Web Semantics: Science, Services and Agents on the World Wide Web 30 (2015) 22–38

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Discovering and understanding word level user intent in Web search queries

Rishiraj Saha Roy [a,*], Rahul Katare [a], Niloy Ganguly [a], Srivatsan Laxman [b,1], Monojit Choudhury [c]

[a] Computer Science and Engineering, Indian Institute of Technology Kharagpur, India
[b] Scibler Technologies Private Limited, India
[c] Microsoft Research India, India

ARTICLE INFO

Article history:
Received 15 July 2013
Received in revised form
27 July 2014
Accepted 28 July 2014
Available online 7 August 2014

Keywords:
Query understanding
Query intent
Intent words
Co-occurrence entropy

ABSTRACT

Identifying and interpreting user intent are fundamental to semantic search. In this paper, we investigate the association of intent with individual words of a search query. We propose that words in queries can be classified as either *content* or *intent*, where content words represent the central topic of the query, while users add intent words to make their requirements more explicit. We argue that intelligent processing of intent words can be vital to improving the result quality, and in this work we focus on intent word discovery and understanding. Our approach towards intent word detection is motivated by the hypotheses that query intent words satisfy certain distributional properties in large query logs similar to function words in natural language corpora. Following this idea, we first prove the effectiveness of our corpus distributional features, namely, word co-occurrence counts and entropies, towards function word detection for five natural languages. Next, we show that reliable detection of intent words in queries is possible using these same features computed from query logs. To make the distinction between content and intent words more tangible, we additionally provide operational definitions of content and intent words as those words that should match, and those that need not match, respectively, in the text of relevant documents. In addition to a standard evaluation against human annotations, we also provide an alternative validation of our ideas using clickthrough data. Concordance of the two orthogonal evaluation approaches provide further support to our original hypothesis of the existence of two distinct word classes in search queries. Finally, we provide a taxonomy of intent words derived through rigorous manual analysis of large query logs.

1. Introduction

Semantic search has attracted a good amount of research in recent years [1–3]. The goal of semantic search is to improve the result relevance by appropriately understanding user intent and using intelligent document retrieval techniques to leverage the knowledge of this intent. Thus, the ability to identify user intent is one of the first steps in semantic search. Most often, the search query is a translation of the user's intent into a short sequence of keywords. This imposes great value on every word in the query from the aspect of a semantic search engine. Past research has mostly focused on inferring the intent of the query as a whole, and the most generic intent classes were found to be informational, navigational and transactional [4–6]. In this research, we take a deeper look at query intent, zooming in on individual words as possible indicators of user intent.

From an information retrieval (IR) perspective, the equivalence of a Web search query with an unordered sequence of words (or a "bag-of-words") has long been challenged, with research on term dependence [7–9] and term proximity models [10–14] showing significant improvements in retrieval performance. Extending this idea of the presence of a *query structure* further, we propose that words or multiword units in queries basically belong to two classes—*content words* that represent the central topics of queries, and *intent words*, which are articulated by users to refine their

[*] Corresponding author. Tel.: +91 9477588851.
E-mail addresses: rishiraj.saharoy@gmail.com, rishiraj@cse.iitkgp.ernet.in (R. Saha Roy), rah.ykg@gmail.com (R. Katare), niloy@cse.iitkgp.ernet.in (N. Ganguly), srivatsan.laxman@gmail.com (S. Laxman), monojitc@microsoft.com (M. Choudhury).
[1] This work was done while the author was at Microsoft Research India.

http://dx.doi.org/10.1016/j.websem.2014.07.010
1570-8268/© 2014 Elsevier B.V. All rights reserved.

information needs concerning the content words. The class of content units include, but are not restricted to named entities (like `brad pitt`, `titanic` and `aurora borealis`)—anything that is capable of being the topic of a query would be the content unit in the context of that query. For example, `blood pressure`, `marriage laws` and `magnum opus` are legitimate examples of content words or units. Intent words or intent units, on the other hand, present vital clues to the search engine regarding the specific information sought by the user about the content units. For instance, intent units like `home page`, `pics` and `meaning`, all specify unique information requests about the content units. The queries `brad pitt website`, `brad pitt news` and `brad pitt videos` all represent very different user intents. It is not hard to see that while content units need to be matched inside document text for relevance, it is possible to leverage the knowledge of intent units to improve user satisfaction in better ways. For example, words like `pics`, `videos` and `map` can all trigger relevant content formats to directly appear on the result page. Words like `near` and `cheap` may be used to sort result objects in the desired order. These ideas motivate us to focus on the discovery and understanding of query intent units in this research.

Appropriately understanding the distinction between the two classes of words and concretizing these notions of intent and content required rigorous manual analysis of large volumes of query logs on our part. During this process, we observed that intent units share corpus distributional properties similar to function words of natural language (NL). NLs generally contain two categories of words—*content* and *function* [15]. In English, nouns, verbs, adjectives and most adverbs constitute the class of content words. On the other hand, pronouns, determiners, prepositions, conjunctions, interjections and other particles are classified as function words. While content words express meaning or *semantic content*, function words express important grammatical relationships between various words within a sentence, and themselves have little lexical meaning. The distinction between content and function words, thus, plays an important role in characterizing the syntactic properties of sentences [16–18]. Distributional postulates that are valid for function word detection, like the co-occurrence patterns of function words being more diverse and unbiased than content words, seemed to be valid for query intent units as well. Following these leads, we first segment queries to identify possible multiword units using a state-of-the-art query segmentation algorithm [19], and compute the relevant distributional properties, namely, co-occurrence counts and entropies, for the obtained query units. We found that the units which exhibit high values of these indicators indeed satisfy our notions about the class of intent units. Subsequently, we systematically evaluated our findings against human annotations and clickthrough data (which represent functional evidence of user intent) and substantiate our hypotheses.

In hindsight, we understand that while NL function words have little describable meaning (like `in`, `of` and `what`) and only serve to specify relationships among content words, well-defined semantic interpretations can be attributed to most intent words (like `map`, `pics` and `videos`). Intent words, even though effectively lacking purpose without the presence of a content word(s) in the same query, carry weight of their own within the query. Thus, content and intent units play slightly different roles in the query from the roles of content and function words in NL sentences. It simply turns out that function words in NL and intent words in queries share similar statistical behavior. Function words and intent words are still not fully comparable, and an important difference between the two is the fact that the definition of a function word is not context-dependent, whereas intent words can also behave as content words depending on the context (Section 4).

The objective of this paper is to identify and characterize intent words in Web search queries, words that are explicit indicators of user intent, and it is organized as follows. In Section 2, we begin with a verification of the efficacy of corpus-based distributional statistics towards function word identification and through rigorous experimentation over five languages, discover that *co-occurrence counts and entropies* are the most robust indicators of function words in NL. Having convinced ourselves of the power of co-occurrence statistics in detecting function words across diverse languages, we apply similar techniques to discover intent units in Web search queries (Section 3). This is followed by a simple algorithm to label intent units in the context of individual queries and subsequent evaluations using human annotations and clickthrough data (Section 4). Observing that co-occurrence statistics locate quite a diverse set of intent units, we attempt to provide a taxonomy of such units based on their relationships with content words that we believe can be very useful in semantic search (Section 5). Finally, we present concluding remarks and open directions for future work (Section 6).

2. Distributional properties of NL function words

Function words play a crucial role in many Natural Language Processing (NLP) applications. They are used as features for unsupervised POS induction and also provide vital clues for grammar checking and machine translation. In this section, we first re-examine this popular hypothesis that the most frequent words in a language are the function words. By *function words or units* we refer to all the closed-class lexical items in a language, e.g., pronouns, determiners, prepositions, conjunctions, interjections and other particles (as opposed to open-class items, e.g., nouns, verbs, adjectives and most adverbs). We note that the statistics presented here are applicable for both single-word (`in, about`) as well as multiword (`how to, because of`) function units from corpora, though the latter demands chunking of the NL text. We perform all the NL experiments on unsegmented (or unchunked) sentences and hence report the results for detection of single word function units. Nevertheless, Web search queries, on which we mainly focus, have been suitably segmented by the state-of-the-art algorithm [19].

2.1. Datasets

For the NL experiments, we shall look at five languages from diverse families: English, French, Italian, Hindi and Bangla. English is a *Germanic* language, French and Italian are *Romanic* languages, and Hindi and Bangla belong to the *Indo-Aryan* family. Therefore, any function word characterization strategy that works across these languages is expected to work for a large variety of languages.

The details of the corpora used for these five languages are summarized in Table 1. The sentences were uniformly sampled from larger datasets. M in the value columns denotes million. S, N, V and F denote the *numbers* of all sentences, all words, unique words (vocabulary size) and function words, respectively. We note that the Indian languages have almost twice as many function words as compared to the European ones. This is due to morphological richness and the existence of large numbers of modal and vector verbs.

2.2. Metric

In a distributional property-based function word detection approach, the output is a ranked list of words sorted in descending order of the corresponding indicator value. Here we adopt a popular metric, *Average Precision* (AP) [20,21], used in IR for the evaluation of ranked lists. More specifically, let $w_1, w_2, \ldots, w_n$ be

Table 1
Details of NL corpora. S, N and V respectively denote the *numbers* of sentences, words and unique words present in the corpus, and F denotes the number of function words present in the gold standard list used.

Language	Corpus source	S	N	V	Function word list source	F
English	Leipzig corpora[a]	1 M	19.8 M	342 157	Sequence Publishing[b]	229
French	*-do-*	1 M	19.9 M	388 221	Built by extracting pronouns, determiners, prepositions, conjunctions and interjections from POS-tagged corpora available at WaCKy[c]	289
Italian	*-do-*	1 M	20 M	434 680	*-do-*	257
Hindi	*-do-*	0.3 M	5.5 M	127 428	Manually constructed by linguists and augmented as above with POS-tagged corpora available at LDC[d]	481
Bangla	Crawl of *Anandabazar Patrika*[e]	0.05 M	16.2 M	411 878	*-do-*	510

[a] http://corpora.informatik.uni-leipzig.de/download.html.
[b] http://www.sequencepublishing.com/academic.html#function-words.
[c] http://wacky.sslmit.unibo.it/doku.php?id=download.
[d] http://www.ldc.upenn.edu (Catalog Nos. LDC2010T24 and LDC2010T16 for Hindi and Bangla respectively).
[e] http://www.anandabazar.com/.

a ranked list of words sorted according to some corpus statistic, say, frequency. Thus, if $i < j$, then frequency of w_i is greater than the frequency of w_j. *Precision at rank k*, denoted by P@k, is defined as

$$\text{P@}k = \frac{1}{k}\sum_{i=1}^{k} f(w_i) \tag{1}$$

where, $f(w_i)$ is 1 if w_i is a function word, and is 0 otherwise. This function can be computed based on the gold standard lists of function words. Subsequently, *average precision at rank n*, denoted by AP@n, is defined as

$$\text{AP@}n = \frac{1}{n}\sum_{k=1}^{n} \text{P@}k. \tag{2}$$

AP@n is a better metric than P@k because P@k is insensitive to the rank at which function words occur in the list. In our tables, we report AP@n averaged over $\mathcal{N}$ corpus sub-samples, which is given by $\frac{1}{\mathcal{N}}\sum_{r=1}^{\mathcal{N}}(\text{AP@}n)_r$ where $(\text{AP@}n)_r$ is the AP@n for the rth sub-sample.

2.3. Frequency as a function word indicator

Frequency (Fr) is often used as an indicator for detecting function words, but the following factors affect its robustness.

