Joint Annotation of Search Queries

Michael Bendersky
Dept. of Computer Science
University of Massachusetts
Amherst, MA
bemike@cs.umass.edu

W. Bruce Croft
Dept. of Computer Science
University of Massachusetts
Amherst, MA
croft@cs.umass.edu

David A. Smith
Dept. of Computer Science
University of Massachusetts
Amherst, MA
dasmith@cs.umass.edu

Abstract

Marking up search queries with linguistic annotations such as part-of-speech tags, capitalization, and segmentation, is an important part of query processing and understanding in information retrieval systems. Due to their brevity and idiosyncratic structure, search queries pose a challenge to existing NLP tools. To address this challenge, we propose a probabilistic approach for performing joint query annotation. First, we derive a robust set of unsupervised independent annotations, using queries and pseudo-relevance feedback. Then, we stack additional classifiers on the independent annotations, and exploit the dependencies between them to further improve the accuracy, even with a very limited amount of available training data. We evaluate our method using a range of queries extracted from a web search log. Experimental results verify the effectiveness of our approach for both short keyword queries, and verbose natural language queries.

1 Introduction

Automatic mark-up of textual documents with linguistic annotations such as part-of-speech tags, sentence constituents, named entities, or semantic roles is a common practice in natural language processing (NLP). It is, however, much less common in information retrieval (IR) applications. Accordingly, in this paper, we focus on annotating search queries submitted by the users to a search engine.

There are several key differences between user queries and the documents used in NLP (e.g., news articles or web pages). As previous research shows, these differences severely limit the applicability of standard NLP techniques for annotating queries and require development of novel annotation approaches for query corpora (Bergsma and Wang, 2007; Barr et al., 2008; Lu et al., 2009; Bendersky et al., 2010; Li, 2010).

The most salient difference between queries and documents is their length. Most search queries are very short, and even longer queries are usually shorter than the average written sentence. Due to their brevity, queries often cannot be divided into sub-parts, and do not provide enough context for accurate annotations to be made using the standard NLP tools such as taggers, parsers or chunkers, which are trained on more syntactically coherent textual units.

A recent analysis of web query logs by Bendersky and Croft (2009) shows, however, that despite their brevity, queries are grammatically diverse. Some queries are keyword concatenations, some are semi-complete verbal phrases and some are wh-questions. It is essential for the search engine to correctly annotate the query structure, and the quality of these query annotations has been shown to be a crucial first step towards the development of reliable and robust query processing, representation and understanding algorithms (Barr et al., 2008; Guo et al., 2008; Guo et al., 2009; Manshadi and Li, 2009; Li, 2010).

However, in current query annotation systems, even sentence-like queries are often hard to parse and annotate, as they are prone to contain misspellings and idiosyncratic grammatical structures.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 102–111,
Portland, Oregon, June 19-24, 2011.

(a)

Term	CAP	TAG	SEG
who	L	X	B
won	L	V	I
the	L	X	B
2004	L	X	B
kentucky	C	N	B
derby	C	N	I

(b)

Term	CAP	TAG	SEG
kindred	C	N	B
where	C	X	B
would	C	X	I
i	C	X	I
be	C	V	I

(c)

Term	CAP	TAG	SEG
shih	C	N	B
tzu	C	N	I
health	L	N	B
problems	L	N	I

Figure 1: **E**xamples of a mark-up scheme for annotating capitalization (L – lowercase, C – otherwise), POS tags (N – noun, V – verb, X – otherwise) and segmentation (B/I – beginning of/inside the chunk).

They also tend to lack prepositions, proper punctuation, or capitalization, since users (often correctly) assume that these features are disregarded by the retrieval system.

In this paper, we propose a novel joint query annotation method to improve the effectiveness of existing query annotations, especially for longer, more complex search queries. Most existing research focuses on using a single type of annotation for information retrieval such as subject-verb-object dependencies (Balasubramanian and Allan, 2009), named-entity recognition (Guo et al., 2009), phrase chunking (Guo et al., 2008), or semantic labeling (Li, 2010).

In contrast, the main focus of this work is on developing a unified approach for performing reliable annotations of different types. To this end, we propose a probabilistic method for performing a *joint query annotation*. This method allows us to exploit the dependency between different unsupervised annotations to further improve the accuracy of the entire set of annotations. For instance, our method can leverage the information about estimated parts-of-speech tags and capitalization of query terms to improve the accuracy of query segmentation.

We empirically evaluate the joint query annotation method on a range of query types. Instead of just focusing our attention on keyword queries, as is often done in previous work (Barr et al., 2008; Bergsma and Wang, 2007; Tan and Peng, 2008; Guo et al., 2008), we also explore the performance of our annotations with more complex natural language search queries such as verbal phrases and wh-questions, which often pose a challenge for IR applications (Bendersky et al., 2010; Kumaran and Allan, 2007; Kumaran and Carvalho, 2009; Lease, 2007). We show that even with a very limited amount of training data, our joint annotation method significantly outperforms annotations that were done independently for these queries.

