Clustering Web Queries

John S. Whissell, Charles L.A. Clarke, Azin Ashkan
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
jswhisse@cs.uwaterloo.ca, claclarke@plg.uwaterloo.ca, aashkan@cs.uwaterloo.ca

ABSTRACT

Despite the wide applicability of clustering methods, their evaluation remains a problem. In this paper, we present a metric for the evaluation of clustering methods. The data set to be clustered is viewed as a sample from a larger population, with clustering quality measured in terms of our predicted ability to discriminate between members of this population. We measure this property by training a classifier to recognize each cluster and measuring the accuracy of this classifier, normalized by a notion of expected accuracy. To demonstrate the applicability of this metric we apply it to Web queries. We investigated a commercially oriented data set of 1700 queries and a general data set of 4000 queries. Both sets are taken from the logs of a commercial Web search engine. Clustering is based on the contents of search engine result pages generated by executing the queries on the search engine from which they were taken. Multiple clustering algorithms are crossed with various weighting schemes to produce multiple clusterings of each query set. Our metric is used evaluate these clusterings. The results on the commercially oriented data set are compared to two pre-existing manual labelings, and are also used in an ad clickthrough experiment.

General Terms

Experimentation, Algorithms

Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering—*algorithms*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering*

Keywords

Clustering, Clustering Evaluation, Query Intent Detection

CIKM'09, November 2–6, 2009, Hong Kong, China.

1. INTRODUCTION

Despite their wide applicability, clustering methods suffer from notable problems, including the evaluation of results. For example, consider the domain of text clustering where many clustering, methods have been applied successfully ([15, 28, 35, 12, 16]). A major issue with clustering evaluation in text domains is that it is typically based on a comparison to ground truth labelings produced manually. Typical evaluation metrics such as normalized mutual information (*NMI*), purity, and f-measures all equate higher similarity to ground truth with higher quality clustering results. These metrics are reasonable measures of quality when one requires clusterings similar in form to the ground truth being used. Problems arise when two or more unrelated, but useful, ground truths exist. Metrics based on ground truth will identify a clustering result as 'good' with respect to at most one such ground truth. In the absence of ground truth; algorithm-specific objective functions are often used for evaluation. However, when such functions are used, it is often unclear how to compare results from clustering algorithms run on different weightings and/or representations of the same data set. In addition, objective functions necessarily have a bias favoring a specific kind of structure, and there are no assurances all useful labelings of a data set will exhibit this structure.

In this paper, we present a metric for evaluating the quality of clustering results that does not require comparison to ground truth or the use of a specific clustering algorithm's objective function. The key to our method is to place clustering in a larger framework: We view the data set to be clustered as a sample of a larger population, and the clustering will be applied in some (as yet unknown) task on the population. Regardless of the task, we can say that discrimination is a key property of clustering in this framework. We present a classifier based solution to measure the expected discrimination quality on the population. We train a classifier to recognize each cluster and take the accuracy of this classifier, normalized by a notion of expected accuracy, as a measure of the overall quality of the clustering. Classifier bias is dealt with by allowing potentially many classifiers to be applied to the same clustering, with only the best (most discriminating) classifier used in the final measure. By measuring clustering quality in this way, not only are ground truths not required (we may find very different clusterings all with good quality) but the problems associated with comparing clusterings based on different weightings/representations of a data set are avoided.

1.1 Clustering Web Queries

We apply our evaluation technique in the context of Web query intent analysis. Classic efforts to detect user goals in Web search depend upon user surveys, interviews, and the manual inspection of query logs. Seminal work by Broder [6], Rose and Levinson [25] and Lee et al. [21] recognizes a fundamental distinction between *navigational* queries and *informational* queries. The intent behind a navigational query is to locate a specific page or site on the Web, which is known or expected to exist. The intent behind an informational query is to learn something about a specific topic, with a lesser regard for the source of that knowledge, provided that it is reliable. More recent efforts to improve sponsored search have recognized a distinction between *commercial* and *non-commercial* queries [9]. The intent behind a commercial query is the possible purchase of product or service, either immediately or in the future.

Queries may be classified along both dimensions. For example, the query "University of Washington" might be classified as a navigational/non-commercial query, seeking the university's home page; the query "cheap air tickets" might be classified as an informational/commercial query, seeking discount travel from any source.

In this paper, we replace manual inspection of query logs with automatic inspection through clustering and the use of our clustering quality metric. While nothing can entirely replace user surveys and interviews for understanding user intent, Web query intent analysis is an ideal area to apply clustering methods and use our metric to select good clusterings as multiple ground truth of intent are already in use (commercial/non-commercial and navigational/informational). We may gain additional insight by applying clustering methods and examining the good clusterings (by our metric) that arise naturally.

In our initial experiments (Sections 2-6), we focus on binary clusterings (which we call "splits") in order to better compare our clustering results with manually labeled splits corresponding to the navigational/informational and commercial/non-commercial dimensions. We use ten clusters in our final experiment, reported in Section 7, where manual labels are not available. This second experiment provides a finer grained view of general Web queries.

1.2 Overview

In Section 2 our clustering algorithms and our commercially-oriented data set of search engine logs are detailed. In Section 3, we perform a standard analysis of our clustering splits, except that two manual labelings are used. We show that many of the splits learned by the clustering algorithms on this data set are similar to the manual commercial/non-commercial split, while none are strongly similar to the manual navigational/informational split. This section highlights the problem with using ground truth to evaluate clustering. In Section 4 we explain our new classification-based metric for assessing clustering quality, comparing them to a traditional measure of internal similarity. We use this metric in Section 5 to select clustering results for further study, showing that two strong learnable splits exist, one of which is related to the manual commercial/non-commercial split, the other of which is not closely related to either manual split. Section 6 discusses an experiment to predict advertising clickthroughs, applying the results of the previous clustering experiments. In these experiments, the best clusterings of each of the two strong learnable splits are incorporated into models of ad clickthrough behavior to illustrate that they may reflect user intent and therefore have external validity. Section 7 details the results of the application of the clustering algorithms and weighting functions in Section 2 to a separate set of general Web queries. Our quality metric from Section 4 is used to select the result to examine. This result is shown to be an easily interpretable partitioning of Web queries, providing further evidence regarding the validity of our metric. Section 8 gives our concluding discussion.

