Mining Acronym Expansions and Their Meanings Using Query Click Log

Bilyana Taneva[1]*, Tao Cheng[2], Kaushik Chakrabarti[2], Yeye He[2]

[1]Max-Planck Institute for Informatics, Saarbrücken, Germany
[2]Microsoft Research, Redmond, WA

[1]btaneva@mpi-inf.mpg.de
[2]{taocheng, kaushik, yeyehe}@microsoft.com

ABSTRACT

Acronyms are abbreviations formed from the initial components of words or phrases. Acronym usage is becoming more common in web searches, email, text messages, tweets, blogs and posts. Acronyms are typically ambiguous and often disambiguated by context words. Given either just an acronym as a query or an acronym with a few context words, it is immensely useful for a search engine to know the most likely intended meanings, ranked by their likelihood. To support such online scenarios, we study the offline mining of acronyms and their meanings in this paper. For each acronym, our goal is to discover all distinct meanings and for each meaning, compute the expanded string, its popularity score and a set of context words that indicate this meaning. Existing approaches are inadequate for this purpose. Our main insight is to leverage "co-clicks" in search engine query click log to mine expansions of acronyms. There are several technical challenges such as ensuring 1:1 mapping between expansions and meanings, handling of "tail meanings" and extracting context words. We present a novel, end-to-end solution that addresses the above challenges. We further describe how web search engines can leverage the mined information for prediction of intended meaning for queries containing acronyms. Our experiments show that our approach (i) discovers the meanings of acronyms with high precision and recall, (ii) significantly complements existing meanings in Wikipedia and (iii) accurately predicts intended meaning for online queries with over 90% precision.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data mining*

Keywords

acronym; acronym expansion; acronym meaning; click log

1. INTRODUCTION

Acronyms are abbreviations formed from the initial components of words or phrases. These components may be individual letters (e.g., "CMU" from "Carnegie Mellon University") or parts of words (e.g., "HTTP" from "Hypertext Transfer Protocol"). Acronyms are used very commonly in web searches as well as in all forms of electronic communication like email, text messages, tweets, blogs and posts. With the emergence of mobile devices, the usage of acronyms is becoming even more common because typing is difficult in such devices and acronyms provide a succinct way to express information.

One key characteristic of acronyms is that they are typically *ambiguious*, i.e., the same acronym has many different meanings. For example, "CMU" can refer to "Central Michigan University", "Carnegie Mellon University", "Central Methodist University", and many other meanings. Consider a web search scenario: given an acronym as a query, it is immensely useful for the search engine to know all its popular meanings, ranked by their popularity. For example, for "CMU", "Central Michigan University" is the most popular meaning followed by "Carnegie Mellon University" and others. The search engine can either modify the original query with these expansions and retrieve more relevant results [7] or simply show them to users so that they can disambiguate it themselves[1].

A second characteristic of acronyms is that they are typically *disambiguated by context*, i.e., the intended meaning is clear when the user provides a few context words. For example, a user searching for "cmu football" is most likely referring to "Central Michigan University" while the one searching for "cmu computer science" is most likely referring to "Carnegie Mellon University". Given an acronym and one or more context words, it is useful for the web search engine to know the most likely intended meaning (or a few most likely intended meanings, ranked by the likelihood). The search engine can then use query alteration techniques to retrieve more relevant results [7].

To enable the above online scenarios, we study the offline mining of acronyms and their meanings in this paper. For each acronym, we discover its various meanings; for each meaning, we output:

- *Expansion*: The complete expanded string of the acronym for the meaning.
- *Popularity score*: A score reflecting how often people intend this meaning when they use the acronym (e.g., how often web searchers intend it when they use only the acronym as the query).

*Part of the work was done during employment at Microsoft Research

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

[1]Google has recently started showing the different meanings of a query on the right hand side of search result page for limited queries. The algorithm used by Google has not been published and is hence not publicly known.

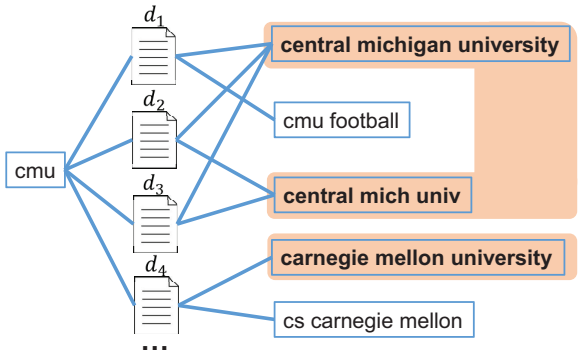

Figure 1: Example illustrating insights.

• *Context words*: A set of words when used in context of the acronym indicates this meaning. Each word has a score reflecting how strongly it indicates this meaning.
For example, for "CMU", we aim to discover the various meanings "Central Michigan University", "Carnegie Mellon University", "Central Methodist University" and so on. The popularity scores should reflect that "Central Michigan University" is more popular compared with the other meanings. Finally, we aim to find context words like "pittsburgh", "computer science", "research", "computing", etc. for the meaning "Carnegie Mellon University". The 1:1 mapping between the output and the meanings is critical to enable the above online scenarios.

There are several efforts in mining expansions of acronyms. We briefly discuss them here; a more detailed discussion can be found in Section 6.

• *Wikipedia*: Wikipedia covers acronyms through its manually edited "disambiguation pages". It has low recall with many meanings not covered. We find from our experiments that roughly two thirds of the meanings of acronyms are not covered in Wikipedia. Furthermore, it does not provide popularity scores.

• *Acronymfinder.com*: Websites such as `acronymfinder.com` list the possible acronym expansions; this is also manually edited. As in Wikipedia, it does not provide popularity scores. Furthermore, it does not provide any context words for most of the expansions.

• *Automatic Mining*: There has been recent work towards automatic mining of acronym expansions using the Web [6]. The main focus of this work is in finding legitimate expansions of a given acronym. However, there is no 1:1 mapping between the outputted expansions and meanings, no popularity scores and no context words.

Due to the above limitations, it is difficult for web search engines to leverage the above approaches to support the online scenarios discussed above.

Our main insight is that acronyms and their various expansions are captured in a search engine query click log. *While some people use acronyms as queries and click on relevant documents, others use their expanded forms as queries and click on the same documents.* Thus, we can find expansions by observing queries that "co-click" on the same documents as the acronym. As shown in Figure 1, by observing other queries co-clicked with query "cmu", we can find acronym expansions such as "central michigan university", "central mich univ" and "carnegie mellon university".

There are several technical challenges in finding the distinct meanings from the co-clicked queries. First, not all co-clicked queries are expansions (e.g., "cmu football" is not an expansion of "cmu" in Figure 1). How do we identify the ones that are expansions? Second, the co-clicked queries that are expansions do not correspond to the distinct meanings. There are several variants that correspond to the same meaning (e.g., "central michigan university" and "central mich univ" in Figure 1). How do we group them such that there is a 1:1 mapping between groups and meanings? Third, how do we identify context words for each meaning? Fourth, co-clicked queries tend to cover the popular meanings of the acronym (e.g., "Massachusetts Institute of Technology" for "MIT") but not the "tail meanings" (e.g., "Mazandaran Institute of Technology", "Maharashtra Institute of Technology", "Mahakal Institute of Technology", etc.). This is because the first few pages of results returned by a search engine for the query "MIT" do not represent the tail meanings. How do we find such tail meanings?

Our main contributions can be summarized as follows:

• We formulate the offline acronym mining problem. The novelty of our problem formulation is to find the distinct meanings, not just the expansions. This is critical to enable the above online scenarios (Section 2).

