Improving Term Weighting for Community Question Answering Search Using Syntactic Analysis

David Carmel, Avihai Mejer, Yuval Pinter, Idan Szpektor
Yahoo Labs
Haifa, 31905, Israel
{dcarmel,amejer,yuvalp,idan}@yahoo-inc.com

ABSTRACT

Query term weighting is a fundamental task in information retrieval and most popular term weighting schemes are primarily based on statistical analysis of term occurrences within the document collection. In this work we study how term weighting may benefit from syntactic analysis of the corpus. Focusing on Community-based Question Answering (CQA) sites, we take into account the syntactic function of the terms within CQA texts as an important factor affecting their relative importance for retrieval. We analyze a large log of web queries that landed on Yahoo Answers site, showing a strong deviation between the tendencies of different document words to appear in a landing (click-through) query given their syntactic function. To this end, we propose a novel term weighting method that makes use of the syntactic information available for each query term occurrence in the document, on top of term occurrence statistics. The relative importance of each feature is learned via a learning to rank algorithm that utilizes a click-through query log. We examine the new weighting scheme using manual evaluation based on editorial data and using automatic evaluation over the query log. Our experimental results show consistent improvement in retrieval when syntactic information is taken into account.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: Term weighting; Part-of-speech tagging; Dependency parsing; Community question answering; Learning to Rank

1. INTRODUCTION

Query term weighting is a fundamental task for information retrieval (IR), and an abundance of weighting schemes have been proposed and studied during the years [33, 32, 21, 2]. The common belief in the IR community is that statistical analysis, mostly based on term counts, is satisfactory for providing highly effective query term weighting for retrieval. Indeed, popular weighting schemes such as tf-idf [33], BM25 [32], statistical language models [21], divergence from randomness [2], and many others, are all primarily based on statistical analysis of term occurrences within the text.

While many attempts have been made in the past to enrich statistical methods for term weighting with linguistic analysis methods [1, 6, 22, 26, 30], standard Natural Language Processing (NLP) methods such as morphological analysis, part-of-speech tagging, dependency parsing, etc., failed to show significant improvement over shallow methods such as stop-wording, stemming, and word proximity analysis. It has therefore become widely accepted in the IR community that the impact of linguistic methods on term weighting is marginal, and that ROI of linguistic analysis, i.e. the expected contribution to retrieval, if any, compared to the computational cost, is not justifiable [35, 36, 4].

We hypothesize that the low impact of NLP-based term weighting methods for retrieval could be attributed to the short queries that most IR systems deal with, especially on the Web. For such queries, the appearance of the query terms in the document is a strong enough indication to its relevancy to the query. However, long queries can potentially benefit from linguistic analysis as the syntactic roles of the query terms, and their inter-relations, affect their relative contribution to relevance estimation.

As a test-case, in this paper we analyze a vertical search scenario where the search engine issues the query against several verticals in addition to searching over the Web [3]. We focus on a Community-based Question Answering (CQA) vertical which searches over CQA collections such as Yahoo Answers , Quora and Baidu Zhidao. We focus on this type of search, since typical Web queries submitted to the CQA vertical are longer and contain more content than general Web queries, and are therefore likely benefit from syntactic analysis of the documents (see Fig. 2 for query length analysis of CQA based Web queries). We emphasize that these queries, even longer, are still typical (telegraphic) Web queries, in contrast to natural language questions that are submitted directly to a CQA site which have an explicit intent to be answered by humans.

Take for example the Web query "*color or paint brush*", in relationship to the CQA texts *t1*="*I would like to brush the color through my hair*", *t2*="*What is the color of this brush?*", and *t3*="*Where can I find a color brush?*". In terms of bag of words, and even when considering word proximity, there is no strong preference to either of the texts. Yet, The part-of-speech of the query term '*brush*' in *t1* is *Verb*, whereas in *t2* and *t3* it is *Noun*. Furthermore, in *t2*, '*color*' functions as a subject, while in *t3* as a modifier in a noun

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2661901.

compound. If we would know that nouns are better indicators for relevancy than verbs, and modifying nouns are better than sentence subjects, then we should point at *t3* as more relevant to the query than *t1* or *t2*.

To test our hypothesis, we suggest to apply part-of-speech tagging and dependency parsing to candidate CQA documents, and to include the part-of-speech categories and syntactic roles of query terms appearing in the text as factors affecting the relative importance of documents for retrieval. We note that part-of-speech tagging and dependency parsing has already been used for IR tasks [1, 18, 13, 6, 26, 22, 30, 31, 41, 29]. Yet, these approaches **parse the query**, and are therefore limited to natural language queries, for which syntactic analysis of the query is feasible and reliable. In contrast, our model is based only on document content analysis and therefore can be applied to any query. Moreover, to the best of our knowledge, our work is the first one to utilize syntactic analysis of the document content for query term weighting.

To this end, we analyze a large dataset of Web queries that resulted in a click on a Yahoo Answers document. By parts-of-speech tagging and by syntactically parsing the title of each clicked document, we notice significant, and sometimes surprising differences in the probability of a title term to appear in a landing (click-through) query given the term's part-of-speech tag, and similarly given its syntactic role in a dependency parse tree.

Following our findings, we propose a novel term weighting method that makes use of the syntactic information available for each query term occurrence in the document, on top of term occurrence statistics. Specifically, we weigh in the relative importance of the part-of-speech tag and the syntactic role of all occurrences of the matched query terms in the document, effectively summing a syntactic weight for the matched terms. These syntactic ranking features are incorporated into the final ranking score of the document, which also includes a rich set of frequency-based scoring features. The relative importance of each feature is learned via a learning to rank algorithm (LTR) that utilizes a click-through query log.

We evaluated the contribution of our syntactic term weighting features for retrieval under two settings: a) large-scale automatic evaluation over a click-through query log; b) manual evaluation of the top retrieved documents for a set of tested queries. We compared our approach to a state-of-the-art LTR model that utilizes only frequency-based term weighting features. Both evaluations show a significant improvement in document ranking when syntactic information is incorporated, demonstrating the potential of our weighting scheme for retrieval.