1.3 Related Web Query Clustering Work

The problem of clustering Web queries has received recent attention. A comparison between manual query intent labelings and unsupervised labelings was reported by Nettleton et al. [23]. In their work a Kohonen SOM was compared against Broder's categories [6]. Baeza-Yates [2] discusses using k-means clustering on web query logs. In later work Baeza-Yates et al. [3] compare the results of *Probabilistic Latent Semantic Analysis* [14] with categories taken from the Open Directory Project[1]. Beeferman and Berger [4] design a hierarchical clustering algorithm for web queries and URLs. Hosseini and Abolhassani [17] design another hierarchical algorithm for URLs and queries, which applies singular value decomposition before clustering.

2. EXPERIMENTAL SETUP

In the following subsections we detail the data set for the experiments reported in Sections 2-6 including its preprocessing and our clustering algorithms.

2.1 Data Set

The data set we used in our initial experiments is a sample of web search engine logs obtained from Microsoft adCenter, sampled over a few months. The data set was designed to investigate commercial search (as we do in Section 6) and includes a record of queries entered, ads displayed and ads clicked. Personally identifying information was removed from this data set. The data includes a sample of roughly 100 million search impressions, where an impression is defined as a single search result page. Queries are assumed to be in the English language. Any extra space at the beginning and end of the queries, and between words of the queries, were removed. All queries were case-normalized. We found about 27 million queries occurring only once in the impression file, mostly with no ads. Such queries were removed from the impression data. Impressions with a duplicate combination of impression id and user session id were removed in order to filter out repeated queries from the same user. The queries subject to manual labeling were the same 1700 queries used in [1], selected through the following procedure: the original impression file was sorted based on the time of the impression. Starting from an arbitrary point in the file (approximately 1/5 of the length of the file from the beginning), 1700 queries were selected for which the ad click frequency of the query was above 10. Because of this ad click based filtering, we refer to this set of 1700 as *commercially-oriented*. In Section 7 we will examine a data set without ad click filtering.

Each selected query was then manually labeled by three researchers in our lab. Working independently, each re-

[1] `dmoz.org`

searcher labeled each query as either navigational or informational, and as either commercial or non-commercial. In the case of disagreement, the majority opinion was taken. These labelings served as two equally valid interpretations of ground truth in our experiments, and will be referred to as *manual splits* throughout the rest of the paper.

For each query, we have two types of features available: i) the contents of a search engine result page (*SERP*) generated by executing the query on a commercial search engine, and ii) the query-specific features extracted from the query string. Since the experiments in Section 6 are based on ad clickthrough, SERP content was filtered to remove ads to avoid tainting the clickthrough prediction experiment, leaving only the organic search results for feature generation. For the majority of the paper, Sections 3-5 and for the additional experiments in Section 7, we use only the SERP features. The query-specific features are used only in Section 6, when user behavior is considered.

Clustering algorithms typically require data sets of real vectors. SERPs may be converted to this form by first removing all tagging and other extraneous information. In this form, a SERP can be represented as an unordered collection of words. A feature vector may then be derived from this collection of words. The set of SERPs, one for each query, may be represented as a set of vectors, each one of the form

$$\langle tf_1, tf_2, \ldots, tf_m \rangle,$$

where tf_i, the *term frequency* of word i, is the number occurrences of word i in the SERP. We apply the above transformation to the SERPs to generate *tf* vectors. We further remove all words entirely from the vectors that occur in less than 4 SERPs.

In addition to working directly with these raw term frequency vectors, we experiment with various weighting functions, each giving a different set of vectors. Weighting functions for text clustering most often follow a *tf-idf* scheme. In such schemes, the final weighted value for term i is some function of its term frequency (*tf*) multiplied by some function of its inverse document frequency (*idf*). A length normalization is typically applied after all terms have been weighted, such that each vector is of unit Euclidian length.

The set of weighting functions we used is summarized in Table 1. Each weighting function was applied to the *tf* vectors to produce eight different weighted data sets. We refer to each weighting method by its short name listed under the column *Short* in Table 1. The *o* weighting function is a variant of Okapi BM25, where b and k_1 are constants. Although normally learned from training data, here we assume values of $b = 0.75$ and $k_1 = 1.2$, which are typical for BM25. Several of the other weighting functions are simplified variations of *o*. *kt* may be obtained by removing the *idf* component and setting $b = 0$. *Kt* and *kti* functions may be derived using other simplifications. The *t* function is the raw SERP vectors. log *tf*-log *idf* is a common weighting function in clustering research. β is binary weighting, 1 if a word occurs once or more, 0 otherwise. *i* is simple *idf* weighting. To our knowledge none of the variants of Okapi BM25 we used here have been previously applied to text clustering. However, the BM25 weighting is known to perform well for search, and many of the reasons for its success in that domain may apply equally well here (and we shall see that it does).

Table 1: Data set weighting methods

Weighting	Short	Function
raw *tf*	*t*	tf_i
binary	β	1 if $tf_i > 0$, 0 otherwise
log *idf*	*i*	$\log(idf_i)$
log *tf*-log *idf*	*ti*	$\log(tf_i)\log(idf_i)$
simplified *k1tf*	*kt*	$\frac{(k_1+1)tf_i}{k_1+tf_i}$
k1tf-log *idf*	*kti*	$\frac{(k_1+1)tf_i}{k_1+tf_i}\log(idf_i)$
k1tf	*Kt*	$\frac{(k_1+1)tf_i}{k_1(1-b+b(d/avgd))+tf_i}$
okapi	*o*	$\frac{(k_1+1)tf_i}{k_1(1-b+b(d/avgd))+tf_i}\log(idf_i)$

2.2 Clustering Algorithms

All the clustering algorithms we selected produce a strict partitioning of a data set. We refer to such a partitioning as a *clustering*, and each group within the partition as a *cluster*. Each of the algorithms takes k, the number of clusters to return, as a parameter. For reasons detailed in the introduction, our selection of clustering algorithms was not based on expected ground truth approximation. Instead, we focused on algorithms that are both relatively simple and have been widely applied to text clustering problems in prior research. Our selections were as follows:

1. *k-means* clustering (*kmeans*) using Lloyd's method [22]
2. *Normalized-Cut Spectral* clustering (*spect*) [27]
3. *UPGMA* clustering (*upgma*) [29]
4. *Single Link* clustering (*slink*) [18]
5. *Complete Link* clustering (*clink*) [18]
6. Document clustering algorithms based on some of the objective functions from Zhao and Karypis [35]. Specifically the *e1*, *i1*, *i2*, *g1*, *g1p*, and *h1* objective functions.

