Query Directed Web Page Clustering

Daniel Crabtree, Peter Andreae, Xiaoying Gao
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
New Zealand
daniel@danielcrabtree.com, pondy@mcs.vuw.ac.nz, xgao@mcs.vuw.ac.nz

Abstract

Web page clustering methods categorize and organize search results into semantically meaningful clusters that assist users with search refinement; but finding clusters that are semantically meaningful to users is difficult. In this paper, we describe a new web page clustering algorithm, QDC, which uses the user's query as part of a reliable measure of cluster quality. The new algorithm has five key innovations: a new query directed cluster quality guide that uses the relationship between clusters and the query, an improved cluster merging method that generates semantically coherent clusters by using cluster description similarity in additional to cluster overlap, a new cluster splitting method that fixes the cluster chaining or cluster drifting problem, an improved heuristic for cluster selection that uses the query directed cluster quality guide, and a new method of improving clusters by ranking the pages by relevance to the cluster. We evaluate QDC by comparing its clustering performance against that of four other algorithms on eight data sets (four use full text data and four use snippet data) by using eleven different external evaluation measurements. We also evaluate QDC by informally analysing its real world usability and performance through comparison with six other algorithms on four data sets. QDC provides a substantial performance improvement over other web page clustering algorithms.

1 Introduction

Web search is difficult because it is hard for users to construct queries that are both sufficiently descriptive and sufficiently discriminating to find just the web pages that are relevant to the user's search goal. Queries are often ambiguous: words and phrases are frequently polysemantic and user search goals are often narrower in scope than the queries used to express them. This ambiguity leads to search result sets containing distinct page groups that meet different user search goals. Often users must refine their search by modifying the query to filter out the irrelevant results. Users must understand the result set to refine queries effectively; but this is time consuming, if the result set is unorganised.

Web page clustering is one approach for assisting users to both comprehend the result set and to refine the query. Web page clustering algorithms identify semantically meaningful groups of web pages and present these to the user as clusters. The clusters provide an overview of the contents of the result set and when a cluster is selected the result set is refined to just the relevant pages in that cluster.

Clustering performance is very important for usability. If cluster quality is poor, the clusters will be semantically meaningless or will contain many irrelevant pages. If cluster coverage is poor, then clusters representing useful groups of pages will be missing or the clusters will be missing many relevant pages. Therefore, improving the performance of web page clustering algorithms is both worthwhile and very important.

This paper presents QDC, a query directed web page clustering algorithm that gives better clustering performance than other clustering algorithms. QDC has five key innovations: a new query directed cluster quality guide that uses the relationship between clusters and the query, an improved cluster merging method that generates semantically coherent clusters by using cluster description similarity in additional to cluster overlap, a new cluster splitting method that fixes the cluster chaining (drifting) problem, an improved heuristic for cluster selection that uses the query directed cluster quality guide, and a new method of improving clusters by ranking the pages by relevance to the cluster.

The next section of the paper sets QDC in context of the other research in the field by describing related work. The following sections describe the algorithm and evaluate QDC by comparing its performance against other clustering algorithms.

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00

2 Related Work

Most clustering algorithms for web pages start by preprocessing the pages in a standard way. Various page elements and words are removed from the pages: HTML tags, punctuation and other similar non-informative text, a set of stop words containing very common and uninformative words such as "the", "it", and "on". Light stemming, using the Porter stemming algorithm [12], is often applied to reduce terms to their root form, for example, "dogs" becomes "dog". This leaves each page represented by a sequence of words.

There are several models used to represent the preprocessed pages, the most common models are the bag of terms and set of terms. Terms can be either words or phrases, although often just words are used. There are also graph based models that preserve the ordering of document terms [13]; one that is quite efficient is the suffix tree model [19].

Researchers have applied all the standard data clustering methods [2, 8, 15] to web page clustering: hierarchical (agglomerative and divisive), partitioning (probabilistic, k-medoids, k-means), grid-based, density-based, fuzzy c-means, Bayesian, Kohonen self-organising maps, and many more. Many algorithms build on the standard methods by using web or document specific characteristics to assist clustering: Suffix Tree Clustering (STC) [18] and Lingo [10, 11] use phrases and some algorithms [17, 9] consider the hyperlinked nature of web pages.

Current web page clustering algorithms produce clusterings of low quality: many clusters are semantically meaningless and the meaningful clusters are often small, missing many relevant pages, and contain irrelevant pages. The problem is that from a textual perspective the algorithms only use properties and statistics of pages from within the result set. Many algorithms such as hierarchical and partitioning algorithms [15] use data similarity measures [2] to construct clusters; when applied directly to page data, the similarity based methods are not effective at producing semantically meaningful clusters.

One way of improving web page clustering algorithms is to make better use of the textual properties of web pages. The semantic relationships between words is very useful information; for example, synonyms, hyponyms, meronyms, etc. [4]. WordNet [4] is a lexical reference system and is one source of this information. However, the data in these systems is incomplete, particularly for commercial, technical, and popular culture word usage.