• We present a novel, end-to-end solution that leverages the query click log to identify expansions, group them into distinct meanings, compute popularity scores and discover context words (Section 3). We leverage two key insights. First, expansions of the same meaning click on the same set of documents, whereas expansions of different meanings click on different documents. We design similarity functions to leverage this insight and perform clustering to group the expansions. Second, co-clicked queries shed light on the context words of respective meanings. For instance, the fact that "cmu football" and "central michigan university" click on the same document hints the relevance of "football" to "central michigan university". We leverage this insight to discover context words.

• We present a novel enhancement to discover tail meanings in addition to the more popular meanings (Section 3).

• We describe how web search engines can leverage the mined information for prediction of intended meaning for queries containing acronyms (Section 4).

• We perform extensive experiments using a large-scale query click log. Our experiments show that our approach (i) discovers acronym meanings with high precision and recall, (ii) significantly complements existing meanings in Wikipedia and (iii) accurately predicts intended meaning for online queries with over 90% precision (Section 5).

To the best of our knowledge, this is the first work on automatic mining of distinct meanings of acronyms.

2. PROBLEM DEFINITION

In this section, we formally define the offline acronym meaning discovery problem and then present our solution overview.

2.1 Problem Definition

We study the following offline acronym meaning discovery problem.

Definition 1. (**Acronym Meaning Discovery Problem**) Given an input acronym, find the set $\{M_1, \ldots, M_n\}$ of distinct meanings associated to it. For each meaning $M_i = (e, p, C)$, find the canonical expansion $M_i.e$, the popularity score $M_i.p$ and the set $M_i.C$ of context words along with scores.

For any meaning, there can be many variants of the expanded string in the query log. For example, for the meaning "Carnegie Mellon University" of the acronym "CMU", the variants include "Carnegie Mellon University", "Carnegie Mellon Univ" as well as several misspellings. $M_i.e$ is the most representative variant; we refer to it as the *canonical expansion*. The popularity score $M_i.p$ measures how often web searchers intend this meaning when they use the acronym in a query. To easily leverage these scores for online meaning prediction, we compute these scores as probabilities. Finally, the set $M_i.C$ of the context words are the words which when used in context of the acronym in web searches indicate this meaning. For example, for the meaning "Carnegie Mellon University", $M_i.C$ = {"pittsburgh", "research", "cs", "science", etc.}. We associate a score with each context word in $M_i.C$ which measures how strongly the word indicates this meaning. Again, to easily leverage these scores for online meaning prediction, we compute these scores as probabilities.

Notice that our problem formulation assumes that the acronym is given. We assume a separate module that identifies the acronyms commonly used in web searches; this can be used as input to the acronym meaning discovery problem. For example, this module can extract all acronyms listed in Wikipedia. Another approach is to treat all words that are not common English words as acronyms. Our framework will be able to find out the meanings of true acronyms, whereas words which are not actual acronyms will not likely produce any meanings.

2.2 Solution Overview

To discover the different meanings of an acronym, we leverage the query click log of a web search engine. Our solution is based on the following insight: while some searchers use acronyms as queries and click on the relevant documents, others use their expanded forms as queries and click on the same documents. We compute the canonical expansions, popularity scores as well as context words for the different meanings of an acronym by observing the set of queries that click on the same documents as the acronym query. We refer to them as "co-clicked" queries.

Query Click Log: The query click log collects the click behavior of millions of web searchers over a long period of time (say, two years). We assume the query log $\mathcal{Q}$ to contain records of the form (q, d, f) where q is a query string, d is a web document, represented by its unique URI, and f is the number of times d has been clicked by web searchers after posing the query q to the search engine.

It is technically challenging to find the distinct meanings from the co-clicked queries. To address this challenge, we develop a novel, end-to-end solution that consists of the following steps:

• **Candidate Expansion Identification:** We first collect the co-clicked queries for the given acronym. Not all co-clicked queries are valid expansions of the acronym. We identify the valid expansions from the co-clicked queries; we refer to them as *candidate expansions*. For this purpose, we use an *acronym-expansion checking function*, which checks if a query can be considered as the complete expanded string of the acronym.

• **Acronym Expansion Clustering:** The candidate expansions do not correspond to the distinct meanings; there are several variants that correspond to the same meaning. We group the candidate expansions such that each group has unique meaning, and no two groups have the same meaning.

• **Enhancement for Tail Meanings:** We observe that co-clicked queries do not cover the tail meanings. To address this problem, we present a novel extension that considers *supersequence queries*. This approach finds significantly more meanings, especially the tail ones. We refer to this algorithm as *Enhanced Acronym Expansion Clustering*.

• **Canonical Expansion, Popularity Score and Context Words Computation:** We select the canonical expansion for each discovered meaning. We compute the popularity score for each meaning, such that more popular meanings receive higher scores. Finally, we assign a set of context words to each meaning. We also assign a score to each context word.

We describe the above steps in details in Section 3. In Section 4, we describe how we can leverage the discovered meanings to predict the intended meanings of online queries.

3. ACRONYM MEANING DISCOVERY

We explain the four steps in details. The output of each step is the input to the subsequent step. We present the input and output of each step followed by the algorithm.

3.1 Candidate Expansion Identification

Input: The acronym a and the query click log $\mathcal{Q}$.
Output: The set $E(a)$ of candidate expansions of a.

Our main insight is that the expansions corresponding to the different meanings of a are included in the set of co-clicked queries for a. We first compute the co-clicked queries for a. Let $D(q)$ denote the set of documents which users clicked when they searched with the query string q as recorded in the query click log. Furthermore, let $Q(d)$ denote the set of queries for which users clicked on web document d as recorded in the query click log. We first compute from the query click log the set of documents $D(a)$ clicked for acronym a. Then, for each document $d \in D(a)$, we collect the set of queries $Q(d)$ for which d was clicked. We thus obtain the set $\cup_{d \in D(a)} Q(d)$ of co-clicked queries for a.

Not all co-clicked queries are valid expansions for a. To identify the valid expansions in $\cup_{d \in D(a)} Q(d)$, we propose an *acronym-expansion checking function*.

Acronym-Expansion Checking Function: We present a function that checks whether a given string can be considered as the expanded string of a given acronym.

Definition 2. (**Acronym-Expansion Checking Function**) Given a string q and an acronym a, the checking function $IsExp : q \times a \rightarrow \{true, false\}$ returns $true$ if q is a valid expansion of a and $false$ otherwise.

For example, $IsExp$("carnegie mellon university", "cmu") should be true while $IsExp$("cmu football", "cmu") should be false.

It is difficult to develop a set of exact rules for matching acronym letters in a query. For example, a common rule is that acronym letters should be the initial letters of the words in the expansion. However, this rule does not always hold: "Hypertext Transfer Protocol" is expansion for "HTTP". On the other hand, stop words are often skipped when constructing acronyms (e.g. "Master of Business Administration" is expansion for "MBA"). However, this does not always hold (e.g., "League of Legends" is expansion of "LOL").

We approach the problem by using heuristics based on dynamic programming. We assign weights to the words and letters of the query string, and modify the longest common subsequence algorithm to find the subsequence with highest score [17]. Pseudo-code is shown in Algorithm 1 in the Appendix, along with explanations of the checking function.

Expansion Identification from Co-clicked Queries: A co-clicked query $q \in \cup_{d \in D(a)} Q(d)$ is a valid expansion of acronym a, iff $IsExp(q, a) = true$; we refer to it as a candidate expansion of a. We compute the set $E(a) = \{q | q \in \cup_{d \in D(a)} Q(d), IsExp(q, a) = true\}$ of candidate expansions of a by checking each co-clicked query using the acronym-expansion checking function.