K-means using Lloyd's method is a classic partitional clustering algorithm that looks for k centroids of a data set such that the total squared Euclidian distance of each data point to its nearest centroid is minimized. The spectral clustering algorithm we used is discussed in Shi and Malik [27].

UPGMA, single link, and complete link clustering are from the same family of agglomerative clustering algorithms. In each of these methods every point begins as a singleton cluster and clusters are progressively merged with the best similarity. In single link similarity between clusters is equal to the similarity of their closest points. In complete link the farthest points are used. In UPGMA the average similarity between all points is used.

We apply the objective functions from Zhao and Karypis in an agglomerative framework: every object begins in its own cluster. Clusters are merged iteratively such that the objective function is optimized for each individual step. The *e1* function is based around minimizing the weighted similarity of cluster centroids from the centroid of the whole data set. The *i2* function is essentially the *k*means objective function except any similarity metric may be used in the calculation. The *i1* function is similar to UPGMA. The *h1* function is *i1*/*e1*.

Both *g1* and *g1p* are graph-cut based objective functions. The *g1* function is based on MinMaxCut [11], while the *g1p* function is a combination of *g1* and *e1*.

For all our clustering algorithms except k-means, cosine similarity was used. Our k-means method used squared Euclidian distance. As we used random starting points in our k-means algorithm it was not deterministic. To compensate for lack of determinism we ran it 20 times with each weighted data set and kept the best result by the k-means objective function for further analysis. Each k-means clustering was run till complete convergence.

The first step in our experiments consisted of clustering each weighted data set using each clustering algorithm detailed above, with two clusters as output. This resulted in 88 different binary clusterings, one for each (algorithm, weighted data set) pairing. Each of these will be referred in the form <weighting>-<algorithm> throughout the paper.

3. SIMILARITY TO MANUAL LABELINGS

Before any attempt was made to evaluate the clusterings without ground truth, we were interested in determining which clusterings were similar, if any, to either manual split (in a typical experiment, this would be done to measure the quality of clustering algorithms). Our intention was to determine if the clusterings naturally fit the manual splits. Similarity to the manual splits was assessed using *normalized mutual information* (*NMI*). NMI between two discrete random variables X and Y is given by the equation

$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}. \quad (1)$$

$I(X;Y)$ is the mutual information between X and Y, defined as

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log(\frac{p(x,y)}{p(x)p(y)}), \quad (2)$$

where $H(X)$ is the entropy of X

$$H(X) = \sum_{x\in X} p(x)\log(p(x)). \quad (3)$$

NMI may be viewed as the reduction in uncertainty of X knowing Y's value. This value is symmetric and between 0 (independent) and 1 (perfectly associated). Here, X is a clustering and x is a cluster. Y is one of the two manual labelings and y is a category for that labeling. Let n be the size of the data set (the number of queries). Then we estimate $p(x) = \frac{|x|}{n}$, $p(y) = \frac{|y|}{n}$, and $p(x,y) = \frac{|x\cap y|}{n}$. Each of our 88 clusterings had its NMI with both manual splits computed. The top 5 NMI clusterings with respect to the manual commercial/non-commercial split are given in Table 2. Table 3 gives similar results for the manual navigational/informational split.

Table 2 shows that some of the clusterings share substantial similarity with the commercial/non-commercial split. A maximum NMI of 0.29432 was obtained by the weighting/algorithim combination *kt-e1*.

The magnitude of similarity is more easily observed with keywords. We use a weighted *pointwise mutual information* formula to quantify the keyword strength of a term to a single cluster or category. *Pointwise mutual information* (*PMI*)

Table 2: Top five NMIs for the commercial/non-commercial split.

	NMI	
Split	**Commercial/ Non-Commercial**	**Navigational/ Informational**
kt-i1	0.29432	0.00483
β-i2	0.27218	0.00268
kt-e1	0.25288	0.00206
o-e1	0.24590	0.00154
kt-spect	0.24317	0.00103

Table 3: Top five NMIs for the navigational/informational split.

	NMI	
Split	**Commercial/ Non-Commercial**	**Navigational/ Informational**
i-i1	0.09616	0.02636
lti-i1	0.08710	0.02123
Kt-clink	0.04730	0.02006
kt-g1p	0.03530	0.01554
lti-spect	0.10080	0.01535

is defined as

$$\mathrm{PMI}(x,y) = log(\frac{p(x,y)}{p(x)p(y)}) \quad (4)$$

Our *weighted pointwise mutual information (WMPI)* metric is defined as

$$\mathrm{WPMI}(x,y) = p(\mathrm{tf}_y > 0|x)\mathrm{PMI}(x,y). \quad (5)$$

Here, x is a cluster from a clustering or a label category from a manual labeling; and y is a word. $p(x) = \frac{|x|}{n}$ and $p(y) = \frac{p(tf_y>0)}{n}$, where $p(tf_y > 0)$ is the estimated probability of a data point having one or more occurrences of word y; $p(x,y)$ is the probability a data point has one or more occurrences of word y and belongs to the category or cluster x ($p(x,y) = \frac{p(tf_y>0, X=x)}{n}$); $p(\mathrm{tf}_y > 0|x)$ is the fraction of points in x where y occurs once or more.