The rest of the paper is organized as follows. In Section 2 we demonstrate several examples of annotated search queries. Then, in Section 3, we introduce our joint query annotation method. In Section 4 we describe two types of independent query annotations that are used as input for the joint query annotation. Section 5 details the related work and Section 6 presents the experimental results. We draw the conclusions from our work in Section 7.

2 Query Annotation Example

To demonstrate a possible implementation of linguistic annotation for search queries, Figure 1 presents a simple mark-up scheme, exemplified using three web search queries (as they appear in a search log): (a) *who won the 2004 kentucky derby*, (b) *kindred where would i be*, and (c) *shih tzu health problems*. In this scheme, each query is marked-up using three annotations: capitalization, POS tags, and segmentation indicators.

Note that all the query terms are non-capitalized, and no punctuation is provided by the user, which complicates the query annotation process. While the simple annotation described in Figure 1 can be done with a very high accuracy for standard document corpora, both previous work (Barr et al., 2008; Bergsma and Wang, 2007; Jones and Fain, 2003) and the experimental results in this paper indicate that it is challenging to perform well on queries.

The queries in Figure 1 illustrate this point. Query (a) in Figure 1 is a wh-question, and it contains

a capitalized concept (*"Kentucky Derby"*), a single verb, and four segments. Query (b) is a combination of an artist name and a song title and should be interpreted as *Kindred — "Where Would I Be"*. Query (c) is a concatenation of two short noun phrases: *"Shih Tzu"* and *"health problems"*.

3 Joint Query Annotation

Given a search query Q, which consists of a sequence of terms $(q_1, \ldots, q_n)$, our goal is to annotate it with an appropriate set of linguistic structures $\mathcal{Z}_Q$. In this work, we assume that the set $\mathcal{Z}_Q$ consists of *shallow* sequence annotations $\mathbf{z}_Q$, each of which takes the form

$$\mathbf{z}_Q = (\zeta_1, \ldots, \zeta_n).$$

In other words, each symbol $\zeta_i \in \mathbf{z}_Q$ annotates a single query term.

Many query annotations that are useful for IR can be represented using this simple form, including capitalization, POS tagging, phrase chunking, named entity recognition, and stopword indicators, to name just a few. For instance, Figure 1 demonstrates an example of a set of annotations $\mathcal{Z}_Q$. In this example,

$$\mathcal{Z}_Q = \{\mathbf{CAP}, \mathbf{TAG}, \mathbf{SEG}\}.$$

Most previous work on query annotation makes the independence assumption — every annotation $\mathbf{z}_Q \in \mathcal{Z}_Q$ is done separately from the others. That is, it is assumed that the optimal linguistic annotation $\mathbf{z}_Q^{*(I)}$ is the annotation that has the highest probability given the query Q, regardless of the other annotations in the set $\mathcal{Z}_Q$. Formally,

$$\mathbf{z}_Q^{*(I)} = \operatorname*{argmax}_{\mathbf{z}_Q} p(\mathbf{z}_Q|Q) \qquad (1)$$

The main shortcoming of this approach is in the assumption that the linguistic annotations in the set $\mathcal{Z}_Q$ are independent. In practice, there are dependencies between the different annotations, and they can be leveraged to derive a better estimate of the entire set of annotations.

For instance, imagine that we need to perform two annotations: capitalization and POS tagging. Knowing that a query term is capitalized, we are more likely to decide that it is a proper noun. Vice versa, knowing that it is a preposition will reduce its probability of being capitalized. We would like to capture this intuition in the annotation process.

To address the problem of joint query annotation, we first assume that we have an initial set of annotations $\mathcal{Z}_Q^{*(I)}$, which were performed for query Q independently of one another (we will show an example of how to derive such a set in Section 4). Given the initial set $\mathcal{Z}_Q^{*(I)}$, we are interested in obtaining an annotation set $\mathcal{Z}_Q^{*(J)}$, which jointly optimizes the probability of *all* the annotations, i.e.

$$\mathcal{Z}_Q^{*(J)} = \operatorname*{argmax}_{\mathcal{Z}_Q} p(\mathcal{Z}_Q|\mathcal{Z}_Q^{*(I)}).$$

If the initial set of estimations is reasonably accurate, we can make the assumption that the annotations in the set $\mathcal{Z}_Q^{*(J)}$ are independent given the initial estimates $\mathcal{Z}_Q^{*(I)}$, allowing us to separately optimize the probability of each annotation $\mathbf{z}_Q^{*(J)} \in \mathcal{Z}_Q^{*(J)}$:

$$\mathbf{z}_Q^{*(J)} = \operatorname*{argmax}_{\mathbf{z}_Q} p(\mathbf{z}_Q|\mathcal{Z}_Q^{*(I)}). \qquad (2)$$

From Eq. 2, it is evident that the joint annotation task becomes that of finding some optimal unobserved sequence (annotation $\mathbf{z}_Q^{*(J)}$), given the observed sequences (independent annotation set $\mathcal{Z}_Q^{*(I)}$).

