Abstract

We present a method for detecting and resolving lexical ambiguity in information retrieval queries. Leveraging existing word sense disambiguation tools, we define a measure of query term ambiguity based on the distribution of word senses in the relevant document set. If a query term is ambiguous, we allow the user to select the correct sense of the query term, in the style of Google’s spelling correction. Secondly, we present a method for results diversification, where one word sense dominates the top results, but there is a sizable number of documents with a second sense. We present a successful qualitative evaluation of our methods which demonstrates the plausibility and applicability of the approach.

1 Introduction

The ambiguity of natural language is commonly thought to be a major hurdle for natural language processing, particularly for information retrieval. However, research has shown [Krovetz and Croft, 1992] that many query terms have a clear sense given other terms in the query. For example, in the query “hospital bill”, we know that “bill” means an amount owned, not a monetary unit. Although this is often the case, word sense ambiguity is still present in many queries. This motivates us to develop methods for detecting and resolving ambiguity in information retrieval.

Consider the query “golf club”. It is unclear just from this query whether the user is interested in golf clubs to swing, or golf clubs to visit and play golf at. Motivated by Google’s automatic spelling correction in the form of “Did you mean ...?”, we have developed a similar tool for resolving query ambiguity. When queried with “golf club”, we wish to ask “Did you mean club as in golf equipment or club as in association?”

Following this motivation, we leverage existing word sense disambiguation tools [Patwardhan et al., 2005], which map words to their WordNet synset [Fellbaum, 1998], to detect the level of ambiguity of each query term. If this ambiguity (defined in Section 3.2) passes a threshold, we prompt the user with the two most likely senses. This sense
information can be used in several ways: we simply filter documents which do not contain the correct sense.

Another direction we take with word sense ambiguity is results diversification. Oftentimes the top ranked documents are dominated by a single sense, but there is a sizable number of lower ranked documents of a less common sense. For instance, in a Google search of “golf club”, golf associations dominate the results. Thus, we would like to detect when there is an underrepresented sense in the top documents and allow the user to see the underrepresented results.

The remainder of the paper is structured as follows. In Section 2 we formally define the problem. Section 3 presents the core of our method: sense disambiguation, ambiguity detection and diversification. We explore related work to ambiguity in information retrieval in Section 4, and present a qualitative evaluation of our methodology in Section 5. We close the paper with future work and conclusion.

2 Problem Definition

Given a query $Q$ that contains one or more terms $q_1, q_2, ..., q_m$ and a ranked collection of documents $D$ that are relevant to the query, we identify the set of potential senses $S = \{s_1, s_2, ..., s_n\}$ for the query $Q$. We then address the following problems.

2.1 Disambiguation Problem

The disambiguation problem is defined as follows. If the set $S$ suggests that $Q$ is ambiguous we allow the user to resolve the ambiguity by identifying the correct sense $s_c \in S$ that was intended. We then present the subset of results from $D$ that match the sense $s_c$.

We know that a subset of queries made will contain some degree of ambiguity. Other queries which may in fact contain ambiguous keywords will be disambiguated by additional keywords that may be present in $Q$. The first step to resolve the ambiguity problem is to identify queries that can benefit from sense disambiguation. In order to detect such queries, we examine the result set $D$ that is returned. The key observation here is that it is not queries that are ambiguous, but the results that are returned. Continuing our previous example, consider the query $golf club$. Here the word $club$ may take one of two meanings and is ambiguous. However, depending on the source of the documents in the collection, the results returned may contain only a single sense of the word $club$. If this is so, it is pointless to attempt to disambiguate $club$ since there are no results that match the alternate sense.

In the event that we detect that a query $Q$ is ambiguous and has multiple senses $S$ we rely on user input to make the decision that resolves the ambiguity. The user is presented with an option to choose between the two most likely senses and selects the correct sense $s_c \in S$. We adopt this approach because determining the exact sense intended by the user is a process that is prone to error – especially when the likelihood of the senses does not differ substantially. Furthermore, the relationship between the

---

1 We have $club$ as in golf equipment, and $club$ as in the place where people play golf.
degree of difficulty in detecting the correct sense of a query and the usefulness of such disambiguation is linear. In cases where a query is ambiguous but one sense dominates the results, the query is relatively easy to disambiguate and the task can be completed with high degree of precision. Conversely, the disambiguation also has low utility to the end user since the majority of the results already represent this sense. When the sense selection problem becomes harder i.e. when no one sense comprises a dominating share of the results, the benefit gained from the ability to select the right sense increases. However, selecting between two equally likely senses raises the likelihood of an error that would defeat the purpose of the entire exercise. Thus we rely on user to make the final decision regarding the correct sense.

2.2 Diversification Problem

The diversification problem is defined as follows. If the distribution of the set of senses $S$ across the top $l$ documents in the results set $D$ differs significantly from the distribution across the entire set, we consider the top $l$ results to be non-representative. The choice of $l$ is made by estimating the number of results the average user is expected to examine before moving to the next query. In this situation we choose the sense $s_u \in S$ that is least represented in the top $l$ results and provide a prompt to the user that allows isolation of results with the sense $s_u$.

The problem arises in situations where the intended sense of a query is rare. The resolution to the ambiguity problem will only provide users with the two most probable alternatives. However, the result set $D$ may contain a rare sense of query in which the user is interested. Secondly, even if a sense is well represented in the result set $D$, the top $l$ results that may not reflect the presence of this sense. In both cases diversification improves the users ability to comprehend the information across the entire result set when compared to examining a small subset that can be displayed on a single page.

3 Methodology

3.1 Sense Disambiguation

We use the word senses defined in WordNet as our disambiguation target. To do so, we use an implementation of the SenseRelate::TargetWord algorithm, presented in [Patarwardhan et al., 2005]. This method uses a WordNet similarity measure [Pedersen et al., 2004] to measure the semantic similarity between a word and its neighbors. Then, a word is assigned the sense that is most related to its context.

However, recent work [Liu et al., 2005] has shown that WordNet senses are often times too fine grained for use in information retrieval tasks (whereas these fine grained distinctions would be needed in natural language understanding tasks). However, our approaches are not tied to a specific method of disambiguation. Other disambiguation approaches define word senses from a specific document collection by clustering similar contexts for a given word (citation needed). This approach has the advantage of a direct correlation between the word senses and the document collection, whereas when using a WordNet word senses, there are oftentimes word senses in the document
collection not covered by WordNet (example: the Cardinals baseball team, rather than the type of bird).

In order to determine the correct sense of a word, the SenseRelate::TargetWord algorithm uses information from the context of a word to determine which sense is being used. From the SenseRelate documentation:

*The sense having the highest relatedness with its context word senses is most likely to be the sense used.*

Ideally, the relevant sentences containing the target word would be used to provide the context information. Since retrieval engines use inverted indexes to locate relevant documents, sentence boundary information will normally not be available. Thus, we use a fixed length window around the word to provide the context of a word to the SenseRelate algorithm.

### 3.2 Ambiguity Detection

In this section we present our methods for ambiguity detection. Formally, we wish to measure the ambiguity of a query term $q_i$ from a query $Q$. Ambiguity, and the idea of a word sense in general, is not a clearly defined concept. Many word sense disambiguation algorithms use a probabilistic method, so each tagging of a sentence is assigned some probability of being correct. The obvious approach would be to take this probability as a measure of sense confidence, where low probability taggings are likely ambiguous. However, this approach is very sensitive to disambiguation noise and would likely not be usable.

Secondly, we are working in an information retrieval setting rather than a general natural language processing setting, so we tailor our approach to this distinction. Instead of defining the ambiguity of a query based solely on the query itself, we define the ambiguity of a query in relation to the top $k$ relevant documents for the query. This approach has several advantages. Firstly, the ambiguity detection is more robust to a single disambiguation error, whereas the naive approach discussed above is extremely sensitive to disambiguation performance. In addition, since we are using these senses to rank relevant documents, our judgments would match the distribution of senses in our document set. For example, if there are no documents about golf clubs as golf equipment, then we would not ask the user if they mean golf equipment.

Following this motivation, we define the ambiguity of a query term as a function of the senses it takes in the relevant documents. For a query term $q_i$ and a set of $k$ relevant documents $D_k$, where $q_i$ takes $n$ senses in $D_k$, we define a maximum likelihood probability distribution $p_{q_i}$ over each sense $s$ as

$$p_{q_i}(s|D_k) = \frac{C(s, q_i, D_k)}{\sum_{j=1}^{n} C(s_j, q_i, D_k)}$$  \hspace{1cm} (1)

Here we define $C(s, q_i, D_k)$ as the number of times term $q_i$ takes sense $s$ in the set of documents $D_k$. From this probabilistic sense distribution, we define the ambiguity of a query term as the entropy of its sense distribution:
\[ A(q_i, D_k) = -\sum_{j=1}^{n} p_{q_i}(s_j|D_k) \log p_{q_i}(s_j|D_k) \]  

(2)

To decide if a given query term \( q_i \) is ambiguous, we calculate a threshold \( \theta_{q_i} \). This threshold is calculated as follows. We pick a base probability \( b \), so that if any one sense has probability greater than \( b \), the query term is not considered ambiguous. To determine the threshold, we calculate the entropy of the sense distribution \( p_n \) where one sense has probability \( b \), and the rest of the probability mass is uniformly distributed over the rest of the senses. Formally, for \( n \) senses,

\[ \theta_{q_i} = -\sum_{j=1}^{n} p_n(s_j) \log p_n(s_j) \]  

(3)

The parameter \( b \) allows a system designer (or user) to dictate how intrusive the user prompting is. However, similar to spelling correction on Google, these messages could be made unintrusive.

There are several modifications to this approach we could also justify. Firstly, the fact that we count every occurrence of a query term in a document means that we bias ourselves towards longer documents. Following the principle that words typically take one sense for an entire document, we could instead define \( C(s, q_i, D_k) \) as the number of documents in \( D_k \) for which \( q_i \) takes sense \( s \). On the other hand, longer documents are more likely to contain information related to the query. Thus, some logarithmic consideration of the term frequency in the document could be used. However, these possible modifications do not largely change our basic approach.

3.3 Diversification

As described previously in Section 2.2, our diversification methodology attempts to identify under-represented query senses in the top \( l \) results that a user is expected to view. We diversify results when the sense of a query term \( q_i \) in the top \( l \) results is less ambiguous than the sense of \( q_i \) in the result set \( D_k \). Formally, given a threshold \( \theta_d \), in terms of Equation 2 we provide a diversification option to the user when

\[ A(q_i, D_k) > A(q_i, D_l) + \theta_d \]

Consistent with the goal of diversification, this metric is intentionally asymmetric. This is because we only wish to diversify when \( q_i \) is more ambiguous in \( D_k \) than in \( D_l \). Choosing the best sense to include as a diversification option may be done in several ways. In our implementation we picked the sense \( s_u \) with the largest decrease in probability when going from \( D_k \) to \( D_l \). In terms of Equation 1 we have:

\[ s_u = \arg \max_s (p_{q_i}(s|D_k) - p_{q_i}(s|D_l)) \]

Other techniques for comparing the distribution of senses across the two collections include KL-divergence, however, it is not immediately obvious how a threshold may be applied there to reflect differences in ambiguity.
4 Related Work

A fair amount of work has studied the effect of lexical ambiguity on information retrieval. Sanderson showed in [Sanderson, 1994] that short queries benefit most from ambiguity resolution, confirming the intuition that longer queries tend to be unambiguous based on query context. This work showed that disambiguation techniques can lead to better performance (albeit a modest increase), countering an original study by [Weiss, 1973], which showed that perfect ambiguity resolution would only lead to a 1% increase in accuracy. However, these works consider disambiguation of all queries, while we focus on only those where ambiguity is highest.

Another body of work has explored specialized word sense disambiguation methods tailored to information retrieval. These works, most notably [Liu et al., 2005], give methods for coarse-grained disambiguation, avoiding the pitfalls of fine grained WordNet senses. The greatest direction for improvement for our results is to utilize these techniques.

5 Qualitative Evaluation

A qualitative evaluation of query disambiguation over six queries is presented in Table 5. Terms highlighted in bold are considered ambiguous since the entropy value is higher than the threshold. We used a base probability $b = 0.8$. The dynamic threshold technique seems to perform well for 10 of the 12 terms.

<table>
<thead>
<tr>
<th>Query</th>
<th>Term</th>
<th>Senses</th>
<th>Entropy</th>
<th>Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>golf club</td>
<td>golf</td>
<td>1</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>club</td>
<td>5</td>
<td>0.6218</td>
<td>1.1219</td>
</tr>
<tr>
<td>conference chair</td>
<td>conference</td>
<td>2</td>
<td>0.8680</td>
<td>0.7219</td>
</tr>
<tr>
<td></td>
<td>chair</td>
<td>2</td>
<td>0.1414</td>
<td>0.7219</td>
</tr>
<tr>
<td>cardinals baseball</td>
<td>cardinals</td>
<td>2</td>
<td>0.3274</td>
<td>0.8680</td>
</tr>
<tr>
<td></td>
<td>baseball</td>
<td>1</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>new bill</td>
<td>new</td>
<td>1</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>bill</td>
<td>5</td>
<td>1.2988</td>
<td>1.1219</td>
</tr>
<tr>
<td>abortion bill</td>
<td>abortion</td>
<td>1</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>bill</td>
<td>4</td>
<td>1.1750</td>
<td>1.0389</td>
</tr>
<tr>
<td>dollar bill</td>
<td>dollar</td>
<td>4</td>
<td>0.5061</td>
<td>1.0389</td>
</tr>
<tr>
<td></td>
<td>bill</td>
<td>4</td>
<td>1.2821</td>
<td>1.0389</td>
</tr>
</tbody>
</table>

Table 1: Qualitative Results

The results for dollar bill are in fact correct as we explain below.
5.1 Error Analysis

We identify two major sources of errors. First, errors in disambiguation lead to false positives for the ambiguity detection as in the case of the query \textit{abortion bill}. A number of occurrences of the word \textit{bill} were tagged with the sense “paper money” which was decidedly incorrect. This lead to a high entropy resulting in an error.

Second, the senses in WordNet are arbitrarily fine grained. In the case of the query \textit{conference chair} the two dominating senses for the word \textit{conference} were “discussion” and “meeting”. For most purposes these senses are hard to distinguish and can easily be merged. In Section 7 we discuss the probability of using more robust sense disambiguation tools to address these problems.

Lastly, some results are surprising. In the case of golf club, it seems we had a false negative, however on closer examination of the data set we find that occurrences of \textit{club} refer to the association, not the equipment\footnote{The diversification algorithm, however, finds an instance of the sense “playing card” and provides a corresponding prompt to the user.}. In the case of \textit{dollar bill} one would expect that the query is unambiguous, however, many documents do contain the word \textit{bill} in its usage as a “legal document”. This demonstrates a strength of our system. By using the result set as our context for disambiguation, we identify ambiguous results where query based tools would generally fail.

6 Conclusion

In conclusion we find that user based lexical query disambiguation can be an effective tool for information retrieval applications. The quality of results suggests that our approach can be successfully applied to larger problems to investigate robustness. Coupled with diversification, ambiguity resolution gives us a powerful mechanism for presenting results to a user in a manner that allows for better comprehension of the contents of the result set.

7 Future Work

The most pressing direction for future work is a quantitative analysis of our methods. We propose such an evaluation here. Given a standard IR system evaluation set, such as TREC, we can manually tag the queries with the correct sense. Then, we can run our system on the queries, using our manually tagged queries to simulate user input. Thus, we assume that users can correctly select the relevant sense for an ambiguous query.

The precision and recall for this experiment could then be compared with the baseline system. We can conduct a similar analysis for the performance of diversification. These studies would not account for how intrusive the sense queries would be. It is difficult to study this aspect without conducting user studies, as the tolerance of actual users is dependent on user group and interface. Another direction of improvement is the use of disambiguation tools that are better suited to IR tasks, namely \cite{Liu et al., 2005}. We could also explore the use of corpus-trained statistical disambiguation tools rather than those based on semantic similarity.
References