3.2 Acronym Expansion Clustering

Input: The set $E(a)$ of candidate expansions of a and the query click log $\mathcal{Q}$.
Output: Grouping $\mathcal{G}(a) = \{G_1, \ldots, G_n\}$ of candidate expansions $E(a)$.

The set $E(a)$ of candidate expansions output by the previous step does not correspond to the distinct meanings. It contains several variants that correspond to the same meaning. For example, for the meaning "Carnegie Mellon University", the variants include "Carnegie Mellon University", "Carnegie Mellon Univ" as well as misspellings like "Carnegie Melon University". They all pass the acronym-expansion checking function. Given the set $E(a)$ of candidate expansions of a, this step clusters them into a set $\mathcal{G}(a) = \{G_1, \ldots, G_n\}$ of groups such that each group has a unique meaning, and no two groups have the same meaning. These groups correspond the desired set $\{M_1, \ldots, M_n\}$ of distinct meanings. We first discuss the distance metrics between the candidate expansions and then the clustering algorithm.

3.2.1 Distance Metric for Candidate Expansions

Candidate expansions that correspond to the same meaning are typically minor spelling variations of each other (e.g., "Carnegie Mellon University" and "Carnegie Melon University") while those that correspond to different meanings are often far in terms of string distance (e.g., "Carnegie Mellon University" and "Central Michigan University"). One obvious approach is to cluster the candidate expansions based on their string distance, say edit distance. However, there are many cases where expansions corresponding to the same meaning have large string distances. For example, expansions like "Mass Inst Tech" and "Massachusetts Institute of Technology" correspond to the same meaning, but their edit distance is high enough to prevent them from being grouped together. On the other hand, expansions like "Manukau Institute of Technology" and "Manipal Institute of Technology" refer to two different meanings but may incorrectly be grouped together due to their low edit distance.

Our key insight is that each document clicked by any of the expansions in $E(a)$ typically corresponds to a single meaning; hence, *the expansions that correspond to the same meaning will click on the same set of documents, whereas expansions corresponding to different meanings will click on different sets of documents.* We design distance metrics to leverage this insight and perform clustering to group the expansions.

Set Distance (Jaccard Distance): One way to measure the distance between two expansions e_i and e_j in $E(a)$ is by the distance between the corresponding sets $D(e_i)$ and $D(e_j)$ of clicked documents. A common way to measure set distance is Jaccard distance: $dist(e_i, e_j) = 1 - \frac{|D(e_i) \cap D(e_j)|}{|D(e_i) \cup D(e_j)|}$.

However, Jaccard distance has a serious limitation. Click logs are known to be noisy and contain many clicks that users performed by mistake (referred to as "mis-clicks"). For example, documents associated with "Massachusetts Institute of Technology" get significant number of mis-clicks for the query "Michigan Institute of Technology". Jaccard distance is not robust to such mis-clicks.

Distributional Distance (Jensen-Shannon Divergence) Our main insight is to leverage the frequency of clicks. The frequency of mis-clicks is typically much lower compared with frequency of clicks on documents that are consistent with the meaning of the expansion. We consider the *distribution* of documents clicked for a given query instead of the *set* of documents. We use a distributional distance metric, square root of *Jensen-Shannon divergence*, to evaluate distance between expansions. This metric is much more robust to mis-clicks.

Denote by $F(q, d)$ the frequency with which d is clicked by q. The click distribution $\Omega(q)$ of a query q over all possible documents is $Pr(\Omega(q) = d) = \frac{F(q,d)}{\sum_{d \in D(q)} F(q,d)}$.

Given click distributions defined by click frequencies, the Jensen-Shanon divergence between two expansions e_i and e_j in $E(a)$ is:
$JSD(\Omega(e_i)||\Omega(e_j)) = \frac{1}{2} KL(\Omega(e_i)||\Omega(\overline{e})) + \frac{1}{2} KL(\Omega(e_j)||\Omega(\overline{e}))$,
where $\Omega(\overline{e}) = \frac{1}{2}(\Omega(e_i) + \Omega(e_j))$ and $KL(X||Y)$ is the Kullback-Leibler divergence between two distributions:

$$KL(X||Y) = \sum_i Pr(X(i)) \log \frac{Pr(X(i))}{Pr(Y(i))}.$$

Then, $dist(e_i, e_j) = \sqrt{JSD(\Omega(e_i)||\Omega(e_j))}$.

3.2.2 Clustering of Candidate Expansions

We cluster the candidate expansions in $E(a)$ based on the above distance metric. We use the bottom-up, average-link hierarchical clustering [14, 5].

3.3 Enhancement for Tail Meanings

While the set of co-clicked queries for the acronym a covers the popular meanings of a, it does not cover many of the less popular meanings (referred to as "tail meanings"). Consider the acronym "MIT". "Massachusetts Institute of Technology" is the dominating meaning for "MIT"; the first few pages of results returned by the search engine for the query "MIT" are dominated by that meaning. Tail meanings for that acronym (e.g., "Maharashtra Institute of Technology", "Mahakal Institute of Technology", "Mazandaran Institute of Technology" and so on) are not represented in the top results. As a result, the co-clicked queries for "MIT" will not cover these tail meanings. As shown in Figure 2, the co-clicked queries for "MIT" (i.e., "massachusetts institute of technology", "mit boston" and "mass institute of tech") all correspond to the dominating meaning. Hence, the above approach misses the tail meanings.

We leverage the following insight to address this problem. Since users searching for tail meanings do not find the desired documents when they use only the acronym as a query, they use additional words to disambiguate the query. For example, a user searching for the meaning "Maharashtra Institute of Technology" (which is located in Pune, India) will issue the query "mit pune" while the one searching for the

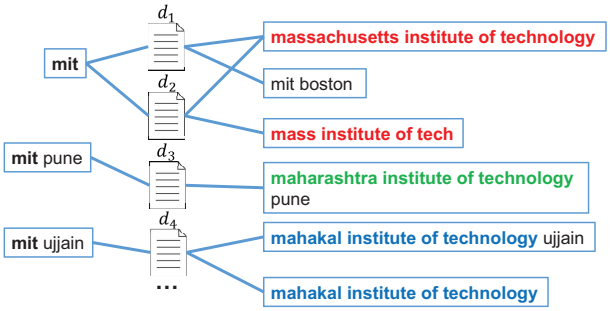

Figure 2: Example illustrating enhancement for tail meanings.

meaning "Mahakal Institute of Technology" (which is located in Ujjain, India) will issue the query "mit ujjain". *Instead of collecting co-clicked queries for the acronym a, we collect co-clicked queries for acronym supersequence queries (ASQ).*

Definition 3. (**Acronym Supersequence Query**) An acronym supersequence query, denoted as $a+s$, for an acronym a is a query in the query click log that contains the string a and an additional sequence of words s either as prefix or as suffix of a.

For example, "mit pune", "mit ujjain" and "mit admission" are ASQs for "mit". Co-clicked queries of ASQs of a contain many more tail meanings of a. As shown in Figure 2, co-clicked of the above ASQs of "mit" cover the tail meanings like "Maharashtra Institute of Technology" and "Mahakal Institute of Technology". We enhance the candidate expansion identification and expansion clustering steps based on the above insight.

Candidate Expansion Identification:
Input: The acronym a and the query click log $\mathcal{Q}$.
Output: The set $E(a)$ of candidate expansions of a.

The goal of this step is to identify the candidate expansions among the co-clicked queries of ASQs of a. However, there is a challenge: unlike in the case of co-clicked queries of a, the candidate expansions may not themselves appear in co-clicked queries of ASQs of a. For example, the co-clicked queries for ASQ "mit pune" does not contain "maharashtra institute of technology". But it contains "maharashtra institute of technology pune". This is because people tend to use acronyms and their expansions interchangeably; so, ASQ $a+s$ may not have co-clicks with e where e is an expansion of a but will have co-clicks with $e+s$. We refer to them as "expansion supersequence queries".