Accordingly, we can directly use a supervised sequential probabilistic model such as CRF (Lafferty et al., 2001) to find the optimal $\mathbf{z}_Q^{*(J)}$. In this CRF model, the optimal annotation $\mathbf{z}_Q^{*(J)}$ is the *label* we are trying to predict, and the set of independent annotations $\mathcal{Z}_Q^{*(I)}$ is used as the basis for the *features* used for prediction. Figure 2 outlines the algorithm for performing the joint query annotation.

As input, the algorithm receives a training set of queries and their ground truth annotations. It then produces a set of independent annotation estimates, which are jointly used, together with the ground truth annotations, to learn a CRF model for each annotation type. Finally, these CRF models are used to predict annotations on a held-out set of queries, which are the output of the algorithm.

Input:	$\mathbf{Q}_t$ — training set of queries. $\mathcal{Z}_{\mathbf{Q}_t}$ — ground truth annotations for the training set of queries. $\mathbf{Q}_h$ — held-out set of queries.
(1)	Obtain a set of independent annotation estimates $\mathcal{Z}^{*(I)}_{\mathbf{Q}_t}$
(2)	Initialize $\mathcal{Z}^{*(J)}_{\mathbf{Q}_t} \leftarrow \emptyset$
(3)	*for each* $\mathbf{z}^{*(I)}_{\mathbf{Q}_t} \in \mathcal{Z}^{*(I)}_{\mathbf{Q}_t}$:
(4)	$\mathcal{Z}'_{\mathbf{Q}_t} \leftarrow \mathcal{Z}^{*(I)}_{\mathbf{Q}_t} \setminus \mathbf{z}^{*(I)}_{\mathbf{Q}_t}$
(5)	Train a CRF model $\mathcal{CRF}(\mathbf{z}_{\mathbf{Q}_t})$ using $\mathbf{z}_{\mathbf{Q}_t}$ as a *label* and $\mathcal{Z}'_{\mathbf{Q}_t}$ as *features*.
(6)	Predict annotation $\mathbf{z}^{*(J)}_{\mathbf{Q}_h}$, using $\mathcal{CRF}(\mathbf{z}_{\mathbf{Q}_t})$.
(7)	$\mathcal{Z}^{*(J)}_{\mathbf{Q}_h} \leftarrow \mathcal{Z}^{*(J)}_{\mathbf{Q}_h} \cup \mathbf{z}^{*(J)}_{\mathbf{Q}_h}$.
Output:	$\mathcal{Z}^{*(J)}_{\mathbf{Q}_h}$ — predicted annotations for the held-out set of queries.

Figure 2: **A**lgorithm for performing joint query annotation.

Note that this formulation of joint query annotation can be viewed as a *stacked classification*, in which a second, more effective, classifier is trained using the labels inferred by the first classifier as features. Stacked classifiers were recently shown to be an efficient and effective strategy for structured classification in NLP (Nivre and McDonald, 2008; Martins et al., 2008).

4 Independent Query Annotations

While the joint annotation method proposed in Section 3 is general enough to be applied to any set of independent query annotations, in this work we focus on two previously proposed independent annotation methods based on either the query itself, or the top sentences retrieved in response to the query (Bendersky et al., 2010). The main benefits of these two annotation methods are that they can be easily implemented using standard software tools, do not require any labeled data, and provide reasonable annotation accuracy. Next, we briefly describe these two independent annotation methods.

4.1 Query-based estimation

The most straightforward way to estimate the conditional probabilities in Eq. 1 is using the query itself. To make the estimation feasible, Bendersky et al. (2010) take a *bag-of-words* approach, and assume independence between both the query terms and the corresponding annotation symbols. Thus, the indepentent annotations in Eq. 1 are given by

$$\mathbf{z}^{*(QRY)}_{Q} = \underset{(\zeta_1,\ldots,\zeta_n)}{\operatorname{argmax}} \prod_{i\in(1,\ldots,n)} p(\zeta_i|q_i). \qquad (3)$$

Following Bendersky et al. (2010) we use a large n-gram corpus (Brants and Franz, 2006) to estimate $p(\zeta_i|q_i)$ for annotating the query with capitalization and segmentation mark-up, and a standard POS tagger[1] for part-of-speech tagging of the query.

4.2 PRF-based estimation

Given a short, often ungrammatical query, it is hard to accurately estimate the conditional probability in Eq. 1 using the query terms alone. For instance, a keyword query *hawaiian falls*, which refers to a location, is inaccurately interpreted by a standard POS tagger as a *noun-verb* pair. On the other hand, given a sentence from a corpus that is relevant to the query such as *"Hawaiian Falls is a family-friendly waterpark"*, the word "falls" is correctly identified by a standard POS tagger as a proper noun.

