Toward a Short Text Classification Framework Based on Background Knowledge Discovery

Isak Taksa
Computer Information Systems, Baruch College, CUNY, New York, NY, USA

Abstract - The ubiquitous, diverse and growing impact of digital living creates a massive amount of short text - a search query, a twit or a caption. Short text frequently presents itself as an arbitrary combination of semantically unconnected words. Using machine learning to classify the corpora of such texts is a challenging task. A large body of research exists in this area, but in this paper we will focus on Background Knowledge (BK) and its role in machine learning for short-text and non-topical classification. More specifically, we present an effort to create a short text classification framework based on Background Knowledge. We propose novel Information Retrieval techniques to construct BK and demonstrate the advantages of Automatic Query Expansion (AQE) vs. basic search. We discuss other results of this research and its implications on the advancement of short text classification.

Keywords: Machine learning, short text, classification, background knowledge, information gain

1 Introduction

Text classification in the framework of machine learning is an active area of research, encompassing a variety of learning algorithms [20], classification systems [19] and data representations [16]. This paper examines the classification of search queries, which is one example of text classification that is particularly complex and challenging. Typically, search queries are short, reveal very few features per single query and are therefore a weak source for traditional machine learning [6].

We examine the issues of non-hierarchical [17] classification and investigate a method that combines limited manual labeling, computational linguistics and information retrieval to classify a large collection of search queries. We discuss classification proficiency of the proposed method on a large search engine query log, and the effect of the variations of this method on the quality and efficiency of short-text classification.

We start with a search engine query log which is viewed as a set of textual data on which we perform classification [13]. Observed in this way each query in a log can be seen as a document that is to be classified according to some pre-defined set of labels, or classes. Viewing the initial log with the search queries as a document corpus \( D = \{d_1, d_2, \ldots, d_m\} \), we create a set of classes that indicate a personal demographic characteristic of the searcher, \( C = \{c_1, c_2, \ldots, c_p, \ldots, c_m\} \). Using Web searches our approach retrieves a set of background knowledge to learn additional features that are indicative of the classes, \( C \). This allows for the categorization of the queries. The approach consists of the following five steps:

I) Select (from print and online media) a short set of manually chosen terms \( T_{init} = \{t_1, t_2, \ldots, t_j, \ldots, t_m\} \), consisting of terms \( t_j \) that are known a priori to be descriptive of a particular class \( c_j \);

II) Use this initial set \( T_{init} \) to classify a small subset of (search queries) set \( D \) thereby creating an initial set of classified queries \( Q_{init} = \{q_1, q_2, \ldots, q_j, \ldots, q_l\} \);

III) Consists of sub-step A and sub-steps B

Sub-step A – Automatic query expansion (AQE) step. Submit these queries \( q_j \) to a commercial search engine and use the snippets of the returned search results to build an initial set of expanded classified queries \( EQ_{init} = \{eq_1, eq_2, \ldots, eq_j, \ldots, eq_l\} \);

Sub-step B – Submit these queries \( eq_j \) to a commercial search engine and use the returned search results (\( n \) for each query \( eq_j \)) to build a temporary corpus of background knowledge \( EB_{temp} = \{eb_1, eb_2, \ldots, eb_j, \ldots, eb_l\} \);

IV) Use an algorithm to select from \( EB_{temp} \) more class related terms \( T \);

V) Use this newly created set \( T \) to classify more documents (search queries) in corpus \( D \) thereby adding more classified queries to set \( Q \).

While steps I and II are executed only once, steps III through V are repeated continuously until the classification process is terminated.

We focus on validating our approach to the classification of a set of short documents, namely search queries. This approach uses a combination of techniques: we first look at developing several methods to obtain relevant background knowledge for a set of web queries; then we build the background knowledge to acquire ranked terms for improved information retrieval; we then investigate the impact of the new terms’ selection algorithms on the effectiveness of the classification process.
2 Background

The Text Classification problem has been studied extensively by machine learning researchers over the last decade. We can define the categorization as follows: Given a set of documents $D$ and a set of $m$ classes (or labels) $C$, define a function $F$ that will assign a value from the set of $C$ to each document in $D$. For example, $D$ might consist of the set of all Classified Advertisements, and $C$ could be the set of types of classified advertisements (automobile, home furnishing, help wanted, etc.) Although the text classification problem can be defined easily, in practice there is often not enough information to find the function $F$.