We identify the candidate expansions of a as follows:
1) We first obtain the set $ASQ(a)$ of all ASQs of a. We compute this by scanning the query click log and identifying queries which contain a and a prefix or suffix string. We consider prefix and suffix strings containing zero, one and two words.
2) For each ASQ $a+s \in ASQ(a)$, we collect the set of co-clicked queries $\cup_{d \in D(a+s)} Q(d)$. e is a candidate expansion for a based on ASQ $a+s$ iff (i) the expansion supersequence query has co-clicks with $a+s$, i.e., $e+s \in \cup_{d \in D(a+s)} Q(d)$ and (ii) the acronym-expansion checking function returns true, i.e., $IsExp(e,a) = true$. We formally define the candidate expansion set $E_s(a)$ of a based on ASQ $a+s$:

$$E_s(a) = \{e | e+s \in \cup_{d \in D(a+s)} Q(d) \wedge IsExp(e,a) = true\}$$

3) We obtain the candidate expansion set $E(a)$ of a by unioning the candidate expansion sets based on the ASQs. We formally define the candidate expansion set $E(a)$ of a by $E(a) = \bigcup_{\forall s, a+s \in ASQ(a)} E_s(a)$.

Note that the prefix/suffix can be empty. So, $ASQ(a)$ includes a and hence the above candidate expansion set subsumes the previously defined candidate expansion set. The new candidate expansion set contains strictly more expansions and hence improves coverage.

Acronym Expansion Clustering:
Input: The set $E(a)$ of candidate expansions of a and the query click log $\mathcal{Q}$.
Output: Grouping $\mathcal{G}(a) = \{G_1, \ldots, G_n\}$ of candidate expansions $E(a)$.

The goal is to group the set $E(a)$ of candidate expansions into groups such that each group corresponds to a distinct meaning. The key insight for expansions also holds for expansion supersequence queries: the expansion supersequence queries that correspond to the same meaning will click on the same set of documents, whereas expansion supersequence queries corresponding to different meanings will click on different sets of documents. For example, "massachusetts institute of technology admissions" and "mass inst of tech admissions" will share clicks but "massachusetts institute of technology admissions" and "maharashtra institute of technology admissions" will not. Hence, we can leverage the same general clustering approach based on distributional distance to perform the grouping. However, the same expansions can have multiple corresponding expansion supersequence queries; we need to compute the distance between two expansions by aggregating the distances between the corresponding expansion supersequence queries. There are multiple ways to perform this aggregation; we present two such options:

Distance Aggregation: One option is to compute the distance for each distinct expansion supersequence query (corresponding to a distinct prefix/suffix string) and then aggregate the distances. Let $ASQ(a, e_i, e_j) = \{a+s | a+s \in ASQ(a), e_i + s \in \mathcal{Q}, e_j + s \in \mathcal{Q}\}$ be the subset of ASQ queries for which both $e_i + s$ and $e_j + s$ are valid queries in the query log $\mathcal{Q}$. For each $a+s \in ASQ(a, e_i, e_j)$, denote by $dist_s(e_i, e_j)$ the distance between two candidate expansions e_i and e_j measured over the same supersequence $a+s$, using the distributional distance between queries e_i+s and e_j+s. This can be defined as $dist_s(e_i, e_j) = dist(e_i + s, e_j + s)$. We then aggregate $dist_s(e_i, e_j)$ over all possible $a+s \in ASQ(a, e_i, e_j)$ to obtain the overall distance $dist(e_i, e_j)$ between candidate expansions e_i and e_j. That is $dist(e_i, e_j) = \frac{1}{|ASQ(a,e_i,e_j)|} \sum_{a+s \in ASQ(a,e_i,e_j)} dist_s(e_i, e_j)$.

Click Frequency Aggregation: Another option is to aggregate clicks instead of distance scores. For each pair of candidate expansions, e_i and e_j, we compute the click distribution of expansion e_i by aggregating over all possible expansion supersequence queries in $ASQ(a, e_i, e_j)$. The aggregated click distribution, denoted by $\Omega_{ij}(e_i)$, is:

$$Pr(\Omega_{ij}(e_i) = d) = \frac{\sum_{a+s \in ASQ(a,e_i,e_j)} F(e_i+s, d)}{\sum_{a+s \in ASQ(a,e_i,e_j)} \sum_{d \in D(e_i+s)} F(e_i+s, d)}.$$

We then compute a distributional distance between e_i and e_j based on the aggregated click distribution: $dist(e_i, e_j) = \sqrt{JSD(\Omega_{ij}(e_i) || \Omega_{ij}(e_j))}$.
As we will show later in our experiments, we did not observe noticeable difference between the two aggregation approaches.

We refer to this approach as *Enhanced Acronym Expansion Clustering*; we refer to the approach discussed in Section 3.2 as *Acronym Expansion Clustering*.

3.4 Canonical Expansion, Popularity, Context

Input: Grouping $\mathcal{G}(a) = \{G_1, \ldots, G_n\}$ of candidate expansions $E(a)$ and the query click log $\mathcal{Q}$.
Output: Meanings $\{M_1, \ldots, M_n\}$ with $M_i.e$, $M_i.p$ and $M_i.C$ for each meaning M_i.

The clustering step outputs a set of groups of expansions $\mathcal{G}(a) = \{G_1, \ldots, G_n\}$. These groups correspond to the desired set $\{M_1, \ldots, M_n\}$ of distinct meanings for the acronym a. In this step we compute, for each meaning M_i, canonical expansion $M_i.e$, popularity $M_i.p$, and context words $M_i.C$.
Canonical Expansion: We posit that the canonical expansion of G_i is the most "popular" expansion, because intuitively the canonical acronym expansion should occur more frequently than non-canonical expansions, or expansions with spelling mistakes.

In our click log data, popularity is measured by the number of clicks. If a document d is clicked by acronym a for a total of $F(a, d)$ times, we want to find out how many of those clicks are intended for each expansion $e_k \in G_i$.

Since there is no way for us to know users' real intent, we approximately distribute clicks $F(a, d)$ to each expansion $e_k \in G_i \in \mathcal{G}$ proportionally by the number of clicks between e_k and d, namely, $F(e_k, d)$. The intuition is that if the document d is clicked by a particular expansion e_k a lot, then a significant portion of the clicks $F(a, d)$ should be credited to e_k.

Given a click between a and $d \in D(a)$, the probability that the click is intended for e_k, denoted as $Pr(e_k, d)$, is computed by the total number of clicks between e_k and d, $F(e_k, d)$, divided by the total number of clicks between d and all possible expansions in $\mathcal{G}$. The probability that a click on document $d \in D(a)$ belongs to expansion e_k is thus:

$$Pr(e_k, d) = \frac{F(e_k, d)}{\sum_{G_l \in \mathcal{G}} \sum_{e_j \in G_l} F(e_j, d)}$$

If we only look at acronym a itself (without supersequence tokens), then the probability of an expansion e_k can be computed by aggregating over all possible acronym-document clicks $F(a, d)$:

$$e_k.p = \frac{\sum_{d \in D(a)} F(a, d) Pr(e_k, d)}{\sum_{d \in D(a)} F(a, d)} \quad (1)$$

However, the probability of an expansion should also include cases where the acronym is mentioned in conjunction with supersequence tokens $a + s$, where the meaning of a is intended for that expansion. Conceptually, the meaning probability of a should be counted regardless of whether a is mentioned alone, or with some other tokens. (If we do not account for supersequence queries, on the other hand, then for certain tail expansions discovered via ASQ that have no co-clicks with a, these expansions would get zero-probability, which is intuitively incorrect).