Corpus size: If the corpus size is not large, many function words will not occur a sufficient number of times. For example, even though `the` and `in` will be very frequent in most English corpora, `meanwhile` and `off` may not be so. As a result, if frequency is used as a function word detector with small datasets, we will have a problem of low recall [21]. In our experiments, we measure corpus size, N, as the total number of words present.

Corpus diversity: If our language corpus is restricted, or sampled only from specific domains, words specific to those domains will have high frequencies and will get detected as function words. For example, the word `government` will be much more frequent in political news corpora than `although`. The number of unique words in a corpus, or the vocabulary size, V, is a good indicator of its diversity. For restricted domain corpora, V grows much more slowly with N than in an open domain corpus.

2.3.1. Experiments and results

For our frequency-based experiments, we create 200 sub-samples from the original corpora. We choose 10 different values of N, and for each N choose 20 different samples such that we get a different V each time. For each sub-sampled corpora, we compute frequency of each word and sort words in decreasing order of frequency. Then we compute AP@200, AP@500 and AP@1000 with respect to the gold standard lists of function words (Table 1). A representative set of results is shown in Fig. 1. We see this same trend for all the languages, as well as for AP@200 and AP@1000. For a fixed N, AP increases with V, which means that the performance of the frequency-based strategy works better when the corpus has high diversity. We also observe that, in general, the performance gets better as N increases. However, for a fixed V, increasing N effectively means increasing the number of sentences without increasing the diversity of the corpus. Regression lines in Fig. 1 suggest that for the same V, a higher N would lead to a lower AP.

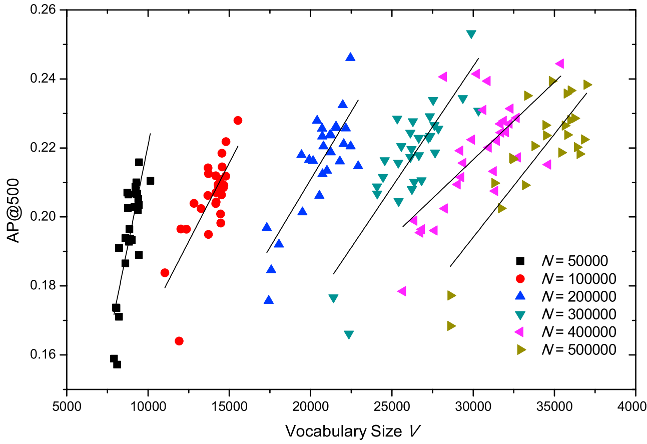

Fig. 1. (Color online) AP@500 with frequency as the function word indicator for English at various V and N, with linear regression lines.

2.4. Co-occurrence statistics as function word indicators

After having a feel of the issues faced when using frequency as a function word indicator, we introduce other properties of function words that may help in more robust detection. We observe the following interesting characteristics about the syntactic distributions of function and content words in NL, which can be summarized by the following two postulates.

Postulate I: Function words, in general, tend to co-occur with a larger number of distinct words than content words. What can occur to the immediate left or right of a content word is much more restricted than that in the case of function words. We hypothesize that even if a content word, e.g., *government*, might have high frequency owing to the nature of the domain, there will be only a relatively few words that can co-occur immediately after or before it. Therefore, the co-occurrence count may be a more robust indicator of function words.

Table 2
Definitions of the different function word indicators.

Indicator	Symbol	Definition
Frequency	Fr	Frequency of a word in the corpus
Left co-occurrence count	LCC	Number of distinct words appearing to the immediate left of a word
Left co-occurrence entropy	LCE	Entropy of the left co-occurrence distribution
Total co-occurrence count	TCC	Number of distinct words appearing to the immediate left and right of a word
Total co-occurrence entropy	TCE	Entropy of the total co-occurrence distribution
Right co-occurrence count	RCC	Number of distinct words appearing to the immediate right of a word
Right co-occurrence entropy	RCE	Entropy of the right co-occurrence distribution

Postulate II: The co-occurrence patterns of function words are less likely to show bias towards specific words than those for content words. For example, and will occur beside several other words like `school`, `elephant` and `pipe` with more or less an equally distributed co-occurrence count with each of these words. In contrast, the co-occurrence distribution of `school` will be skewed, with more bias towards `to, high` and `bus` than `over, through` and `coast`, with the list of words occurring beside `school` also being much smaller than that for and.

In order to test Postulate I, we measure the number of distinct words that occur to the immediate left, right and either side of each unique word in the sub-sampled corpora. We shall refer to these statistics as *left*, *right* and *total co-occurrence counts* (LCC, RCC and TCC) respectively. To test Postulate II, we compute the *entropy* [22] of the co-occurrence distributions of the words occurring to the *left*, *right* and either side (i.e., *total*) contexts of a word w:

$$\text{Entropy}(w) = -\sum_{t_i \in \text{context}(w)} p_{t_i|w} \log_2(p_{t_i|w}) \qquad (3)$$

where, $\text{context}(w)$ is the set of all words co-occurring with w either in the left, the right or the total contexts, and $p(t_i|w)$ is the probability of observing word t_i in that specific context.

Context: In this paper, the left, right and total *contexts* of a word w respectively denote the immediately preceding (one) word, immediately succeeding (one) word and both the immediately preceding and the immediately succeeding words for w respectively, in sentences of the corpus. The definition of context (i.e., whether it includes the preceding or the succeeding one or two or three words) will change the absolute values of our results, but all the trends in the results are expected to remain the same.

This probability in Eq. (3) can be computed simply by counting the frequency of the appropriate bigrams normalized by the frequency of w. We shall refer to these statistics as *left, right* and *total Co-occurrence Entropy* (LCE, RCE and TCE respectively). We would expect LCC, RCC or TCC of function words to be higher than that of content words due to *Postulate* I; similarly, due to *Postulate* II we can expect the LCE, RCE or TCE to be higher for function words than for content words. The definitions of these indicators are summarized in Table 2.

2.4.1. *Experiments and results*

We now sort the list of all words in descending order of each of the seven indicators. We then compute metrics AP@200, AP@500 and AP@1000 for these seven lists. To bring out the performance difference of each of the six co-occurrence features with respect to frequency, we plot (in Figs. 2 and 3) the following performance measure against N:

$$\text{Value plotted} = \frac{\text{Metric for indicator} - \text{Metric for Fr}}{\text{Metric for Fr}}. \qquad (4)$$

The x-axis can now be thought of as representing the performance of frequency. In Fig. 2, for a particular N, the data points were averaged over all (N, V) pairs (we had 20 (N, V) pairs for each N). For Fig. 3, we kept N fixed at 500 000. The general trends were the same for AP@500 and AP@1000. The observations (both N and V variation) for French and Italian were similar to that of English, while those for Hindi and Bangla were similar to each other. Table 3 reports AP values for all statistics for the five languages. From Table 3, we see that for all the languages, AP for some of the co-occurrence statistics are higher than AP obtained using frequency.

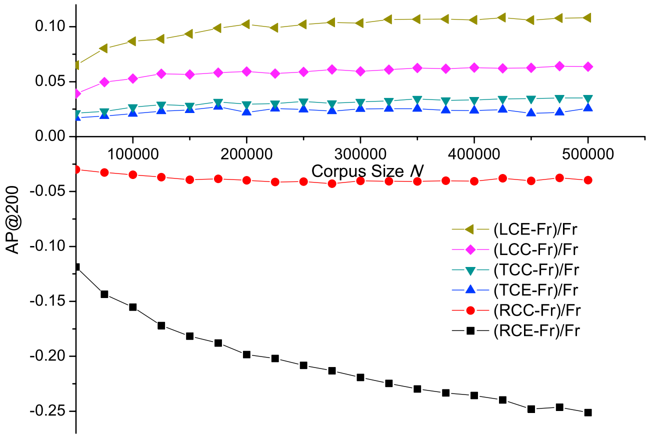

Fig. 2. (Color online) Performance of co-occurrence statistics with respect to frequency for AP@200 (English) (variation with N).

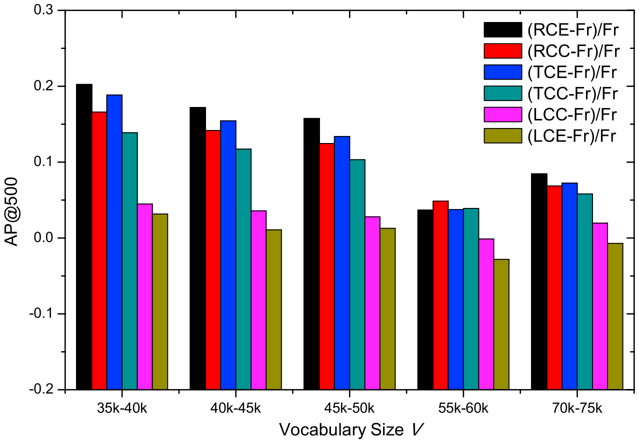

Fig. 3. (Color online) Performance of co-occurrence statistics with respect to frequency for AP@500 (Bangla) (variation with V; N fixed at 500 000).

Regular improvements over frequency: From the plots and Table 3, it is evident that some of the co-occurrence statistics consistently beat frequency as indicators. In fact, as evident from Figs. 2 and 3, the use of co-occurrence statistics results in systematic improvement over frequency with variations in N and V, and hence, are very robust indicators. Among the co-occurrence statistics, entropy is generally observed to be more powerful than simple counts. This justifies that Postulate II is indeed a stricter characteristic of function units.

The best indicator depends upon language typology: A very interesting fact that came out of these experiments is that the left co-occurrence statistics (LCE and LCC) generally outperform the

Table 3
AP for frequency and co-occurrence statistics, averaged over 200 (N, V) pairs for each language.

Language	Metric	Typology	Fr	LCC	LCE	TCC	TCE	RCC	RCE
English	AP@200	Pre-	0.663	0.702*	**0.729***	0.684*	0.679*	0.637	0.527
	AP@500		0.453	0.477*	**0.493***	0.468*	0.464*	0.439	0.365
	AP@1000		0.314	0.328*	**0.336***	0.324*	0.319	0.305	0.259
French	AP@200	Pre-	0.594	0.642*	**0.648***	0.615*	0.611*	0.553	0.501
	AP@500		0.390	0.430*	**0.438***	0.405*	0.398	0.357	0.313
	AP@1000		0.264	0.290*	**0.296***	0.273	0.269	0.242	0.212
Italian	AP@200	Pre-	0.611	0.639*	**0.645***	0.636*	0.620	0.606	0.601
	AP@500		0.422	0.433*	0.423	**0.438***	0.423	0.411	0.395
	AP@1000		**0.299**	0.295	0.290	**0.299**	0.291	0.282	0.268
Hindi	AP@200	Post-	0.682	0.614	0.510	0.698*	0.694*	**0.716***	0.713*
	AP@500		0.497	0.458	0.394	0.511*	0.505	**0.523***	0.521*
	AP@1000		0.368	0.345	0.306	0.379*	0.371	**0.383***	0.380*
Bangla	AP@200	Post-	0.648	0.684*	0.691*	0.730*	**0.763***	0.741*	0.757*
	AP@500		0.522	0.543*	0.537*	0.579*	0.599*	0.589*	**0.603***
	AP@1000		0.415	0.428*	0.422	0.454*	0.470*	0.463*	**0.475***

The highest value in a row is marked in **boldface**. The paired t-test was performed and the null hypothesis was rejected if p-value < 0.05.

* Statistically significant improvement over frequency.

right for English, French and Italian, whereas the reverse is true for Hindi and Bangla (RCE and RCC are the best). This is due to the fact that English, French and Italian are prepositional languages whereas Hindi and Bangla are postpositional. In a prepositional language, function words generally precede content words. Therefore, the lexical categories (and hence the exact numbers of lexical items) that can succeed a function word is restricted. For instance, only nouns or articles can follow words like `in` and `of` in English. On the other hand, there is no restriction on the class of words that can precede a function word. Hence function words in a prepositional language can be expected to have significantly higher left co-occurrence counts (and hence higher entropies). Similarly, the opposite is valid for postpositional languages.

Total co-occurrence: a safe choice. It is not always possible to know the typology of a language in advance. Thus, it may not be clear *a priori* whether to depend on left or right co-occurrence statistics. The nice point here is that the total co-occurrence statistics (TCE and TCC) are almost always better than frequency (Table 3). This makes them safe indicators to rely on when not much is known about the language syntax.

2.5. Inverse document frequency

A *stop word* is a term that is popular in IR which is used to denote a word that does not have sufficient discriminative power which can be used by the retrieval system to distinguish between relevant and non-relevant documents. Even though the concepts of stop words in IR and function words in NL understanding are fundamentally different in function, it nevertheless turns out that there is a significant level of overlap among these sets. See, for example, one of the lists of English stop words, used in the popular SMART IR system [23], at http://bit.ly/8vBrVF. We note that the overlap is caused by general domain stop words.

Thus, it is worthwhile to explore techniques used in stop word detection to our problem. The concept of Inverse Document Frequency (IDF) is traditionally used to mark stop words in IR systems. The IDF of a word w is defined as

$$\mathrm{IDF}(w) = \log_{10} \frac{1 + |d|}{|d_w|} \tag{5}$$

where $|d|$ is the number of documents in the document collection d, and $|d_w|$ is the number of documents containing w. In the SMART system [23], a combination of term frequency (TF) $(\mathrm{Fr}(w))$ and IDF, known as TF–IDF, is measured for every word-document pair (w, d):

$$\text{TF-IDF}(w, d) = \begin{cases} 0 & \text{if } \mathrm{Fr}(w, d) = 0 \\ \mathrm{TF}(w, d) \times \mathrm{IDF}(w) & \text{otherwise} \end{cases} \tag{6}$$

where $\mathrm{TF}(w, d)$ is the normalized term frequency of w in document d and is defined as

$$\mathrm{TF}(w, d) = 1 + \log_{10}(1 + \log_{10}(\mathrm{Fr}(w, d))) \tag{7}$$

where $\mathrm{Fr}(w, d)$ is the raw frequency of w in d. The higher number of documents that a word is present in, the lower is its IDF. Stop words, by virtue of their relative abundance, have low IDF and hence low TF–IDF values. For measuring the effectiveness of TF–IDF of a word as a corpus-level indicator, we generalize it from being a document-specific value by computing the mean TF–IDF for every document containing that word.

EuroParl corpus: The existence of multiple documents is necessary for computing IDF-related measures, i.e. the NL corpus should be segmented into discrete documents. The Leipzig Parallel Corpora used for the previous experiments contain all the sentences in a single large document, which deems it unfit for evaluating the performance of IDF. Fortunately, in one of the previous versions (five and earlier) of another widely used NL corpus, the EuroParl[2] dataset [24], the corpus is fragmented into thousands of documents (approximately 5000 documents for each language). However, the EuroParl corpus, being Parliament proceedings of European countries, does not contain datasets for Hindi and Bangla. Hence, we report findings on English, French and Italian only.

Experiments and results: For a fair evaluation, we need to recompute AP values for all the indicators for the EuroParl dataset and contrast them with IDF and TF–IDF. We note that while the ranked lists for frequency and co-occurrence statistics were obtained by sorting the words in descending order of these

[2] http://www.statmt.org/europarl/.

Table 4
Comparison of IDF with other indicators for AP@200.

Indicator	English	French	Italian
IDF	**0.435**	**0.360**	**0.400**
TF–IDF	**0.035**	**0.020**	**0.030**
Fr	0.571	0.564	0.547
LCC	0.678	0.667	0.633
LCE	0.722	0.649	0.648
TCC	0.648	0.623	0.601
TCE	0.673	0.609	0.593
RCC	0.592	0.524	0.540
RCE	0.492	0.454	0.508

The two minimum values in a column are marked in **boldface**.

indicators, a reverse sorting (ascending) is necessary for IDF and TF–IDF (stop words have low IDF). We summarize our results in Table 4 (AP@200). The trends observed for AP@500 and AP@1000 are exactly the same.

From Table 4, we see that TF–IDF performs the worst, followed by IDF. But even for IDF, the difference in performance with the next better indicator is always substantial. Thus, we infer that these measures are clearly unsuitable for function word detection. On manually analyzing the ranked lists for understanding the poor performance of IR measures, the reason was clearly understood. IDF and TF–IDF pull out stop words that do not offer discriminating evidence for ranking documents in response to a query. A majority of these words at the top positions, apart from the most frequent function words like `the` and `and`, turn out to be content words (`resume, declare, adjourns, vote, president` and `minutes`). We note that the corpus is from a restricted domain (Parliament proceedings), and the domain-specific stop words negatively impact the performance of IDF-based measures. We recollect that the same reason is one of main drawbacks of using frequency as an indicator (Section 2.3). The best performance again comes from co-occurrence statistics (mostly entropy), highlighting their robustness even in restricted domain datasets.

3. Intent units of Web search queries

Web search queries are issued by users to *communicate* their information needs to search engines. Thus, their function is similar to languages [25,26]. Past research [27–29] suggests that Web queries have a distinct structure where the units are not always single words but segments comprising one or more words. For example, not all permutations of the query `nokia n96 gprs config telstra australia` are meaningful—only three permutable units make sense, which are `nokia n96`, `gprs config` and `telstra australia`. Complex network based analysis of co-occurrence networks derived from query logs demonstrate both similarities and differences with NL [30]. These findings, along with other observations, have led researchers to propose that queries can be regarded as a language of their own, which is evolving at a fast pace [31,25,26,32].

On the other hand, linguistic or computational attempts to characterize the structure of Web search queries have primarily focused on the application of English NLP tools and notions from English syntax on queries. For instance, Barr et al. [33] describe a study on the POS tagging of Web search queries where a state-of-the-art POS tagger that achieves approximately 97% accuracy for English achieves only about 48% accuracy on queries if trained on NL corpora. Training on annotated queries significantly increases the tagging accuracy to almost 79%. This suggests that English Web search queries are not really "English" and that attempts to project notions of Standard English morpho-syntax on queries can often fail. A *noun* (e.g., `wikipedia`, `article`) or *verb* (e.g., `download`, `compare`) in English language are so-called because of their specific distributional and functional characteristics. The same words when used in a query need not retain similar distributional characteristics and need not assume similar functional roles. Therefore, there is a need to learn the distributional properties of the units and define the lexical and functional categories present in Web search queries from the first principles. In this section, we apply our robust function word identification strategies to query logs and observe the resultant partitioning of words. We find that the top ranking words according to co-occurrence statistics align well with our notion of intent units (Section 1).

3.1. Dataset

For all our experiments on queries, we use a log sampled from Bing Australia[3] in May 2010. This raw data slice consisted of 16.7 M (M = Million) queries. We subsequently extracted 11.9 M queries from the raw data such that the queries were composed of ASCII characters only and were of length between two and ten words. The justification for imposing a filter based on query length is as follows. One word queries do not contribute to co-occurrence statistics. Very long queries (having more than ten words) are typically computer generated messages or excerpts from NL text, and need separate query processing techniques. There are 4.7 M unique queries among the extracted 11.9 M queries—but in order to preserve data properties arising out of the natural power law frequency distribution of queries (Pass et al. [34] and analysis on own log), duplicate queries were retained for all experiments.

3.2. Operational definitions

As mentioned earlier, segments or multiword units are the basic building blocks of queries [27,28]. Query segmentation has been shown to improve IR performance [9,35,19]. Therefore, instead of single words, we study and classify *segments* (units) for Web search queries. In our study, we used the state-of-the-art query segmentation algorithm presented in Saha Roy et al. [19], which uses query logs and Wikipedia titles as the input resources. Segment boundaries are marked by parentheses in this text, like `(public schools) (new york)`. We apply the segmentation algorithm on all the queries and compile a list of unique units (1,311,025 in number) that occur in our query log. For each unit, we measure its frequency, the three co-occurrence counts and the corresponding entropies.

To give a feel of the units that are pulled up, we present some examples in Table 5 when sorted in descending order of TCE. Only 26 out of the top 100 units for queries are function words of English. We understand that it can be hard to make a definite distinction between content and intent units solely on a qualitative basis. So before we can have any further quantitative evaluation of our indicators, we must have in place *operational definitions* of content and intent units in queries that can help concretize the notion of a word being content or intent with respect to a query. An empirical validation of the proposed operational definitions is presented in Section 4.3.

Content units in Web search queries: They carry the core information requirement within a Web search query. Just like the role of content units in NL sentences, removing these units makes the query lose its central idea. For this reason, content units need to be *matched* within the documents for effective retrieval. For example, `titanic`, `age of empires` and `ford cars` are all content units.

Intent units in Web search queries: They specify user intent in Web search queries. They *need not match* exactly at the document side, and the search engine can have intelligent techniques for

[3] http://www.bing.com/?cc=au.

Table 5
Sample units at top ranks when sorted in descending order by TCE.

Ranks 1–10	Ranks 11–20	Ranks 21–30	Ranks 51–60	Ranks 91–100
in	with	for sale	home	time
the	lyrics	is	de	your
and	by	what is	pictures of	book
for	from	best	music	show
of	2010	vs	uk	la
free	online	video	jobs	myspace
to	new	2009	black	baby
on	at	my	song	james
how to	2008	pictures	news	cheap
a	download	school	about	does

using such units to increase the relevance of result pages. For example, `music`, `online`, and `for sale` are some commonly encountered intent units. Analogous to NL, removal of these units removes vital details about query semantics. We note that function units in NL (like `and`, `of` and `in`) can play similar roles in queries, and hence fall under this category.

These definitions of content and intent words, and the condition of matching in document text, are extremely vital to principles in semantic search. We emphasize that the definitions of content and intent are always necessarily operational—content segments need to be matched in the document text during the retrieval process, while the search engine can have intelligent techniques to process intent segments to improve relevance of result pages. Thus, what has to be treated as content today can become an intent segment after a few years if the (semantic) search system develops a more improved way to handle that segment than searching for it in the document text. This is where it differs from other similar frameworks, which are static and more like the entity–attribute model [36].

For example, in a query like `london wedding` or `london population`, we would treat `wedding` (or `population`) to be a content word and not an intent word (`london` would be a content too), because in the current search scenario, there is almost no way to infer the "intent" `wedding` or `population` from a page without matching the term within the document text. `Population` could become an intent word the day when annotations or other features of a Semantic Web enable the engine to infer the answer (i.e. the population of a city or country) even if the word does not appear on the retrieved page. But `population` is, and would always remain, an attribute of a country or a city (which is the entity). Current search engines provide direct answers to queries like `london population` today but those are summaries generated from a document pool created by traditional matching. In contrast, for queries like `london weather`, `london place` and `london life` (generally all Web queries are in lowercase), `london` would be content (as it is the topic of the information need) and `weather, place` or `life` would be intent as there exist ways today (search engines may use them or not) to infer information relevant to these contexts without direct matching. Say, for example, knowledge graphs enable the search engine to know that temperature, rainfall, and humidity are aspects of weather (as can be employment, poverty and cleanliness aspects of city life) and can be scraped off pages to provide consolidated information on weather and life. Intent words like place or location can be used to understand the preferred content type, like bringing up relevant maps. In summary, the collection of all intent words or units is a *dynamic set* completely defined for a particular span of time by the state-of-the-art (semantic) search technologies available during that span of time.

3.3. Experimental results

We note that it is not possible to build an exhaustive list of such intent units for queries. So in order to have a suitable gold standard set created by humans for future validation of results, we first need a representative sample unit set. These can be manually classified as intent units (or content units). To avoid bias towards any particular indicator, we took the union of the top 1000 units when sorted by each indicator. We asked three human annotators A, B and C to mark these 1215 query segments as "intent" or "content" with the above operational definitions as guidelines. All of our annotators were graduate students in the age group of 25–35 years and were well-acquainted with Web search, each issuing about 20–30 queries per day. Out of the 1215 segments, A, B and C marked 607, 646 and 548 units as intent respectively. Now we assume the units marked as "intent" by each annotator separately as the gold standard. Then, similar to the method followed in NL, we sort the list of all units in descending order of each of the seven indicators and compute the AP@200, AP@500 and AP@1000 for these ranked lists. Results are presented in Table 6.

Superiority of total co-occurrence: Just like NL, co-occurrence statistics consistently beat the performance of frequency. When the ranked list is small (200 units), the right (A and C) or left (B) co-occurrence statistics gives the best accuracy. On the other hand, for longer lists (500 and 1000 units), the total co-occurrence count (A) and entropy (B) always perform the best. In general, total co-occurrence statistics are generally the best or the second-best, with improvements over frequency in all cases. These trends are observed across all the annotators, thus underlining the adequacy of the operational definitions. We observed that C was more strict in labeling units as intent (markedly lower AP values than A and B). This can be understood from the following example units that are marked as intent by A and B but not by C—`driver, kids, tutorial, program` and `custom`. All of these do carry user intent in queries, but not in a direct fashion like the more general units like `movies`, `define` and `games` (labeled as intent by all three).

Intent units which tend to occur at the beginning of the query have low LCC and LCE (e.g. `how to`, `what does` and `define`). Similarly, there are examples like `mp3`, `for sale` and `blog`, which typically occur only at the end in queries, displaying the opposite behavior. Such extreme cases are rare in NL, because words that begin or end a sentence also frequently occur at other positions. Thus, left or right co-occurrence alone are insufficient for extracting intent units in queries, highlighting the importance of total co-occurrence statistics.

Rank adjustments by co-occurrence statistics: In Table 7, we compare the ranks of a few units with respect to the seven different statistics. Content units like `wedding` can have very high frequency owing to the popularity of the event or concept; however, co-occurrence statistics help push such candidates lower down the list (from Rank 138 in frequency to out of the top-200 by all other indicators). Next, we see that intent units like `blog` and `define`, which rank around 500 by frequency move much higher up the ranked list when appropriate co-occurrence statistics are used. Hence, average precision is generally observed to increase for co-occurrence-based features. We note that the rank of `make`

Table 6
Average precision of each of the indicators for intent unit detection in Web queries.

Annotator	Metric	Fr	LCC	LCE	TCC	TCE	RCC	RCE
A	AP@200	0.622	0.654	0.639	**0.696**	0.653	**0.701**	0.668
	AP@500	0.462	0.495	0.498	**0.548**	**0.519**	0.513	0.479
	AP@1000	0.335	0.348	0.331	**0.421**	**0.400**	0.343	0.305
B	AP@200	0.719	0.812	**0.854**	0.850	**0.852**	0.793	0.777
	AP@500	0.528	0.617	0.631	**0.665**	**0.674**	0.590	0.567
	AP@1000	0.381	0.416	0.408	**0.488**	**0.491**	0.388	0.363
C	AP@200	0.434	0.458	0.488	0.490	0.494	**0.542**	**0.535**
	AP@500	0.338	0.361	0.359	**0.401**	0.385	**0.392**	0.381
	AP@1000	0.252	0.261	0.253	**0.322**	**0.308**	0.260	0.243

The two highest values in a row are marked in **boldface**.

Table 7
Ranks assigned to intent units of Web queries by the seven different statistics.

Unit	Fr	LCC	LCE	TCC	TCE	RCC	RCE
`for sale`	16	24	30	27	58	119	2216
`pictures`	48	39	35	56	45	93	53
`mp3`	109	75	93	115	221	487	1712
`blog`	490	164	87	294	127	1323	945
`biography`	824	278	110	561	171	5567	4009
`how to`	4	80	77	8	32	2	11
`wedding`	138	363	377	295	438	240	447
`make a`	188	3953	209 164	213	923	66	40
`what does`	316	2275	1 517	174	734	56	294
`define`	503	1727	1 098	199	51	70	22

Intent units in the upper and lower halves are pulled up higher by left and right co-occurrence statistics respectively. Total co-occurrence statistics are seen to have a moderating effect between the two extremes.

`a` by LCE is 209 164. This is because `make a` is preceded by only a handful of segments like `how to` or `way to`. Thus, it has a very restricted left co-occurrence distribution and hence a very low LCE. This pushes its rank by LCE so far down. Other indicators are seen to have balancing effects on words with such skewed distributions.

A note on segmentation errors: First names like `james` co-occur with several different family names and acquire a high rank (Table 5). We would not have observed them this high up in the lists had the segmentation algorithm always been able to group together entire names. For example, popular figures like *james bond* and *james cook* do get grouped together, and as units they do not have such high co-occurrence statistics.

A note on IDF for queries: The concept of IDF (Section 2.5) cannot be explored in the context of intent word detection in Web queries (Section 3) because even though each query can be considered as a sentence, the concept of a (coherent) *document* is not well-defined. The only notion that comes close is grouping the queries from a single user *session* as a document. However, session segmentation of a query stream is an active area of research [37,38] and is beyond the scope of this work.

4. Labeling intent units in query context

A segment can act as content or intent in a query depending upon the context. For example, while the segment `video` behaves as an intent unit in most queries, like, `(us open) (video)` (specifying that the desired content type is a video), it is the content unit in the query `(definition of) (video)`. Thus, a labeling scheme is practically useful only if it can label segments as content or intent within a query, and not just in a context-agnostic standalone fashion. We note here that this is not true for NL function words. The concept of function words is independent of sentence context. This is an important point of difference between the concepts of function and intent words. In this work, for simplicity, we restrict ourselves to labeling two-segment queries; extension to multi-segment queries is an important future work. Interestingly, two-segment queries (derived from the output of the segmentation algorithm in Saha Roy et al. [19]) form a significant proportion of our Bing log (≃44%) (Section 3.1).

As the first step, we define an *intent-ness score* $\mathrm{IS}(u)$ for every unit u that appears in the query log. Since all our indicators hold clues towards the *intent-ness* of a unit, this score is calculated as a simple log-linear combination of the indicators as

$$\begin{aligned}\mathrm{IS}(u) &= \log_2(\mathrm{Fr}(u)) + \log_2(\mathrm{LCC}(u)) + \mathrm{LCE}(u) \\ &\quad + \log_2(\mathrm{TCC}(u)) + \mathrm{TCE}(u) + \log_2(\mathrm{RCC}(u)) + \mathrm{RCE}(u). \quad (8)\end{aligned}$$

Logarithms of Fr, LCC, TCC and RCC are taken to make them comparable in value to the entropies (cf. Eq. (3)), which are already in logarithmic space. Since intent units are expected to obtain higher individual feature values than content units, the former is also expected to achieve higher intent-ness scores. However, we understand that there could be more appropriate methods of feature combination [39] like learning weights with linear regression models, but such methods require supervision (while all the techniques used in this research are unsupervised) and will require detailed experimentation.

Algorithm: The segment with the lower IS in a query is marked as content (\c). The intuition behind this is that a query must have at least one content unit, and the IS of an average content

Table 8
General examples of segmented and labeled queries.

Human labeled query	Machine labeled query
`(roger federer)\c (pics)\i`	`(roger federer)\c (pics)\i`
`(cranes)\c (for sale)\i`	`(cranes)\c (for sale)\i`
`(star trek)\c (wikipedia)\i`	`(star trek)\c (wikipedia)\i`
`(britney)\c (biography)\i`	`(britney)\c (biography)\c`
`(ethan hawke)\c (movies)\i`	`(ethan) (hawke) (movies)`[a,b]
`(adobe flash)\c (download)\i`	`(adobe flash)\c (download)\i`
`(free)\i (video converters)\c`	`(free video)\i (converters)\c`[a]
`(hotels)\c (near)\i (airport)\c`	`(hotels) (near) (airport)`[b]

[a] Error in segmentation algorithm.
[b] Machine unable to label more than two-segment queries.

unit is expected to be lower than that of an intent unit. If the score of the other unit in the query exceeds that of a user-defined threshold δ, it is marked as intent (\i). Otherwise, the second unit is also labeled as content. Since the absolute number of intent units in the query log is expected to be low in comparison to the number of content units, simply labeling the unit with the higher IS as intent, without a threshold, would result in too many false positives. We note that if the intent-ness score (IS) of a segment is below the threshold δ, it will always be labeled as content. Obtaining an intent-ness score below the threshold essentially means that there is insufficient evidence in the query log for labeling this unit as intent. Thus, our tagging algorithm labels two-segment queries as *either* content segment–intent segment (equivalently intent segment–content segment), *or* as content segment–content segment. We denote the first set of queries as *content–intent queries* (like `(brad pitt)\c (home page)\i`, `(pictures of)\i (digestive system)\c` and `(how to)\i (paraglide)\c`) and the second set of queries to be *content–content queries* (like `(brad pitt)\c (villa costanza)\c`, `(digestive system)\c (enzymes)\c` and `(paraglide)\c (safety equipment)\c`).

4.1. Evaluating in-query labeling using human annotations

Experiment. Our test data comprised of 2600 unique two-segment queries (segmented by the algorithm in Saha Roy et al. [19]), randomly sampled from all the two-segment queries in our entire Bing log. These queries were not used for training. We asked our three annotators A, B and C, who had previously annotated individual segments (Section 3.3), to annotate 1000 queries each by marking the segments as content or intent units, as they deem fit, in accordance with the operational definitions. The annotators were asked to label content and intent segments *in queries* according to our operational definitions. A segment was to be labeled as *content* by the annotator if: (a) the segment represented the core information need or the topic of the query; (b) removing the segment made the query lose its central idea; and (c) if it was necessary that the segment had to matched in the document text for relevance. A segment was to be labeled as *intent* by the annotator if: (a) they specifically carried user intent about the other segment; and (b) if relevant pages can be found even if the segment does not match exactly in the document text (the annotator had to conceive of some way of fetching relevant pages without exact matching). Additionally, it was mandated that a query must have at least one content segment. If the segmentation was incorrect, they were supposed to provide the correct segmentation and then mark the content and intent units. Queries that had more or less than two segments after annotation were not considered for further steps. In order to measure inter-annotator agreement (IAA), we had ensured that there are 200 queries common for all the annotators A, B and C ($(1000 - 200) \times 3 + 200 = 2600$) queries. Some general sample annotations, not restricted to this dataset of two-segment queries, are shown in Table 8.

For content unit labeling in queries, in general, our method can be improved by using rules and resources for identifying names of people (like `ethan hawke`), organizations (like `world health organization`), places (like `isle of wight`), etc. using named entity (NE) lists such as Yago, DBpedia and Freebase. But for practical applications, it would be imperative to fine-tune the algorithm using such rules and named entity recognition (NER) in queries [40]. Usually lists will work only for the relatively well-known entities, and if our segmentation algorithm can correctly group (rare or popular) entities, our content–intent tagger will also make the correct decision most of the time as such entities will have restricted co-occurrence distributions and will be correctly marked as content, even if it does not appear on the popular NE lists.

4.1.1. Results and observations

Percentage Inter-annotator Agreement (IAA) on the labels, i.e., percentages of units on which annotators agree on the content–intent labels, are 83.99, 77.06 and 77.32 for $A - B$, $B - C$, and $C - A$ respectively. For all annotators, about 70% of the units are marked as content and the rest 30% as intent. The corresponding values for Cohen's Kappa (κ) [41], a stricter metric for IAA that considers the effect of chance agreements, are 0.62, 0.45 and 0.46. A κ close to 0.5 indicates statistically significant IAA between annotators.

For simplicity, from now on we use only the 2400 queries for which we have exactly one annotation, for our analysis. Out of these 2400 queries, 1356 queries (56.5%) were labeled as content–intent and 1044 queries (43.5%) were labeled as content–content by our annotators. We first compute our labeling accuracy by penalizing cases where our algorithm predicts an opposite set of labels for content–intent queries. Results show that our algorithm achieves a labeling accuracy of 78.79% (82.28% for A, 78.67% for B, and 75.43% for C) ($\delta = 13$, as determined through experiments presented later). This means that we predict the opposite set of labels only about 20% of the times; to be specific, for 271 queries (out of 1356 queries). This is particularly high considering that the IAA is also roughly 80%. The mistakes typically occur in those cases where the content unit is very popular and achieves a significantly high intentness score, while the intent unit is relatively uncommon. For example, in the query `(finland) (bed and breakfast)`, `finland` is marked as intent by the annotator and `bed and breakfast` as content, while our algorithm labels wrongly as the reverse. According to our framework, `bed and breakfast` is the main topic of the query and hence acts as content, whereas the location `finland` represents user intent (see *source* specifiers, Section 5).

Effect of threshold: We evaluated the labeling algorithm against the test set at different values of δ. For this purpose, we computed the precision (*Prec*), recall (*Rec*) and F-Score [20] for intent and content units, as defined below.

$$\text{Prec(Intent units)} = \frac{\#\text{(Units correctly labeled as intent)}}{\#\text{(Units labeled as intent)}} \qquad (9)$$

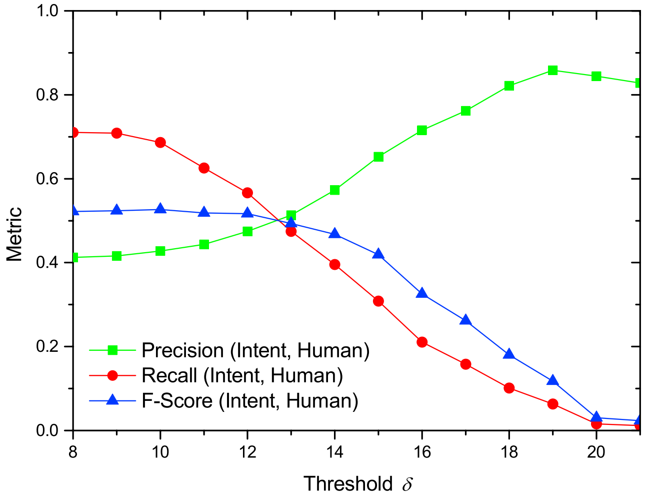

Fig. 4. (Color online) Precision, recall and F-Score for intent units at different thresholds when machine output is evaluated against human annotations.

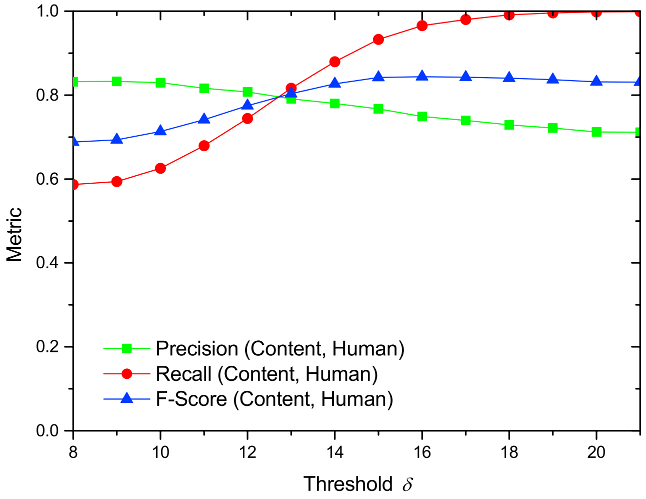

Fig. 5. (Color online) Precision, recall and F-Score for content units at different thresholds when machine output is evaluated against human annotations.

$$\text{Rec(Intent units)} = \frac{\#\text{(Units correctly labeled as intent)}}{\#\text{(Units labeled as intent by annotators)}} \quad (10)$$

$$\text{F-Score(Intent units)} = \frac{2 \times \text{Prec(Intent units)} \times \text{Rec(Intent units)}}{\text{Prec(Intent units)} + \text{Rec(Intent units)}}. \quad (11)$$

The precision, recall and F-score for content units are defined similarly. We note that these metrics are computed by looking at the aggregate pool of content–intent and content–content queries, i.e. all the 2400 queries. Figs. 4 and 5 show the curves obtained when these metrics are plotted by varying δ for intent and content unit detection, respectively. The optimum δ turns out to be about 13 (value used in the previous experiments for computing labeling accuracies). Our content labeling has a much higher precision than intent labeling, but this is correlated to the fact that the natural proportion of content units in a query log is expected to be much higher than that for intent units. As one would expect, there is a trade-off between precision and recall. The precision of intent units increases with δ and vice versa. This indicates that the general theory of our intent-ness score is working well. The opposite trends are observed for content unit detection.

4.2. Evaluating in-query labeling using clickthrough data

Till now, we have postulated and identified the distributional characteristics of the lexical categories of the query language, i.e. content units and intent units. Like in NL, lexical categories in queries must also have their specific *functions*. In fact, our notions of content and intent units are based on their functions, which is *a content unit denotes the core information need of the user* and *an intent unit further modifies the information need in one of many possible ways* (Section 3.2). Can we mathematically model and compute the functional characteristics of these units and provide further evidence for their existence? One possible way to study the functions of the units is to analyze click data. A click is representative of the function or the role of the unit in a query because it leads to the purpose of issuing the query, i.e. land on a (possibly) relevant page.

Human judgments can often be very expensive to obtain on a Web scale. Fortunately, clickthrough logs can also help us in large-scale automatic evaluation of our content–intent labeling algorithm. The basic idea is as follows: Consider two content units c_1 and c_2 (say `tom cruise` and `anjelina jolie`) and two intent units i_1 and i_2 (say `movies` and `home page`). The queries c_1, c_1 i_1 and c_1 i_2 (or c_2, c_2 i_1, and c_2 i_2) are closely related because the core information need, which is c_1 (or c_2), is the same for all of them. Therefore, we can expect to see a good amount of overlap among the URLs clicked for each of them. On the other hand, the queries i_1 (if it makes sense), c_1 i_1 and c_2 i_1 (or i_2, c_1 i_2 and c_2 i_2) are very different in their information needs. Hence, we can expect very little, if not zero, overlap among the URLs clicked for them. Thus, one way to define the *information content* of a unit u is to collect all queries containing u and compute the overlap between clicked URLs for these queries. A low overlap would imply that u is usually an intent unit, and a high overlap indicates that u is generally a content unit. This concept is illustrated through an example in Fig. 6. The exact procedure of using clickthrough logs to arrive at a labeling of a two-segment query is explained next.

4.2.1. Modeling click overlap

A precise quantification of the amount of overlap between two sets of URLs is non-trivial because exact string match to compare URLs is unreliable. For instance, the pair of URLs www.puzzle.com and www.puzzle.com/demo/help.html are very closely related, but do not match exactly at string level. On the other hand, partial string-level matches can also be misleading. For example, URLs en.wikipedia.org/wiki/fox and en.wikipedia.org/wiki/guitar have no logical overlap. Therefore, we first show how to identify the overlaps between pairs of URLs (with respect to a particular query, as in Fig. 6), and then use these overlap values to compute the overlap between two *sets* of URLs. Let a URL $\mathcal{U}$ be created by the concatenation of a number of strings $s_{\mathcal{U}_i}$. Drawing upon intuition, we propose that the overlap between a pair of URLs $\mathcal{X} \equiv s_{\mathcal{X}_1}/s_{\mathcal{X}_2}/s_{\mathcal{X}_3}/\cdots/s_{\mathcal{X}_k}/\cdots/s_{\mathcal{X}_{n_1}}$ and $\mathcal{Y} \equiv s_{\mathcal{Y}_1}/s_{\mathcal{Y}_2}/s_{\mathcal{Y}_3}/\cdots/s_{\mathcal{Y}_k}/\cdots/s_{\mathcal{Y}_{n_2}}$ depends on the following factors: the length (as measured by the number of strings delimited by slashes) of the prefix up to which the URLs match exactly (k), the number of times the URLs have been clicked for the query under consideration (click counts $c_{\mathcal{X}}$ and $c_{\mathcal{Y}}$), the lengths of the URLs n_1 and n_2 (as measured by the number of strings delimited by slashes), and a quantity we term as the Inverse URL frequency (IUF). This last factor is helpful in identifying very general domain prefixes such as which should contribute minimally to the overlap score (cf. the concept of IDF in Section 2.5). We define the IUF of a URL prefix s as follows (cf. Eq. (5) for justification):

$$\text{IUF}(s) = \log_{10} \frac{1 + |U|}{|U_s|} \quad (12)$$

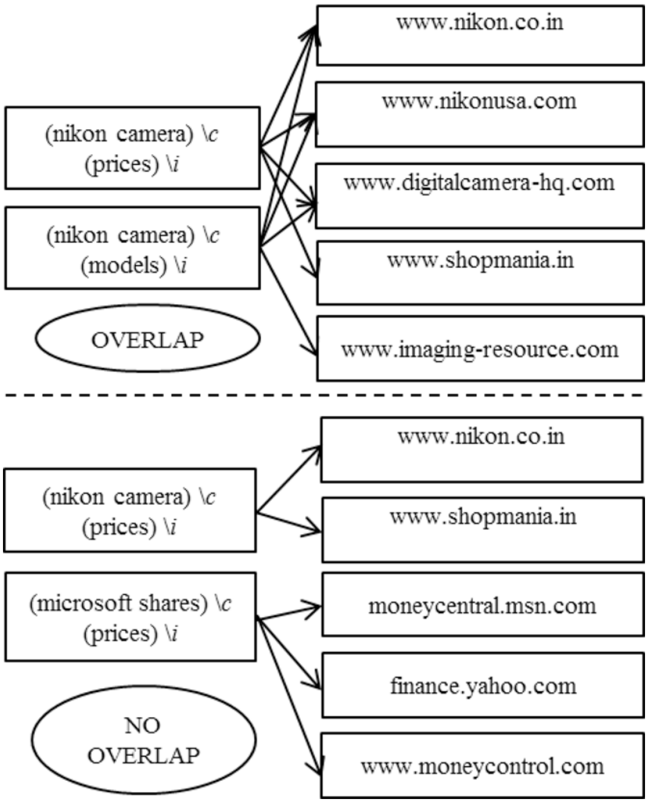

Fig. 6. Illustrating difference in click overlaps in query sets with common content and intent units respectively with a simple example.

where $|U|$ is the number of distinct URLs in our log and $|U_s|$ is the number of distinct URLs with prefix s. The overlap o between $\mathcal{X}$ and $\mathcal{Y}$ is directly proportional to the IUF of the first string of the common prefix ($s_{\mathcal{X}_1}$ or $s_{\mathcal{Y}_1}$), the number of *common* clicks obtained by *both* the URLs ($\min(c_{\mathcal{X}}, c_{\mathcal{Y}})$), and the length of the common prefix (k). On the other hand, it is inversely proportional to the sum of n_1 and n_2, i.e. the sum of the lengths of the two URLs (in terms of constituent strings). For the last factor, we use the mean length of the two URLs as the combining factor, i.e. $\frac{n_1+n_2}{2}$. We thus define the overlap o between $\mathcal{X}$ and $\mathcal{Y}$ as a simple combination of the factors as (assuming the constant of proportionality to be one)

$$
\begin{aligned}
o(\mathcal{X}, \mathcal{Y}) &= \mathrm{IUF}(s_{\mathcal{X}_1}) \times \min(c_{\mathcal{X}}, c_{\mathcal{Y}}) \times k \times \frac{1}{\frac{n_1+n_2}{2}} \\
&= \mathrm{IUF}(s_{\mathcal{X}_1}) \times \min(c_{\mathcal{X}}, c_{\mathcal{Y}}) \times \frac{2k}{n_1 + n_2}. \qquad (13)
\end{aligned}
$$

The contributing factors could be combined in a better way to define the resultant overlap as future work. To compute the click overlap of a set of URLs S, we compute the mean of the pairwise overlaps of all URLs in S. For each content or intent unit u, a value of o can thus be derived. For a given two-segment query q, the unit with the lower overlap $o(u)$ is treated as an intent unit, and the one with the higher $o(u)$ as content.

4.2.2. *Results and observations*

The identification of the different behaviors of click overlaps for content and intent units opens up the possibility of not being tied to manual annotations for evaluation. We checked the percentage IAA of labeling done using click overlap formulation (unit with lower overlap is intent, the other is content) with the manual annotators and found it to be 73.09%, 71.65% and 68.23% for A, B and C respectively, which are similar to our earlier IAA values (Section 4.1). We then checked the precision, recall and F-Score (Eqs. (9) through (11)) for our labeling algorithm with the output produced by click data modeling. The definition of recall, however, is appropriately modified to

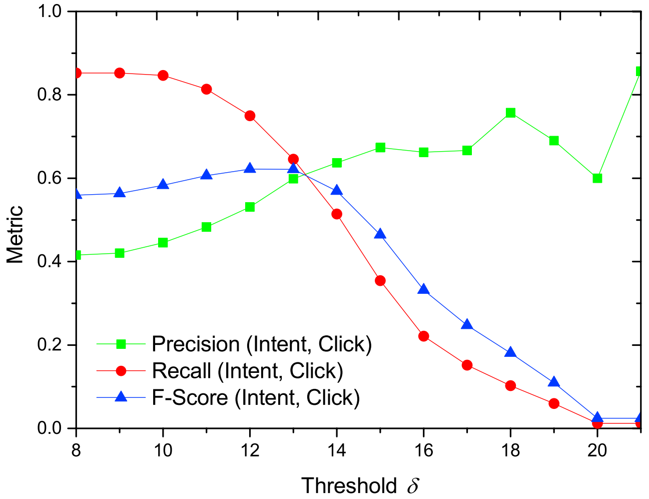

Fig. 7. (Color online) Precision, recall and F-Score for intent units at different thresholds when machine output is evaluated against click data.

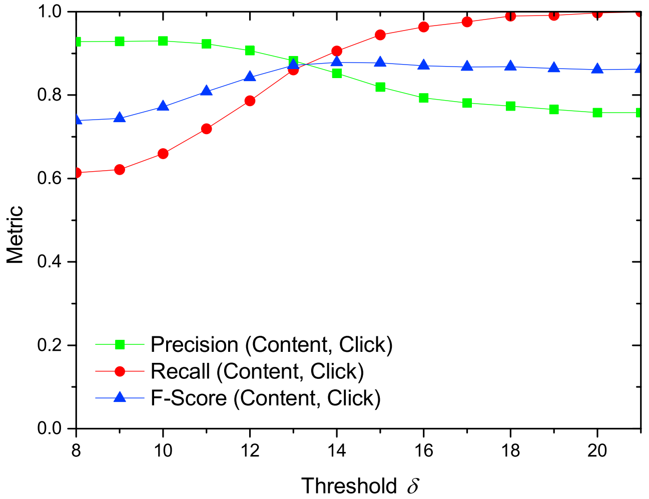

Fig. 8. (Color online) Precision, recall and F-Score for content units at different thresholds when machine output is evaluated against click data.

$$
\mathrm{Rec(Intent\ units)} = \frac{\#(\text{Units correctly labeled as intent})}{\#(\text{Units labeled as intent by click data})}. \qquad (14)
$$

A similar change is made for content recall. Figs. 7 and 8 show corresponding plots obtained by varying threshold δ, for intent and content units respectively. These results are markedly similar to the results produced by evaluating against human annotated data (Figs. 4 and 5), which justifies our choice of using clickthrough data as an alternative evaluation strategy.

We also computed click overlap ratios during our experimentation. We found that the click overlap values range from a minimum of 0.0011 to a maximum of 1.7825 (mean = 0.0818, standard deviation = 0.1664). The differences in overlap values are found to lie in the range $[-1.7725, 1.5830]$ with mean and standard deviation as 0.0453 and 0.2303 respectively. The range for the overlap ratios turned out to be $[0.0205, 417.5886]$ and the mean and standard deviation were found to be 9.6975 and 24.7345 respectively. Thus, the difference had a much smaller range than the ratio.

However, we note that neither the difference nor the ratio helps us in taking a decision on the segment labels; it is only the magnitudes of the overlaps that can help. The hypothesis remains that the content unit will have a higher overlap and the intent unit a lower one, implying $overlap(c)/overlap(i) > 1$ (for click overlap ratios) or $overlap(c) - overlap(i) > 0$ (for click overlap differences).

We wish to clarify that even in using the principle of labeling the unit with higher overlap as content and the other as intent, context-dependence is taken into account. As an example, let us take the real query `(most popular) (youtube video)`. While `most popular` had an overlap score of 0.0141, `youtube video` had a score of 0.0992. Thus, `youtube video` was labeled as content and `most popular` as intent here (we believe rightly so). On the other hand, the click overlap score for the unit `murray river` was found to be 0.1550. Thus, if the query had been `(murray river) (youtube video)`, `murray river` would have been labeled as content, and now `youtube video` would get the intent label (again, rightly so).

Choosing the segment with the higher overlap score as content precludes the possibility of both segments getting labeled as content. The situation of both segments getting labeled as content, however, can only be incorporated with the help of a *threshold* on the difference or the ratio. Since our intention was to use click logs as a gold standard, the only way we could learn an appropriate threshold was to use a new and independent source of information, which we have not considered in this research. This can definitely be one of the points of future improvement for this work.

4.3. Verification of the operational definitions

While every relevant document for a query must contain the content units, this is not necessarily true for intent units. For example, in the query `(jaguar x8) (for sale)`, the user expects every relevant document to contain the content unit `jaguar x8`, but this is not true of the intent unit `for sale`. This was the basis on which our operational definitions for content and intent units were formulated. We verified the validity of this notion on a very recent corpus released by Saha Roy et al. [19]. The dataset[4] consists of 500 Web search queries with associated documents and relevance judgments (RJs) (approximately 30 per query, 0–2 scale, average rating of three annotators). The corpus consists of a total of 13 959 documents. Since this dataset contains queries accompanied by relevant documents, it is appropriate for verifying our operational definitions. This dataset was constructed for evaluating query segmentation and thus also contains segmentation annotations from various algorithms and humans.

We used the segmented versions of the queries as output by the algorithm in Saha Roy et al. [19], and subsequently labeled the 383 two-segment queries with content and intent tags using our algorithm. We now wish to observe the presence, and the distribution, of content and intent segments in relevant documents associated with each query. Since exact string matching for segments in documents can often be misleading (the segment `hp aio printers` can be present in the document as `hp printer aio`), we formulated the following three-point (0–2) scoring criteria for *approximate* segment matches in documents. Exact and exact stemmed matches (each word of the segment stemmed by the Porter Stemmer [42]) would be rated as SM 2 (Segment Match Grade 2). SM 1 is awarded if the stemmed segment was present in a *modified* form in the document. We define segment *modification* as a 1-insertion, a 1-substitution, a 1-deletion or a 1-transposition at *one position* of the stemmed form. The above operations, as applied to a multiword expression $\mathcal{M} = \langle a\ b\ c\ d\rangle$, are explained next. We note that there are some overlaps among these sets, but since all are assigned the same score (SM 1), it does not make a difference. We do not deal with n-modifications in this work, where $n > 1$.

Table 9
Segment match versus document relevance for content units.

Content	SM 0	SM 1	SM 2
RJ 0	19.113	2.647	7.285
RJ 1	21.566	3.844	13.491
RJ 2	15.328	3.022	13.704

Table 10
Segment match versus document relevance for intent units.

Intent	SM 0	SM 1	SM 2
RJ 0	20.458	0.360	4.530
RJ 1	30.747	0.390	6.959
RJ 2	27.127	0.630	8.799

- 1-insertions: All new segments formed by inserting one new word in an intermediate position of the original segment. 1-insertions for $\mathcal{M} = \{\langle a\ x\ b\ c\ d\rangle, \langle a\ b\ x\ c\ d\rangle, \langle a\ b\ c\ x\ d\rangle\}$, where x is any word.
- 1-substitutions: All new segments formed by substituting one word in the original segment by a new word. 1-substitutions for $\mathcal{M} = \{\langle x\ b\ c\ d\rangle, \langle a\ x\ c\ d\rangle, \langle a\ b\ x\ d\rangle, \langle a\ b\ c\ x\rangle\}$, where x is any word.
- 1-deletions: All new segments formed by deleting one word from the original segment. 1-deletions for $\mathcal{M} = \{\langle b\ c\ d\rangle, \langle a\ c\ d\rangle, \langle a\ b\ c\rangle, \langle a,\ b,\ d\ \rangle\}$.
- 1-transpositions: All new segments formed by swapping the positions of one pair of adjacent words in the original segment. 1-transpositions for $\mathcal{M} = \{\langle b\ a\ c\ d\rangle, \langle a\ c\ b\ d\rangle, \langle a\ b\ d\ c\rangle\}$.

If the segment is not present in the documents in any of these forms, the case is rated as SM 0. Next, each segment of each query is searched in the document text of each document in the query pool (average pool depth for this dataset is about 30) and the subsequent match (or non-match) is rated as SM 0, SM 1 or SM 2. As mentioned earlier, each document in the pool is associated with an RJ of 0 (non-relevant), 1 (partially relevant) or 2 (relevant). Since each segment searched is tagged as content or intent, we can now build the following 3 × 3 matrices for degree-of-match versus degree-of-relevance, accumulated for all segments of a particular type (Tables 9 and 10).

The absolute counts of the specific cases in the matrix cells were normalized by the sum of the values in the entire table, and converted into percentages. The first rows of the tables are grayed out because matches in non-relevant documents are not of interest to us. The second and the third rows imply that the document was at least partially relevant to the query. If we consider exact and partial matches (SM 1 or SM 2) for these two rows, we see that the corresponding total percentage for content units ($\simeq$34%) is almost double of that for intent units ($\simeq$17%). Moreover, we note how the absence of segments affects document relevance. For content segments, in only $\simeq$37% cases was the document at least partially relevant (RJ 1 or RJ 2) when the segment was absent in the document, while the corresponding number for intent segments is as high as $\simeq$58%. Both of these observations indicate that while matching a content segment in a document is crucial to improving IR performance, an intent segment *need not* always match (exactly or partially) for the document to be relevant—thus validating our operational definitions. It is important to note that the way current Web documents and commercial search engines are designed (emphasizing presence or match of keywords), it is very difficult to obtain substantial evidence for pages that do not contain the intent units and yet are relevant to the query. However, it is intuitive that such pages exist on the Web, and one of the main objectives of semantic search is to discover these pages.

[4] http://bit.ly/ZS0ybI.

4.4. Use of content and intent labeling in IR

Labeling segments as content or intent is only half of the work required for our ideas to be useful in a practical scenario. The second half is the functional aspect, i.e. to be somehow able to use these labels during the IR process for better ranking or result presentation. We note that there can be several ways of doing this, and search engines are possibly doing some of these today. For example, specifying `video` or `pics` or `map` with content units almost certainly puts video or image or map content at the top, instead of the usual "ten blue links".

We devise a simple and generic application for our labeling strategy, in line with our operational definitions of content and intent. Our intuition lies in the definitions themselves: while content segments need to be matched exactly within documents, intent units need not match exactly in the document text for relevance. Current search engines support use of the double quotes operator ("...") to force exact phrase match in the document. Exact match refers to perfect ordering of segment words in the document, without word insertions, deletions, transpositions, substitutions or other linguistically informed flexible matching criteria (like synonyms). However, it is known that users rarely use quotes in their queries to use this feature (only about 8% of queries in our Bing log), while a much larger fraction of queries (about 71% as reported in Guo et al. 2009 [40]) do have named entities or multiword expressions (`roger federer`, `summa cum laude`) within them. It could also be detrimental to put quotes indiscriminately around all segments. In our opinion, for example, it would be harmful to ensure exact match for intent segments like `how to` or `difference between`, because a page can contain the procedure for something or comparison between items (say, as a table) without having these exact words. Thus, developing an automatic selective quoting strategy based on content and intent markup could be a good way of putting our work to use. To summarize, we state that content units must be quoted while intent units should not be enclosed within double quotes during the search process. Note that quoting for ensuring exact word ordering is meaningful only for multiword segments, as quoting single word units only differentiates between stemmed and unstemmed word forms (like `brown` and `browning`). In our previous work [19], we had shown with an oracle-based approach that quoting helps improve IR performance, but a deterministic quoting strategy is yet to be discovered. We believe that content–intent labeling is the first step towards such a strategy.

With reference to the state-of-the-art, we note here that none of the four state-of-the-art researches that tag queries with content–intent like labels [43–46] provide an IR-based evaluation for their approaches. Moreover, schemes that do use some sort of query tagging to improve retrieval, do not report results on a single dataset so as to be comparable among each other. Thus, we select Microsoft Bing Web Search, a commercial search engine, as our state-of-the-art baseline, accessible through its API.[5] This provides a very challenging baseline, and if we are able to show IR improvement on a reasonable proportion of queries over the Bing API, our method can be said to have substantial practical significance.

We run experiments with a dataset[6] that contains 500 queries (5–8 words), a corpus of 13 959 documents, and about 30 relevance judgments per query (0–2 scale; three annotators). This dataset was released along with our previous work [19] (mentioned in Section 4.3). Most queries (383 out of 500) consist of two segments only (according to our segmentation algorithm), which are labeled for content and intent segments by our method. 123 out of these 383 queries have one intent segment and one content segment (remaining 260 are content–content), which forms our final evaluation set. For each of these 123 content–intent queries, we generate the following query variants: (a) both content and intent segments are in quotes (`c-q i-q`); (b) content segment is in quotes and intent segment is unquoted (`c-q i-u`); and (c) content segment is in quotes and the intent segment is deleted (`c-q i-d`). Among these, `c-q  i-u` and `c-q  i-d` can be said to be "our" methods as `c-q i-q` can be generated without the tagging step by simply quoting both segments. We subsequently use the Microsoft Bing Search API to search our document collection. Essentially, we use the Bing search API to retrieve the top-10 URLs from the Web for our query versions (three quoting variants and the original query) and then search our corpus for these URLs and their corresponding relevance judgments. Since the original corpus was also constructed using the Bing API [19], all the documents and most of the corresponding relevance judgments were found in the dataset. Next, we compute well-established IR metrics of Normalized Discounted Cumulative Gain (nDCG) [47] and Mean Average Precision (MAP)[7] [20] for each query, and report averaged values in Table 11. Specifically, nDCG is computed after observing the first ten results only, as happens in a typical Web search scenario, and hence we report nDCG@10. The results are computed for each of the three annotators (named X, Y and Z) and their mean rating, all of which are available in our dataset. We compute the following three statistics for each quoting variant (represented by the three sets of columns in Table 11): (a) percentage of queries on which the variant improves over the original query; (b) mean metric value (nDCG@10 or MAP) for the variant; and (c) the mean metric gain over the original query for improved queries. In addition to the three variants, we compute these values for the column `Max(c-q i-u, c-q i-d)` that represents the better of the two variants `c-q  i-u` and `c-q  i-d` in terms of the metric value (nDCG@10 or MAP). If this strategy gives the best results among the rest, we can say that content–intent labeling has the *potential* for producing substantial improvement over the original query, even with a very strong baseline. `Max(c-q  i-u,  c-q i-d)` can be considered as a disjunction of `c-q  i-u` and `c-q i-d`. The percentage of queries improved over the original version for `Max(c-q i-u, c-q i-d)` corresponds to the percentage of queries that improved either with `c-q i-u` or with `c-q i-d`.

We make the following important observations from Table 11: (a) `c-q  i-u` and `c-q  i-d` together can improve nDCG for more than 50% queries (64 out of 123 queries for mean rating) has the best performance; (b) `c-q  i-u` and `c-q  i-d` together result in higher metric gain (both metrics) over the original query than `c-q  i-q`; and (c) `c-q  i-u` generally has the highest IR performance among the three variants. For case (b), we note that taking `Max(c-q  i-u,  c-q  i-d)` increases the number of improved queries, and hence the mean of `Max(c-q  i-u,  c-q  i-d)` can fall below the mean for `c-q  i-d`. For both metrics, for a large majority of the cases (17 out of 24 cases), the `Max(c-q  i-u,  c-q i-d)` version achieves the best results, and the gains are often statistically significant (applicable for the second and the third sets of columns, 8 out of 16 cases). These results show that tagging segments as content or intent can be leveraged for good IR performance. It is heartening to see that our deterministic `c-q  i-u` variant generally achieves the second best performance for the left and the middle sets of columns of percentage queries improved and mean metric values (i.e., the best among the first three columns of deterministic variants in each set) (13 out of 16 cases).

[5] http://datamarket.azure.com/dataset/bing/search.

[6] http://bit.ly/1i6F4Q0.

[7] For computing MAP, relevance judgment ratings of 2 were treated as "relevant" while ratings of 0 and 1 were considered as "non-relevant".

Table 11
Retrieval-based evaluation of content–intent labeling for two-segment queries using the Bing API.

Annotator	Percentage of queries improved over original version				Mean metric value of quoting strategy				Mean metric gain over original version			
nDCG@10	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`
X	34.150	34.960	24.390	**47.970**	0.771	0.807	0.542	**0.882**[a]	0.196	0.176	**0.227**[a]	0.212
Y	39.840	39.020	22.760	**50.410**	0.709	0.755	0.552	**0.818**[a]	0.101	0.105	**0.202**[a]	0.154
Z	34.960	39.020	18.700	**50.410**	0.723	0.797	0.531	**0.858**[a]	0.145	0.157	**0.213**[a]	0.185
Mean	34.960	40.650	21.140	**52.030**	0.772	0.830	0.575	**0.890**[a]	0.151	0.132	**0.195**[a]	0.161
MAP	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`	`c-q i-q`	`c-q i-u`	`c-q i-d`	`Max(c-q i-u, c-q i-d)`
X	42.280	38.210	14.630	**47.150**	0.567	0.589	0.230	**0.625**[a]	0.176	0.182	0.170	**0.184**
Y	31.710	30.890	14.630	**40.650**	0.343	0.370	0.145	**0.392**[a]	0.102	0.098	**0.126**	0.109
Z	38.210	46.340	15.450	**56.100**	0.478	0.528	0.217	**0.568**[a]	0.126	0.116	**0.142**[a]	0.128
Mean	32.520	34.150	8.940	**39.020**	0.359	0.380	0.112	**0.397**[a]	**0.126**	0.105	0.090	0.108

The highest value within each set of columns is marked in **boldface**. The 2-tailed paired t-test was performed and the null hypothesis was rejected if $p < 0.05$ (applicable only for the middle and the right sets of columns).

[a] Statistical significance of the highest value within a set of columns over the next best.

This is a direct validation of the success of our operational definition that intent segments need not match exactly within text of relevant documents. We also see evidence that intent segments are not always "deletable" and while they need not match exactly in document text, or can even be absent, they can be used by the search engine in other several different ways. This is apparent from the result that even though the `c-q i-d` variant on its own generally performs the poorest among the three variants (16 out of 16 cases), yet for the queries that it improves upon (21%–24% on nDCG, 9%–15% on MAP), the gain is quite substantial. This is seen from the performance of this variant in the third set of columns, where it is usually the best among the four variants (6 out of 8 cases).

5. A taxonomy of intent units in Web search queries

Roles of units: In order to better understand the roles of intent units in queries, we went through the list of intent units and several hundreds of queries in which they occur. Our study reveals that intent units in Web search queries can be broadly thought of as performing one of two tasks, namely, *restrict* or *rank*. The *restrict* task is concerned with filtering the pool of relevant documents from which the final results are presented. The *rank* task determines the order in which the final results are displayed. These broad categories can be further subdivided into classes as shown in Fig. 9 and Table 12. In some cases, the distinction between restrict and rank tasks begins to blur, and consequently, the table also presents examples for the *restrict* + *rank* category.

The restrict class: In the *restrict* category, *context specifiers* act as disambiguators for the rest of the query (`book, movie`). Similarly, *operation specifiers* are generic action units that specify some action to be performed on or with the content unit(s) (`download, install`). They act like an operator with one or more content units as arguments, thus often behaving like unary, binary or multi-nary relations. The intent units in the *other aspects* sub-category mainly specify aspects of particular classes of content (like medicines (`side effects`) and songs (`lyrics`)), in which the user is interested.

The rank class: In the *rank* category, *sort order specifiers* indicate that results can be ranked by a parameter of the content unit(s). For example, `near` or `cheap` specifies that results can be ranked in order of some distance or price respectively. *Time specifiers* are used when users have a preference about *when* the pages were published (the `latest` news or `recent` updates about events or products). Most adjectives fall in the rank category, e.g., `free`, `public` and `printable`. These intent units specify the user's preference as to which of the retrieved pages must be ranked higher in the final results' list.

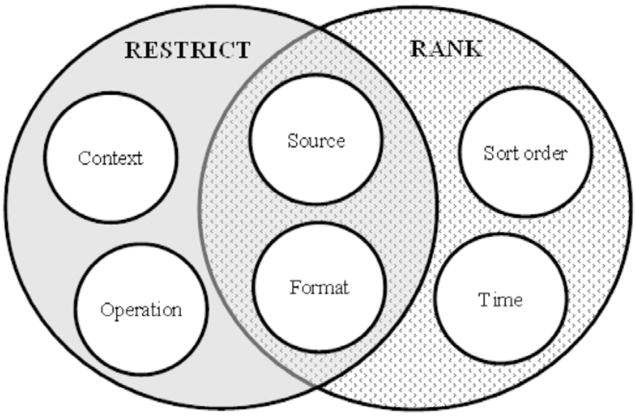

Fig. 9. A Venn diagram for the intent unit taxonomy.

The intersection class: The intent units in the intersection of these classes can help in both restrict and rank tasks. For example, *source specifiers* indicate from where the user wants result pages to be retrieved from. Real source specifiers are geographical locations (mostly names of countries like `germany` or `australia`). Similarly, virtual source specifiers indicate online sources (like `wikipedia` or `ebay`). *Format specifiers* indicate explicit output formats for the results. They may be direct (file extensions like `pdf` or `mp3`) or indirect (`photos of` and `videos of`). We propose that these units belong to the *restrict* + *rank* category because while they try to *restrict* pages to the desired source or type, they also help in the *ranking* of the other results (lower than desired pages). If the desired pages are not available, then the other pages are ranked higher. In either case, the user (generally) still only *prefers* pages of the desired type, and will often look at alternative sources or types if the earlier content was not satisfactory. For example, consider a common source specifier unit such as `wikipedia`. The user may be only interested in Wikipedia articles (restrict task). Alternatively, the user may just *prefer* a Wikipedia article, but is willing to consider results from other sources as well (rank task).

Table 12
Examples of intent units from each class of our intent taxonomy.

Restrict			Restrict + Rank		Rank		
Context	Operation	Other aspects	Source	Format	Sort order	Time	Other preferences
book	how to	side effects	wikipedia	pdf	near	latest	online
movie	what is	benefits	youtube	mp3	cheap	recent	free
game	where are	reviews	espncricinfo	slides	fast	2012	downloadable
tv show	download	biography	ebay	videos	large	new	public
ps2	compare	obituary	bestbuy	pictures	close to	current	exclusive
soap	difference between	history	facebook	photos	high-res	last 24 h	private
windows	buy	applications	linkedin	images	shortest	today	black
scientist	upload	recipe	australia	ppt	budget	now	best
footballer	install	lyrics	india	map	popular	last month	printable
actor	who is	cheats	us	torrent	best-selling	this week	widescreen

Discussion: We observe that intent words play very important and diverse roles in Web search queries. Sometimes this distinction of intent from content can become ambiguous. For example, take the query (facebook) (wikipedia), where the user could be looking for the Facebook (content) entry in Wikipedia (intent), or the Wikipedia (content) page on Facebook (intent). Therefore, detection of intent units and understanding their role is very important for IR. A particularly useful scenario for applying our methods is *enterprise search*, i.e., searching the entire collection of documents belonging to a particular enterprise (mostly) by its employees. The collection of user intents (and consequently the set of intent units and its distribution) is expected to vary from one enterprise to another. Since additional information such as clickthrough data may not be available (or may be very sparse), often query logs are the only resources for intent analysis in enterprise search. Classification of intent units according to our taxonomy can help in identifying the most important needs within the enterprise. Moreover, our taxonomy can also be used for intent diversification, triggering advertisements in sponsored search, and generating query suggestions. Since the relevance of this taxonomy is mostly application-centric, an *evaluation* of the taxonomy is best conducted through appropriate end-to-end applications by the administrators of the deploying systems.

5.1. Related work

We emphasize that our notion of intent units does not contradict but supports or subsumes much of the related efforts in this area, which use Web documents, query logs and knowledge bases. One such line of research is on the automatic acquisition of *attributes* of *classes* or *instances* [36,48–55]. Our method captures several attributes, like side effects (of medicines), biography (of important people) and recipes (of dishes). However, our technique also detects intent units like compare and how to, which do not fit in with the current framework of *class-instance-attribute*. Similarities can be observed in the nomenclature of Li [44], where the author states that noun phrase queries are composed of *intent heads* (like cast) and *intent modifiers* (like alice in wonderland). Intent heads are closely related to attributes and our intent units. Our framework is not limited to noun phrase queries, and can explain other queries like (how to)\i (meditate)\c. A framework and taxonomy using *entities* and *intent phrases* have been proposed for understanding name entity queries in Yin and Shah [45]—but our framework is more generic in the sense that it is not restricted to name entity queries only. The motivation of our work is also fundamentally different from the previous studies. Our notion of intent units largely agrees with the term *intent words* [45,46], proposed for specific domains like actors, musicians, cities and national parks. Similar is the case with *modifiers* [43], which are proposed to carry user intent within queries (as opposed to the query *kernel*). Again, our framework applies for all domains of queries and our unsupervised method using co-occurrence statistics can be considered as a low-cost open-domain [48,56] information extraction technique to detect all categories of such attributes, intent words, intent phrases, intent heads and modifiers.

5.1.1. Intent units as explicit facet indicators

It has recently been proposed that *query intent* can be represented through a set of *facets*, like spatial and time sensitivities, genre, topic and scope [57,58]. These are aspects that can be attributed to the query as a whole. Proper identification of facets has been shown to improve query intent classification [58]. We argue that query intent units mined through our technique are actually *words or segments that the user has included in the query to explicitly indicate his or her intent*, and there is often a one-to-one correspondence between the facets [58] and our intent units. It is important to note that a segment when behaving as an intent unit can indicate multiple facets at the same time. For example, the unit mp3 can tell us both that the query is from the *topic* of *music* and that the user has the *objective* of finding a *resource*. Similarly, presence of *imdb* indicates the facets {*topic*: *movies*} and {*authority sensitivity*: *yes*}. We believe that intent units can be very useful features for query intent classification, and can deepen our understanding of user intent. Thus, classification of our intent units into various facet classes and using them as features for intent classification are promising directions for future research.

6. Conclusions and future work

In this paper, we have proposed that intents units can act as indicators of user intent in Web search queries. We have shown that co-occurrence distributions of units can be leveraged for unsupervised mining of intent units from query logs. We have established the effectiveness of our method by using similar techniques for detecting function words in NL text, which share similar corpus distributional properties with intent words of search queries. As our techniques do not use any specific domain knowledge, they are very suitable for open domain information extraction [56,54,48,49]. Results obtained by our generic and lightweight method have been validated by independent evaluations with human annotations and clickthrough data. A more principled way of combining our different features for computing the intent-ness score remains an important future work. A comprehensive classification scheme for mined intent units has been presented, providing readers with a qualitative analysis of the nature of such units. We have proposed that intent units broadly serve two important functions in IR—*restrict* and *rank* final result pages.

This paper aims at consolidating several ongoing works on associating intents with query words by providing an overarching framework, and opens up several major avenues for directing future efforts. These can broadly be classified into two areas: (a) seamlessly integrating intelligent techniques into search systems that allow for special treatment of intent units to serve better pages; and (b) developing automatic classifiers for assigning detected intent units to their respective categories. Like all aspects of semantic search, problems of vagueness and evaluation pose stiff challenges in these directions. Our paper attempts to be a stepping stone in pinning down such difficulties to focused areas and making them addressable by the concerted efforts of the community.

Acknowledgments

The first author was supported by Microsoft Corporation and Microsoft Research India under the Microsoft Research India Ph.D. Fellowship Award (Grant No. 8901901/IITKGP-Fship-2011). We would like to thank Nikita Mishra, Amritayan Nayak, Dastagiri Reddy and Anusha Suresh (Indian Institute of Technology Kharagpur, India), and Pavan Kumar and Rohit Kumar (National Institute of Technology Durgapur, India), for valuable inputs at various stages of this work. Finally, we thank anonymous reviewers of this paper whose insightful comments helped us a lot in improving this work.

References

[1] P. Haase, D. Herzig, M. Musen, T. Tran, Semantic Wiki search, in: The Semantic Web: Research and Applications, Springer, 2009, pp. 445–460.

[2] D. Tumer, M.A. Shah, Y. Bitirim, An empirical evaluation on semantic search performance of keyword-based and semantic search engines: Google, Yahoo, MSN and Hakia, in: Internet Monitoring and Protection, 2009, ICIMP'09, Fourth International Conference on IEEE, 2009, pp. 51–55.

[3] S. Ferré, A. Hermann, Semantic search: Reconciling expressive querying and exploratory search, in: The Semantic Web–ISWC 2011, Springer, 2011, pp. 177–192.

[4] A. Broder, A taxonomy of Web search, SIGIR Forum 36 (2002) 3–10.

[5] U. Lee, Z. Liu, J. Cho, Automatic identification of user goals in Web search, in: Proceedings of the 14th International Conference on World Wide Web, ACM, 2005, pp. 391–400.

[6] B.J. Jansen, D.L. Booth, A. Spink, Determining the informational, navigational, and transactional intent of Web queries, Inf. Process. Manage. 44 (3) (2008) 1251–1266.

[7] J. Gao, J.-Y. Nie, G. Wu, G. Cao, Dependence language model for information retrieval, in: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'04, ACM, New York, NY, USA, 2004, pp. 170–177. http://dx.doi.org/10.1145/1008992.1009024.

[8] D. Metzler, W.B. Croft, A Markov random field model for term dependencies, in: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'05, ACM, New York, NY, USA, 2005, pp. 472–479. http://dx.doi.org/10.1145/1076034.1076115.

[9] M. Bendersky, W.B. Croft, D.A. Smith, Two-stage query segmentation for information retrieval, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'09, ACM, New York, NY, USA, 2009, pp. 810–811. http://dx.doi.org/10.1145/1571941.1572140.

[10] T. Tao, C. Zhai, An exploration of proximity measures in information retrieval, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 295–302.

[11] R. Cummins, C. O'Riordan, Learning in a pairwise term–term proximity framework for information retrieval, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2009, pp. 251–258.

[12] R. Song, M.J. Taylor, J.-R. Wen, H.-W. Hon, Y. Yu, Viewing term proximity from a different perspective, in: Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval, ECIR'08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 346–357. URL http://dl.acm.org/citation.cfm?id=1793274.1793317.

[13] J. Bai, Y. Chang, H. Cui, Z. Zheng, G. Sun, X. Li, Investigation of partial query proximity in Web search, in: Proceedings of the 17th International Conference on World Wide Web, WWW'08, ACM, New York, NY, USA, 2008, pp. 1183–1184. http://dx.doi.org/10.1145/1367497.1367717.

[14] B. He, J.X. Huang, X. Zhou, Modeling term proximity for probabilistic information retrieval models, Inform. Sci. 181 (14) (2011) http://dx.doi.org/10.1016/j.ins.2011.03.007.

[15] J.L. Morgan, K. Demuth, Signal to Sintax: Bootstrapping from Speech to Grammar in Early Acquisition: [chapters Presented at a Conference Held Feb. 19-21, 1993, Brown University, Providence, RI], Psychology Press, 1996.

[16] R. