

Query Term Ranking based on Dependency Parsing of Verbose Queries

Jae-Hyun Park and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA, 01003, USA
{jhpark,croft}@cs.umass.edu

ABSTRACT

Query term ranking approaches are used to select effective terms from a verbose query by ranking terms. Features used for query term ranking and selection in previous work do not consider grammatical relationships between terms. To address this issue, we use syntactic features extracted from dependency parsing results of verbose queries. We also modify the method for measuring the effectiveness of query terms for query term ranking.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Query formulation

General Terms

Algorithm, Experimentation, Performance

Keywords

Dependency Parse, Query Reformulation, Query Term Ranking

1. INTRODUCTION

Most search engines have a tendency to show better retrieval results with keyword queries than with verbose queries. Verbose queries tend to contain more redundant terms and these terms have grammatical meaning for communication between humans to help identify the important concepts.. Search engines do not typically use syntactic information.. For example, given a verbose query, “*Identify positive accomplishments of the Hubble telescope since it was launched ...*”, search engines cannot recognize that “*Hubble telescope*” is the key concept of the query whereas “*accomplishments*” should be considered as a complementary concept, while people can readily identify this by analyzing the grammatical structure of the query. Therefore, search engines potentially need a method for exploiting this structure.

In this work, we rank terms in a verbose query and reformulate a new query using selected highly ranked terms. Good selection methods should be able to leverage the grammatical roles of terms within a query. To do this, we use syntactic features extracted from dependency parsing trees of queries. In addition, we suggest a new method for measuring the effectiveness of terms for query term ranking.

Copyright is held by the author/owner(s).
SIGIR’10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

2. QUERY TERM RANKING

2.1 Features extracted from Dependency Parsing

We use syntactic features extracted from dependency parsing to capture the grammatical properties of terms for a query. Features used by previous work in query term ranking [1, 6] are inadequate to reflect these characteristics. The limitation of these features is that they are based on individual terms. Features such as tf, idf, part-of-speech (PoS) tag, etc. will not change even if the role of the term changes according to the syntactic structure of queries. Even features for sub-queries [5] are also unlikely to reflect grammatical characteristics because they are not affected by the structure of queries.

Therefore, we propose to overcome this limitation by using dependency parsing trees. A typed dependency parse labels dependencies with grammatical relations [3]. Figure 1 shows an example of a typed dependency parse tree. Dependency parsing tree fragments of terms can provide grammatical information about terms in queries [2].

It is infeasible to use all dependency parse tree fragments as syntactic features. We limit the number of arcs in syntactic features to two arcs. Even if we limit the number of arcs, some of collected tree fragments are too specific to

Sentence: Identify positive accomplishments of the Hubble telescope since it was launched in 1991.

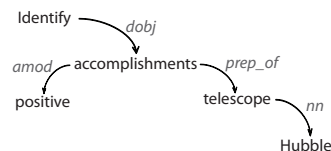


Figure 1: An example of dependency parsing trees. Labels attached to arcs are types of dependencies.

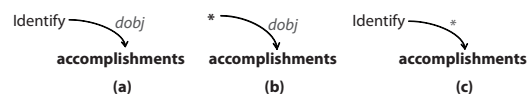


Figure 2: Three types of syntactic features for the term “*accomplishments*”. (a) An original syntactic feature (b) The word is generalized to a * (c) The type of the dependency is generalized to a *

have a reliable amount of training data and not all of them are useful. We generalize syntactic features which consist of arcs labeled with dependency types and nodes representing words which are dependent. Figure 2 shows an example of an original syntactic feature and its generalized features. In the figure, “*” means any word or any type of dependency.

2.2 Query Term Ranking

Our approach aims to rank terms in a query and to reformulate the query using the ranking. To build training data for a ranking model, Bendersky and Croft [1] manually annotate the concept from each query that had the most impact on effectiveness. For given terms $T = \{t_1, t_2, \dots, t_n\}$, they used labeled instances (t_i, l_i) , where l_i is a binary label, as training data. However, queries can have more than one effective term or concept. In addition, it is difficult for annotators to judge the effectiveness of a term. Therefore, we estimate the effectiveness of terms, i.e., the labels for training data, by evaluating the search results of terms in training data. By using these estimated scores, we expect that a ranking model can take account of all terms in a query and consider how effective they are.

Lee et al. [6] point out the importance of underlying correlations between terms. Previous work has evaluated the effectiveness of groups of terms instead of individual terms to capture these relationships [5, 6]. The problem is that the number of unique groups will grow exponentially with the size of the term groups and it will cause a data sparseness problem. We used the following equation for $E(t)$, the effectiveness of a term to reflect the effects of relationships between terms in training labels.

$$E(t) = \frac{1}{N} \cdot \sum_{c \in C_m} (\varphi(c, t) - \varphi(c)), \quad (1)$$

where C_m is all possible combinations of m terms except t . N is the number of elements in C_m and $\varphi(\alpha)$ is the search performance of α . Eq. (1) estimates the effectiveness of term t through aggregating the impacts of term t on effectiveness when using it with other terms in C_m . Thus, the scores of Eq. (1) reflects the correlations between t and other terms.

3. EXPERIMENTS AND ANALYSIS

We evaluated our proposed method using two TREC collections: Robust 2004 (topic numbers are 301-450 and 601-700) and Wt10g (topic numbers are 450-550). The average number of nouns, adjectives and verbs in queries of Robust2004 and Wt10g are 8.7 and 6.5 per a query, respectively. We used the language model framework with Dirichlet smoothing (μ set to 1,500). Indexing and retrieval were conducted using the Indri toolkit.

To rank query terms, we used RankSVM [4]. We trained query term ranking models for each query using leave-one-out cross-validation in which one query was used for test data and the others were used for training data. We labeled training data based on Key concepts [1] and the effectiveness measured by Eq. 1 in which we chose nDCG as the performance measure. We used syntactic features in addition to tf, idf, and PoS tag features.

When we combined selected terms with original queries, we used two approaches. First, we assigned uniform weights to selected terms (**binary**). Alternatively, we used query term ranking scores as the weight for selected terms (**weight**).

Table 1: Mean Average Precision (MAP) of Robust04 and Wt10g collections, Key-Concept: using key concept [1] as labels of training data, Auto: using effectiveness in retrieval as labels of training data

		Robust04	Wt10g
<title>		25.17	18.55
<desc>		24.07	17.52
Key-Concept	binary	23.98	18.55
	weight	24.24	19.45
Auto	binary	25.40	17.91
	weight	26.21	19.15

Experimental results in Table 1 shows that selected terms by using query term ranking have better performance than description queries except for one result in which we used key concepts and uniform weighting. In this case, only the most important concepts in queries are labeled, whereas the effectiveness in retrieval is measured for all terms in queries. This difference makes the method using the effectiveness of terms (**Auto**) superior for the relatively longer queries in Robust2004, and the method using key concepts (**Key Concept**) better for the shorter queries in Wt10g.

4. CONCLUSIONS

In this paper, we propose a query term ranking method that uses syntactic features extracted from dependency parsing trees. By using syntactic features, we can take into account grammatical relationships between terms. We also modify the query term ranking method to measure the effectiveness of terms based on combinations of terms. Experimental results showed that the terms selected by the query term ranking method improved retrieval performance.

5. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0711348. Any opinions, findings and conclusions or recommendations expressed in this material are the authors’ and do not necessarily reflect those of the sponsor.

6. REFERENCES

- [1] M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In *Proc. ACM SIGIR*, pages 491–498, 2008.
- [2] A. Chanen. *A comparison of human and computationally generated document features for automatic text classification*. PhD thesis, The University of Sydney, 2009.
- [3] M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proc. LREC 2006*, 2006.
- [4] T. Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proc. ACM KDD*, pages 133–142, 2002.
- [5] G. Kumaran and V. Carvalho. Reducing long queries using query quality predictors. In *Proc. ACM SIGIR*, pages 564–571, 2009.
- [6] C. Lee, R. Chen, S. Kao, and P. Cheng. A term dependency-based approach for query terms ranking. In *Proc. CIKM*, pages 1267–1276, 2009.