A text document can be expressed as a feature-value vector, where the features correspond to particular words (or phrases), and the value corresponds to the presence/absence of the word (or some weight that corresponds to the word). When documents are expressed in this fashion the machine learning community can apply well-known algorithms to this problem [15]. The straightforward approach deals with text classification problems as supervised learning problems. In this case the human expert simply has to label a set of examples with appropriate classes. This set of labeled examples is called the training set, which we will refer to as the set $T$. Once a training corpus of correctly labeled documents is available, there are a variety of techniques that can be used to create a set of rules or a model of the data that will allow future documents to be classified correctly. The techniques can be optimized and studied independently of the domains and specific problems that they will be used to address. A plethora of different learning algorithms including Bayesian classifiers [7], nearest neighbor [23], and support vector machines [1], have been applied to many different representations of textual documents, successfully allowing for the classification of documents in varied domains.

Topical text categorization problems, such as the example given above, that have a sufficient number of training examples, are now well-understood by machine learning researchers, however, non-standard problems have been the focus of more recent research. A common problem when using machine learning for text classification is dealing with an insufficient number of training examples to correctly classify instances with unknown classes. If there are too few examples, machine learning algorithms often cannot represent the classes properly, and therefore have a high error rate when attempting to classify new examples.

There are a number of approaches that may be taken to aid in the creation of more accurate classifiers. Researchers have noted that although it is often the case that there are very few labeled examples, there are often many unlabeled examples readily available [4]. An approach that has been taken by a number of researchers has been to choose, in some way, a small number of additional training examples that should be hand-labeled in order to add particular examples to the labeled training set that will improve learning. These hand labeled examples then become part of the training corpus. In this way fewer examples must be given to an expert to be labeled than if the examples were simply randomly sampled.

Other approaches have been taken in these hard-to-classify domains. There have been studies on the incorporation of domain knowledge by the selection and creation of cross-referencing query [21], and domain documents [3], or reweighing of features using related information such as ontologies [9] or user feedback [5]. Domain knowledge has also been incorporated into text classifiers by modification of the classifiers to include prior results [24]. There has also been work done using query-expansion type techniques to incorporate additional knowledge into text classifiers [22] and query formulation techniques using terms found in previously retrieved documents [14].

Often, with short text classification problems, there are related textual documents that are not examples that can be classified. We term this set of related documents research background knowledge and use it to aid a short-text classifier. Background knowledge has been previously used [12] to improve classification of unknown instances. These sets of background text are not of the same length and form as the training and unlabeled examples, but can be used to find common co-occurrences of terms, as well as terms that are indicative of specific classes.

Short text classification is a challenging type of classification because very little information (i.e. words) is known for each example that is to be classified. Researchers have recognized that since short text examples tend to share few terms, it is particularly difficult to classify new instances and common comparisons between texts often yield no useful results. Simply comparing a training set to unknown examples using traditional methods such as cosine similarities can therefore be useless. An example of short text classification that is receiving interest lately is the query classification [21].

3 Methodology

Our approach in this research is different from the traditional machine learning approaches described above. Instead of actually incorporating the background knowledge set into the learning algorithm, we use background knowledge for the purpose of finding previously unknown class related terms. As described earlier, we begin with only a small set of manually selected class related terms (or phrases). These terms are used to label a small set of documents – search queries extracted from a large Excite query log collected in the morning and afternoon hours of one day, and contains close to 2.5 million queries. This small set of labeled documents is then used as search queries to retrieve a much larger set of longer, related documents. We analyze the larger
set of related documents to learn additional class related terms for the classification task.

3.1 Bootstrapping from known class-related terms

To create the set of classes we used Levinson’s Life Structure Theory. After studying a group of men and women Levinson introduced his theory [10] as consisting of equilibrium/disequilibrium periods during which a man builds/questions his life structure. At the center of his theory is the life structure, the underlying pattern of an individual’s life at any particular time. For our classes we use Levinson’s seasonal cycles – C (Early adulthood, Adult world, Settle down), Middle adulthood), Culmination, Late adulthood) or simpler C = {EA, AW, SD, MA, CL, LA} which corresponds to the following age brackets A {17-22, 23-33, 34-45, 46-55, 56-65, 66+}.