WPMI rewards a word for being common in a category or cluster and for having high PMI. Table 4 shows the highest WPMI words for each category in the commercial/non-commercial split and its two most similar clusterings from Table 2.

A commercial cluster is fairly obvious in both of the clustering results in Table 4. The non-commercial clusters also match the non-commercial category to a reasonable extent. We suggest that the very similar keywords but only moderate NMI indicate that the commercial clusters in our clustering results correspond to queries that evoke strongly commercial responses in search engines. This is different from the manual commercial/non-commercial split — in that split commercial was defined as the query itself having commercial intent. Many clustering results were similar to the commercial/non-commercial split. This suggests that commercial/non-commercial or some variation on that theme may be a natural split for commercially-oriented Web queries (at least from the standpoint of SERP features).

Table 3 shows that no clusterings shared significant NMI with the navigational vs. informational split. The highest

Table 4: The highest WPMI words for commercial vs. non-commercial split and its two most similar clusterings. Each row is one category/cluster within the respective labeling/clustering.

Commercial vs. Non-Commercial		accessories, prices, discount, store, shop, buy, deals, company, offers, our, sale, stores
		news, video, videos, games, web, download, live, people, game, play, this, what, downloads
i1-kt	**A**	prices, discount, deals, buy, store, accessories, sale, stores, shop, selection, vacation
	B	news, video, games, videos, access, page, web, this, can, as, what, information, wikipedia
i2-β	**A**	prices, discount, buy, accessories, deals, selection, store, shop, car, shopping, our, shipping
	B	news, information, wikipedia, video, encyclopedia, free, page, games, as, this, access, web

NMI value obtained was 0.02636 (*i-i1*). Despite little evidence in our clusterings, the navigational/informational split has been shown to exist quite strongly in practice [6, 25]. Further, in the following Section we show that SERPs of the data set do contain the necessary information to build a navigational/informational classifier (80.4% accuracy). Non-SERP text features of search engine logs may contain stronger indications of the navigational/informational split. These features include dwell time — the latency between SERP display and the first click — and the shape of clickthrough curves [21].

We make a final observation in this section. In the preceding experiment we had two manual splits — a navigational/informational labeling and a commercial/non-commercial labeling. These were almost entirely unrelated (NMI of 0.0305). If we treat these as ground truth to evaluate the clustering algorithms we see significantly different results depending on the ground truth. This highlights the problem of using ground truth similarity as a quality measure when multiple ground truth are/may be valid. If we pick a certain ground truth as our standard of quality, we may or may not see any good results, regardless of how well it performs with respect to other ground truths.

4. CLASSIFICATION QUALITY METRIC

From the previous section, it can be said ground truth metrics evaluate algorithms by determining if an algorithm can find a certain structure. This is not the same task as evaluating the clusterings themselves. Every clustering algorithm has an internal objective function which it attempts to optimize. These objective functions represent the algorithm's notion of a good clustering; they do not use true labels, except in semisupervised situations. Variations on k-means [5, 22, 34] use deviation from centroids as quality measures. Likelihoods and model fitting are common objective functions [10, 7]. Information bottleneck methods such as Gondek et al.'s [12] are based around preserving important information in the clustering. SVM based objective functions have also been designed [31].

A specific objective function may be used as an evaluation metric to compare clusterings. Unfortunately, it is usually unclear how to compare objective function evaluations on clusterings based on different representations of the same data set. If we measure the average similarity of objects in the same cluster and use that value as a quality measure, we might get wildly different results for each weighting scheme (we show that this is the case later in the section). Also, objective functions have a bias to certain structures, and there is no reason to believe in general that all useful clusterings may exhibit that same structure.

It is worth noting that selecting the number of clusters is related to evaluating clusterings (in that the clustering with the 'correct' number of clusters is better). There has been significant research on selecting the best number of clusters. Pelleg and Moore [24] use an information criterion approach to iteratively split a data set in to the best number of clusters. Hamerly and Elkan [13] perform something similar except confidence from a Gaussian fitness test is used to select the number of clusters. These two methods may be viewed as wrappers of single algorithms, and like standard algorithms their objective functions are not really comparable across clusterings on different weightings/representations of a data set. Gap statistic methods [30, 33], which select the number of clusters using some form of internal similarity, have a similar issue.

Information criterions (such as BIC [19]) may be applied to evaluate general clusterings of a single data set by balancing the fitness of the clustering (to a certain model) versus the number of parameters used in the fit. Again though, if different representations of the same data set are used in a single experiment comparison in this method becomes unclear. Stability-based methods for comparing clusterings such as Lange et al.'s [20] might be applicable across multiple algorithms, representations, and numbers of clusters, however, some research [26] has shown that stability may not be a suitable measure of clustering quality.

Caruana et al. [8] recognize very different clusterings may be useful for different tasks. They construct a tree of clusterings based on their labeling similarities with encouraging results (the trees are useful for the tasks shown). However, evaluation in their method is a manual process, users must search through the tree. To automate the investigation of the tree would require some ground truth or specific objective function, introducing their commensurate problems.

To avoid using ground truth and specific objective functions for evaluating clustering, we first consider the purpose of clustering. We have a data set. Often, the data set is a sample of some larger population (in our case, we have a sample of Web queries). If this data set is being clustered, we assume that the clustering will be applied in some (as yet unknown) task on the population. Nothing specific is known about the task. Knowing such limited details restricts how we can assess clustering quality. We can, however, say that that regardless of the specific task, being able to discriminate members of the population will be valuable. We can use this discriminative ability as a measure of quality in the absence of a known task. While evaluating in this way does not ensure fitness for specific tasks, it can direct users to clusterings with a definable generic property that is likely useful in a number of tasks.

Ideally manual assessment would be used to estimate the discriminative ability of clustering on the population by observing the clusters themselves. Those assessors would examine the clusters and determine if they can tell them apart.

However, there is a time expense associated with this method. Therefore, we opt to use classifiers in place of people. A classifier may be trained to recognize clusters in a clustering. Using crossfold validation of the classifiers we obtain estimates of the discriminative ability of the clustering on the population in the form of classification accuracy.

We denote the ten-fold crossvalidation accuracy of a certain classifier c as acc_c. Different classifiers may yield very different acc_c.

Unfortunately, it is easy to engineer a situation in which classification accuracy will be arbitrarily high despite the clustering being effectively worthless. This may happen in two ways. First, the data set may be reduced to a trivial representation before clustering. For this paper, we assume that representations are not trivialized in this manner. Secondly, when one cluster contains most of the points in a data set, an extremely good classifier may be designed trivially by assigning all points to the largest cluster.

To factor cluster size balance into accuracy we define a *trivial classifier*. A trivial classifier is one that assigns all points to the largest cluster/category. The average crossfold validation accuracy of a trivial classifier, denoted T_a, is computed as

$$T_a = \arg\max_{x \in X} \frac{|x|}{n}. \tag{6}$$

where X is a clustering or labeling.

Our measure of clustering quality, with respect to the classifier c, then becomes

$$Q_c = \frac{\text{acc}_c}{T_a}. \tag{7}$$

Our final measure of clustering quality, called *normalized accuracy* (N_a), is then defined as

$$N_a = \arg\max_{c \in C} Q_c \tag{8}$$

where C is the set of all classifiers used. Note that only the best classifier is used in computing N_a. This is appropriate, as we are only concerned with our ability to discriminate the clusters, not how the discrimination occurs or if certain classifiers can not discriminate the clustering. Clusterings that maximize N_a will be both balanced in terms of size distribution and have high accuracy under at least one classifier.

To show that N_a can overcome the comparability limitations of objective functions between clusterings on different data representations we illustrate a correlation between N_a using a linear SVM and *internal similarity*. Internal similarity, or within cluster consistency, is a measure of similarity between points in a single cluster. Although optimal clusterings rarely have the best internal similarity possible, it is generally regarded that higher internal similarity indicates a better clustering. The exact definition of internal similarity we use here, which we denote as ISIM-cos, is defined as

$$\text{ISIM-cos} = \sum_{x \in X} \frac{2}{|x|(|x|-1)} \sum_{i,j \in x} \cos(i,j) \tag{9}$$

where X is the clustering, x is a cluster, i and j are individual data points and $\cos(i,j)$ is the cosine between i and j. This makes ISIM-cos the average cosine between points in the same cluster. A linear SVM was used with ISIM-cos as they have related biases. Note that in general there is no assurance that an objective function and a classifier will be correlated in the way we show for ISIM-cos and linear SVM. The experiment we present here is intended only to show that for reasonably matched classifiers and objective functions, comparison across data set representations is not a problem. Classifier/objective functions that do not match will obtain lower acc_cs and not be used in the final computation of N_a (only the most accurate classifier is used in Equation 8), so they will not damage the actual process of evaluating N_a.

Table 5: The top five clustering N_as and the top N_as for each manual labeling.

Split	acc	$\mathbf{N}_a$
β-spect	95.2%	1.88
kt-kmeans	96.6%	1.82
e1-i	90.2%	1.78
Kt-g1p	91.0%	1.78
β-kmeans	95.5%	1.77
Com. vs. Non-Com.	87.4%	1.52
Nav. vs. Inf.	80.4%	1.35

We computed N_a and ISIM-cos for each of the 88 clusterings. We further took the two manual labelings and computed their N_a and ISIM-cos with respect to each of our 8 weighted data sets. Figure 1 shows plots of N_a vs. ISIM-cos for four of our weighted data sets. A positive correlation between N_a and ISIM-cos is visible in each plot. The four other weighted data sets show similar trends but are omitted for space reasons. Notice that ISIM-cos values vary greatly across the four different weightings shown, a comparison using ISIM-cos across these weightings is meaningless. Similar problems would be observable using a large number of objective functions. However, the N_a values across the weightings are of the same scale, suggesting N_a can be used to compare between clusterings on different representations.

Finally, we note that as the number of clusters in a solution increases, the accuracy of a trivial classifier will drop ($T_a = 1/k$ with k perfectly balanced clusters). This results in larger N_a values as k increases unless accuracy drops to match. From our experiments, this does not happen, thus some penalty must be associated with larger number of clusters. However, since the number of clusters is fixed in each experiment, this extension is left for future research.

5. SPLIT DISCOVERIES

The N_a values computed in the previous section were used to analyze our clusterings. We believe the computation of N_a using a linear SVM is appropriate in this case as linear SVMs have been shown to work reasonably well in many text applications, although in general multiple classifiers could be used. Table five shows the top 5 N_a clusterings. The top N_a for each manual labeling is shown as well for comparison purposes. Table 6 shows the highest WPMI keywords for our clustering results. The clusterings are ordered from highest to lowest N_a. Again, the two manual labelings are included for comparison purposes. Their positioning in Table 6 is based on their optimal N_a.

The 22 trivial clustering results mentioned in Section 4 had N_a close to one. There were also a number of non-trivial clusterings with low N_a (such as *kt-g1p* in Table 6). Given their low N_a, all of these were not considered further.

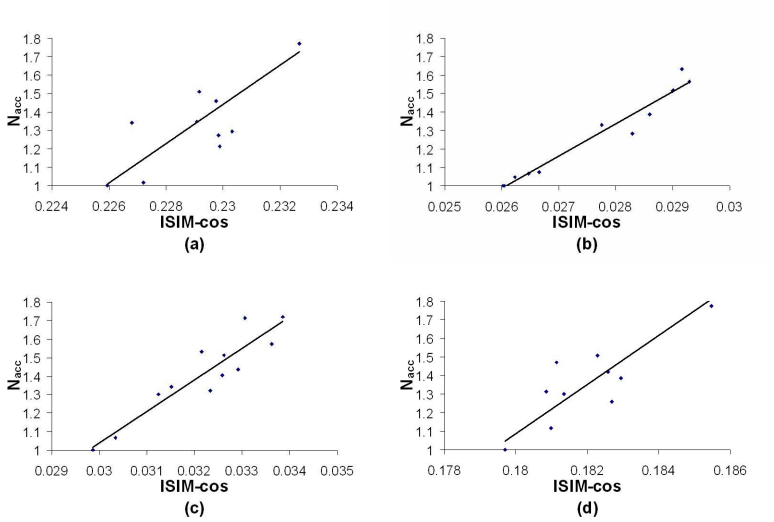

Figure 1: Plots of N_a vs. ISIM-cos for our weighted data sets. (a) is *kt*. (b) is *ti*, (c) is *kti*, and (d) is *i*.

Table 6: The highest WPMI keywords for the 88 clusterings and the manual labelings.

Split	N_a	Cluster/Category	Keywords
β-spect	1.88	A	wikipedia, encyclopedia, news, free, official, information, american, video, latest, 2008, as, an
		B	we, our, your, prices, you, at, great, online, find, selection, discount, are, buy, store, or, have
kt-kmeans	1.82	A	wikipedia, encyclopedia, official, news, free, information, american, 2008, as, an, national
		B	your, online, find, you, we, our, best, at, great, search, have, discount, account, prices, buy
i-e1	1.78	A	news, games, video, latest, videos, internet, downloads, yahoo, music, web, download,
		B	prices, our, offers, discount, store, products, locations, at, we, buy, sale, deals, shop
Kt-g1p	1.78	A	news, games, video, latest, videos, entertainment, play, tv, sports, download, internet, travel
		B	products, our, store, prices, we, are, buy, accessories, shop, selection, or, shipping, stores, sale
β-kmeans	1.77	A	wikipedia, encyclopedia, free, official, an, as, news, american, was, company, its, states
		B	we, your, you, find, prices, great, easy, search, my, best, our, discount, buy, selection, offer
⋮	⋮	⋮	⋮ ⋮ ⋮ ⋮
Commercial vs. Non-Commercial	1.52	A	accessories, prices, discount, store, shop, buy, deals, company, offers, our, sale, stores
		B	news, video, videos, games, web, download, live, people, game, play, this, what, downloads
⋮	⋮	⋮	⋮ ⋮ ⋮ ⋮
i-clink	1.42	A	travel, tickets, us, information, vacation, vacations, reservations, hotel, hotels, airline
		B	videos, music, downloads, new, share, internet, windows, games, download, software
⋮	⋮	⋮	⋮ ⋮ ⋮ ⋮
Navigational vs. Informational	1.35	A	company, service, search, business, contact, account, inc, please, us, mail, careers, home
		B	pictures, guide, source, tips, other, dictionary, symptoms, from, definition, many, natural
⋮	⋮	⋮	⋮ ⋮ ⋮ ⋮
kt-g1p	1.25	A	service, that, our, services, help, use, offers, are, or, you
		B	videos, news, video, games, photos, latest, stats, game, sports, official
⋮	⋮	⋮	⋮ ⋮ ⋮ ⋮

The high N_a results proved interesting. Clusterings that strongly resemble the commercial/non-commercial labeling occur in the top results, namely *i-e1* and *Kt-g1p* (the third and fourth lines of Table 6). They share NMIs of 0.146 and 0.055 respectively with the commercial/non-commercial manual labeling. Despite the NMIs, the keywords are very similar.

Three of the top five clusterings in Table 5 and 6 contain a cluster with "wikipedia" as the first keyword. Queries in these clusters have a strong informational flavor. We refer to these clustering as "Wikipedia splits". Nonetheless, they have NMI close to zero with respect to the navigational/information labeling and only a slightly positive NMI with respect to the commercial/non-commercial labeling.

The majority of higher N_a clusterings were Wikipedia splits or commercial/non-commercial splits. The only other high N_a clusterings contain a strong travel-related cluster and a strong media-related cluster. Such a clustering is illustrated by *i-clink* in Table 6; *kti-clink* had similar keywords and an N_a of 1.30.

To further interpret our results, we performed meta clustering somewhat similar to that in Caruana et al. [8]. Each of the 88 clusterings and the 2 manual labelings formed the objects for clustering. Vectors of NMIs were used to represent each object. The ith index of the jth object's vector was the NMI between the ith and jth object. K-means clustering using Lloyd's method was applied to this data set (as opposed to generating a tree of meta-clusters as in Caruana et al. [8]). For this experiment, we specified a value of $k = 4$ for the number of clusters to be returned. Results for other values were consistent.

To minimize confusion, we refer to a cluster of clusterings as a *meta-cluster*. All of the 22 trivial clusterings were put in to a single meta-cluster. A second meta-cluster contained only five low N_a (< 1.05) clusterings. A third meta-cluster contained a pure collection of 34 of commercial/non-commercial clusterings. The average N_a of the clusterings in this meta-cluster was 1.47. The final meta-cluster contained all the Wikipedia splits (15 of them), the two strong travel/media splits, all the miscellaneous splits (10 in total), and the two manual labelings. This meta-cluster had a lower average N_a of 1.36 because of the weak miscellaneous splits.

We were surprised that the manual commercial/non-commercial labeling did not appear in the meta-cluster with the commercial/non-commercial clusterings. This outcome may further indicate that our clusters are somehow fundamentally different from the manual labeling.

The results of this section indicate that commercial/non-commercial split and Wikipedia split are two main learnable binary splits using commercial-oriented SERPs. Additional strong learnable binary splits may exist, examining this via other clustering algorithms and/or weighting methods is an avenue of further research. Increasing the number of clusters might also produce more informative, finer grained splits.

6. CLICKTHROUGH ANALYSIS BASED ON DETECTED QUERY CATEGORIES

To determine if the splits our unsupervised evaluation method gave high scores to had external validity we used two of them to train ad clickthrough prediction models in this section. This experiment was selected as the data set was collected to investigate commercial behaviors. Recall that the labels for the strongest Wikipedia split came from *β-spect*, while the labels for the strongest automatic commercial/non-commercial split came from *i-e1*. These two labelings, and the commercial/non-commercial manual labeling, became the subject of the experiments in this Section. As in Section 4, clusterings are used to generate classifiers here, so we will refer to clusters as categories.

Table 7: The clickthrough rates for categories in the manual commercial/non-commercial labeling and the best automatic commercial/non-commercial (*i-e1*) and Wikipedia split (*β-spect*).

Split	Cluster/Category	Clickthrough Rate
Manual	Non-Commercial	5.91%
	Commercial	24.50%
β-spect	A	6.95%
	B	20.05%
i-e1	A	7.11%
	B	18.54%

Initially, a linear SVM classifier was built for each labeling. This was done using both the SERP features and the query features. To test these classifiers for clickthrough prediction properties a distinct test set of 45K unique queries, along with their SERP contents, was used (from the same source as the training data set [1]). For our purposes, the average clickthrough rate of a set of queries is the percentage of queries in that set that had an ad click. We first calculated the average clickthrough rate for the objects in the test set without classifying them and found a clickthrough rate of 10.03%. Then the test set was classified using the three classifiers. From these classifications we obtained average clickthrough rates for each category. Table 7 gives these values.

Examining Table 7 shows the unsupervised splits have real meaning in terms of ad clickthrough prediction. This provides further evidence of our algorithm selecting good clusterings (although the predictions are not as strong as the manual split, the unsupervised splits may well be suited for other tasks as well). It is interesting to note that the category with higher clickthrough rates in both unsupervised splits has commercial keywords, such as *price*, *discount*, and *buy*. Unsurprisingly, commercial keywords are associated with a greater frequency of ad clicks.

7. GENERAL WEB QUERY CLUSTERING

We considered the task of clustering general Web query SERPs, as opposed to the commercially-oriented ones used in previous Sections. To do this, we created a sample data set from the same source as the data set in Section 2.1. The filtering procedure on the data set, the collection of the SERPs, and their conversion in to vectors was identical to the procedure described in Section 2.1, except the filtering based on ad clicks was removed, making the queries general as opposed to commercially-oriented. For this experiment we used 4000 Web queries.

All the weighting functions in Table 2 were applied to the vectors to produce eight weighted data sets, each of these of data sets was clustered using the clustering algorithms discussed in Section 2.2. K-means and spectral clustering were omitted from this test as we desired a hierarchy of clusters and only the other nine methods produced such a structure.

This gave 72 (clustering algorithm, weighted data set) pairs. Ten clusters were created in each clustering to give a finer grained view. We computed N_a for each clustering result in the same manner as in the previous experiment. The best N_a value of the 72 clustering results was obtained by the $g1p$ clustering algorithm with o weighting. It scored 6.80 (a perfect result would be 10 in this case). The hierarchy of clusters obtained by this combination is illustrated in Figure 7. Each rectangle is a cluster, with the number on left indicating the percentage of all SERPs in the cluster, and the words on the right being the top WPMI words for that cluster.

The results illustrated in Figure 7 scarcely need deep interpretation. Clusters corresponding to travel, lotteries, finance, education, medicine, cars, shopping, games, entertainment and software are all apparent in the leaves of the hierarchy. The easy interpretability of the clustering our clustering quality metric considers best provides a further indication that the quality metric is a useful method for selecting meaningful clusterings in an entirely unsupervised fashion. Considering the best result in isolation, it may represent a meaningful division of general Web query types and user intents, at least as reflected by search engine results.

8. CONCLUDING DISCUSSION

Existing approaches to the evaluation of clustering methods have several limitations. When clustering methods are compared using ground truth, different interpretations of ground truth will favor different algorithms. In other words, the best clustering method will vary from one problem to another. On the other hand, when using specific objective functions for evaluation, differing data representations and weighting functions confounds the comparison.

Our proposed metric views the target data set to be clustered as having been drawn from a larger source population, with the intention of applying the clustering to some as yet unknown task on that source population. Under this framework, we cluster objects using multiple representations and algorithms, treating the ability to discriminate between clusters as the property to maximize. Classification accuracy, normalized by a form of expected accuracy, is used to measure this discriminative ability, and hence, the quality of a clustering. We call this quality normalized accuracy (N_a).

Normalized accuracy is comparable over different representations of the data set and does not depend on ground truth. In two distinct experiments on Web queries, we show that high N_a clusterings have external validity. In the first experiment, high N_a clusterings of commercially-oriented Web queries were applied to the problem of clickthrough prediction in commercial search. In the second experiment, the optimal N_a clustering of a general Web query set is easily interpretable as a reasonable division of Web content.

One future direction for our research is to extend our metric to select the number of clusters. Investigating the metric's results on other tasks is another avenue, as is using multiple classifiers.

9. REFERENCES

[1] A. Ashkan, C. L. A. Clarke, E. Agichtein, and Q. Guo. Classifying and Characterizing Query Intent. In *Proceedings of the 31st European Conference on IR*, pages 578–586, 2009.

[2] R. Baeza-Yates. Applications of Web Query Mining. In *Lecture Notes in Computer Science: Advances in Information Retrieval*, volume 3408, pages 7–22. Springer Berlin, 2005.

[3] R. Baeza-Yates, L. Calderán-Benavides, and C. González-Caro. The intention behind web queries. In *Lecture Notes in Computer Science: SPIRE*, volume 4209, pages 98–109. Springer Berlin, 2006.

[4] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knolwedge dicovery and data mining*, pages 407–416, 2000.

[5] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press.

[6] A. Broder. A taxomony of web search. ACM SIGIR *Forum*, 36:3–10, 2002.

[7] R. M. Castro, M. J. Coates, and R. D. Nowak. Likelihood based heirarchical clustering. *IEEE Transactions on Signal Processing*, 52:2308–2321.

[8] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith. Meta Clustering. In *Proceedings of the Sixth International Conference on Data Mining*, 2006.

[9] H. Dai, L. Zhao, Z. Nie, J. Wen, L. Wang, and Y. Li. Detecting online commercial intention (OCI). In *15th International Conference on the World Wide Web*, pages 829–837, 2006.

[10] A. Dempster, N. Laird, and D. Rubin. Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society of Britian, B*, 39:1–38, 1977.

[11] C. Ding, X. He, H. Zha, M. Gu, and H. Simon. Spectral min-max cut for graph partitioning and data clustering. Technical Report TR-2001-XX, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA., 2001.

[12] D. Gondek and T. Hofmann. Conditional information bottleneck clustering. In *3rd IEEE International Conference on Data Mining, Workshop on Clustering Large Data Sets*, pages 36–42, 2003.

[13] G. Hamerly and C. Elkan. Learning the k in k-means. In *Neural Information Processing Systems*, 2003.

[14] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[15] T. Hofmann. Probabilistic latent semantic indexing. In *ACM SIGIR 1999*, 1999.

[16] J. Hu, L. Fang, Y. Cao, H. Zeng, H. Li, Q. Yang, Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *ACM SIGIR 2008*, pages 179–186, 2008.

[17] M. Hosseini and H. Abolhassani. *Hierarchical Co-clustering for Web Queries and Selected URLs*.

[18] A. K. Jain, M. N. Murthy, and P. J. Flynn. Data clustering: A review. *ACM Computing Reviews*, pages 264–323, 1999.

[19] R. E. Kass and L. Wasserman. A reference Bayesian test for nested hypothesis and its relationship to the Schwarz criterion. *Journal of the American Statistical Association*, 90(431):928-934, 1995.

[20] T. Lange, M.L. Braun, V. Rother, and J.M. Buhmann. Stability-Based Model Selection. In *Advances in Neural Information Processing Systems* 15, 2003.

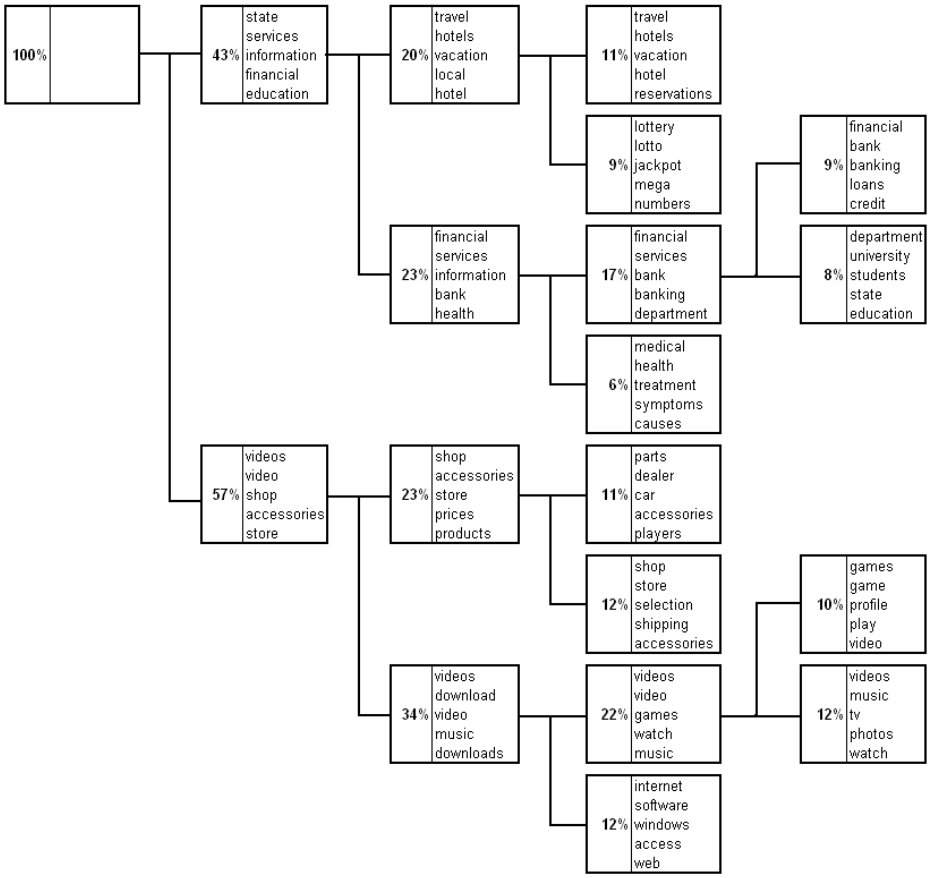

Figure 2: The clustering hierarchy produced by the $g1p$ algorithm on the o weighted data set.

[21] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in Web search. In *Proceedings of the 14th International World Wide Web Conference*, pages 391–400, Chiba, Japan, 2005.

[22] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[23] D. Nettleton, L. Calderán-Benavides, and R. Baeza-Yates. Analysis of web search engine query session and clicked documents. In *Advances in Web Mining and Web Usage Analysis*, pages 207–226, 2007.

[24] D. Pelleg and A. Moore. X-means: Extending k-means with an efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning*, pages 727–734, 2000.

[25] D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proceedings of 13th International World Wide Web Conference*, pages 13–19, 2004.

[26] S. Ben-David, U. von Luxburg, and D. Pal. A Sober Look at Stability of Clustering. *COLT*, 2006.

[27] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 2000.

[28] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In ACM SIGIR 2000, pages 208–215, 2000.

[29] R. R. Snokal and P. H. Sneath. *Numerical Taxonomy*, pages 230–234. W. H. Freeman and Company, San Francisco, 1973.

[30] R. Tibshirani, G. Walter, and T. Hastie. Estimating the number of data clusters via the gap statistic. *Journal of the Royal Statistical Society, Series B* 63, 2001.

[31] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems 17*, pages 1537–1544, 2005.

[32] W. Xu, X. Liu, Y. Gong. Document clustering based on non-negative matrix factorization. In ACM SIGIR 2003, pages 267–273, 2003.

[33] M. Yan and K. Ye. Determining the Number of Clusters Using the Weighted Gap Statistic. *Biometrics* 63.

[34] B. Zhang, M. Hsu, and U. Dayal. Generalized k-harmonic means - boosting in unsupervised learning. Technical Report HPL-1999-124, Hewlett-Packard Labs, 1999.

[35] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *Data Mining and Knowledge Discovery*, 2002.