An alternate source, although less accurate and less informative, is to use global document analysis and term co-occurrence statistics to identify whether terms are related or unrelated. The number of pages in multi-term search result sets can approximate term co-occurrence statistics. Google distance [4] and the Rough Set based Graded Thesaurus [5] are two techniques that use these statistics to determine term similarity and both have been shown to be effective on various tasks, such as hierarchical word clustering [4] and web query expansion [5].

QDC uses term relationships to provide a dramatic improvement in clustering performance. Specifically, QDC uses normalized Google distance (NGD) [4]:

$$NGD(i,j) = \frac{\max(\ln(f(i)), \ln(f(j))) - \ln(f(i \wedge j))}{\ln(M) - \max(\ln(f(i)), \ln(f(j)))}$$

where i and j are terms, $f(t)$ is the approximate web frequency of some term or terms, and M is the approximate total number of pages.

Some algorithms represent pages using more advanced models than a bag of words, but their performance is still inadequate. One of the best web page clustering algorithms is STC, which uses the suffix tree model to identify base clusters consisting of all pages containing one phrase. While there are some high quality base clusters, many are too broad and are ambiguous even within the context of the result set and introduce many irrelevant pages into the final clusters and degrade clustering performance. Advanced models alone are inadequate for producing good clustering performance. QDC uses a simple set of words model and removes low quality base clusters using the relationship between cluster descriptions and the user's query.

Base clusters often have poor coverage as they miss many relevant pages. STC addresses this by merging clusters using a single-link clustering algorithm [8] with cluster overlap as the similarity measure. But cluster overlap may merge semantically unrelated clusters, which lowers cluster quality, unless the overlap threshold is set very high. However, this leaves many related clusters separate, which limits cluster coverage. QDC uses cluster description similarity in addition to cluster overlap to provide a more effective similarity measure for merging clusters.

Some clustering algorithms, including single-link clustering and STC, are susceptible to cluster chaining (drifting) [19]. In a sequence of clusters, each cluster may be similar to its immediate neighbours, but completely dissimilar from clusters further away in the sequence. Clusters obtained by merging such sequences are often of low quality and are not semantically meaningful. The improved similarity measure used for merging in QDC limits this significantly, but does not stop it entirely. QDC solves the cluster chaining (drifting) problem by making a second pass over merged clusters and splitting those that have been joined inappropriately.

Algorithms that construct many clusters, like STC, must select a subset (no more than about ten) to show the user, as the user cannot comprehend too much at one time. Extended Suffix Tree Clustering (ESTC) [6], an extension of STC, uses a cluster selection method that considers page

coverage and cluster overlap in a heuristic hill-climbing search with look-ahead and enhanced branch and bound pruning. QDC improves the heuristic by additionally considering cluster quality and the number of selected clusters.

Providing the most relevant pages earlier in the results can reduce the time users spend searching [1]. Most clustering algorithms order the pages in the clusters by their position in the search results [3]. Such an ordering fails to use the additional information about the user's search goal, provided by the user selecting the cluster, so the most relevant pages may not be shown first. QDC orders the pages within each cluster according to their relevance to the cluster.

3 Algorithm - QDC

This section describes QDC, a query directed web page clustering algorithm with five stages, which roughly match our five key innovations.

3.1 Base Cluster Identification

A base cluster is described by a single word and consists of all the pages containing that word. Equivalently, base clusters are single word search refinements based on the current search results. After standard page pre-processing, QDC constructs a collection of base clusters, one for every word that is in at least 4% of the pages. Using a lower threshold will increase clustering performance at the cost of algorithm speed.

Many base clusters are useless and only serve to contaminate the final clusters. Removing these useless clusters would improve the clustering, but selecting the right clusters to prune requires some guide to cluster quality. The user's query is the best, and often the only, specification of the information desired by the user. QDC uses the relationship between query terms and cluster descriptions as one part of a cluster quality guide. QDC computes the query distance of each base cluster — the distance from the query, using NGD as defined in section 2. The query distance from a base cluster to the query is the minimum of the NGD between the word specifying the base cluster and any query term.

Terms with a low query distance tend to be very specific and are often unambiguous in the context of the query, while terms with a high query distance tend to be quite broad and are often ambiguous, even in the context of the query. Ambiguous clusters are often of poor quality as they combine multiple distinct ideas of which only one is normally of interest to a given user. QDC removes these low quality clusters by removing clusters whose query distance is too large. Our experiments use cutoffs of 1.5 when using full text data and 2.5 when using snippet data. This step removes most low quality clusters, but if the cutoff is too low, high quality clusters may be removed as well; using a higher cutoff removes fewer clusters.