Accordingly, the document corpus can be bootstrapped in order to better estimate the query annotation. To this end, Bendersky et al. (2010) employ the *pseudo-relevance feedback* (PRF) — a method that has a long record of success in IR for tasks such as query expansion (Buckley, 1995; Lavrenko and Croft, 2001).

In the most general form, given the set of *all* retrievable sentences r in the corpus $\mathcal{C}$ one can derive

$$p(\mathbf{z}_Q|Q) = \sum_{r\in\mathcal{C}} p(\mathbf{z}_Q|r)p(r|Q).$$

Since for most sentences the conditional probability of relevance to the query $p(r|Q)$ is vanishingly small, the above can be closely approximated

[1] `http://crftagger.sourceforge.net/`

by considering only a set of sentences R, retrieved at top-k positions in response to the query Q. This yields

$$p(\mathbf{z}_Q|Q) \approx \sum_{r \in R} p(\mathbf{z}_Q|r)p(r|Q).$$

Intuitively, the equation above models the query as a mixture of top-k retrieved sentences, where each sentence is weighted by its relevance to the query. Furthermore, to make the estimation of the conditional probability $p(\mathbf{z}_Q|r)$ feasible, it is assumed that the symbols ζ_i in the annotation sequence are independent, given a sentence r. Note that this assumption differs from the independence assumption in Eq. 3, since here the annotation symbols are *not independent* given the query Q.

Accordingly, the PRF-based estimate for independent annotations in Eq. 1 is

$$\mathbf{z}_Q^{*(PRF)} = \underset{(\zeta_1,\ldots,\zeta_n)}{\operatorname{argmax}} \sum_{r \in R} \prod_{i \in (1,\ldots,n)} p(\zeta_i|r)p(r|Q). \quad (4)$$

Following Bendersky et al. (2010), an estimate of $p(\zeta_i|r)$ is a smoothed estimator that combines the information from the retrieved sentence r with the information about unigrams (for capitalization and POS tagging) and bigrams (for segmentation) from a large n-gram corpus (Brants and Franz, 2006).

5 Related Work

In recent years, linguistic annotation of search queries has been receiving increasing attention as an important step toward better query processing and understanding. The literature on query annotation includes query segmentation (Bergsma and Wang, 2007; Jones et al., 2006; Guo et al., 2008; Hagen et al., 2010; Hagen et al., 2011; Tan and Peng, 2008), part-of-speech and semantic tagging (Barr et al., 2008; Manshadi and Li, 2009; Li, 2010), named-entity recognition (Guo et al., 2009; Lu et al., 2009; Shen et al., 2008; Paşca, 2007), abbreviation disambiguation (Wei et al., 2008) and stopword detection (Lo et al., 2005; Jones and Fain, 2003).

Most of the previous work on query annotation focuses on performing a particular annotation task (e.g., segmentation or POS tagging) in isolation. However, these annotations are often related, and thus we take a joint annotation approach, which combines several independent annotations to improve the overall annotation accuracy. A similar approach was recently proposed by Guo et al. (2008). There are several key differences, however, between the work presented here and their work.

First, Guo et al. (2008) focus on *query refinement* (spelling corrections, word splitting, etc.) of short keyword queries. Instead, we are interested in *annotation* of queries of different types, including verbose natural language queries. While there is an overlap between query refinement and annotation, the focus of the latter is on providing linguistic information about existing queries (after initial refinement has been performed). Such information is especially important for more verbose and gramatically complex queries. In addition, while all the methods proposed by Guo et al. (2008) require large amounts of training data (thousands of training examples), our joint annotation method can be effectively trained with a minimal human labeling effort (several hundred training examples).

An additional research area which is relevant to this paper is the work on joint structure modeling (Finkel and Manning, 2009; Toutanova et al., 2008) and stacked classification (Nivre and McDonald, 2008; Martins et al., 2008) in natural language processing. These approaches have been shown to be successful for tasks such as parsing and named entity recognition in newswire data (Finkel and Manning, 2009) or semantic role labeling in the Penn Treebank and Brown corpus (Toutanova et al., 2008). Similarly to this work in NLP, we demonstrate that a joint approach for modeling the linguistic query structure can also be beneficial for IR applications.