2. RELATED WORK

2.1 IR for CQA

Significant research efforts have been conducted over the years in attempt to improve information retrieval over CQA sites [19, 16, 40, 9, 41, 38]. Most of these works focus on finding similar archived questions for input natural language questions. Jeon et al. [19] and Xue et al. [40] incorporated a translation-based retrieval model to find semantically similar questions to the user input question, with the goal of overcoming the lexical gap and the vocabulary mismatch between question variations and between questions and answers. Cao et al. [9] proposed a category-based framework that exploits the category meta data within Q&A pages. Duan et al. [16] identified the question topics and the question focus of an input question, following shallow parsing. They showed how to incorporate them into the language model used for retrieval. Cai et al. [8] incorporate latent topic similarity as a complementary frequency-based approach to word-based translation language models. Wang et al. [37] use a tree kernel to measure the similarity between the syntactic parse trees of the input question and a candidate retrieved question. Zhang et al. [41] proposed a term re-weighting scheme which assumes that strongly dependent terms in the queried question should be assigned with similar weights.

In contrast to the work described above that assume natural language questions as input, our work deals with queries issued in Web search engines. Such queries are typically shorter, keyword based, and lack a well formed syntactic structure [24]. Additionally, most of these works focused on statistical-based term weighting methods, while we apply syntactic analysis for term weighting, on top of statistical-based features.

Only few papers address searching CQA archives with Web queries. Wu et al. [38] target short Web queries, of length 3 or shorter, which are rather underspecified. The authors identify the query intent from several sources such as the question description (the body field), query log, and search results and incorporate it in retrieval. Still, [38] points out that short queries are not the majority of queries landing on CQA sites, in agreement with our analysis shown in Fig. 2. Since syntactic analysis mostly help long queries (see Section 7.1.2), our work complements this approach. Liu et al. [23] utilized several statistical-based query/question matching features as part of their searcher satisfaction classification approach (including BM25, TFIDF and LM). These features are included in our baseline model (see Section 5.1.1), and we propose to complement them with syntactic-based features.

2.2 Syntactic Analysis for IR

Another direction that is relevant to our work is the attempt to improve information retrieval using NLP techniques [35, 36, 1, 18, 6, 22, 26, 30, 31]. Out of this large body of works, we present those that employ the syntactic analysis of queries and documents that are most relevant to our work.

Allan and Ragahavan [1] applied part-of-speech (POS) tagging for query disambiguation. By extracting common patterns of POS tags, and by identifying patterns that are frequently found around the query terms in the corpus, they were able to propose meaningful natural language questions for query disambiguation. Shah and Croft [34] used parts-of-speech to assist in identifying the focus of the query. Barr et al. [6] investigated the applicability of parts-of-speech to typical Web queries. They showed that proper nouns and common nouns together constitute over 70% of the query terms, and that the majority of queries are noun-phrases. They showed that matching the POS tag of a word in the query with the POS tag of the same word in the document is a significant feature in a LTR framework, though overall no statistically significant increase in retrieval performance was shown.

Some works captured long-distance dependencies between query terms using dependency parsing, in contrast to tradi-

tional proximity features, which are typically defined based on term co-occurrence in a fixed window size [28]. Gao et al. [18] proposed a dependence language model in which term dependencies are generated based upon the linkage structure of the query and the document. The query is generated from the document dependency language model in two stages: the linkage is generated first, and then each term is generated in turn depending on other previously generated terms according to the linkage.

Several attempts applied query syntactic parsing for query term re-weighting. Lee et al. [22] weighted query terms by detecting long-distance dependencies using a linguistic parser. POS tags and term dependencies features were integrated into a regression model used for query term re-weighting. Lu et al. [26] derived semantic features of the query using part-of-speech tagging and named-entity recognition. These features were integrated with many other signals to construct a ranking function using LTR techniques. Results showed that syntactic features improve performance particularly for long queries. Park and Croft [30] selected the most important terms in a verbose query using syntactic features extracted from the query's dependency parsing tree. Term weights were determined by taking into account grammatical relationships between the query terms, in addition to traditional statistical based term features. Park et al. [31] align the syntactic parsed trees of the query and the content via matches between different types of syntactic relations in the document and the query.

Dependency parsing and parse tree matching have been widely used for passage retrieval for question answering [13, 29]. Cui et al. [13] proposed to measure the degree of overlap between dependency relations in candidate passages with their corresponding relations in the input question. Syntactic analysis of the question and passages was also employed by the IBM DeepQA project in order to validate candidate answers [29]. The degree of the match between the syntactic graph of the modified input question and the syntactic graph of the passage constitutes the candidate retrieval score.

All of the algorithms and models described above syntactically analyze the query for term re-weighting, term dependency detection, and parse tree matching. General Web query parsing is still an open challenge, and as far as we know, there are no existing mature parsers for this task. This is probably one of the reasons that only few works tackle the challenge of syntactic parsing of general Web queries [26, 6]. We suggest to circumvent this challenge by re-weighting terms based on parsing the content rather than the query. Therefore, our model does not suffer from the lack of appropriate query parsing tools and can be applied to any query type.

3. YAHOO ANSWERS

In this paper we perform our analysis and experiments on a document collection taken from Yahoo Answers . We chose Yahoo Answers since it is the largest and most popular CQA web-site to date, containing hundreds of millions of questions about diverse topics, such as sports, healthcare, entertainment, politics, science and many others. In Yahoo Answers , askers post questions that consist of a *title*, a short summary of the question, and a *body*, containing a detailed description of the question and even additional details. During a four day period, the question can be answered by other Yahoo Answers users. During this period, the asker may choose a best answer, but if they do not, the task of selecting a best answer is delegated to the community for an indefinite time. Once a best answer is chosen, the question is said to be "resolved." Finally, any question that is not answered at all within four days is removed from the site.

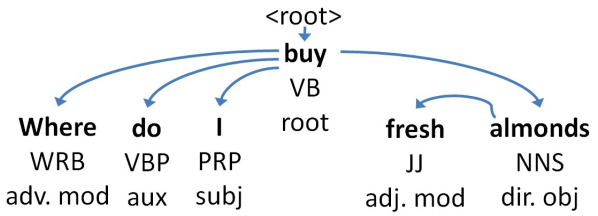

Figure 1: POS tagging and dependency parse tree for the question "*Where do I buy fresh almonds?*". The upper label of each token is its POS tag and the lower label is its syntactic role.

Each Yahoo Answers question page contains the text of the question being asked, including the title and the body of the question. It also includes all the answers provided for this question, and if a best answer was chosen for the question, it is appropriately highlighted in the page. Given a query issued by a searcher, search-engines expose mainly the question's title on the search result page. Therefore the relevancy of the title to the query is one of the main reasons for a user to click on a specific Yahoo Answers link. Following this reasoning, we chose to focus on the analysis and modeling of the grammatical structure of only the title of a Yahoo Answers question, leaving the body and answer syntactic analysis for future work.

The dataset used for this study contains 54 million question pages of Yahoo Answers and 500,000 Web queries, each one landed on one of these pages. We use this dataset for analyzing the relative importance of different syntactic information for term weighting, and for learning our novel ranking model.

4. SYNTACTIC BEHAVIOR ANALYSIS

We next study whether query terms correspond more to specific part-of-speech tags and dependency relations. If such behavior does occur, it is a good indication that a term weighting scheme may benefit from incorporating the syntactic information of a term within the final term weight.

To this end, we analyzed the 500,000 queries in our dataset and the documents they landed on, viewed as {*query*, *clicked-question*} pairs. We syntactically analyzed the title of each clicked question in the dataset using the Stanford Parser[1] [20, 14] under the "all typed dependencies" setting. From this analysis we extracted for each token in the title its part-of-speech tag and its *syntactic role* – the dependency relation in which this term is the child. In addition, each query was tokenized using the Stanford tokenizer, in order to have a consistent tokenization between each paired query and question. We finally lower cased all texts, but did not apply any additional transformation (*e.g.* stemming). Especially, we emphasize that the queries themselves were not POS-tagged or parsed.

[1] `http://nlp.stanford.edu/software/lex-parser.shtml`

part of speech	Pr(in title)	Pr(in query | in title)
proper noun, singular	0.412	0.490
noun, singular	0.780	0.469
adjective	0.500	0.435
noun, plural	0.389	0.413
verb, base	0.552	0.366
number	0.124	0.354
verb, participial	0.104	0.346
verb, gerund	0.143	0.330
verb, 3rd-person present	0.342	0.300
preposition	0.579	0.298
verb, past	0.109	0.291
determiner	0.510	0.278
pronoun	0.392	0.270
verb, present	0.351	0.266
adverb	0.246	0.258
modal	0.189	0.224
coordinating conjunct	0.194	0.172

Table 1: POS tag statistics in 500,000 {query, clicked-question} pairs

As an example for this process, consider the query "*fresh almonds cheap*" that landed on the Yahoo Answers question page with the question title "*Where do I buy fresh almonds?*". The POS tags and dependency parse tree for this question title are shown in Fig. 1. The token '*fresh*' in the title, for which a matched token was found in the query, is tagged with the POS tag ADJECTIVE and has the syntactic role ADJECTIVAL MODIFIER. In a similar manner, token '*almonds*' was tagged with PLURAL NOUN and has the syntactic role DIRECT OBJECT.

For our analysis, we inspected title terms that appeared in the corresponding query. We then looked for differences in occurrence statistics between such title terms, *e.g.* '*fresh*' and '*almonds*', compared to title terms that did not appear in the corresponding query, *e.g.* '*buy*'. Query terms that do not appear in the title, such as '*cheap*', are ignored in this analysis. The next subsections depict our findings for part-of-speech tags and syntactic roles.

4.1 Part-Of-Speech Analysis

For each of the possible 42 part-of-speech tags [27] we collected all the titles in which at least one word is tagged with the target POS tag. We report the percentage of such titles in our datasets, denoted by Pr(*in title*). Next we counted, for each collected title, the proportion of tokens that are tagged by the target POS tag and also appear in the corresponding query, out of all the tokens with this tag. We report this proportion averaged across all of the collected titles, denoted by Pr(*in query*|*in title*). This statistics corresponds to the empirical probability of a word that appears in a specific tag in the title to also appear in a clickthrough query. The two statistics per POS tag are shown in Table 1. For easier reading, we only show the results for frequent POS tags, which appear in at least 10% of the titles.

From the table we see that noun classes, especially proper names, are very likely to appear in the query when they appear in the title. This should come as no surprise, since nouns often refer to entities, which are the focus of many queries. In opposition, grammatical classes that only serve for syntactic soundness of proper sentences, and thus serve little purpose in conveying content, are likely to be lost in the correspondence. These include determiners ('*a*', '*the*', etc.), modal verbs ('*can*', '*would*'), conjuncts ('*and*', '*or*') and pronouns ('*I*', '*you*'). These findings reinforce the general practice of treating such words as search-engine stopwords, removing them from incoming queries.

syntactic role	Pr(in title)	Pr(in query | in title)
noun as modifier	0.463	0.517
adjective as modifier	0.411	0.444
direct object	0.516	0.444
object of preposition	0.545	0.432
noun as subject	0.790	0.369
sentence root	0.998	0.368
conjunct	0.184	0.335
parataxis	0.014	0.224
discourse	0.008	0.157

Table 2: Syntactic role statistics in 500,000 {query, clicked-question} pairs.

Another finding is the differences between different forms of verbs. While the base form (*e.g.* '*rest*') is preserved at roughly baseline proportions (0.366 vs. 0.363), other verb forms such as past ('*rested*') or 3rd person singular ('*rests*') are substantially less likely to appear in the corresponding query (0.300, 0.266 respectively). While our analysis does not include stemming, and therefore may miss token matches between different verb forms in the title and the query, we think that this result points at a common use of base verb form in conveying required actions in queries, such as "*can I find...*", "*where to buy...*". This behavior is echoed to some extent in the more verbose question writing. On the other hand, other verb forms are used more for describing events and personal experience, which are related to the context of the question, but not directly to the information asked for, *e.g.* in "*I was sleeping when...*" or "*I worked hard to...*". Therefore they do not appear in queries, in which such descriptive context is removed for the sake of brevity.

4.2 Dependency Parsing Analysis

Similarly to the statistics collected for the various part-of-speech tags, for each of the 48 possible syntactic roles we collected all the titles in our dataset that contain at least one token that is annotated with the target syntactic role. We then measure the same statistics that are described in Section 4.1, namely Pr(*in title*) (the percentage of such titles in the dataset) and Pr(*in query*|*in title*) (the chances of seeing a token annotated with the target syntactic role also in a corresponding clickthrough query). For the sake of clarity, we present these statistics in Table 2 only for the more interesting syntactic roles in our dataset.

It is no surprise that syntactic roles that are related to noun tokens have higher probability to appear in a corresponding query than syntactic roles associated with other parts of speech, as a corollary of the previous subsection's results. Yet, the statistics in Table 2 draw a clear distinction between various noun-related syntactic roles (all these differences are statistically significant with $p<0.0001$). Specifically, to our surprise, nouns have more chances of appearing in the corresponding query when they act as modifiers to other nouns, such as the word '*science*' in '*a science book*', rather than as any other syntactic role, including sentential subject and direct object, which are the more common syntactic roles.

Interestingly, the inclusion of modifiers in queries is not unique to nouns. The other type of noun modifiers – adjectives (as '*red*' is in "*a red book*") – is the second most likely

modifier	head noun
Gatwick	Airport
radiator	question
transmission	problems
Yahoo	password

Table 3: Examples of noun phrase parts in question titles where the head noun did not appear in the corresponding query

type of tokens to be seen in a corresponding query. These findings are not trivial, as it might have been expected that the three core elements of the sentence, the subject, main predicate ("root") and direct object, would be those that are intended most by queries. After examining a sample of the data, the reason can be stated with confidence: when a noun phrase is constructed, many times the main noun describes a rather general category, while the modifier specifies a type. In addition, many times the modifier already captures the semantics of the category, as in '*Gatwick Airport*'. When constructing a short Web query, the searcher would add first the terms that most accurately capture his/her information need, therefore choosing modifiers over head nouns. The sample in Table 3 exemplifies which of the parts is more crucial to the searcher.

In Table 2 we also present some low-probability syntactic roles: the parenthetical elements marked by the parser as parataxis ("*Where can I find,* ***my brother asks****, the best pizza in Chicago?*") and discourse ("*Was that a great game or what,* ***eh****?*"). These are parts of the sentence we would most likely not associate with the query, even though the tokens within them may be considered important according to standard statistical features such as inverse document frequency.

Finally, we so far discussed only different types of incoming dependency edges, that is edges in which the target token is the child. These types, normally only one per token, represent the syntactic role of the token in the sentence. For completeness, we conducted a similar analysis for outgoing edges which are not known to have any intrinsic syntactic merit. Indeed, our analysis, omitted here, showed no interesting results.

The analysis conducted in this section showed encouraging signals that the searcher's choice of words in a submitted query is also derived from the grammatical information each word is expected to carry in relevant documents, at least as a proxy to their expected semantic role in such documents. We next propose one approach for including this information in IR tasks, namely for query term weighting.

5. SCORING MODEL

Our novel scoring model for term weighting is based on syntactic analysis of the document text (*e.g.* the question's title in our experiments), taking into account the POS tag and the syntactic role of each occurrence of the query terms within the document. The scoring formula integrates the syntactic information associated with the occurring query terms together with statistical-based measures of the similarity between the document and the query. The relative weight of each component in the scoring formula is determined using a learning to rank (LTR) method based on click-through data.

Feature	Formulation
L1	$\sum_{q_i \in q \cap d} c(q_i, d)$
L2	$\sum_{q_i \in q \cap d} log(c(q_i, d) + 1)$
L3	$\sum_{q_i \in q \cap d} \frac{c(q_i, d)}{\|d\|}$
L4	$\sum_{q_i \in q \cap d} log(\frac{c(q_i, d)}{\|d\|} + 1)$
L5	$\sum_{q_i \in q \cap d} log(\frac{\|C\|}{df(q_i)})$
L6	$\sum_{q_i \in q \cap d} log(log(\frac{\|C\|}{df(q_i)}))$
L7	$\sum_{q_i \in q \cap d} log(\frac{\|C\|}{c(q_i, C)} + 1)$
L8	$\sum_{q_i \in q \cap d} log(\frac{c(q_i, d)}{\|d\|} log(\frac{\|C\|}{df(q_i)}) + 1)$
L9	$\sum_{q_i \in q \cap d} c(q_i, d) log(\frac{\|C\|}{df(q_i)})$
L10	$\sum_{q_i \in q \cap d} log(\frac{c(q_i, d)}{\|d\|} \frac{\|C\|}{c(q_i, C)} + 1)$
H1	BM25 score
H2	log(BM25 score)
H3	LMIR (with Dirichlet smoothing) score

Table 4: LETOR features: $c(t, X)$ is the term frequency of t in X; $df(t)$ is the document frequency of t; $|X|$ is the total number of terms in X.

It is important to note that once LTR training phase is completed, the term features required by our term weighting model are computed only once at indexing time, including any required syntactic analysis. Therefore, at retrieval time, these term features are efficiently utilized like any other statistical features stored in the index per document term occurrence.

In the following we detail the scoring features used by the scoring formula and the learning method applied for training the model.

5.1 Feature Extraction

The input to our scoring formula is a query document pair (q, d). We next detail the three types of features induced as (q, d) representation. In the first type of features we include the common statistical-based features. Then, we describe our two novel feature types, which are derived from the syntactic analysis of the document text, namely the POS tags and the dependency parse tree of this text.

5.1.1 Statistical-based features

We extract the common statistical-based similarity features in the learning-to-rank (LTR) literature, as detailed in [25]. Specifically, we extract features L1-L10 and H1-H3, but we do not extract any hyperlink features, which are not available under our experimental framework. Table 4 summarizes these features. We note that H1 is the known $BM25$[2] score [32] of the document as calculated by the search engine for a given query . We use it both as one of the features, as well as one of the baseline scorers to compare with in the experiments described in Section 6. H3 is the language modeling score of the document for the query based on the query likelihood score derived from the document language model with Dirichlet smoothing.

5.1.2 Part-of-Speech features

We derive our first novel feature type from POS tags. To this end, we parse the title of each CQA document using the

[2]We used the $BM25$ implementation provided by *Apache Lucene* (*http://lucene.apache.org*), using default parameter setting ($k1 = 1.2$, $b = 0.75$).

Stanford parser. For each word, a tag is assigned out of the 42 POS tags in the Penn tagset. We maintain two features for each tag in this set:

$\mathbf{POS_{bin}(q, d, p)}$: Given a POS tag p, this feature counts the number of occurrences of query terms in the document title that are tagged with p by the POS tagger:

$$POS_{bin}(q, d, p) = \sum_{t \in q} \sum_{o_t \in Occ(t, d.T)} \delta(POS(o_t) = p)$$

where q is the query, d is the document, $Occ(t, d.T)$ is the list of occurrences of term t in the document title $d.T$, $POS(o_t)$ is the POS tag of term t in occurrence o_t, and $\delta()$ is the indicator function.

$\mathbf{POS_{idf}(q, d, p)}$: The second feature sums the *idf* values of the query terms, while traversing over all occurrences of the query terms tagged with p in the document title:

$$POS_{idf}(q, d, p) = \sum_{t \in q} \sum_{o_t \in Occ(t, d.T)} idf(t) \cdot \delta(POS(o_t) = p)$$

where $idf(t) = \log(N/N_t)$ stands for the term's inverse document frequency; N is the number of documents in the collection and N_t is the number of documents containing the term t.

We chose these two variations since the POS_{idf} feature considers the rareness of a term, as reflected by its idf, in addition to its POS tag, while the POS_{bin} feature is more coarse. Our experiments showed that both types of features are useful for the ranking function.

Additionally, we also maintain features for coarse grained POS tags (CPOS). We used the *two-letter* tag method proposed by Collins et al. [11] to integrate a family of POS tags to a CPOS. For example, the CPOS tag NN integrates all noun tags without distinguishing between singular/plural and common/proper nouns (i.e., NN, NNS, NNP, NNPS). Following this schema, the original 42 POS tags are reduced to 23 CPOS tags. Similarly to the features induced from POS tags, for each CPOS tag *cp* we maintain two features; a binary feature and an idf-based feature, denoted $CPOS_{bin}(q, d, cp)$ and $CPOS_{idf}(q, d, cp)$ respectively. The POS features capture subtleties that the CPOS features ignore, yet the CPOS features are more robust to tagger errors and provide some generalization. Our experiments showed that both types of features are useful for the scoring function.

5.1.3 Syntactic role features

Features of our second novel type are derived from the syntactic role of the query terms within the document title, or more specifically, the dependency relationship type of the term's incoming edge in the dependency parse tree. Looking at Figure 1, for example, the syntactic role of the word *fresh* is *adjectival modifier* while the syntactic role of *almonds* is *direct object*.

There are 48 possible syntactic roles when using the Stanford parser. As with the POS features, we maintain two features for each syntactic role sr, one with a binary value and one with idf value:

$$DP_{bin}(q, d, sr) = \sum_{t \in q} \sum_{o_t \in Occ(t, d.T)} \delta(Role(o_t) = sr)$$

$$DP_{idf}(q, d, sr) = \sum_{t \in q} \sum_{o_t \in Occ(t, d.T)} idf(t) \cdot \delta(Role(o_t) = sr)$$

where $Role(o_t)$ is the syntactic role of term t in occurrence o_t. As in the case of POS features, $DP_{bin}(q, d, sr)$ counts the number of occurrences of the query terms in the document title which have an sr role, according to the parser. $DP_{idf}(q, d, sr)$ sums the *idf* values of the query terms, while traversing over all occurrences of query terms tagged with sr in the document title.

We note that for both POS features and syntactic role features we tested if normalizing by the title's length would affect the performance of the above features. We found that title normalization did not improve performance and therefore we skip these results in our experiments.

5.2 Model Weight Learning

We experimented with three state-of-the-art LTR algorithms in order to determine the final scoring formula: LambdaRank [7, 15], ListMLE [39] and SVMRank [10]. For LambdaRank, we experimented both with optimizing NDCG and MRR. Additionally, for LambdaRank and ListMLE we tested both an underlying linear regressor and a two-layer neural network. We used AdaGrad [17] in all these gradient-based schemes. For SVMRank, we implemented an online variant, which we further detail below.

Our training dataset consists of a random sample of 57,000 queries out of the queries in our dataset (see Section 3). We maintain for each query the Yahoo Answers document it landed on as a query/document pair. As the CQA collection on which the search task is performed, we use the 54 million Yahoo Answers documents in our dataset, indexed by Lucene. Additionally, a small held-out validation set of 5,000 query/document pairs was sampled, on which the hyper-parameters of the learning procedures were optimized.

In our experiments, the best performing algorithm was the online SVMRank. It outperformed the other algorithms consistently over all our test sets, and therefore in the next sections we report results only for this algorithm.

5.2.1 Online SVMRank

Our online variant of SVMRank searches for a linear weight vector that should rank the clicked document for each training query higher than other top scored documents for this query.

The training algorithm begins with a zero weight vector and updates it for each example using the AROW online learning procedure [12], which showed comparable performance to SVM. Specifically, for each query example our algorithm first reranks the top 100 documents retrieved by Lucene using the currently learned ranker. Then, it selects document pairs consisting of the clicked document feature vector v_c and the feature vector v_d of each of the top K ranking documents. Following the original SVMRank algorithm, for each pair the algorithm generates the difference vector $v_c - v_d$ as a training example for the linear ranker.

The optimized parameters for this algorithm, based on the validation set, are: 12 training rounds, K set to 5, AROW hyper-parameter r set to 1000. Additionally, for the language model feature H3 in Table 4, the Dirichlet smoothing hyper-parameter was set to 10.

6. EXPERIMENTAL SETUP

6.1 Ranking Models

In order to measure the impact incurred by the different feature types, we evaluated several re-ranking models using different combination of features.

baseline: Our baseline ranking model is the $BM25$ scoring function as provided by Lucene.

letor: Adds the other 12 LETOR features to the baseline score (see Table 4).

pos: Adds to the baseline score the entire family of POS related features for the question's title (see Section 5.1.2). This feature family consists of 130 features.

dp: Adds to the baseline score the family of dependency relationship features for the question's title (see Section 5.1.3). This feature family consists of 96 features.

pos+dp: Adds to the baseline score all the syntactic features, *i.e.* both the POS and the dependency relationship features.

all: Adds to the baseline score all the syntactic features and LETOR features.

For every combination of features we trained a separate ranking model using SVMRank, based on the training dataset described in Section 5. We evaluated the different models under two test sets: the first is based on a large-scale clicked-data and the second is based on manual judgments, as detailed below.

6.2 Automatic Evaluation

We conducted a large scale automatic evaluation by sampling 100,000 query document pairs as a test set, conditioned that each target document is found among the top 100 results retrieved for its landing query by the $BM25$ baseline scoring function. Additionally, the queries in the test set do not intersect with the queries in the training set, which was used for learning the model parameters. We note that since most of our queries in our dataset are long (6 words on average, see Section 7.1.2), these are long tail queries and therefore the chances of finding similar queries, in terms of content, in the training and test sets is very slim.

For each query we retrieved the top 100 results from the collection using Lucene with the $BM25$ scoring function. We consider this ranked list of results as our baseline. We then re-ranked the list using each of the tested combinations of the syntactical and statistical features.

6.3 Manual Evaluation

For manual evaluation, we randomly sampled 1,000 queries from our dataset described in Section 3. The queries in the test set do not intersect with the queries in the training set used for learning the weight models. Since clickthrough queries of length 1-2 are scarce in CQA and syntactic features are of no interest in such short queries (see the analysis in Section 7.1.2), we only sampled queries of length 3 words and above in this test-set.

For each query we collected a pool of 15 documents, constructed from the results retrieved for the query by a variety of ranking methods from our collection. Then, professional editors assessed the relevance of each result in the pool on five Likert-scale levels, from non-relevant (1) to highly-relevant (5). We note that the ranking methods we used for collecting the pool of results do not consider the syntactic features proposed in this work.

Next, we used the manually judged dataset to evaluate the proposed re-ranking scheme. As for the automatic evaluation, for each query we retrieved the top 100 results from the collection using Lucene with the $BM25$ scoring function. Then, we re-ranked the list of results using the different tested models.

Model	MRR	R@1	R@3	R@5	R@10
baseline	0.484	0.360	0.541	0.622	0.726
letor	0.507	0.376	0.572	0.657	0.763
dp	0.496	0.365	0.560	0.645	0.753
pos	0.501	0.371	0.565	0.650	0.758
pos+dp	0.500	0.369	0.565	0.652	0.760
all	**0.513**	**0.381**	**0.582**	**0.666**	**0.773**

Table 5: Results for the automatic evaluation. All differences are statistically significant with $p < 0.001$.

7. RESULTS

We next present the results on the different test-sets and provide additional analysis and insights for the usage of syntactic analysis for term re-weighting.

7.1 Automatic Evaluation

For the automatic test-set, we evaluated the quality of the retrieved results of the various ranking models using MRR and Binary-Recall $R@k$ (the relative number of queries with $P@K > 0$) [5]. Table 5 presents the results of the automatic evaluation. We note that all differences between models are statistically significant with $p < 0.001$.

Looking at the table, we first see that LETOR features within a LTR framework improve over the basic Lucene BM25 ranking function. This is well known for general Web queries and medical queries [25, 15, 39], but, to the best of our knowledge, was not shown for CQA-related queries before. In our testset, the improvement of incorporating the LETOR statistical features is 4.8% for MRR, and for R@K ranges from 4.4% (R@1) and up to 5.7% (R@3).

If instead of statistical features we take the syntactic features as input to LTR, we still gain an improvement over the baseline BM25 score, but it does not reach that of the statistical features. For example, MRR is increased by 3.5% with *pos* and by 2.5% with *dp*. Similar results are shown for R@K, *e.g.* R@3 is increased by 4.4% with *pos* and by 3.5% with *dp*. In this experiment, it seems that part-of-speech tags provide more useful information for ranking than syntactic roles, thereby achieving higher results in all ranking measures. In addition, their combination does not provide any additional gain. This result may indicate that Web searchers click on a CQA page mainly because of the appearance of query words in the title, with the intended part-of-speech tags, while not thoroughly verifying their intended syntactic roles.

The main hypothesis of this paper is that syntactic features derived from categorical and syntactical roles of words in the text help retrieval over CQA collections. So far, we showed that while such features convey additional information compared to the baseline, better performance can be achieved using statistical signals instead. It is thus the com-

Feature Name	Relative Importance
nouns	14.4
verbs	7.0
WH-adverbs	6.2
adjectives	3.8
pronouns	3.1
prepositions	3.1
WH-pronouns	1.9
modals	1.3
determiners	0.8
adverbs	0.7

Table 6: Relative importance of top features in the *pos* ranking method (out of 22 features)

Feature Name	relative importance
noun as subject	5.4
direct object	4.9
object of preposition	4.1
adverb as modifier	4.1
auxiliary verb	3.3
adjective as modifier	3.1
preposition	3.1
noun as modifier	2.6
possessive	2.3
determiner	0.9

Table 7: Relative importance of top features in the *dp* ranking method (out of 48 features)

bination of syntactic and statistical features that is most interesting: is some information about the word's relative importance only captured by syntactic features and not by statistical features? Looking at our full *all* model, we see that indeed, syntactic analysis of words provides some complementary information to that of occurrence-based statistics. This model consistently improves over all other models, including *letor*. For example, the improvement in MRR compared to the baseline is 6%, a relative increase of 25% compared to the improvement achieved using only statistical features. The improvement gap is even larger at the top results. Looking at R@1 and R@3, the improvement compared to the baseline is 5.8% and 7.8% respectively, a relative increase of 33% and 30% compared to the *letor* model.

We illustrate the improvement effect of considering the POS and syntactic roles information with two examples from our evaluation set. In the first example, the query is "*american pie like*". The baseline model ranked "*Is college life really **like** in american pie?*" higher than the clicked question "*Who else here doesn't **like** american pie?*". Yet, in the first question the term '*like*' functions as a preposition, while in the second question it functions as a verb. Our model gives higher weight to verbs than to prepositions (see Section 7.1.1) and therefore ranked the clicked question higher.

In our second example, the query is "*does mass change*". The baseline model ranked "*How does density change according to changes in the **mass**?*" higher than the clicked question "*Does the **mass** of an object change as the distance from center of gravity?*". In both titles the term '*mass*' is a noun. Yet, the syntactic role of the term '*mass*' in the first title is *object of preposition*, while in the second title it is the *subject* of the sentence. As our model gives the *subject* role higher weight than to the *object of preposition* (see Section 7.1.1), it swaps the ranking order between the two.

7.1.1 Feature Analysis

We next analyze the behavior of the models that were learned. We start with inspecting the importance of the various features in these models. The top features and their relative importance in the *pos* and *dp* models[3] are presented in Tables 6 and 7 respectively. Looking at POS features, we see that, as expected, nouns, verbs and adjectives, which are the main content indicators, are at the top. Interestingly, joining them as the third most important feature is the part-of-speech *WH-adverbs*, which stands for WH words such as '*how*', '*why*' and '*where*'. These words capture part of the question type, and when specified in the query as well, they are important disambiguators with respect to the type of information requested for entities, events, processes etc. The rest of the features have relatively low weight and indicate word families that do not appear in queries. These are stop words, such as *determiners* and *modals*. Yet, some POS tags, such as *adverbs*, may also refer to infrequent words, such as '*arcanely*' and '*calculatedly*', which would receive high weights under frequency-based term weighting.

Looking at prominent dependency features (Table 7), we see at the top noun syntactic roles, which typically appear more in corresponding queries. These include frequent types, such as *subject*, *direct object* and *object of preposition*. Yet, modifiers are also ranked high, including nouns and adjectives, as expected based on our analysis in Section 4. On the other hand, frequent types that do not convey important information, *e.g. determiners*, and thus typically do not appear in a query, received low weights. Most notably of these types is the sentence *root*, which is not in the top 10 features. This is quite surprising, since the sentence root usually refers to the main predicate. It can be explained by the fact that rather empty main predicates are quite frequent in CQA questions, such as '*get*' in "*where can I get good ski boots?*" or '*think*' in "*do you think that Michael Jackson is the best singer ever?*".

Using the feature analysis we can also exemplify the complementing information between POS tags and syntactic roles. On one side, POS tags refer to all nouns as one category, while noun occurrences are separated into several syntactic roles. On the other side, the syntactic role *adverb as modifier* refers to two POS tags, *WH-adverbs* and *adverbs*, which have significantly different expectancy to appear in a related query, as discussed above.

7.1.2 Query Length Analysis

We further measured the change in performance of the best performing models compared to the baseline with respect to different query lengths. The number of queries of each length in our test set is summarized in Fig. 2, and the MRR results are summarized in Fig. 3.

Fig. 3 draws a clear picture, in which the longer the query is, the more effective syntactic analysis is for document ranking. For short queries, with one, two and even three terms, syntactic analysis does not help to improve the ranking quality. This result echoes the common knowledge in IR for the inadequacy of NLP for Web queries, since short queries typically refer to one noun, an entity or a reference mention (*e.g.* "*David Bowie*" and "*Pluto*"). In such queries no complex re-

[3]We analyze the *pos* and *dp* models and not the combined model *all*, since the weights of features with overlapping information, as is the case of part-of-speech tags and syntactic roles, are not easily interpreted.

Model	NDCG	MAP	P@1	P@3	P@5	P@10	R@1	R@3	R@5	R@10
baseline	0.447	0.270	0.235	0.142	0.108	0.071	0.235	0.356	0.408	0.458
letor	0.454	0.281	0.246	0.144	0.112	0.071	0.246	0.362	0.418	0.469
dp	0.456	0.281	0.249	0.149	0.113	0.072	0.249	0.372	0.420	0.471
pos	0.457	0.282	0.250	0.148	0.115	0.071	0.250	0.370	0.421	0.471
pos+dp	0.459	0.286	0.251	0.150	**0.116**	**0.072**	0.251	**0.378**	**0.428**	**0.474**
all	**0.460**	**0.288**	**0.260**	**0.150**	0.112	0.070	**0.260**	0.374	0.422	0.466

Table 8: Results for the manual evaluation

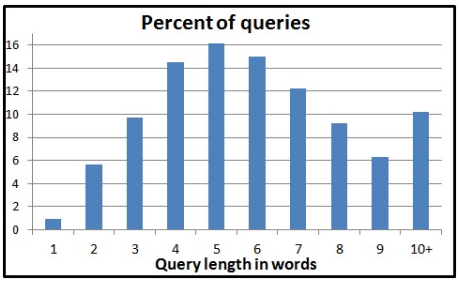

Figure 2: % of CQA-related Web queries of each length in the test set. Label '*10+*' refers to queries of length 10 and above.

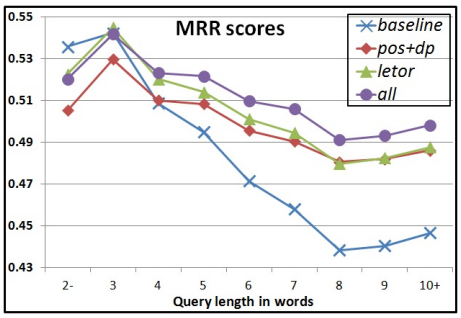

Figure 3: MRR scores of various models, broken down by query lengths. Label '*2-*' refers to queries of length 1 and 2. Label '*10+*' refers to queries of length 10 and above.

lations between words is expressed, and relevant documents are ones that include the entity or reference in different syntactic roles with no specific preference. However, as shown in Fig. 2, such short queries are rather infrequent in CQA retrieval.

The situation is reversed for queries with 4 terms or more, which are frequent in Web search that relates to CQA collections (about 70% of the sampled queries are of length 5 or more). Such queries require relevant documents to reflect a more complex relationship between the query terms, and here syntactic analysis helps in identifying the more likely roles and categories that are dimmed relevant. Specifically, for queries of length 6 and above there is a consistent improvement of 2% in MRR score when adding syntactic features on top of LETOR features. This improvement is statistically significant at $p < 0.001$.

7.2 Manual Evaluation

Under the manual evaluation setting, where gold standard annotations are provided, we evaluated the performance of each tested ranking model using Mean Average Precision (MAP), NDCG, Precision and Binary-Recall at k ($P@k$, $R@k$) [5]. The results of the manual experiment are summarized in Table 8.

Looking at the table, we see the usefulness of the syntactic features in this test-set. While the *letor* model improves over the baseline, it trails behind any model that takes syntactic features into account. For example, NDCG improves by 1.6% with the *letor* model, compared to 2.7% with syntactic features (*pos+dp*). Similarly, MAP is increased by 4% compared to 5.9% for the *letor* and *pos+dp* models respectively. Surprisingly, in this experiment combining all features together (*all* model) has no conclusive improvement over just taking a syntactic features. We note that the differences between *all* and *letor* in all measures and between *pos+dp* and *letor* in {R,P}@3 are statistically significant at $p < 0.05$.

One reason for this difference in model behavior compared to the automatic evaluation may be that in the automatic evaluation, clicked documents were sought. These are documents that were most likely shown to web-searchers as part of the top ten results for the issued queries. If we assume that search engines utilize mainly statistical features in their ranking function, then the retrieved results, from which users chose what to click on, introduce a bias towards statistical features. It is therefore encouraging that incorporating syntactic features contribute to a significant improvement in performance even with this bias. In our manual evaluation, on the other hand, all top results retrieved by the different models were evaluated, therefore reducing this bias.

An open question which our work does not deal with is how the level of title quality affects the effectiveness of our ranking approach. We can reasonably argue that since the title of a CQA page is in fact the focus of all page content in most cases, its deep analysis is extremely important for search over CQA data. Whether other domains with high quality focused titles (e.g. news), or in contrast domains with low quality titles, can similarly benefit from term weighting using syntactic analysis, is an interesting direction for further research.

8. CONCLUSIONS

In this paper we study how term weighting may benefit from syntactic analysis of the documents. Taking as a test-case the task of Web search over Community-based Question Answering collections, we showed that syntactic analysis, such as part-of-speech tagging and dependency parsing, complement statistical-based methods for term weighting. In a large scale analysis over pairs of queries and clicked CQA pages, we showed significant differences, sometimes quite surprising, between the chances of page title terms to appear in the corresponding query given their part-of-speech tag or their syntactic role in a dependency parse tree. Following this analysis, we proposed a novel term weighting model that incorporates both statistical informa-

tion and syntactic information of the term, learning the relative importance of each signal using LTR on a collection of queries/clicked-pages pairs. We conducted a manual evaluation and a large-scale automatic evaluation to test our hypothesis, and the results of both experiments indicate that term weighting with syntactic information significantly improves retrieval quality.

We see this work as a first step towards showing the benefit of syntactic analysis for advanced term weighting techniques. In future work, we would like to investigate the effect of our term weighting approach in domains different than CQA, such as news and blogs. In addition, we would like to develop syntactic analysis techniques that are specific for queries in order to see if they could provide additional leverage for IR. Finally, we are interested in testing the contribution of semantic analysis, such as semantic role labeling, for the task of term weighting.

9. REFERENCES

[1] J. Allan and H. Raghavan. Using part-of-speech patterns to reduce query ambiguity. In *Proceedings of SIGIR*, pages 307–314. ACM, 2002.

[2] G. Amati, V. Rijsbergen, and C. Joost. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4), Oct. 2002.

[3] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proceedings of SIGIR*, pages 315–322. ACM, 2009.

[4] R. Baeza-Yates. Challenges in the interaction of information retrieval and natural language processing. In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 2945, pages 445–456. Springer Berlin Heidelberg, 2004.

[5] R. A. Baeza-yates and B. A. Ribeiro-neto. *Modern Information Retrieval, Second Edition*. Addison-Wesley Professional, 2011.

[6] C. Barr, R. Jones, and M. Regelson. The linguistic structure of English Web-search Queries. In *Proceedings of EMNLP*, pages 1021–1030. ACL, 2008.

[7] C. J. Burges, R. Ragno, and Q. V. Le. Learning to rank with nonsmooth cost functions. In *NIPS*, volume 6, pages 193–200, 2006.

[8] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community QA. In *IJCNLP*, volume 11, pages 273–281, 2011.

[9] X. Cao, G. Cong, B. Cui, C. S. Jensen, and C. Zhang. The use of categorization information in language models for question retrieval. In *Proceedings of CIKM*, pages 265–274. ACM, 2009.

[10] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Proceedings of SIGIR*, pages 186–193. ACM, 2006.

[11] M. Collins, L. Ramshaw, J. Hajič, and C. Tillmann. A statistical parser for Czech. In *Proceedings of ACL*, pages 505–512. ACL, 1999.

[12] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. *Machine Learning*, 91(2):155–187, 2013.

[13] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua. Question answering passage retrieval using dependency relations. In *Proceedings of SIGIR*, pages 400–407. ACM, 2005.

[14] M.-C. De Marneffe, B. MacCartney, C. D. Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[15] P. Donmez, K. M. Svore, and C. J. Burges. On the local optimality of lambdarank. In *Proceedings of SIGIR*, pages 460–467. ACM, 2009.

[16] H. Duan, Y. Cao, C.-Y. Lin, and Y. Yu. Searching questions by identifying question topic and question focus. In *Proceedings of ACL*, pages 156–164, 2008.

[17] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[18] J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proceedings of SIGIR*, pages 170–177. ACM, 2004.

[19] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of CIKM*, pages 84–90. ACM, 2005.

[20] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of ACL*, pages 423–430. ACL, 2003.

[21] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of SIGIR '01*, pages 120–127. ACM, 2001.

[22] C.-J. Lee, R.-C. Chen, S.-H. Kao, and P.-J. Cheng. A term dependency-based approach for query terms ranking. In *Proceedings of CIKM*, pages 1267–1276. ACM, 2009.

[23] Q. Liu, E. Agichtein, G. Dror, E. Gabrilovich, Y. Maarek, D. Pelleg, and I. Szpektor. Predicting web searcher satisfaction with existing community-based answers. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 415–424. ACM, 2011.

[24] Q. Liu, E. Agichtein, G. Dror, Y. Maarek, and I. Szpektor. When web search fails, searchers become askers: Understanding the transition. In *Proceedings of SIGIR*, pages 801–810. ACM, 2012.

[25] T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, pages 3–10, 2007.

[26] Y. Lu, F. Peng, G. Mishne, X. Wei, and B. Dumoulin. Improving web search relevance with semantic features. In *Proceedings of EMNLP*, pages 648–657. ACL, 2009.

[27] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[28] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of SIGIR*, pages 472–479. ACM, 2005.

[29] J. W. Murdock, J. Fan, A. Lally, H. Shima, and B. Boguraev. Textual evidence gathering and analysis. *IBM Journal of Research and Development*, 56(3.4):8–1, 2012.

[30] J. H. Park and W. B. Croft. Query term ranking based on dependency parsing of verbose queries. In *Proceedings of SIGIR*, pages 829–830. ACM, 2010.

[31] J. H. Park, W. B. Croft, and D. A. Smith. A quasi-synchronous dependence model for information retrieval. In *Proceedings of CIKM*, pages 17–26. ACM, 2011.

[32] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, Apr. 2009.

[33] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.

[34] C. Shah and W. B. Croft. Evaluating high accuracy retrieval techniques. In *Proceedings of SIGIR*, pages 2–9. ACM, 2004.

[35] A. F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In *Natural language information retrieval*, pages 99–111. Springer, 1999.

[36] E. M. Voorhees. Natural language processing and information retrieval. In *Information Extraction: Towards Scalable, Adaptable Systems*, pages 32–48. Springer-Verlag, 1999.

[37] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based QA services. In *Proceedings of SIGIR*, pages 187–194. ACM, 2009.

[38] H. Wu, W. Wu, M. Zhou, E. Chen, L. Duan, and H.-Y. Shum. Improving search relevance for short queries in community question answering. In *Proceedings of WSDM*, pages 43–52. ACM, 2014.

[39] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of IMCL*, pages 1192–1199. ACM, 2008.

[40] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *Proceedings of SIGIR*, pages 475–482. ACM, 2008.

[41] W. Zhang, Z. Ming, Y. Zhang, L. Nie, T. Liu, and T.-S. Chua. The use of dependency relation graph to enhance the term weighting in question retrieval. In *Proceedings of Coling*, pages 3105–3120, 2012.