After pruning using query distance, there are still many low quality clusters. The relationship between the pages and the clusters can be used to further prune the collection of base clusters. The distribution of web pages tends to follow the frequency of user interest in the page topics. Therefore, larger clusters have a greater probability of being useful refinements and cluster size is an indication of cluster quality. QDC removes the worst clusters according to a measure proportional to cluster size and inversely proportional to query distance. The number of clusters kept is proportional to the total number of pages being clustered. Keeping a lower number of clusters will increase algorithm speed but lowers clustering performance, but if too many clusters are kept, low quality clusters are not pruned and may contaminate the merging process.

Removing this many clusters would normally have a negative effect on clustering performance, but because the query directed heuristics give a reliable guide to cluster quality, the low quality clusters that would later contaminate the merging process are removed, and the performance actually improves.

3.2 Cluster Merging

QDC constructs larger clusters by merging clusters together. Each cluster (c) is constructed from a set of base clusters ($base(c)$), and a cluster is described by the word that describes the cluster's largest base cluster. However, the set of pages in a cluster is not necessarily all the pages in its base clusters. A page is only included in the cluster if it is present in enough of the base clusters in the cluster. This threshold should increase with the number of base clusters in the cluster, but should not increase steeply. QDC uses a log function. A cluster is a set that contains the pages that are in at least $\log_2(|base(c)| + 1)$ of the cluster's base clusters.

Initially there is a singleton cluster for each base cluster. QDC merges clusters using single-link clustering over a relatedness graph. Single-link clustering merges together all clusters that are part of the same connected component on the graph. The relatedness graph has the clusters as vertices and has an edge between any two clusters that are sufficiently similar.

Previous methods use cluster content similarity and often merge unrelated clusters. Merging unrelated clusters decreases cluster quality by introducing irrelevant pages. The problem is exacerbated by cluster chaining (drifting): clusters that are closely related to one of the unrelated clusters but not the others are often merged in too, bringing further irrelevant pages with them.

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE

QDC defines two clusters to be sufficiently similar only if both the cluster contents and cluster descriptions are sufficiently similar. Requiring the cluster descriptions to match in addition to the contents dramatically reduces the merging of semantically unrelated clusters and increases cluster quality. Additionally, the cluster contents similarity threshold can be significantly reduced, which allows more semantically related clusters to merge (increasing cluster coverage).

The cluster contents are sufficiently similar if enough of the pages in one cluster are also in the other cluster (*i.e.*, if there is enough overlap between the clusters):

$$\frac{|c_1 \wedge c_2|}{\min(|c_1|, |c_2|)} > 0.6$$

The cluster descriptions are sufficiently similar if the pair of cluster descriptions occur together on the web significantly more frequently than would be expected if the pair were unrelated (*i.e.*, if their appearances were independent):

$$\frac{Mf(d_1 \wedge d_2)}{f(d_1)f(d_2)} > 4$$

Where d_1 and d_2 are the cluster descriptions, and $f(t)$ and M are as per NGD in section 2.

Decreasing either the cluster content or the cluster description similarity threshold will increase cluster coverage at the cost of greater cluster overlap.

3.3 Cluster Splitting

Each cluster now contains at least all the base clusters that relate to one idea; this is assured as single-link clustering merges all related clusters. But single-link clustering, even with our improved similarity function, can produce clusters containing multiple ideas and irrelevant base clusters due to cluster chaining (drifting). Such clusters need to be split. Interestingly, it is easier to split such a compound cluster than to prevent its formation in the first place; because the splitting can take into account the final cluster, whereas the merging process cannot.

QDC uses a hierarchical agglomerative clustering algorithm to identify the sub-cluster structure within each cluster. The algorithm uses a distance measure to build a dendrogram for each cluster starting from the base clusters in the cluster. Each cluster is split by cutting its dendrogram at an appropriate point — when the distance between the closest pair of sub-clusters falls below a threshold (our experiments use -2). This threshold means that any groups of base clusters that are not tightly interconnected with each other will be split. Using a higher threshold will lower the split point and increase the splitting frequency.

QDC uses a distance measure with three components: the number of paths between the two sub-clusters on the relatedness graph of length one (onelinks), or of length two (twolinks), and the average distance from base clusters in one sub-cluster to base clusters in the other sub-cluster.

$$dist(c_1, c_2) = onelinks + 0.5\, twolinks - avgdist(c_1, c_2)$$

$$avgdist(c_1, c_2) = \frac{\sum_{b_1 \in base(c_1)} \sum_{b_2 \in base(c_2)} len(b_1, b_2)}{|base(c_1)||base(c_2)|}$$

Where $len(b_1, b_2)$ is the path length between two base clusters in the relatedness graph.

3.4 Cluster Selection

At this stage, QDC has a small set of coherent clusters. However, there will still be more clusters than can be presented to the user. QDC needs to select the best subset of the clusters to present to the user. Ideally, these clusters should be high quality clusters that cover all the pages in the original set with minimal overlap.

QDC uses the ESTC cluster selection algorithm [6] with an improved heuristic, $H(C)$, to select a set of clusters to show the user. The ESTC cluster selection algorithm uses the heuristic with a 3-step look-ahead hill-climbing search to select a set of clusters to present to the user. To evaluate a candidate set of clusters, C, the new heuristic considers the number of pages covered by the clusters (C_P), the number of distinct pages covered by the clusters (C_D), the number of pages not covered by any of the clusters (C_O), and the quality of each cluster ($q(c)$).

$$H(C) = \left(\sum_{c \in C} q(c)\right) - \alpha C_O - \beta(C_P - C_D)$$

The new heuristic has two parameters that enable control of characteristics of the clustering: α (our experiments use 0.2) and β (our experiments use 0.3). α controls coverage and increasing α will generate clusterings with greater coverage at the cost of cluster quality. β controls overlap and increasing β will lead to clusterings with fewer pages in multiple clusters at the cost of page coverage.

The quality of a cluster ($q(c)$) controls the number of clusters selected and places a bias towards high quality clusters. Because of the logarithm, high quality clusters have above average quality and therefore positive quality values, whereas low quality clusters have below average quality and therefore negative quality values.

$$q(c) = \log_2\left(\frac{quality(c)}{\text{average cluster quality}}\right)$$

The quality measure for a cluster is extended from the base cluster quality measure to take the number of base clusters into account as well as the cluster size and query distance. The more base clusters that form a cluster, the

greater the evidence that the cluster represents a semantically meaningful group of pages. But the increase in evidence with each additional base cluster decreases. So we need a function with a monotonically decreasing 1st derivative; QDC uses the logarithm. The query distance of a cluster to the query, QD(c), is the average query distance of its base clusters.

$$quality(c) = \log_2(|base(c)| + 1)\frac{|c|}{QD(c)}$$

3.5 Cluster Cleaning

Base clusters are sometimes formed from polysemous words and therefore clusters can contain pages that cover different topics. Since the clusters should relate to only one topic, pages from other topics are irrelevant. QDC computes the relevance of each page in each cluster and removes irrelevant pages.

The relevance of a page to a cluster is based on the number and size of the cluster's base clusters of which it is a member. Page relevance varies between 0 and 1, with 0 being a page that is completely irrelevant to the cluster. Page relevance is computed as the sum of the sizes of the cluster's base clusters of which it is a member, divided by the sum of the sizes of all of the cluster's base clusters.

$$relevance(p, c) = \frac{\sum_{\{b|b\in base(c)\wedge p\in b\}} |b|}{\sum_{b\in base(c)} |b|}$$

QDC proceeds to remove irrelevant pages from clusters where two requirements are met: the page has relevance below a threshold (our experiments use 0.1) and the page has higher relevance in another cluster. A higher threshold will remove additional irrelevant pages but will also remove relevant pages, but the threshold is not very sensitive as the second requirement limits the pages that can be removed.

Page relevance also provides a ranking on the pages with respect to a cluster. QDC sorts and displays the pages in each cluster according to relevance. This improves cluster quality from the user's perspective as any remaining irrelevant pages are frequently near the bottom of clusters and so users rarely see them.

4 Evaluation

4.1 Algorithm Speed

QDC is on the order of ten times faster than STC and on the order of one hundred times faster than K-means. QDC achieves a significant increase in algorithm speed by pruning many base clusters during base cluster construction using the new query directed cluster quality measure.

4.2 Algorithm Performance

We used 11 measurements to compare the clustering performance of QDC against four other web page clustering algorithms (STC, ESTC, K-means, and Random Clustering) on eight data sets: search results of four different queries ("salsa", "jaguar", "gp", and "victoria university") using both full-page and snippet data. The queries are of varying clustering difficulty. The simplest is "salsa", which has two distinct clusters (both large) and few outliers. "jaguar" is more challenging with five distinct clusters (one large, three medium, and one small) and some outliers. "gp" is harder with five distinct clusters (two large, and three small) and many outliers. "victoria university" is the hardest with five very similar clusters (two large, one medium, and two small) and few outliers.

We compared the algorithms under an external evaluation methodology using a gold standard method [16, 7]. The method uses a rich ideal clustering structure and QC4 measurements (quality and coverage) [7], as this is well suited for web page clustering evaluation. The QC4 measurements provide a fair measure of clustering performance as they do not have any bias towards particular clustering characteristics. In particular the clustering granularity may be coarse or fine; the clusters may be disjoint or the clusters may overlap, so that the same page may appear in several clusters; the clustering may be "flat" so that all clusters are at the same level, or the clustering may be hierarchical so that lower-level clusters are sub-clusters of higher level clusters. Additionally, the QC4 measurements do not have the problems that other measurements have with extreme or boundary case clusterings, such as the extreme case of having all pages in one large cluster. On actual clusterings the QC4 measurements are also more expressive, as they give random clusterings much lower performance: the information capacity of the measurements is larger as the range of informative values is larger. The unreliability of precision and recall can be seen in the experiments where the recall measure gives a higher ranking (almost 60%) to the random clustering than to one of the clearly better algorithms.

In addition to QC4 measurements, we present the standard precision, recall, entropy, and mutual information measurements [16, 7] to provide further evidence for the results. All measurements come in both average and cluster-size weighted varieties (except mutual information for which averaging is not applicable), providing 11 measurements in total. Average measurements treat all clusters as equally important, while weighted measurements treat larger clusters as more important. Note that there is a trade-off between different measurement pairs: quality *vs* coverage, precision *vs* recall, and entropy *vs* recall. Mutual information provides a single measure that combines both aspects. For all measurements, higher is better, except entropy, for which

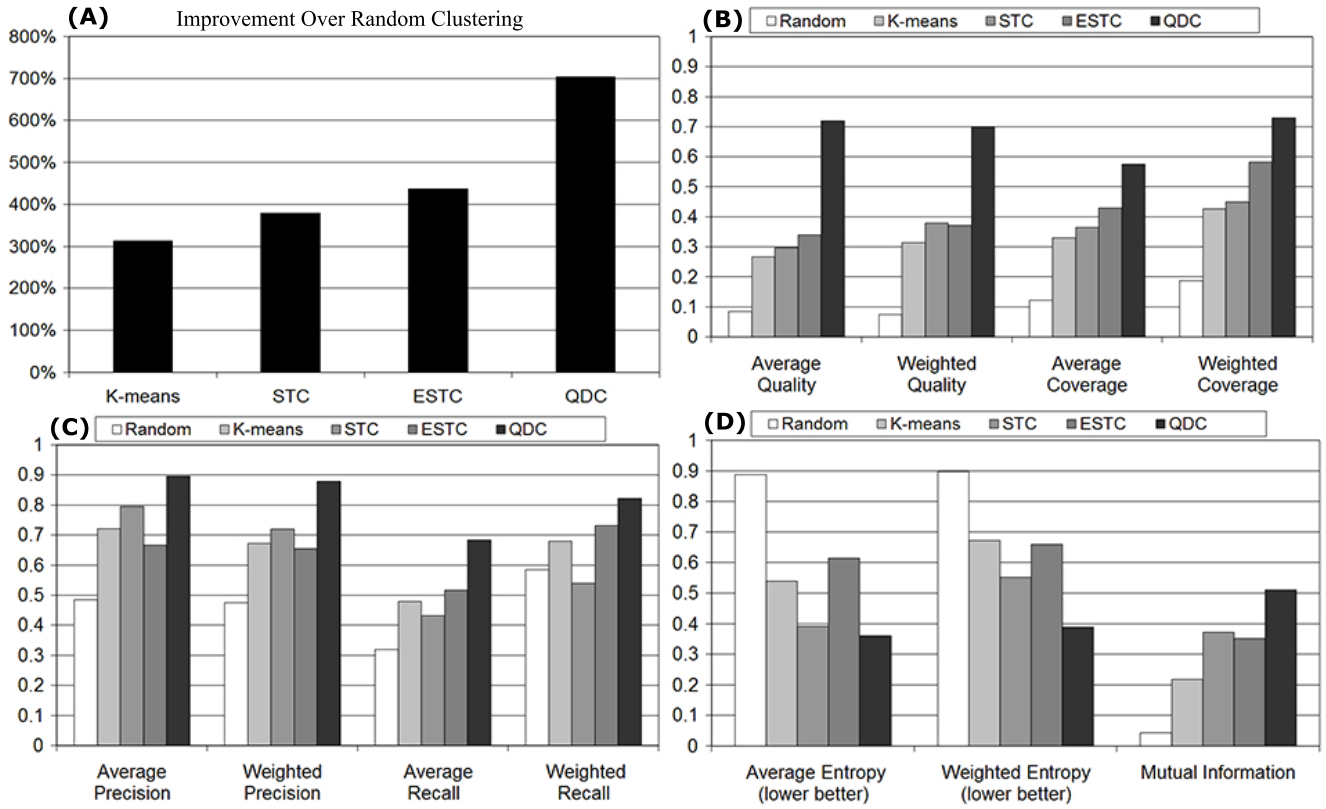

Figure 1. Full Text Results Averaged Over All Data Sets: (A) Combined QC4 Measure, (B) Individual QC4 Measures, (C) Precision and Recall, (D) Entropy and Mutual Information

lower is better.

On average QDC performs substantially better than the other algorithms. Figure 1 shows that for full text data, on average, QDC outperforms all of the other algorithms on all measurements by convincing margins. Figure 1 (A) shows the overall percentage improvement each algorithm makes over the random clustering using the combined QC4 measure $\frac{1}{2}(H(AQ, AC) + H(WQ, WC))$, where H is the Harmonic Mean, AQ is Average Quality, AC is Average Coverage, WQ is Weighted Quality, and WC is Weighted Coverage.

A more detailed investigation of all test cases shows that QDC was almost universally better than the other algorithms. In 40 of the 44 full text test cases (11 measurements on each of 4 data sets), QDC was significantly better than all the other algorithms. In the four cases where QDC was worse, QDC had second best performance. The four cases were for the "salsa" data set, which was the easiest search as all algorithms performed comparatively well on this data set. In all cases where QDC performed worse, the advantage of the other algorithms was very marginal (typically a few percent). Furthermore, when considering the trade-offs, it was clear that QDC performed better overall. When QDC had slightly worse average and weighted precision and entropy than STC, it had significantly better average and weighted recall and would be significantly better on a combination score that balanced both factors in the trade-off.

We also evaluated the performance of QDC against the other algorithms at clustering just snippet data. Figure 2 shows the 11 measurements averaged across the four data sets and shows the percentage improvement each algorithm makes over the random clustering. The results show that QDC offers a very large and significant improvement in performance over other clustering algorithms. QDC has better performance in all but the unreliable recall measurements (where QDC is slightly outperformed by K-means), but QDC does significantly better on precision and entropy and would be significantly better on a combination score that balanced both factors in the trade-off.

As with the full text, QDC was almost universally better than the other algorithms on the snippet data sets. In 38 of the 44 snippet test cases, QDC was significantly better than all the other algorithms. In five of the six cases

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE

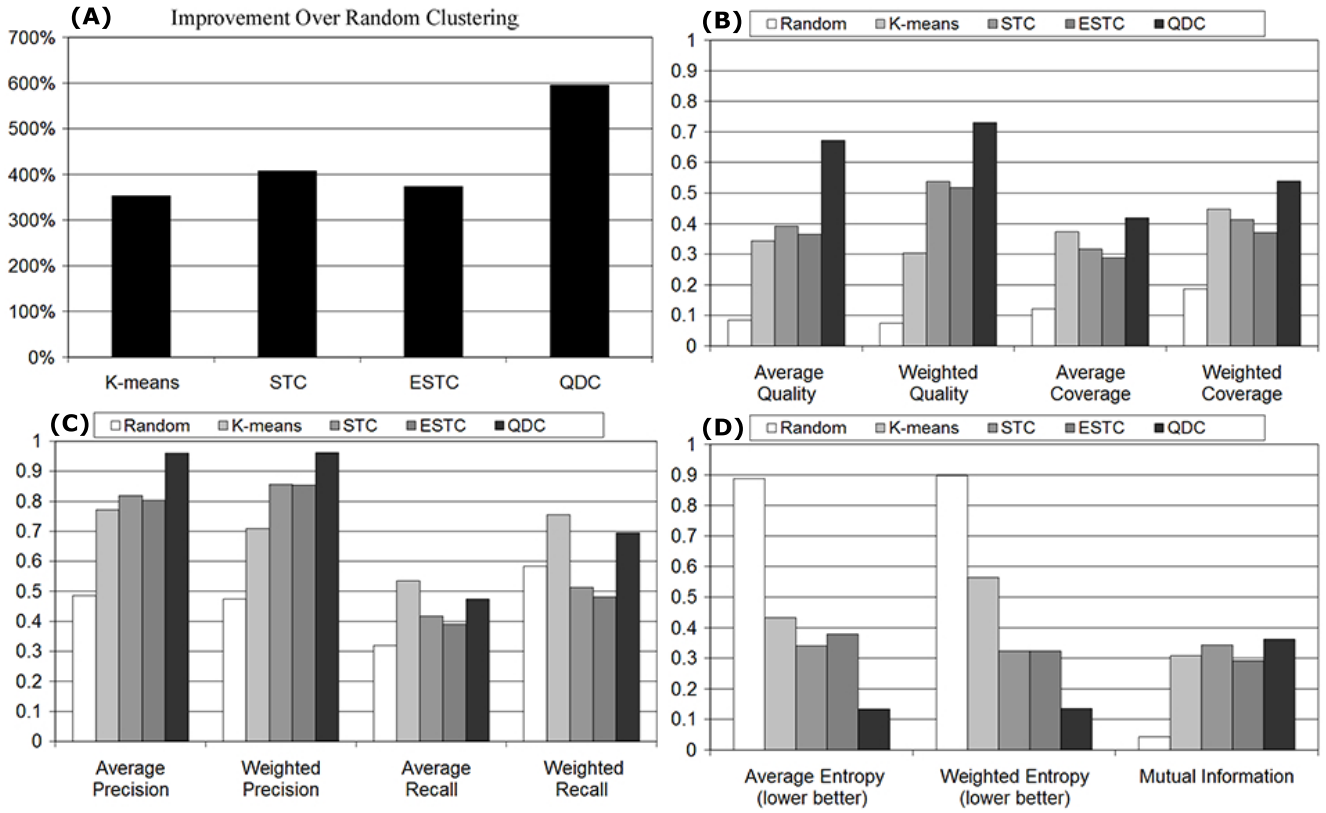

Figure 2. Snippet Results Averaged Over All Data Sets: (A) Combined QC4 Measure, (B) Individual QC4 Measures, (C) Precision and Recall, (D) Entropy and Mutual Information

where QDC was worse, QDC had second best performance. Four of the cases were for weighted recall, a particularly unreliable measure that often gave better performance to the random clustering than to other algorithms. The other two cases were the coverage for the "salsa" data set. In all six cases the coverage or recall were only slightly worse (a few percent), but the quality, precision, and entropy were much better (twice as good in five of the six cases).

4.3 Stage Performance and Sensitivity

We conducted experiments to discover the relative importance of each of the five innovations, which roughly correspond to the five stages of the algorithm. Each innovation and stage of the algorithm individually has a positive effect on clustering or algorithm performance, though not as much as the combination of all five.

The query directed cluster quality guide has a large impact on performance. The pruning it enables in the first stage of the algorithm dramatically improves algorithm speed and clustering performance. Using the cluster description similarity in the second stage significantly improves both cluster quality and cluster coverage. The new cluster splitting method in the third stage solves the cluster chaining (drifting) problem and improves cluster quality; we independently verified this by applying the cluster splitting method to synthetic test cases, including cases that exhibited cluster chaining or drifting.

The improved heuristic of the fourth stage improves cluster selection speed significantly over ESTC and makes far better selections. The improvement to the heuristic is an additional result of the query directed cluster quality guide. The ranking method used in the final stage improves cluster quality, but does not contribute much to the external evaluation, as the ordering is not taken into account. We conducted an independent analysis of the ranking method and found that most of the pages that were hurting cluster quality were placed very close to the bottom of the cluster rankings; when sorted by search position, they were distributed randomly throughout the cluster rankings.

The heuristics in QDC use quite a number of parameters. For the experiments above, we did minimal tuning of the parameters on one of the eight data sets. We ran further experiments to explore the sensitivity of the results to

the parameter values and found that with one exception all parameters were able to be shifted in either direction by at least 20% without making more than a ±2% difference in clustering performance. The query distance threshold in the first stage of the algorithm was more sensitive: shifting this by 20% could make up to a ±6% difference in clustering performance. This is a further indication of the importance and effect of query distance in QDC. It may be worth tuning this parameter.

4.4 Real World Usability

The results of the external evaluation are impressive, but the real test of a web page clustering algorithm is end user usability. While we acknowledge a formal user study would best confirm the results from the external evaluation, at this stage, we can only provide an informal analysis and comparison with other clustering algorithms. The analysis used the same four queries as the external evaluation ("salsa", "jaguar", "gp", and "victoria university") and indicates the results from the external evaluation may have underestimated the real world usability and performance of QDC. The remainder of this section presents the informal analysis of the "jaguar" query (the results were similar for the other queries).

Table 1 shows the cluster names and number of pages in each cluster produced by QDC, K-means, ESTC, Lingo, and Vivisimo [14] for the search "jaguar", sorted by size. STC (due to result similarity with ESTC) and Random clustering (due to its obviously poor performance) are excluded here, but were included in our analysis. Lingo results are from http://carrot.cs.put.poznan.pl and Vivisimo from http://vivisimo.com. Unlike the other algorithms, Lingo and Vivisimo clustered snippets instead of full-page data and used different data sets of 200 and 228 pages respectively. We made several minor changes to the Lingo and Vivisimo clusters: normalizing cluster sizes to account for the different data set sizes, and truncating overly long cluster names. For Lingo, we display only the ten largest clusters of twenty-five.

An informal analysis of the clusters produced by the algorithms shows that QDC finds larger, broader clusters such as "Car", while the other algorithms find smaller more specific clusters such as "Locate a Used Car" and "Jaguar Auto Parts". A problem with capturing topics that are more specific than necessary is that topics of interest to some users may not be covered at all. Showing broader topics both maximizes the probability of a user being able to refine their query and simplifies the user's decision process. The decision process is simpler as there are fewer choices, and it is less likely that there are multiple relevant choices. While the smaller clusters that relate to sub-topics of "Car" are valid and semantically meaningful, they are better left for refinements of a more specific search, for instance, "jaguar car". With "jaguar", there are more obvious refinements that should be made first, and they are exactly those captured by QDC.

Table 1. Clusters for "'jaguar'

QDC		K-means		ESTC	
Car	109	Include	115	Car	56
Cat	48	Car	22	OS 10	33
Other	40	OS	17	Panthera onca	21
Apple	35	Free	16	Online	9
Atari	18	Largest	14	Pictures	9
		Type	13	System	8
		Atari	12	Racing	7
		Service	12	Prices	7
		Panthera	9	Auto	7
				Wildlife	7

Lingo		Vivisimo	
Other	68	Club	48
Locate a Used Car	29	Parts	46
Mac OS Jaguar	24	Jaguar Cars	41
Cat the Jaguar	20	Photos	32
Jaguar Auto Parts	18	Classic	16
Safety Information	16	Animals	7
Jaguar Club	15	Mark Webber	7
Home Page	13	Maya	5
Official Web	13	Enthusiast	4
Amazon.com Books	11	Panthera onca	4

The informal analysis also shows that QDC finds fewer semantically meaningless clusters compared with the other algorithms. For instance, QDC found none when clustering "jaguar", whereas K-means found three ("include", "free", and "service"), ESTC and Lingo each found two, and Vivisimo found one.

The informal analysis also indicates that the usability and performance of QDC is even better than is shown by the external evaluation, because the evaluation did not penalise the creation of overly specific clusters since the gold standard included them. What the external evaluation does show is that of the clusters produced by each algorithm, those produced by QDC had fewer irrelevant pages and covered additional relevant pages.

5 Conclusions and Future Work

This paper has presented QDC, a new query directed web page clustering algorithm that has five key innovations. Firstly, it identifies better clusters using a query directed cluster quality guide that considers the relationship between a cluster's descriptive terms and the query terms. Secondly,

it increases the merging of semantically related clusters and decreases the merging of semantically unrelated clusters by comparing the descriptions of clusters in addition to comparing the overlap of page contents between clusters. Thirdly, it fixed the cluster chaining (drifting) problem using a new cluster splitting method. Fourthly, it chooses better clusters to show the user by improving the ESTC cluster selection heuristic to consider the number of clusters to select and cluster quality. Finally, it improves the clusters by ranking the pages according to cluster relevance.

The gold standard evaluation used QC4 measurements of cluster quality and cluster coverage, and the standard measurements of precision, recall, entropy, and mutual information on eight data sets (four queries using full text data and four queries using snippet data) to show that QDC provides a substantial improvement over Random, K-means, STC, and ESTC clustering algorithms. Additionally, an informal usability evaluation showed that QDC performs very well when compared with Random, K-means, STC, ESTC, Lingo, and Vivisimo and the gold standard evaluation may have underestimated the performance of QDC.

While the results are already very impressive, QDC only considers single words; STC, Lingo, and other clustering algorithms have shown that using phrase information can provide a dramatic improvement. One obvious direction for future work is to extend QDC to use phrases rather than just words. Another direction for future improvement is to consider multiple terms from the cluster descriptions when merging clusters instead of just considering the most descriptive term.

Acknowledgment

Daniel Crabtree is supported by a Top Achiever Doctoral Scholarship from the Tertiary Education Commission of New Zealand.

References

[1] J. Back and C. Oppenheim. A model of cognitive load for ir: implications for user relevance feedback interaction. *Information Research*, 6(2), 2001.

[2] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[3] H. Chen and S. Dumais. Bringing order to the web: automatically categorizing search results. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, 2000.

[4] R. Cilibrasi and P. M. B. Vitanyi. Automatic meaning discovery using google. `www.cwi.nl/paulv/papers/amdug.pdf`, 2004.

[5] M. D. Cock and C. Cornelis. Fuzzy rough set based web query expansion. *International workshop on Rough Sets and Soft Computing in Intelligent Agent and Web Technologies*, pages 9–16, September 2005.

[6] D. Crabtree, X. Gao, and P. Andreae. Improving web clustering by cluster selection. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 172–178, September 2005.

[7] D. Crabtree, X. Gao, and P. Andreae. Standardized evaluation method for web clustering results. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 280–283, September 2005.

[8] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[9] F. Menczer. Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology*, 55(14):1261–1269, December 2004.

[10] S. Osiński, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference*, Advances in Soft Computing, pages 359–368, Zakopane, Poland, 2004. Springer.

[11] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, May/June 2005.

[12] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[13] A. Schenker, M. Last, H. Bunke, and A. Kandel. A comparison of two novel algorithms for clustering web documents. In *Proceedings of the 2nd International Workshop on Web Document Analysis (WDA 2003)*, pages 71–74, Edinburgh, Scotland, August 2003.

[14] A. Spink, S. Koshman, M. Park, C. Field, and B. J. Jansen. Multitasking web search on vivisimo.com. In *International Conference on Information Technology: Coding and Computing (ITCC'05)*, volume II, pages 486–490, 2005.

[15] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[16] A. Strehl. *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, Faculty of the Graduate School of The University of Texas at Austin, 2002.

[17] Y. Wang and M. Kitsuregawa. On combining link and contents information for web page clustering. In *13th International Conference on Database and Expert Systems Applications DEXA2002, Aix-en-Provence, France*, pages 902–913, September 2002.

[18] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998.

[19] S. M. zu Eissen, B. Stein, and M. Potthast. The suffix tree document model revisited. In *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 2005)*, Graz, Austria, 2005.