6 Experiments

6.1 Experimental Setup

For evaluating the performance of our query annotation methods, we use a random sample of 250 queries[2] from a search log. This sample is manually labeled with three annotations: *capitalization*, *POS tags*, and *segmentation*, according to the description of these annotations in Figure 1. In this set of 250 queries, there are 93 questions, 96 phrases contain-

[2]The annotations are available at `http://ciir.cs.umass.edu/~bemike/data.html`

CAP		
	F1 *(% impr)*	MQA *(% impr)*
i-QRY	0.641 *(-/-)*	0.779 *(-/-)*
i-PRF	0.711^{*}*(+10.9/-)*	0.811^{*}*(+4.1/-)*
j-QRY	$0.620_{\dagger}$*(-3.3/-12.8)*	0.805^{*}*(+3.3/-0.7)*
j-PRF	**0.718****(+12.0/+0.9)*	**0.840**$^{*}_{\dagger}$*(+7.8/+3.6)*
TAG		
	Acc. *(% impr)*	MQA *(% impr)*
i-QRY	0.893 *(-/-)*	0.878 *(-/-)*
i-PRF	0.916^{*}*(+2.6/-)*	0.914^{*}*(+4.1/-)*
j-QRY	0.913^{*}*(+2.2/-0.3)*	0.912^{*}*(+3.9/-0.2)*
j-PRF	**0.924****(+3.5/+0.9)*	**0.922****(+5.0/+0.9)*
SEG		
	F1 *(% impr)*	MQA *(% impr)*
i-QRY	0.694 *(-/-)*	0.672 *(-/-)*
i-PRF	0.753^{*}*(+8.5/-)*	0.710^{*}*(+5.7/-)*
j-QRY	$0.817^{*}_{\dagger}$*(+17.7/+8.5)*	**0.803**$^{*}_{\dagger}$*(+19.5/+13.1)*
j-PRF	**0.819**$^{*}_{\dagger}$*(+18.0/+8.8)*	**0.803**$^{*}_{\dagger}$*(+19.5/+13.1)*

Table 1: **S**ummary of query annotation performance for capitalization (CAP), POS tagging (TAG) and segmentation. Numbers in parentheses indicate % of improvement over the *i-QRY* and *i-PRF* baselines, respectively. Best result per measure and annotation is boldfaced. * and $_{\dagger}$ denote statistically significant differences with *i-QRY* and *i-PRF*, respectively.

ing a verb, and 61 short keyword queries (Figure 1 contains a single example of each of these types).

In order to test the effectiveness of the joint query annotation, we compare four methods. In the first two methods, *i-QRY* and *i-PRF* the three annotations are done independently. Method *i-QRY* is based on $\mathbf{z}_Q^{*(QRY)}$ estimator (Eq. 3). Method *i-PRF* is based on the $\mathbf{z}_Q^{*(PRF)}$ estimator (Eq. 4).

The next two methods, *j-QRY* and *j-PRF*, are joint annotation methods, which perform a joint optimization over the entire set of annotations, as described in the algorithm in Figure 2. *j-QRY* and *j-PRF* differ in their choice of the initial independent annotation set $\mathcal{Z}_Q^{*(I)}$ in line (1) of the algorithm (see Figure 2). *j-QRY* uses only the annotations performed by *i-QRY* (3 initial independent annotation estimates), while *j-PRF* combines the annotations performed by *i-QRY* with the annotations performed by *i-PRF* (6 initial annotation estimates). The CRF model training in line (6) of the algorithm is implemented using CRF++ toolkit[3].

[3] `http://crfpp.sourceforge.net/`

The performance of the joint annotation methods is estimated using a 10-fold cross-validation. In order to test the statistical significance of improvements attained by the proposed methods we use a two-sided Fisher's randomization test with 20,000 permutations. Results with p-value < 0.05 are considered statistically significant.

For reporting the performance of our methods we use two measures. The first measure is classification-oriented — treating the annotation decision for each query term as a classification. In case of capitalization and segmentation annotations these decisions are binary and we compute the precision and recall metrics, and report F1 — their harmonic mean. In case of POS tagging, the decisions are ternary, and hence we report the classification accuracy.

We also report an additional, IR-oriented performance measure. As is typical in IR, we propose measuring the performance of the annotation methods on a per-query basis, to verify that the methods have uniform impact across queries. Accordingly, we report the *mean of classification accuracies per query* (MQA). Formally, MQA is computed as

$$\frac{\sum_{i=1}^{N} acc_{Q_i}}{N},$$

where acc_{Q_i} is the classification accuracy for query Q_i, and N is the number of queries.

The empirical evaluation is conducted as follows. In Section 6.2, we discuss the general performance of the four annotation techniques, and compare the effectiveness of independent and joint annotations. In Section 6.3, we analyze the performance of the independent and joint annotation methods by query type. In Section 6.4, we compare the difficulty of performing query annotations for different query types. Finally, in Section 6.5, we compare the effectiveness of the proposed joint annotation for query segmentation with the existing query segmentation methods.

