs via a query term in common. Coselection represents the existence of a non-ambiguous semantic relationship between URLs. A number of coselections between any two URLs indicates semantic similarity between them. Note however that the converse is not true - a lack of coselections does not imply a negative result, i.e. that any two URLs are unrelated semantically. This is because users are not compelled to choose all relevant URLs from a set of search results, and they may have not selected a specific result because another result met their information need, or because of many other reasons.

Finally, a key observation about coselections is that they manifest semantic singularity, because users generally select results only for one sense of the term they search on (this is discussed further below). This sense-singularity means that clusters formed by using coselections as part of the similarity measure are going to be non-ambiguous (Ashman et al. 2011, Truran et al. 2005). Hence any cluster comparison is not going to be confounded by ambiguity, which is recognised as a significant hurdle in automatic translation methods (see 2.1 below).

[1] From this point, we will use 'synonym' to mean any form of semantic similarity between distinct terms, of any length, whether in an external reference or not, and over any language.

1.2 Data sources

The main limitation of the method lies in the data input into the process. The method takes Web search logs, and extracts coselections. However it can only extract coselections for those query terms chosen by the user and can then only find semantic similarity across search terms occurring in the underlying data. On the other hand, the limitations of any one dataset need not preclude the formation of a more comprehensive collection of synonyms, which could be achieved by aggregation of numerous smaller collections generated from different data sources, without those raw data sources needing to ever be shared.

The method also requires 'enough' coselections for any given term in order to be able to form and subsequently use meaningful clusters. In prior experimentation using the Microsoft RFP 2006 clickthrough data collection, we found too few coselections in the raw data to be able to form useful clusters (Smith et al. 2009). In contrast, the data used in this work has a high proportion of clicks although over a limited number of search terms, making it feasible to create significant clusters. This is attributable to the specific user community from which the data is sourced, being the set of all Web searches and selections from a Computer Science school of a UK university for over six years ("the Teesside data"). There were numerous searches on the same topics by many students seeking information for completing assignments. While this user community naturally manifests a low coverage of topics (predominantly assignment-related ones) the click rate and hence the coselection rate is very satisfactory. Also there is often no temporal disalignment between URLs selected, as the users tend to all perform the searches at the same time, due to assignment deadlines. These characteristics have made it possible to perform the experiments reported here and elsewhere (Ashman et al. 2011).

2 Related Work

In this section, related work on semantic similarity detection is discussed. Ashman et al. (2011) details related work on clickthrough data and will not be repeated here.

2.1 Semantic similarity detection

Much work on detecting semantic similarity has occurred in cross-language information retrieval and querying with some work specifically focusing on synonym detection. This section highlights the primary points while Zhou et al. (to appear) provides a more extensive survey.

2.1.1 Synonym discovery

The closest work to that of this paper is that which proposes to use clickthrough data for synonym discovery. The discovery of synonyms using search logs was first proposed by Beeferman and Berger (2000). Search query clustering mines query logs to provide a measure of similarity between queries (Beeferman and Berger, 2000, Cui et al. 2003, Gao et al. 2007, Wen et al., 2002 and Xue et al. 2004). This is achieved by mining clickthough data, i.e. a query is said to be related to a document if it is selected as a result of a search. If two queries have enough of the same resources linked to them via clickthrough data then the two terms are then deemed to be similar (Wen et al. 2002) - this uses overlap between the two collections of URLs to determine their similarity. A “live thesaurus” is then developed by applying a threshold ranking based on the similarity metric (Gao et al. 2007) or by using the query-document and document-document term correlations to link the query with document terms (Cui et al. 2007).

This use of clickthrough data employs a very similar principle to that of this paper, namely to aggregate URLs according to their relevance to a query term, then to compare those aggregations for common URLs. However there is with one key difference. Normal clickthrough data cannot distinguish between ambiguous terms and as a result will create clusters of URLs where URLs are not all mutually relevant, in particular where the query term of the clusters is an ambiguous term. In contrast, Ashman et al. (2011) demonstrated that coselections represent a reliable indicator of mutual relevance between URLs. This means that clusters created using coselections have greater semantic consistency that clusters created by plain clickthrough data. As a result, semantic similarity detected over coselection-based clusters will be significantly greater than that over clickthrough-based clusters, as only those URLs participating in the correct meaning of the query term will be used in the comparison, and no spurious semantic associations will be drawn between URLs from the 'wrong' sense of an ambiguous term.

Furthermore, the accuracy of the semantic similarity comparison using coselection-based clusters will be better than clickthrough-based clusters, as non-relevant URLs will not be present in clusters. This latter point may be important when endeavouring to measure the magnitude of semantic similarity, especially when seeking to determine what proportion of URLs from a given cluster are present in another cluster.

2.1.2 Cross-language 'synonym' discovery (translation candidates)

Cross-language information retrieval involves searching for documents in a target language(s) based on a query in a source language. Generally, translation is not performed on entire document collections but query translation is more common (Christof and Bonnie 2005 and Kishida 2005) and can be either a generalised dictionary-based query translation or machine translation approach. Braschler et al. (2000) found that machine translation techniques suffer from lack of context in short queries, have trouble dealing with the informal grammar a query typically contains, and are prone to meaning loss due to the selection of only a single query as an output. Alternatively, dictionary-based approaches, can be categorised as i) static bilingual dictionary methods, which produce a number of target translations for each term; ii) corpora-based, which use probabilities that words translate via analysis of parallel corpora, and; iii) internet-based, which involves mining Internet resources for translations.

Automatic translation has problems such as the out-of-vocabulary problem and ambiguity (Kishida 2005). One

approach, monolingual disambiguation followed by translation with sense-singular dictionaries has been proposed by Gracia et al. (2006), as well as query log analysis (Gao et al. 2007). In Gracia et al. (2006), the researchers first create a monolingual sense-singular dictionary from multiple ontologies, which is used to expand the query based on each semantic meaning and use frequency statistics from Google searches to disambiguate the monolingual query. The disambiguated monolingual query can then be used to translate the query using a sense-singular multilingual dictionary. However those multiple ontologies must be explicitly created and maintained, unlike the use of coselection data which requires no human-made reference materials.

The second approach is to use parallel corpora to find equivalent terms, creating a similarity thesaurus. Parallel corpora are collections of directly translated documents, such as translations of the Bible (Chew and Abdeladi 2007) and the Europarl collection. Bilingual countries such as Canada translate parliamentary proceedings and official records (Koehn 2005). While such methods have shown good results, they are limited by the availability of parallel corpora (Kishida 2005) both in terms of the language pairs, and in content domain. Extracting parallel corpora from the Internet has been proposed by Jian-Yun and Jian (2001) but uses static, hand selected 'anchor text' (e.g. "Chinese version") which must be developed for each language.

Comparable corpora do not require an exact translation, only approximate translations, e.g. news reports published in multiple languages (Tuomas et al. 2007). Two major approaches to mining comparable corpora have been proposed. The first looks for comparable sections of the documents (Munteanu and Marcu 2006) while the second uses statistical methods, with co-occurrence statistics being popular (e.g. Diab and Finch 2000 and Wai, Shing-Kit and Ruizhang 2007). Both start with corpora that are known to be on the same topic, and are limited to domains where approximate translations exist. Once again a major distinction between the approach in this paper and corpus-based approaches is that use of coselection data works without human-created external reference materials.

Internet-based approaches such as Gracia et al. (2006) make use of existing materials, such as leveraging ontologies to get a larger coverage of language. However, this is limited by the number of specifically-created instances. Other approaches avoid external reference material, e.g. exploiting 'courtesy translations' or manual translations of terms by the content developer which can be found using techniques such as searching for the term to be translated only in pages of the target language (Wen-Hsiang, Lee-Feng and His-Jian 2002). Statistical analysis can be performed in order to extract translations (Ying and Vines 2004). Such techniques are fairly error prone and a hybrid method using linguistic patterns alongside concurrence measures was proposed by Zhou et al. (2008).

2.2 Prior work on synonym discovery with coselections

In Ashman et al. (2011) we assessed the viability of discovering synonyms by comparing coselection-based clusters. A simple overlap-based method was used, so that if two queries had enough URLs in common in their clusters, they were deemed to be semantically similar.

This work used coselection data generated from the first two years of the Teesside data. The clustering method used was very basic, comprising a simple vertex and edge thresholding, and case sensitivity meant that some clusters were separated when they should not have been (at least semantically), for example in figure 1, *Castle Pernstejn* is distinct from *castle Pernstejn*. Also the method used for semantic similarity comparison was merely an overlap, not taking into account the magnitude of either cluster in any comparison. However in spite of these limitations, it was evident that the principle of synonym detection over coselection-based clusters warranted further investigation, for example *pernstejn* was clearly related to *castle Pernstejn*. In the prior work, we conjectured that with improved clustering methods and semantic similarity detection methods, it would be feasible to detect synonyms reliably.

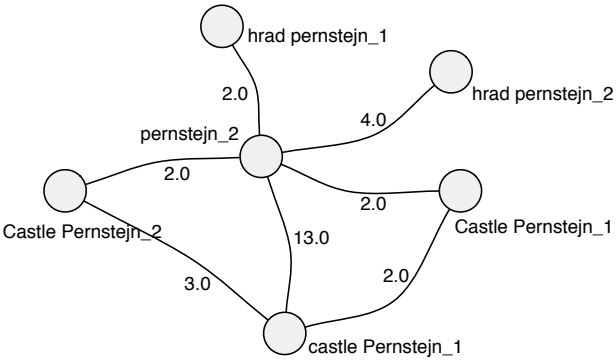

Figure 1: cluster overlap for variants on *Pernstejn* (Ashman et al. 2011)

3 Methodology

The method evaluated in this paper operates in three stages. After extracting a set of coselections from the raw log data, the clustering algorithm creates *term graphs*, namely aggregates of URLs with weighted edges between them which represent the number of coselections between any given pair of URLs. This is discussed in 3.1. Essentially this is the search term's cluster represented in graph format (see Ashman et al. 2011). Notably, the term graph may contain more than one cluster for a given search term, and where this occurs, it may be because the term is ambiguous, although it may also be because the data is sparse (see 3.3). However if the term is indeed ambiguous, there will be more than one cluster, and the semantic coherence within clusters will be reliable (see Ashman et al. 2011). If the distinct clusters are due to sparse data, we can 'correct' the clusters using this same semantic overlap method.

The final step of the method is the assessment of cluster overlap, described in 3.2. The data itself will affect the outputs, for example sparse data will give few

clusters. The input data is considered in 3.3. The evaluation method is discussed in 3.4.

3.1 Clustering

Density Based Spatial Clustering in Applications with Noise (DBSCAN) is a well-established algorithm for clustering spatial data. DBSCAN uses two parameters to cluster spatial data, *epsilon* and *minimum nodes*. It works by considering a single data point and testing whether there are any other data points within a Euclidian distance of epsilon, referred to as the Epsilon neighborhood. Points found are added to the cluster and their epsilon neighbourhood is assessed to see if any further points can be added. Once the cluster is complete it is tested against the minimum nodes parameter, if it fails then it is discarded as noise.

Whilst most of the concepts informing the construction of this algorithm are valid in this situation, some modifications need to be made since our dataset is graph-based and cannot be represented in Euclidean space.

In spatial data sets, the distance between an outlying point and the nearest point of a cluster can be used to identify the similarity between the outlier and the cluster. DBSCAN determines which points should be added to the cluster by comparing this distance to the epsilon parameter of the algorithm. In graph-based data sources an alternative metric must be used for epsilon evaluation as Euclidean distance is undefined in graph space. Using the DBSCAN algorithm over non-spatial data sets presents some additional issues because the lowest edge weight cannot adequately represent the relationship between the outlying point and a cluster. This is especially going to be the case when a URL has low individual edge weights to any other URL, but when there are very many other URLs - in such a situation, the URL would be left out of the cluster despite being linked to a large number of others.

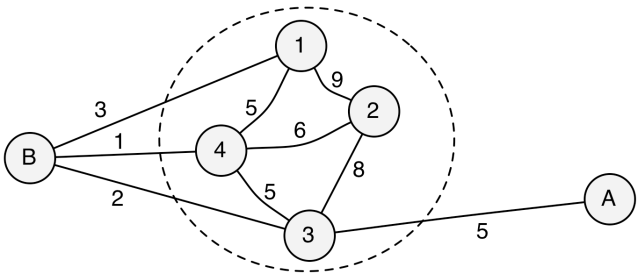

Figure 2: An example cluster with outlying vertices

We used the method of defining the distance between a vertex and a cluster according to the weight of the most significant edge connecting the vertex to the cluster. We then incorporated the vertex if the weight is equal or greater than the epsilon parameter. If fig. 2 is evaluated using an epsilon of 5 using this method, vertex A would be included in the cluster but B would be rejected. This means of evaluation rejects candidates that are widely interlinked to a cluster but lack a large single linking edge, and this is the subject of ongoing work.

3.2 Semantic similarity calculation of clusters

In Ashman et al. (2011) we tested the plausibility of the synonym detection idea by considering the absolute number of URLs occurring in both clusters. This however does not take into account the proportion of URLs that may make up this overlap. Hence we have developed a normalising algorithm for determining the strength of the relationship.

For each vertex in each term graph, the total weight of edges originating/terminating is counted. The mean interlinking strength is then calculated for that term graph, then each vertex is assigned a prominence value equal to its interlinking strength divided by the term graph's mean interlink strength. This normalises the prominence of vertices within that graph such that an average vertex has a prominence of 1.

When calculating the similarity between two clusters, the set of URL matches between the clusters is found. For each member of this overlap set, the prominence values associated with that URL in each term graph are averaged. The total cluster similarity is then found by summing the mean prominences for each member of the overlap set.

We determined an appropriate similarity threshold through experimentation combined with the human evaluation (see 3.4). Even using a relatively weak clustering algorithm still resulted in very useful data at a threshold of 1, with only 6 identified errors even though 90% of the total synonyms found were represented. We varied the similarity threshold for values 1, 2, 3 and 4 to determine which maximised the precision and recall (see 4.2.2).

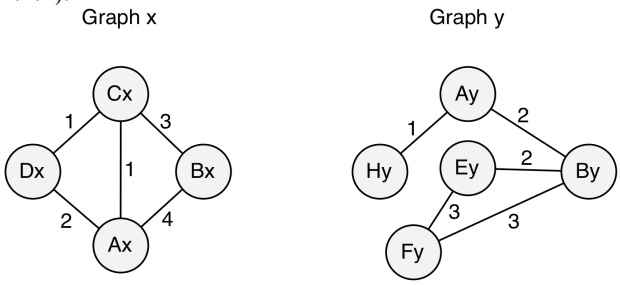

Figure 3: Graphs for worked example

For graph a in the example above, the total edge strengths of each vertex are calculated by adding the weight of each edge originating/terminating at each vertex. This gives us prominence values for each vertex as Ax:7, Bx:7, Cx:5, Dx:3. The normalised prominence of each vertex can be found by dividing by the average of these values (5.5). This gives us normalised values of Ax:1.3, Bx:1.3, Cx:0.91, Dx:0.55.

To find the cluster similarity between two clusters, x and y, the overlap set must first be determined. In this example we will say that there are two URLs shared between the clusters, with Ax and Bx related to Ay and By respectively. The strength of the link Ax – Ay is 0.93, found by the average of Ax's normalised prominence of 1.3 and Ay's normalised prominence of 0.55. By the same method the link Bx – By has a strength of 1.3. The total cluster similarity is calculated by summing the values of all links, so in this case it would be 2.3.

3.3 Data

The data used in this experiment is the same as used for the prior overlap experiment (see 2.2), namely the first two years of Teesside data. This data partly comprises text coselection data, i.e. coselections between two URLs as a result of a text search.

Other researchers found that traditional text search clickthrough is not reliable (see Ashman et al. (2011) for detailed related work on this). However, we found this not to be the case in an experiment assessing the relevance of search terms to clicked URLs for both text (Smith, Antunovic and Ashman 2009) and images (Ashman et al. 2009), and postulated that perhaps the ambiguity of queries coupled with the experimental design in other work may have contributed to the poor results (Ashman et al. 2011). Hence we believe that the use of text search clickthrough to be a valid experimental platform, and the results we report in section 4 support its use, suggesting that the clickthroughs and coselections from text searches are accurate enough, especially after clustering, to support semantic similarity comparisons.

We additionally look at image coselection data. In prior work we found that image clickthrough is a reliable indicator of mutual relevance between search term and image URL (Smith, Brien and Ashman (to appear) and Ashman et al. 2009) and also between coselected images (Ashman et al. 2011). We aim to compare synonym discovery between image coselection data and text coselection data. While there are not large quantities of image coselection data, the results are indicative, if not conclusive.

Finally, it is important to note that while coselections are a reliable indicator of mutual relevance between two URLs, a lack of coselections does not indicate the opposite, i.e. that they are not relevant. This has implications for clustering, as there may be URLs that are semantically similar but which are never coselected, either because they do not both appear in the first few pages, or because they appear in the top pages at different times. Given that users tend to select primarily from the top ten results of any search, and that the top ten is, at least for some search engine interfaces, quite volatile (Truran, Schmakeit and Ashman 2011), it is likely that many pairs of URLs will never be coselected, irrespective of their semantic similarity. That is, the recall of any method using coselection-based clustering is going to be very difficult to measure, and will itself be the result of the search engine's own recall performance, as well as of its ranking algorithm.

While this does not prevent the discovery of synonyms using this method, it may not be meaningful to measure recall for this experiment, and it becomes confusing to speak of precision in this context while not also using recall. For this reason, we instead use the notation of false positive when speaking of the proportion of retrieved documents which are relevant.

3.4 Evaluation method

The accuracy of the synonym discovery method is assessed using groundtruthing by human evaluators. Each pair of query terms claimed to be synonymous by the method was assessed for whether they were genuinely semantically similar. Any pair of query terms not semantically similar classified as a false positive.

A match was considered acceptable if the terms had some significant relevance to each other. For each analysis method and data input, the number of associations found by the semantic similarity calculation is recorded. The result set was then assessed against various threshold values to determine the number of positive and false positive results. As the threshold is lowered, more results are found but the error rate increases. The aim was to find a threshold that provides an acceptable quantity of results whilst minimising the false positive rate.

4 Results and Discussion

The following tables summarise the results, varying the similarity threshold T. In each table we compare the quality and quantity of synonyms detected. The quantity is self-evident, with more synonyms being better. The quality is measured by the proportion of false positives.

Table 1 overview the results. For a threshold of 2 or more, there are no false positives at all, for any variation. Since we are aiming to maximise the number of associations found while minimising the false positives, T=2 is the highest needed. So for the subsequent sections, we consider only T being 0, 1 and 2.

The Result Set column refers to which of the result sets is under question, as we performed the clustering and semantic similarity comparison for both image coselection and text search coselection data, and used a stronger and weaker pair of parameters for DBSCAN. The subsequent columns represent the false positive rate for thresholds T of varying levels, 0, 1, 2, 3 and 4, with 4 being the strictest, expressed as the absolute number of false positives divided by the total number of associations for that specific threshold. T=0 is effectively the set of all associations as found by the algorithm with no filtering.

Result Set	T=0	T=1	T=2	T=3	T=4
Images(3,2)	16/44	6/28	0/6	0/6	0/6
Images(4,3)	0/14	0/10	0/6	0/6	0/6
Text(3,2)	0/296	0/274	0/166	0/166	0/152
Text(4,3)	0/20	0/16	0/8	0/8	0/8

Table 1: DBSCAN with epsilon evaluated using most significant edge

Note that in this section we discuss 'associations' rather than synonyms, so as not to imply that the associations found are necessarily synonyms.

4.1 Varying DBSCAN parameters

First we look at the variation in DBSCAN parameters. Recall that the parameters are the epsilon value and the minimum nodes for a cluster to exist. We selected two sets of parameters, (3,2) and (4,3), expecting (3,2) to be noisier than (4,3) due to the smaller number of nodes for a cluster to exist and the lower epsilon threshold.

The weaker parameters have two effects, they find more associations but have a higher false positive rate. For images, there are at least twice as many associations found with (3,2) for T=0 or 1, than with (4,3), but with higher false positives, going from 0% to 20%.

For text coselections, there is a much greater discrepancy between (3,2) and (4,3) than with images. For every threshold there are roughly 10 times as many associations found by Text(3,2) than by Text(4,3), however the false positive rate is very low, remaining below 3% of the total associations found for all thresholds. There are no false positives for Text(4,3).

	T=0	T=1	T=2
all (3,2) correct	0.95	0.98	1.00
all (3,2) false	0.05	0.02	0.00
all (4,3) correct	0.93	1.00	1.00
all (4,3) false	0.07	0.00	0.00

Table 2: comparison of (3,2) outputs versus (4,3) outputs

What this implies is that the stronger clustering parameters do manifest much more reliable accuracy, with no false positive observed either for Text(4,3) or Image(4,3). However there are clearly a large number of genuine associations found by Text(3,2) and Image(3,2) that were validated by the human evaluators but not discovered by the stronger clustering parameters, thus the recall is impaired. With the proportionally low error rate of (3,2), it seems that the stronger clustering parameters achieve a small improvement for T=1 specifically while losing many genuine associations.

4.2 Varying the threshold

Next, we look at the effect of varying the threshold at which associations are rated as being genuine synonyms. The total associations value T is the number of associations generated altogether by the algorithm, while the remaining columns filter out associations whose cluster similarity falls below the threshold, for threshold values of T=0, 1 and 2 respectively.

As noted above, for all results sets tested is that there is no difference in the false positive rate once T=2 or more. That is, a cluster similarity value of 2 appears to be high enough to filter out all false positives. In fact for DBSCAN parameters (4,3), there are no false positives even for T=0, while only images(3,2) shows a significant proportion of false positives at T=1, with all other result sets showing false positives of 0 at T=1, except text(3,2) with a false positive rate of under 1%.

There is however a difference in the number of associations remaining after filtering with the threshold. There is little or no difference between T=2 and T=3, and T=3 to T=4 except in Text(3,2) which loses around 8% of its associations (all genuine).

The differences are more interesting between T=1 and T=2. In half of the result sets, notably the (4,3) sets, there were no false positives for T=1, so the higher threshold represents an unmitigated penalty. In one of the (3,2) result sets, there were false positives for T=1, but only in images(3,2) was the proportion problematic, at close to 20% even at T=1. We discuss the images versus text contrast in the next section.

We also consider what happens if no threshold is applied, namely whether all associations found by the algorithm are valid. Interestingly both of the Text(4,3) results sets showed no false positives at all while Text(3,2) showed only around 1.3% false positives. Images however showed a more troublesome false positive rate, as discussed in the next section.

	T=0	T=1	T=2
all correct	0.95	0.98	1.00
all false	0.05	0.02	0.00

Table 3: comparison of output accuracy for threshold values T=0, 1 and 2

4.3 Image coselections versus text coselections

Finally we compare the result sets generated by image coselections versus text coselections.

One clear difference is that text coselections generate many more associations. For the weaker clustering parameters (3,2), text coselection clustering generated more than 6.4 times as many potential associations than image coselections. However since the raw source data shows only about 5% of all searches being image searches, this discrepancy is perhaps not surprising.

	T=0	T=1	T=2
text correct	644	586	362
images correct	92	70	30
text false	4	2	0
images false	38	12	0

Table 4: total associations comparison of text coselections input versus image coselections inputs

What is more interesting however is that the false positive rates for text coselection result sets are the same or better than for image coselection result sets, for both clustering parameter pairs. In fact, for T=1, the false positive rate for Text(3,2) is under 1% while the false positive rate for Images(3,2) is around 20%. For T=0, the difference is even more marked, with false positive rates of around 1% for text versus around 29% for images respectively.

	T=0	T=1	T=2
text correct	0.99	1.00	1.00
images correct	0.71	0.85	1.00
text false	0.01	0.00	0.00
images false	0.29	0.15	0.00

Table 5: accuracy comparison of text coselections input versus image coselections inputs

This is a very interesting outcome, as text clickthrough has been deemed unreliable in other research while image clickthrough appears to be more reliable (Ashman et al. 2011). While coselection data is not quite the same as clickthrough data, as it indicates a relationship between two URLs rather than between a URL and a search term, it is implicit that the judgements made by the searchers in creating those coselections are not as error-prone as has been claimed in the past.

5 In Conclusion

This experiment has demonstrated that it is indeed possible to use cluster overlap to reliably identify synonyms from both text and image search, based on searchers' interactions with search engines. A key novel

feature of this work is that the synonyms identified are not compromised by the ambiguity of search terms because the underlying clusters are themselves not ambiguous. This is achieved through a combination of the natural discrimination exercised by searchers and noise-reducing clustering algorithms.

So far we have found good results even with fairly weak parameters. The level of false positives has in most cases been low, especially for traditional text searches. In fact, the level of reliability in the text coselections has been higher than published literature would imply. We plan a further experiment that will assess the reliability of searchers' interactions, testing for cluster coherence by selecting pairs of URLs from clusters that have been associated using the method above, i.e. are synonymous. One URL will be randomly selected from each cluster of any two that are synonymous and human evaluators will be asked to determine whether the two URLs are mutually relevant. This is similar to the mutual relevance ranking experiment performed by Ashman et al. (2011) and will also test for inter-ranker consistency to reduce noise from user error.

Finally, while the numbers of synonyms found has been modest, this is a direct outcome of the raw data input into the process - there needs to be enough coselections to generate reasonable clusters over an adequate number of URLs. Also term coverage is dictated by searchers - only those terms for which users submit searches and make at least two selections are going to become part of the process. However the process is shown to be sound, and with enough data and cooperation among the community, a much wider coverage of terms, not just those appearing in formal lexicons, will be semantically linkable.

6 References

Ashman, H., Antunovic, M., Chaprasit, S., Smith, G. and Truran, M. (2011): Implicit association via crowd-sourced coselection. *Proc. Hypertext 2011*, New York, USA, 7-16, ACM.

Ashman, H, Antunovic, M., Donner, C., Frith, R., Rebelos, E., Schmakeit, J.-F., Smith, G. and Truran, M. (2009): Are clickthroughs useful for image labeling? *Proceedings of IEEE/WIC/ACM Web Intelligence 2009 (WI09).*

Ashman, H., Zhou, D., Goulding, J., Brailsford, T. and Truran, M. (2007): The Global Perpetual Dictionary of Everything. *Proc. Ausweb 2007,* <http://ausweb.scu.edu.au/aw07/papers/refereed/ashman/paper.html>.

Beeferman, D. and Berger, A (2000): Agglomerative clustering of a search engine query log. *Proc.SIGKDD,* 407-416.

Braschler, M., Krause, J., Peters, C. and Schauble, P. (2000): Cross-Language Information Retreval (CLIR) Track Overview. *Proceedings of TREC8,* 26-34.

Chew, P. and Abdeladi, A. (2007): Benefits of the 'Massively Parallel Rosetta Stone': Cross-Language Information Retrieval with over 30 Languages. *Annual meeting Assoc. for Computational Linguistics,* **45**:872.

Christof, M. and Bonnie, J.D. (2005): Iterative translation disambiguation for cross-language information retrieval. *Proceedings of SIGIR 05,* ACM.

Cui, H., Wen, J.-R., Nie, J.W. and Ma, W.Y. (2003): Query expansion by mining user logs. *Transactions on Knowledge and Data Engineering,* **15**:829-839, IEEE.

Diab, M. and Finch, S. (2000): A statistical word-level translation model for comparable corpora. *Conference on Content-based multimedia information access (RIAO).*

Gao, W., Niu, C., Nie, J.-Y., Zhou, M., Hu, J., Wong, K.F. and Hon, H.W. (2007): Cross-lingual query suggestion using query logs of different languages. *Proc SIGIR 2007,* Amsterdam, The Netherlands, ACM.

Gracia, J., Trillo, R., Espinoza, M. and Mena, E. (2006): Querying the web: a mulitontology disambiguation method. *Proc. 6th international conference on Web Engineering,* Palo Alto, California, USA, ACM.

Jian-Yun, N. and Jian, C. (2001): Filtering noisy parallel corpora of web pages. *Intl Conf on Sys, Man and Cyb,* IEEE.

Kishida, K. (2005): Technical issues of cross-language information retrieval: a review. *IP&M,* **41**:433-455.

Koehn, P. (2005): Europarl: A parallel corpus for statistical machine translation. *MT Summit X,* Thailand.

Munteanu, D.S. and Marcu, (2006) Extracting parallel subsentential fragments from non-parallel corpora. *Proc. of the 21st Intl Conf. on Computational Linguistics,* Assoc. for Computational Linguistics

Smith, G., Antunovic, M. and Ashman, H. (2009): Classifying Images with Image and Text Search Clickthrough Data. *Proc. Int. Conf. on Active Media Technology.*

Smith, G., Brailsford, T., Donner, C., Hooijmaijers, D., Truran, M., Goulding, J. and Ashman, H. (2009): Generating unambiguous URL clusters from web search. *Proc. Ws on Web Search Click Data,* 28-34, ACM.

Smith, G., Brien, C. and Ashman, H. (to appear): Evaluating implicit judgments from image search clickthrough data. *Journal of the American Society for Information Science and Technology*.

Truran, M., Goulding, J. and Ashman, H. (2005): Co-active Intelligence for Information Retrieval. *Proceedings of ACM Multimedia '05,* 547-550, ACM.

Truran, M., Schmakeit, J.-F. and Ashman, H. (2011): The Effect of User Intent on the Stability of Search Engine Results. *Journal of the American Society for Information Science and Technology,* **62**(7).

Tuomas, T., Jorma, L., Kalervo, J., Rvelin, M., Martti, J. and Heikki, K. (2007): Creating and exploiting a comparable corpus in cross-language information retrieval. *ACM Trans. on Information Systems,* **25**(4).

Wai, L., Shing-Kit, C. and Ruishang. H. (2007): Named entity translation matching and learning: with Application for mining unseen translations. *ACM Trans. on Information Systems,* **25**(2).

Wen, J.-R., Nie, J. and Zhang, H.J. (2002): Query clustering using user logs. *ACM TOIS,* **20**:59-81.

Wen-Hsiang, L., Lee-Feng, C. and Hsi-Jian, L. (2002): Translation of web queries using anchor text mining. *Transactions on Asian Language Processing,* **1**(2):159-172.

Xue, G.-R., Zeng, H.-J., Chen, Z., Yu, Y., Ma, W.-Y., Xi, W. and Fan, W. (2004): Optimizing web search using web clickthrough data. *Proc CIKM 04,* ACM.

Ying, Z. and Vines, P. (2004): Using the web for automated translation extraction in cross-language information retrieval. *Proc ACM SIGIR 04.* Sheffield, United Kingdom, ACM.

Zhou, D., Truran, M., Brailsford, T. and Ashman, H. (2008): A hybrid technique for English-Chinese Cross Language Information Retrieval, *Transactions of Asian Language Processing,* **7**(2), ACM.

Zhou, D., Truran, M., Brailsford, T., Wade, V. and Ashman, H. (to appear): Translation Techniques in Cross-Language Information Retrieval. *ACM Computing Surveys,* ACM.