We define $Pr_s(e_k, d)$ for each $a + s \in ASQ(a)$ by:

$$Pr_s(e_k, d) = \frac{F(e_k + s, d)}{\sum_{G_l \in \mathcal{G}} \sum_{e_j \in G_l} F(e_j + s, d)}$$

Then the probability of an expansion e_k, denoted as $e_k.p$, can be computed by aggregating clicks credited to e_k, divided by the total number of query clicks containing a:

$$e_k.p = \frac{\sum_{a+s \in ASQ(a)} \sum_{d \in D(a+s)} F(a + s, d) Pr_s(e_k, d)}{\sum_{a+s \in ASQ(a)} \sum_{d \in D(a+s)} F(a + s, d)} \quad (2)$$

As in our previous notations, $a \in ASQ(a)$ because s can be empty. Notice, Equation (2) is a generalization of Equation (1). If supersequence queries are not considered, then it essentially becomes Equation (1).

With that, the canonical expansion of G_i is simply the expansion with the highest probability:

$$M_i.e = \operatorname*{argmax}_{e_k \in G_i} e_k.p$$

where $e_k.p$ is computed in Equation (2).
Meaning Group Popularity: The second output is the probability of each meaning group $M_i.p$. Since we have already computed $e_k.p$ in Equation (2), we can simply aggregate for all $e_k \in G_i$ to obtain $M_i.p$:

$$M_i.p = \sum_{e_k \in G_i} e_k.p$$

Context Words: Let $D(G_i)$ be the set of documents clicked by expansions in group G_i for meaning M_i. We assign context words to each meaning M_i:
$M_i.C = \{w \mid w \text{ is a word in } q, q \in \cup_{d \in D(G_i)} Q(d)\}$. We assign to each word in $M_i.C$ a probability score, which measures how strongly the word indicates the meaning. Let $F(w, G_i)$ be the frequency of a word w in group G_i, given by $F(w, G_i) = \sum_{w \in q, q \in \cup_{d \in D(G_i)} Q(d), d \in D(G_i)} F(q, d)$. We compute the probability that a word w is indicative for M_i by:

$$Pr(w|M_i) = \frac{F(w, G_i)}{\sum_{w' \in M_i.C} F(w', G_i)}.$$

4. ONLINE MEANING PREDICTION

Acronym queries are very common in Web search. Often users provide some context, in addition to the acronym, which can be one or more other words. In such cases, the user experience can be greatly enhanced if the correct meaning of the acronym can be predicted by the search engine. Then the search results for the query will be more relevant and focused.

We propose a solution to such prediction task: given an acronym and a context, predict the correct meaning of the acronym. We assume that we are given a set of meanings $\{M_1, M_2, \ldots, M_n\}$ for an acronym, found using our offline approach from Section 3. Each meaning M_i is also associated with a set of context words as described in Section 3.4. To predict the correct meaning of an acronym, given context words, we leverage (1) the popularity of each meaning, and (2) the relatedness between each meaning and the context words. In case there are no context words given, to predict the meaning of an acronym, we use only the popularity scores of its meanings, $M_i.p$, as computed in Section 3.4.

For each meaning M_i and context word w, we compute the probability $Pr(M_i|w)$. Applying Bayes' theorem we obtain:

$$Pr(M_i|w) = \frac{Pr(w|M_i) Pr(M_i)}{Pr(w)}$$

Here, $Pr(w|M_i)$ is computed as in Section 3.4, and $Pr(M_i)$ is given by $M_i.p$. $Pr(w)$ can be any dictionary-based probability of the word w. Note that since $Pr(w)$ is the same for all meanings M_i, it is sufficient to consider only the numerator in the above formula.

We compute $Pr(M_i|w)$ for each meaning of the acronym. To predict which is the correct meaning, we consider the meaning with the highest probability score.

This prediction task can be further generalized in case we have more than one context word in addition to the acronym:

$$Pr(M_i|w_1,\ldots,w_k) = \frac{Pr(w_1,\ldots,w_k|M_i)Pr(M_i)}{Pr(w_1,\ldots,w_k)}$$

where $Pr(w_1,\ldots,w_k|M_i) = \prod_j Pr(w_j|M_i)$ by considering that all words are independent and identically distributed. $Pr(w_1,\ldots,w_k)$ is computed analogously.

5. EXPERIMENTS

We present an experimental evaluation of the solution proposed in the paper. The goals of the study are:

• To study the effectiveness of our clustering algorithm and our enhanced clustering algorithm in discovering expansions and grouping expansions into meanings;

• To compare the above algorithms with clusterings based on edit distance and Jaccard distance in terms of cluster quality;

• To compare the meanings available in Wikipedia with those discovered by our clustering algorithm;

• To study the effectiveness of online meaning prediction algorithm for acronym+context queries.

5.1 Experimental Setup

5.1.1 Data

To evaluate the proposed approach of clustering acronym expansions, we randomly sampled 100 pages from Wikipedia disambiguation pages. We filtered out pages which do not represent acronyms (e.g., the disambiguation page about "Jim Gray"), and pages of unambiguous acronyms with a single meaning. This resulted in a set of 64 acronyms, on which we perform our experiments. To collect candidate expansions, we use the query log from Bing from 2010 and 2011[2].

5.1.2 Compared Methods

We compare the following methods, all based on standard bottom-up hierarchical clustering with average link and threshold 0.8:

• **Edit Distance based Clustering (EDC):** Clustering, which uses edit distance between candidate expansions.

• **Jaccard Distance based Clustering (JDC):** Clustering, which uses Jaccard distance between expansions.

• **Acronym Expansion Clustering (AEC):** Our approach from Section 3.2, which uses only acronym queries to collect candidate expansions (no supersequence queries). Square root of Jensen-Shannon divergence is used for distance between expansions.

• **Enhanced Acronym Expansion Clustering (EAEC):** Our enhanced approach from Section 3.3, which uses supersequence queries to collect candidate expansions, square root of Jensen-Shannon divergence as distance, and click frequency based aggregation.

5.1.3 Ground Truth Meanings of Acronyms

We use two sets of ground truth meanings for acronyms:

• **Wikipedia Meanings:** Meanings listed in the Wikipedia disambiguation pages of the acronyms.

[2]Note that due to proprietary and privacy concerns we cannot share more details about the query log.

By analyzing the results from our clustering approach on a set of acronyms, we noticed that the meanings discovered from the query log, and the meanings listed in Wikipedia for the same acronyms, are very different. That is why, in addition to the Wikipedia meanings, we compile a second set of ground truth meanings:

• **Golden Standard Meanings:** We consider for each acronym all queries from the click log, which (1) are legitimate w.r.t the acronym-expansion check (see Section 3.1), and (2) have co-clicks with acronym or acronym supersequence queries. Then, we manually label the different meanings/expansions of the acronym. In the Golden Standard we have one or more different expansions for each distinct meaning. For example, we can have two expansions: "central michigan university" and "central mich univ" referring to the same meaning.

Note that some acronym expansions are not meaningful, even though they are legitimate with respect to our acronym-expansion check. One such example is the expansion "computer processor upgrade" for "CPU", since we speculate people never mean "computer processor upgrade" when they mention "CPU". In our Golden Standard set we do not consider such expansions. Since not all legitimate expansions are included in the ground truth, not all groups from our clustering approaches have specific ground truth meaning. We measured the number of the groups which have ground truth meanings, divided by the total number of groups in the clustering, and on average, 82% of the groups have ground truth meanings.

5.1.4 Evaluation Measures

We use the following measures: *Purity, Normalized Mutual Information (NMI),* and *Recall*.

Our algorithms output a set of groups $\mathcal{G}(a) = \{G_1,\ldots,G_n\}$ which maps to golden standard meanings $\mathcal{M} = \{M_1,\ldots,M_k\}$ for a given acronym a. We map each group of expansions G_i to one or more meanings from $\mathcal{M}$ using the top-5 expansions from G_i, ranked by their probabilities. A group can be mapped to one or more meanings, and multiple groups can be mapped to one meaning. For example, a group with expansions, "carnegie mellon university" and "central michigan university", is mapped to 2 distinct meanings, while a group with expansions, "central michigan university" and "central mich univ", is mapped only to one meaning. By a *group-meaning mapping* we consider a meaning, to which a group can be mapped.

• **Purity:** The Purity measures the accuracy of the group-meaning mappings. Let $N = \sum_{i=1}^{n}\sum_{j=1}^{k}|G_i \cap M_j|$ be the total number of group-meaning mappings. The Purity measure counts the number of groups, which are mapped to some meaning, and divides this number by N:

$$\text{Purity}(\mathcal{G},\mathcal{M}) = \frac{1}{N}\sum_{i=1}^{n}\max_j |G_i \cap M_j|$$

Good clusterings have Purity close to 1, and bad ones – close to 0. Since high Purity is easy to achieve, by simply having all expansions in separate groups, in addition to Purity, we use Normalized Mutual Information, described below.

• **Normalized Mutual Information (NMI):**

$$\text{NMI}(\mathcal{G},\mathcal{M}) = \frac{\text{MI}(\mathcal{G};\mathcal{M})}{[\text{H}(\mathcal{G}) + \text{H}(\mathcal{M})]/2}$$

Here $\text{MI}(\mathcal{G};\mathcal{M}) = \sum_i\sum_j Pr(G_i \cap M_j)\log\frac{Pr(G_i \cap M_j)}{Pr(G_i)Pr(M_j)}$ is the Mutual Information of the clusters and the Golden Stan-

	Purity	NMI
EDC	0.956	0.862
JDC	0.999	0.918
AEC	0.998	0.999

Table 1: Evaluation for EDC, JDC, and AEC.

	Purity	NMI	Recall
AEC	0.993	0.994	0.801
EAEC	0.996	0.995	0.996

Table 2: Evaluation for AEC and EAEC.

dard meanings. $\mathrm{H}(\mathcal{G}) = -\sum_i Pr(G_i) \log Pr(G_i)$ is the entropy of the clusters, and $\mathrm{H}(\mathcal{M})$, computed analogically, is the meanings entropy. NMI is a number between 0 and 1, where good clusterings have NMI close to 1, and bad ones – close to 0. Since, the cluster entropy $\mathrm{H}(\mathcal{G})$ increases with the number of groups, clusterings with many groups have low NMI scores. This is why NMI considers the trade-off between the quality of the clusters and their total number.

• **Recall:** We compute Recall only with respect to our ground truth meanings. In practice, it is very difficult to find all possible meanings for a given acronym. The Recall w.r.t our Golden Standard, is the number of meanings, which are found in the clustering, divided by the total number of meanings in the Golden Standard. Furthermore, if a group is mapped to multiple meanings, we consider only one of them, assuming only a single meaning per group.

5.2 Acronym Meaning Discovery Results

We first compare the effectiveness of the Edit Distance based (EDC) and Jaccard Distance based (JDC) clusterings with our Acronym Expansion clustering (AEC) using a subset of 20 acronyms. The results are presented in Table 1.

We notice that both methods, EDC and JDC, have lower cluster quality than AEC, especially in terms of NMI. EDC often fails to cluster expansions by semantic meaning, since expansions with the same meaning can have very large string distance. In such cases, they are incorrectly assigned to different groups. For example "central michigan university", "central mich univ", and "central mi univ" belong to different groups from the EDC clustering. In contrast, the AEC method groups these expansions in a single group, since it does not rely on string distance.

The JDC method has better quality than EDC, but it is still inferior to the AEC method. Since JDC uses set distance between expansions, it is very difficult to find a threshold for similarity. If the threshold is high, then distinct meanings can be easily grouped together due to mis-clicks; if the threshold is low, then identical meanings are not grouped together because sometimes there are not enough clicks. In contrast, the AEC method addresses these problems by using distributional distance metric over click frequencies.

5.2.1 Comparison between AEC and EAEC

To discuss the effectiveness of our clustering approaches, AEC and EAEC, we first present some intuitive examples. In Table 3 we show the top-5 meanings, their probabilities, and a few context words for "CMU", "MBA", and "RISC", using our enhanced clustering EAEC. Each of the three acronyms has one or two dominant meanings, with very high probabilities, and other meanings with much lower probabilities. We also notice that the context words of each meaning are very descriptive. For example, "concrete masonry unit" has context words "cinder", "cement", "construction", etc.

N_W	N_{GS}	$\lvert N_W \cap N_{GS} \rvert$	$\frac{\lvert N_W \cap N_{GS} \rvert}{N_W}$	$\frac{\lvert N_W \cap N_{GS} \rvert}{N_{GS}}$
15.859	15.781	4.922	0.351	0.339

Table 4: Number of meanings in Wikipedia (N_W), in the Golden Standard (N_{GS}), and shared ($N_W \cap N_{GS}$).

We systematically compare AEC and EAEC in Table 2, using the Golden Standard meanings and our complete data set. We first notice that the quality of the two methods is very good, in terms of both, Purity and NMI. The methods succeed in grouping together expansions referring to the same semantic meaning, even if they have large string distances. Furthermore, due to the choice of distributional distance metric for the clustering, the mis-clicks do not influence the clustering.

From the results we also notice that the Recall improves significantly for the enhanced clustering (EAEC), compared to AEC. Using EAEC with supersequence queries, we discover significantly more new tail meanings. Furthermore, since the enhanced clustering EAEC achieves 0.996 Recall w.r.t the Golden Standard, we succeed to output 99% of the meanings in the Golden Standard to the end users.

In addition to using click frequency based aggregation, we also tried out distance based aggregation as described in Section 3. Distance based aggregation yields a purity of 0.996, NMI of 0.997 and recall of 0.994, which is similar to that of click frequency based aggregation.

5.2.2 Wikipedia vs. Golden Standard Meanings

An important result from our study is the comparison between the Wikipedia meanings and the Golden Standard meanings. First, in Table 5 we present the meanings of "CMU", "RISC" and "MBA" which belong only to Wikipedia, only to the Golden Standard, or are shared by both. We notice, that the amount of shared meanings is relatively small, and that both sets, Wikipedia and Golden Standard meanings have meanings not covered by the other.

In Table 4 we present the average number of meanings in Wikipedia, in the Golden Standard, and their shared meanings. In Figure 3 all acronyms in our data set are presented with their meaning counts from the three meaning sets.

From these results we see that only 35% from the meanings in Wikipedia are found using our clustering aproach. This means that a lot of meanings listed in Wikipedia are typically not used in their abbreviated form. It can be because they are extremely tail meanings, or because they are mostly encyclopedic or domain-specific, e.g., medical or mathematical terms. In such cases there is not enough evidence in the query log that these acronyms refer to the corresponding meanings. For example, for "MBA" our method did not find the meaning "main belt asteroid" (see Table 5), and for "LRA" our method did not find the meaning "leukotriene receptor antagonist".

On the other hand, our acronym mining approach discovers many meanings, currently not present in Wikipedia: in the Golden Standard set only 34% of the meanings belong to Wikipedia. More importantly, we discover acronym meanings which are frequently used by common users. Typically, we discover many company names, associations, universities, events, etc. which are used together with their abbreviated form. By finding such new, valid and widely used acronym

	Meaning	Probability	Context Words
CMU	central michigan university	0.615	michigan, university, athletics, campus, edu, football, chippewas
	carnegie mellon university	0.312	mellon, carnegie, pittsburgh, university, library, computer, engineering
	concrete masonry unit	0.045	block, concrete, cmu, masonry, cinder, cement, construction
	central methodist university	0.017	methodist, university, fayette, central, missouri, baseball
	canton municipal utilities	0.004	canton, court, municipal, docket, case, clerk, records
RISC	reduced instruction set computer	0.737	risc, instruction, set, computer, processor, architecture
	rice insurance services company	0.143	insurance, rice, risceo, services, real, estate
	rna induced silencing complex	0.046	complex, rna, silencing, gene, protein
	reinventing schools coalition	0.037	schools, coalition, inventing, alaska
	recovery industry services company	0.022	recovery, certified, specialist, matrix, educational
MBA	master of business administration	0.868	mba, business, gmat, administration, harvard, programs, degree
	mortgage bankers association	0.069	mortgage, bank, implode, amerisave, bankers, rates
	montgomery bell academy	0.022	bell, montgomery, academy, nashville, mba, school, edu
	metropolitan builders association	0.015	builders, homes, association, wisconsin, milwaukee
	military benefit association	0.006	military, armed, association, benefits, insurance, veterans

Table 3: Top-5 meanings for CMU, RISC, and MBA, ranked by probability, and some of their context words.

	Only Wikipedia Meanings	Only Golden Standard Meanings	Shared Meanings
CMU	caribbean medical university	central methodist university	carnegie mellon university
	chiang mai university	canton municipal utilities	central michigan university
	california miramar university	centrul medical unirea	canadian mennonite university
	colorado mesa university	case management unit	concrete masonry unit
	coffman memorial union	central mindanao university	couverture maladie universelle
	college music update	central missouri university	
	complete music update		
	communication management unit		
RISC	rural infrastructure service commons	rice insurance services company	reduced instruction set computing
	research institute for symbolic computation	reinventing schools coalition	rna induced silencing complex
		recovery industry services company	
		rhode island statewide coalition	
MBA	maldives basketball association	metropolitan builders association	master of business administration
	marine biological association	military benefit association	mortgage bankers association
	metropolitan basketball association	master builders association	montgomery bell academy
	media bloggers association	mississippi basketball association	mountain bothy association
	milwaukee bar association	mountain bike action	
	monterey bay aquarium	massachusetts bar association	
	macbook air	mariana bracetti academy	
	main belt asteroid	missionary baptist association	
	market basket analysis	morten beyer agnew	
	miss black america	mind body awareness	
	misty's big adventure	memphis business academy	

Table 5: Meanings from Wikipedia and Golden Standard sets.

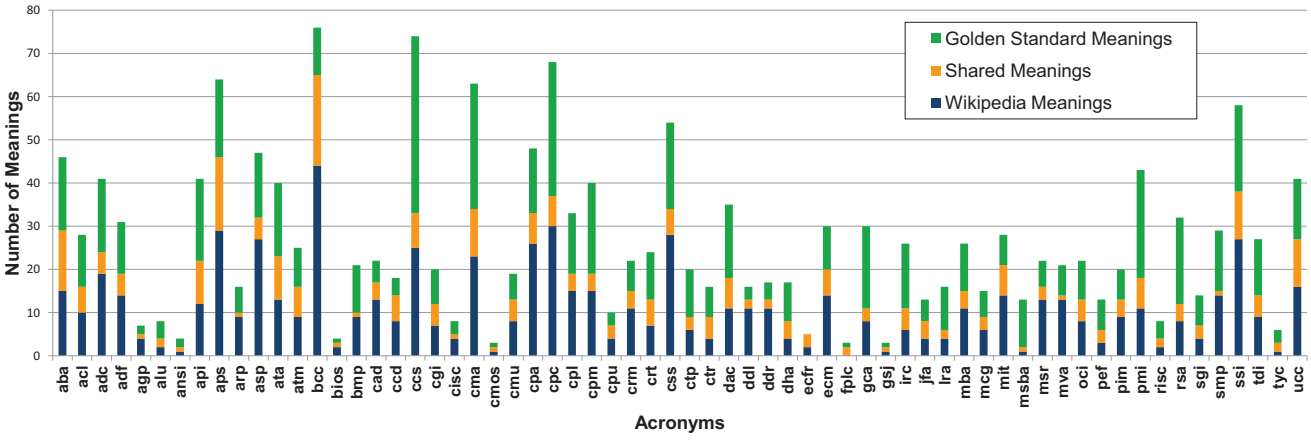

Figure 3: Number of meanings from Wikipedia, from the Golden Standard, and shared between the two sets.

Query	Label
cmu michigan	central michigan university
cmu robotics institute	carnegie mellon university
cmu pittsburgh	carnegie mellon university
cmu fayette missouri	central methodist university

Table 6: Examples for "acronym+context" queries and their labels.

expansions, we can significantly extend the meaning lists in Wikipedia disambiguation pages.

5.3 Online Meaning Prediction Results

Here we consider the online application scenario from Section 4: given acronym+context queries, predict the meaning of the acronym considering the provided context.

We use a set of 7612 acronym+context queries, randomly sampled from the query click log, which refer to the acronyms in our data set. For example, for "CMU" we use queries like "cmu football", "cmu pittsburgh", "cmu robotics institute", etc. Human users label these queries with the meaning they consider most probable, by looking only at the query. For example, "cmu michigan" is labeled by "central michigan university" (see Table 6 for more examples). Queries, for which the additional context does not disambiguate the meaning, are not labeled. For example, "cmu university" can refer to multiple meanings, and hence it is not labeled.

We apply the prediction approach from Section 4 to the labeled queries, without considering their labels. The output is the most probable meaning of the acronym, given the context words in the query.

For each acronym, we compute the number of correctly predicted meanings (by comparing to their labels), divided by the total number of labeled queries for this acronym. The average precision is **0.941**. This means that the assigned context words to each acronym meaning are highly indicative and can be used to predict meanings for online acronym+context queries effectively.

6. RELATED WORKS

While there have been many works and systems available on acronyms, we believe our work has the following unique distinctions compared with the state of art. First, we solve the general acronym meaning discovery problem in a comprehensive way. This is different from other works which either look at domain specific acronyms (e.g., medical domain), or only focus on certain aspects of the problem (e.g., only interested in finding expansions). Second, to the best of our knowledge, this is the first work on acronym expansion and meaning discovery leveraging query click log by exploiting the acronym co-click behaviors. Third, due to the nature of query click analysis, our method is language agnostic. This is different from pattern based discovery in text for instance, where people look for NLP patterns (e.g., "Carnegie Mellon Univeristy, also referred to as CMU, is ...") and therefore the patterns and methods are very language dependent. We now study related works in details.

Wikipedia covers many acronyms and their different meanings through its "disambiguation pages". These pages are manually edited by one or a few editors. First, our experiments show that many meanings are not covered in Wikipedia disambiguation pages; there are almost twice more meanings used in web search queries but not covered in Wikipedia. Second, it does not provide popularity scores. Furthermore, the meanings in Wikipedia are not necessarily the most popular meanings; our experiments show that roughly 65% of the meanings of acronyms on Wikipedia are rarely or never expressed in Web search queries (Section 5). Finally, our work heavily taps into the wisdom of crowds, to discover acronym expansions, understand their meanings and popularity, and mine their corresponding context. Tapping data contributed by millions of end users is a significant and necessary step forward.

Websites such as `acronymfinder.com` list many possible acronym expansions; this is also manually edited. As in Wikipedia, it does not provide popularity scores. Furthermore, it does not provide any context words for most of the acronym expansions. While it does offer a large number of meanings for a large number of acronyms, our study shows that it suffers significantly from the quality problem: (1) many expansions listed are actually near duplicates ("Reduced Instruction Set Computer" and "Reduced Instruction Set Computing" for "RISC"); (2) many expansions are actually meaningless, in the sense people rarely or never use its acronym form to refer to it (e.g., "More Bad Advice" for "MBA").

Recently, there have been a few works on automatic mining of acronym expansions by leveraging Web data [8, 9, 6]. While some aspects are complementary to ours (e.g., in [6] subsequent queries in query sessions are exploited), our study covers many more aspects, including meaning discovery through clustering analysis, popularity computation and context words mining. Our study heavily relies on query click log, and it is not clear how other works can be adapted to support effective clustering, popularity and context words mining without the help of query click log.

Another line of related work is on mining synonyms [3, 4, 15, 11, 1, 2]. Existing studies on synonyms are mostly focused on unambiguous synonyms. Acronym is a special type of synonym, which is highly ambiguous and context dependent. This work can be regarded as a first attempt at addressing the ambiguity problem in synonyms, with a focus on acronyms only.

There have been many works on acronym expansion discovery in vertical domains (mainly in medical), e.g., [12, 10, 16, 13]. These works mainly rely on text analysis to discover acronym expansions, and tend to be optimized for their respective domains. This is different from both the general acronym mining aspect, as well as the query click log analysis angle of this work.

7. CONCLUSION

In this paper, we introduce the problem of finding distinct meanings of each acronym, along with the canonical expansion, popularity score and context words. We present a novel, end-to-end solution to this problem. We describe how web search engines can leverage the mined information for online acronym and acronym+context queries.

Our work can be extended in multiple directions. There are other ambiguous queries besides acronyms like people and place name queries. For example, the query "Jim Gray" can refer to the computer scientist, the sportscaster in addition to many other less famous Jim Grays. Can our techniques be adapted to find all the distinct meanings of such queries? Furthermore, it will also be interesting to look into data sources other than query click log for the mining task.

8. REFERENCES

[1] S. Chaudhuri, V. Ganti, and D. Xin. Exploiting web search to generate synonyms for entities. In *WWW Conference*, 2009.

[2] S. Chaudhuri, V. Ganti, and D. Xin. Mining document collections to facilitate accurate approximate entity matching. *PVLDB*, 2(1), 2009.

[3] T. Cheng, H. W. Lauw, and S. Paparizos. Fuzzy matching of web queries to structured data. In *ICDE*, 2010.

[4] T. Cheng, H. W. Lauw, and S. Paparizos. Entity synonyms for structured web search. *TKDE*, 2011.

[5] D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

[6] A. Jain, S. Cucerzan, and S. Azzam. Acronym-expansion recognition and ranking on the web. In *Information Reuse and Integration*, 2007.

[7] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW*, 2006.

[8] L. S. Larkey, P. Ogilvie, M. A. Price, and B. Tamilio. Acrophile: an automated acronym extractor and server. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 205–214, 2000.

[9] D. Nadeau and P. D. Turney. A supervised learning approach to acronym identification. In *Proceedings of the 18th Canadian Society conference on Advances in Artificial Intelligence*, pages 319–329, 2005.

[10] S. Pakhomov. Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

[11] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *EMNLP*, 2009.

[12] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, and M. Morrell. Automatic extraction of acronym-meaning pairs from medline databases. *Studies in health technology and informatics*, 84(1):371–375, 2001.

[13] J. Pustejovsky, J. Castano, B. Cochran, M. Kotecki, M. Morrell, and A. Rumshisky. Extraction and disambiguation of acronym-meaning pairs in medline. 2004.

[14] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[15] P. D. Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *CoRR*, cs.LG/0212033, 2002.

[16] J. Wren, H. Garner, et al. Heuristics for identification of acronym-definition patterns within text: towards an automated construction of comprehensive acronym-definition dictionaries. *Methods of information in medicine*, 41(5):426–434, 2002.

[17] M. Zahariev. Efficient acronym-expansion matching for automatic acronym acquisition. In *International Conference on Information and Knowledge Engineering*, 2003.

A. APPENDIX: ACRONYM-EXPANSION CHECKING FUNCTION

Let a and e be an acronym and a query, respectively. The acronym-expansion checking function returns *true* if e is an expansion of a, and *false* otherwise. We modify the longest common subsequence algorithm by assigning weights to the words and letters of the query string. We find the subsequence with the highest score and use heuristics to decide if e is an expansion of a as follows.

To increase the chance of matching an acronym letter at the beginning of a word, we assign weights $w_{ns} = 2$ to the initial letters of normal non-stop words, $w_s = 1$ to the initial letters of stop words, and $w_{ni} = 0.1$ to all other letters (i.e. the non-initial letters of all words). The score s of a match is the sum of the scores of the participating letters (see Line 15). Then we check if $s \geq 0.68 \cdot |a| \cdot w_{ns}$, where $|a|$ denotes the number of letters in the acronym and 0.68 is an empirically set threshold parameter.

A further requirement is that all words in the query contain acronym letters. However, as mentioned earlier, often stop words are not considered when acronyms are formed (e.g., "Master of Business Administration" is an expansion for "MBA"). To solve this we use weights for the non-stop query words ($w_{ns} = 2$) and for the stop words ($w_s = 1$). We use another threshold to check if $s \geq 0.8 \cdot T$, where T is defined in Line 2.

If both conditions in Line 16 are satisfied, the acronym-expansion check returns *true*, and otherwise *false*. For example, the score of "master of business administration" for the acronym "MBA" is 6 and both inequalities in Line 16 are satisfied. The query "master of business administration education" is not expansion of "MBA" as the second inequality is not satisfied ($6 \not\geq 0.8 \cdot 9$). Finally, two initial requirements are that the acronym consists of one word only, and that the query contains at least two words.

Algorithm 1 Acronym-Expansion Checking Function

Input: Acronym a; Query e; Weights $\{w_{ns}, w_s, w_{ni}\}$
Output: True if e is expansion of a, false otherwise

1: **function** IsExpansion$(a, e, w_{ns}, w_s, w_{ni})$
2: $T \leftarrow \sum_{t \in e} w(t)$, $w(t) = \begin{cases} w_{ns}, & t \text{ is non-stop word} \\ w_s, & t \text{ is stop word} \end{cases}$
3: Let R be $(|a|+1) \times (|e|+1)$ array of zeros
4: **for** $i \leftarrow 1$ **to** $|a|$ **do**
5: **for** $j \leftarrow 1$ **to** $|e|$ **do**
6: $p \leftarrow \max(R[i-1, j], R[i, j-1])$
7: **if** $a[i-1] = e[j-1]$ **then**
8: $w \leftarrow \begin{cases} w_{ns}, & \text{non-stop word starts at } j-1 \text{ in } e \\ w_s, & \text{stop word starts at } j-1 \text{ in } e \\ w_{ni}, & \text{else} \end{cases}$
9: $R[i, j] \leftarrow \max(p, R[i-1, j-1] + w)$
10: **else**
11: $R[i, j] \leftarrow p$
12: **end if**
13: **end for**
14: **end for**
15: $s \leftarrow R[|a|, |e|]$ ▷ Score of the best match
16: **return** $s \geq 0.68 \cdot |a| \cdot w_{ns}$ **and** $s \geq 0.8 \cdot T$
17: **end function**