6.2 General Evaluation

Table 1 shows the summary of the performance of the two independent and two joint annotation methods for the entire set of 250 queries. For independent methods, we see that *i-PRF* outperforms *i-QRY* for

CAP	*Verbal Phrases*		*Questions*		*Keywords*	
	F1	MQA	F1	MQA	F1	MQA
i-PRF	**0.750**	**0.862**	0.590	0.839	0.784	0.687
j-PRF	0.687*(-8.4%)	0.839*(-2.7%)	**0.671***(+13.7%)	**0.913***(+8.8%)	**0.814** (+3.8%)	**0.732*** (+6.6%)

TAG	*Verbal Phrases*		*Questions*		*Keywords*	
	Acc.	MQA	Acc.	MQA	Acc.	MQA
i-PRF	**0.908**	**0.908**	0.932	0.935	0.880	0.890
j-PRF	0.904 (-0.4%)	0.906 (-0.2%)	**0.951*** (+2.1%)	**0.953*** (+1.9%)	**0.893** (+1.5%)	**0.900** (+1.1%)

SEG	*Verbal Phrases*		*Questions*		*Keywords*	
	F1	MQA	F1	MQA	F1	MQA
i-PRF	0.751	0.700	0.740	0.700	0.816	0.747
j-PRF	**0.772** (+2.8%)	**0.742***(+6.0%)	**0.858***(+15.9%)	**0.838***(+19.7%)	**0.844** (+3.4%)	**0.853***(+14.2%)

Table 2: **D**etailed analysis of the query annotation performance for capitalization (CAP), POS tagging (TAG) and segmentation by query type. Numbers in parentheses indicate % of improvement over the *i-PRF* baseline. Best result per measure and annotation is boldfaced. * denotes statistically significant differences with *i-PRF*.

all annotation types, using both performance measures.

In Table 1, we can also observe that the joint annotation methods are, in all cases, better than the corresponding independent ones. The highest improvements are attained by *j-PRF*, which always demonstrates the best performance both in terms of F1 and MQA. These results attest to both the importance of doing a joint optimization over the entire set of annotations and to the robustness of the initial annotations done by the *i-PRF* method. In all but one case, the *j-PRF* method, which uses these annotations as features, outperforms the *j-QRY* method that only uses the annotation done by *i-QRY*.

The most significant improvements as a result of joint annotation are observed for the segmentation task. In this task, joint annotation achieves close to 20% improvement in MQA over the *i-QRY* method, and more than 10% improvement in MQA over the *i-PRF* method. These improvements indicate that the segmentation decisions are strongly guided by capitalization and POS tagging. We also note that, in case of segmentation, the differences in performance between the two joint annotation methods, *j-QRY* and *j-PRF*, are not significant, indicating that the context of additional annotations in *j-QRY* makes up for the lack of more robust pseudo-relevance feedback based features.

We also note that the *lowest* performance improvement as a result of joint annotation is evidenced for POS tagging. The improvements of joint annotation method *j-PRF* over the *i-PRF* method are less than 1%, and are not statistically significant. This is not surprising, since the standard POS taggers often already use bigrams and capitalization at training time, and do not acquire much additional information from other annotations.

6.3 Evaluation by Query Type

Table 2 presents a detailed analysis of the performance of the best independent (*i-PRF*) and joint (*j-PRF*) annotation methods by the three query types used for evaluation: verbal phrases, questions and keyword queries. From the analysis in Table 2, we note that the contribution of joint annotation varies significantly across query types. For instance, using *j-PRF* always leads to statistically significant improvements over the *i-PRF* baseline for questions. On the other hand, it is either statistically indistinguishable, or even significantly worse (in the case of capitalization) than the *i-PRF* baseline for the verbal phrases.

Table 2 also demonstrates that joint annotation has a different impact on various annotations for the *same* query type. For instance, *j-PRF* has a significant positive effect on capitalization and segmentation for keyword queries, but only marginally improves the POS tagging. Similarly, for the verbal phrases, *j-PRF* has a significant positive effect only for the segmentation annotation.

These variances in the performance of the *j-PRF* method point to the differences in the structure be-

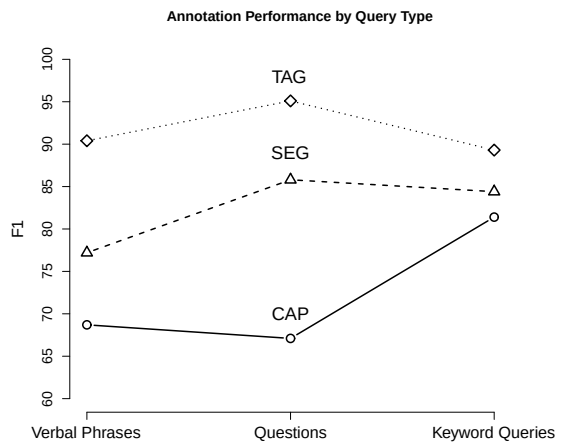

Figure 3: **C**omparative performance (in terms of F1 for capitalization and segmentation and accuracy for POS tagging) of the *j-PRF* method on the three query types.

tween the query types. While dependence between the annotations plays an important role for question and keyword queries, which often share a common grammatical structure, this dependence is less useful for verbal phrases, which have a more diverse linguistic structure. Accordingly, a more in-depth investigation of the linguistic structure of the verbal phrase queries is an interesting direction for future work.

6.4 Annotation Difficulty

Recall that in our experiments, out of the overall 250 annotated queries, there are 96 verbal phrases, 93 questions and 61 keyword queries. Figure 3 shows a plot that contrasts the relative performance for these three query types of our best-performing joint annotation method, *j-PRF*, on capitalization, POS tagging and segmentation annotation tasks. Next, we analyze the performance profiles for the annotation tasks shown in Figure 3.

For the capitalization task, the performance of *j-PRF* on verbal phrases and questions is similar, with the difference below 3%. The performance for keyword queries is much higher — with improvement over 20% compared to either of the other two types. We attribute this increase to both a larger number of positive examples in the short keyword queries (a higher percentage of terms in keyword queries is capitalized) and their simpler syntactic structure (adjacent terms in these queries are likely to have the same case).

SEG	F1	MQA
SEG-1	0.768	0.754
SEG-2	**0.824***	0.787*
j-PRF	0.819* *(+6.7%/-0.6%)*	**0.803*** *(+6.5%/+2.1%)*

Table 3: **C**omparison of the segmentation performance of the *j-PRF* method to two state-of-the-art segmentation methods. Numbers in parentheses indicate % of improvement over the *SEG-1* and *SEG-2* baselines respectively. Best result per measure and annotation is boldfaced. * denotes statistically significant differences with *SEG-1*.

For the segmentation task, the performance is at its best for the question and keyword queries, and at its worst (with a drop of 11%) for the verbal phrases. We hypothesize that this is due to the fact that question queries and keyword queries tend to have repetitive structures, while the grammatical structure for verbose queries is much more diverse.

For the tagging task, the performance profile is reversed, compared to the other two tasks — the performance is at its worst for keyword queries, since their grammatical structure significantly differs from the grammatical structure of sentences in news articles, on which the POS tagger is trained. For question queries the performance is the best (6% increase over the keyword queries), since they resemble sentences encountered in traditional corpora.

It is important to note that the results reported in Figure 3 are based on training the joint annotation model on *all* available queries with 10-fold cross-validation. We might get different profiles if a separate annotation model was trained for each query type. In our case, however, the number of queries from each type is not sufficient to train a reliable model. We leave the investigation of separate training of joint annotation models by query type to future work.

6.5 Additional Comparisons

In order to further evaluate the proposed joint annotation method, *j-PRF*, in this section we compare its performance to other query annotation methods previously reported in the literature. Unfortunately, there is not much published work on query capitalization and query POS tagging that goes beyond the simple query-based methods described in Sec-

tion 4.1. The published work on the more advanced methods usually requires access to large amounts of proprietary user data such as query logs and clicks (Barr et al., 2008; Guo et al., 2008; Guo et al., 2009).

Therefore, in this section we focus on recent work on query segmentation (Bergsma and Wang, 2007; Hagen et al., 2010). We compare the segmentation effectiveness of our best performing method, *j-PRF*, to that of these query segmentation methods.

The first method, *SEG-1*, was first proposed by Hagen et al. (2010). It is currently the most effective publicly disclosed *unsupervised* query segmentation method. *SEG-1* method requires an access to a large web n-gram corpus (Brants and Franz, 2006). The optimal segmentation for query Q, S^*_Q, is then obtained using

$$S^*_Q = \underset{S \in \mathcal{S}_Q}{\operatorname{argmax}} \sum_{s \in S, |s|>1} |s|^{|s|} count(s),$$

where $\mathcal{S}_Q$ is the set of all possible query segmentations, S is a possible segmentation, s is a segment in S, and $count(s)$ is the frequency of s in the web n-gram corpus.

The second method, *SEG-2*, is based on a successful supervised segmentation method, which was first proposed by Bergsma and Wang (2007). *SEG-2* employs a large set of features, and is pre-trained on the query collection described by Bergsma and Wang (2007). The features used by the *SEG-2* method are described by Bendersky et al. (2009), and include, among others, n-gram frequencies in a sample of a query log, web corpus and Wikipedia titles.

Table 3 demonstrates the comparison between the *j-PRF*, *SEG-1* and *SEG-2* methods. When compared to the *SEG-1* baseline, *j-PRF* is significantly more effective, even though it only employs bigram counts (see Eq. 4), instead of the high-order n-grams used by *SEG-1*, for computing the score of a segmentation. This results underscores the benefit of joint annotation, which leverages capitalization and POS tagging to improve the quality of the segmentation.