We then acquired terms that are indicative of each of these classes. In particular, we obtained the terms (words or phrases) from well-known printed publications (Seventeen, Parenting, Family Circle, American Association of Retired Persons Magazine, etc.) and popular social media sites (Facebook, Linkedin, TopMommyBlogs, etc.). For each of the classes in set C, we manually (and arbitrarily) selected a list of words and phrases that are indicative of each of these classes, e.g. \( T_{init} = \{ EA(\text{Cliff notes}), AW(\text{Wedding}), SD(\text{Investments}), MA(\text{Eldercare}), CL(\text{Inheritance}), LA(\text{Pain}) \} \).

We restricted search results to documents written in the English language. Google returned the top results of the search on the classified query that were downloaded and stored. We saved the textual sections of the pages that were downloaded, and each one became a document, classified according to the class of the query that generated it. We limited our search results to the top 10 results returned for each query since users are generally satisfied if the desired page is found within the top 10 results [8].

After downloading search results we had a set of ten text documents for each one of our queries. These were then used as a corpus for analysis. This method allows for the retrieval of documents that are class related, but are much longer than the original queries. The queries from the search log are an average length of ~3 words, whereas the new documents that were downloaded had an average length of several thousand words, and hence more could be learned from them.

3.3 Finding new class-related terms

Each page that was returned by Google was labeled with the class category of the query that produced it. This set of pages can be looked at as a new and different document training corpus with known labels. The training set \( T \) consists of the returned search pages, and the classes \( C \) are the classes that were used to label the original small set of hand-labeled queries. However, the properties of this training corpus are markedly different than the original query training set. Essentially, this newly created training corpus does not consist of short-text examples. As opposed to our original data set, where examples were queries only a few words long, this larger returned corpus contains entries that are web-page length. Hence there is much more generalization that we can draw from the words in this larger returned document corpus.

What is especially interesting is the new, larger, document corpus vocabulary. A serious disadvantage of short-text corpora is that they do not contain a rich enough vocabulary to facilitate learning, however, with a longer document corpus we can learn much about the domain from the set of words that are in it. In essence, our method of page retrieval allows us to swap a short-text corpus for one with longer entries from which we can learn.

Our approach studies the set of terms that compose the returned document corpus to find those particular terms that are related to our classification problem. We began by using the information gain (IG) criterion to rank all terms in the corpus; no stemming was used to facilitate query creation later. For a supervised text classification task, each term that is present in the training corpus can be seen as a feature that can be used individually for classification. For example, suppose that the term \( \text{investment} \) occurs in the training corpus. We can partition the training corpus into two disjoint subsets, one of which contains the word \( \text{investment} \), and one of which does not. Given the training set of classified examples, \( T \), we can partition it by the presence or absence of each term, \( t \) that exists in these examples. We can then determine how closely related this term is to the classification task.

To do this, we borrow a concept from Information Theory, called information gain, which has been used by
machine learning researchers for the purposes of classification (Quinlan, 1986). Given a probability distribution \( P = (p_1, p_2, \ldots, p_n) \) then the information conveyed by this distribution, also called the entropy of \( P \), is:

\[
\text{entropy}(P) = -(p_1 \log(p_1) + p_2 \log(p_2) + \ldots + p_n \log(p_n)) \quad (1)
\]

Essentially, this measure is a measure of the randomness of the distribution. High entropy signifies that the distribution is random, whereas low entropy signifies that there is some pattern in the data. In the field of information theory, the entropy is a measure of how many bits it takes to transmit a message with the probability distribution \( P \). If we wish to discover the entropy of a training set \( T \), then the probability distribution \( P \) is simply the set of probabilities that a training example fits into any of the classes of set \( C \). From these training set probabilities we can compute entropy \( (T) \).

Each term \( t \) gives a partition of the training set \( T, \{T_0, T_1\} \), where \( T_0 \) consists of those training examples that contain the term \( t \), and \( T_1 \) consists of those training examples that do not contain the term \( t \). For each of these subsets, we can compute individual entropies, and the summation of those entropies, weighted by the probability distribution gives us the information needed to identify the class of a training example after the partition is done. The information gain (IG) for a term \( t \) tells us how much information is gained by partitioning the training set \( T \) on the term \( t \). It is defined as the subtraction:

\[
IG(t) = \text{entropy}(T) - \left( \frac{|T_0|}{|T|} \times \text{entropy}(T_0) + \frac{|T_1|}{|T|} \times \text{entropy}(T_1) \right) \quad (2)
\]

Terms with high information gain create partitions of the original training set that overall have lower entropy, and therefore are reflective of the particular classification scheme.

The computation of the IG value for each of these terms allows us to learn important features in this background corpus. However, our challenge was to determine which of these features best reflected each class. To discover which terms give us information about particular classes, we sorted all terms in the corpus in descending order based upon the IG value. We labeled each of the terms with the class whose training examples most reflected this term, i.e. whose training examples actually most often contained that term.

We then chose the top terms for each of the classes. At this point we selected a list of fifty terms (per class) to classify queries that were not classified before. contain no terms that can be deemed class related, or may contain terms that fit two or more classes. It would be impossible to classify these types of queries.

The other important factor that affects the match of queries in the log and terms in the background set, is the fact that we are using today’s Web collection and search engine to produce the background set, but the query log was collected several years ago. The Web collection grew exponentially; search engines are fine-tuned to return results that reflect contemporary culture and language [2]. It is important to note that some of the text documents did not contain the terms that were associated with their class. We are not concerned with this fact, however, because we are simply looking for good indicative terms that are related to particular classes.

### 4 Discussion

Four parameters describe every scenario: \( S_{\text{eval.}} = \{\text{Number of queries to retrieve background knowledge (NBK), Query selection process (QSP), Number of top classification terms (NCT), and Terms selection process (TSP)}\} \).

- **NBK** represents the number of queries that are selected from the queries classified in the prior iteration of our algorithm. These queries are submitted to Google to retrieve “background knowledge”. We use a variable number of queries to examine whether increasing the size of the retrieved background knowledge would generate better quality classification terms.
- **QSP** specifies the process of selecting new search queries from amongst the newly classified queries. The selection criterion is the query length.
- **NCT** represents the number of new terms that will be used in the next classification iteration. After calculating the Information Gain (IG) for every term in background knowledge, we sort the list in descending order of IG. We use various size lists (between 30 and 100 terms).
- **TSP** specifies the selection of new classification terms from the sorted list produced by IG calculations. The selection criterion is the term frequency in the Google collection. In some scenarios we use “light” terms (low term frequency) while in others we use “heavy” terms (high term frequency).

#### 4.1 Appraising classification results

- To bootstrap all the scenarios we used a manually selected set consisting of 60 terms (10 per each class). Even though this set is negligible in size, it allowed for classification of over 8% of the query log.
- Increasing the number of queries above five doesn’t produce significant classification improvements.
- Longer queries produce better background knowledge for some of the classes.
- “Heavy” queries produce more noise (cross-class classification).
- “Light” queries produce IG lists with smaller entropy on top of the list.
Furthermore, as we reflect on the nature of the data, there are several objective factors that make this classification task a difficult one. First, according to a topical study of the same log, a large number of queries are intra-class in nature (e.g. 20.3% People and places, 7.5% Sex and pornography, 6.8% Non-English or unknown) [18], and therefore are not easily classifiable according to our original set of classes, $C$. In particular, many of the queries in the Excite log may

5 Conclusion

Starting with a small, manually selected set of terms, we develop and present an approach that classifies a set of web queries. This text classification task is difficult for three reasons: the dataset does not contain many labeled examples, the text examples are extremely short, and classification is non-topical by characteristics of the users. By iteratively applying our approach and improving performance of the ranking algorithm we are able to classify many queries in a large query log. However, analysis of several distinct background discovery scenarios did not produce a clear winner.

An essential area of further research is how to evaluate the efficiency of the classification methods (relatively easy) and the quality of the classification results (much more complicated due to its subjectivity). To accomplish the first task we plan to utilize the $k$-Means algorithm. There are several ways to explore the second task. We can use a limited number of the classified queries to retrieve web sites and manually compare these sites to well-known age-related sites. Alternatively, we can use a large number of the classified queries to retrieve web sites and algorithmically compare these sites to a set of age-related web sites to check for cross-classification of results.

6 References