When compared to the *SEG-2* baseline, *j-PRF* and *SEG-2* are statistically indistinguishable. *SEG-2* posits a slightly better F1, while *j-PRF* has a better MQA. This result demonstrates that the segmentation produced by the *j-PRF* method is as effective as the segmentation produced by the current supervised state-of-the-art segmentation methods, which employ external data sources and high-order n-grams. The benefit of the *j-PRF* method compared to the *SEG-2* method, is that, simultaneously with the segmentation, it produces several additional query annotations (in this case, capitalization and POS tagging), eliminating the need to construct separate sequence classifiers for each annotation.

7 Conclusions

In this paper, we have investigated a joint approach for annotating search queries with linguistic structures, including capitalization, POS tags and segmentation. To this end, we proposed a probabilistic approach for performing joint query annotation that takes into account the dependencies that exist between the different annotation types.

Our experimental findings over a range of queries from a web search log unequivocally point to the superiority of the joint annotation methods over both query-based and pseudo-relevance feedback based independent annotation methods. These findings indicate that the different annotations are mutually-dependent.

We are encouraged by the success of our joint query annotation technique, and intend to pursue the investigation of its utility for IR applications. In the future, we intend to research the use of joint query annotations for additional IR tasks, e.g., for constructing better query formulations for ranking algorithms.

8 Acknowledgment

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Niranjan Balasubramanian and James Allan. 2009. Syntactic query models for restatement retrieval. In *Proc. of SPIRE*, pages 143–155.

Cory Barr, Rosie Jones, and Moira Regelson. 2008. The linguistic structure of english web-search queries. In *Proc. of EMNLP*, pages 1021–1030.

Michael Bendersky and W. Bruce Croft. 2009. Analysis of long queries in a large scale search log. In *Proc. of Workshop on Web Search Click Data*, pages 8–14.

Michael Bendersky, David Smith, and W. Bruce Croft. 2009. Two-stage query segmentation for information retrieval. In *Proc. of SIGIR*, pages 810–811.

Michael Bendersky, W. Bruce Croft, and David A. Smith. 2010. Structural annotation of search queries using pseudo-relevance feedback. In *Proc. of CIKM*, pages 1537–1540.

Shane Bergsma and Qin I. Wang. 2007. Learning noun phrase query segmentation. In *Proc. of EMNLP*, pages 819–826.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1.

Chris Buckley. 1995. Automatic query expansion using SMART. In *Proc. of TREC-3*, pages 69–80.

Jenny R. Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proc. of NAACL*, pages 326–334.

Jiafeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. 2008. A unified and discriminative model for query refinement. In *Proc. of SIGIR*, pages 379–386.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proc. of SIGIR*, pages 267–274.

Matthias Hagen, Martin Potthast, Benno Stein, and Christof Braeutigam. 2010. The power of naive query segmentation. In *Proc. of SIGIR*, pages 797–798.

Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. 2011. Query segmentation revisited. In *Proc. of WWW*, pages 97–106.

Rosie Jones and Daniel C. Fain. 2003. Query word deletion prediction. In *Proc. of SIGIR*, pages 435–436.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proc. of WWW*, pages 387–396.

Giridhar Kumaran and James Allan. 2007. A case for shorter queries, and helping user create them. In *Proc. of NAACL*, pages 220–227.

Giridhar Kumaran and Vitor R. Carvalho. 2009. Reducing long queries using query quality predictors. In *Proc. of SIGIR*, pages 564–571.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proc. of SIGIR*, pages 120–127.

Matthew Lease. 2007. Natural language processing for information retrieval: the time is ripe (again). In *Proceedings of PIKM*.

Xiao Li. 2010. Understanding the semantic structure of noun phrase queries. In *Proc. of ACL*, pages 1337–1345, Morristown, NJ, USA.

Rachel T. Lo, Ben He, and Iadh Ounis. 2005. Automatically building a stopword list for an information retrieval system. In *Proc. of DIR*.

Yumao Lu, Fuchun Peng, Gilad Mishne, Xing Wei, and Benoit Dumoulin. 2009. Improving Web search relevance with semantic features. In *Proc. of EMNLP*, pages 648–657.

Mehdi Manshadi and Xiao Li. 2009. Semantic Tagging of Web Search Queries. In *Proc. of ACL*, pages 861–869.

André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proc. of EMNLP*, pages 157–166.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL*, pages 950–958.

Marius Paşca. 2007. Weakly-supervised discovery of named entities using web search queries. In *Proc. of CIKM*, pages 683–690.

Dou Shen, Toby Walkery, Zijian Zhengy, Qiang Yangz, and Ying Li. 2008. Personal name classification in web queries. In *Proc. of WSDM*, pages 149–158.

Bin Tan and Fuchun Peng. 2008. Unsupervised query segmentation using generative language models and Wikipedia. In *Proc. of WWW*, pages 347–356.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34:161–191, June.

Xing Wei, Fuchun Peng, and Benoit Dumoulin. 2008. Analyzing web text association to disambiguate abbreviation in queries. In *Proc. of SIGIR*, pages 751–752.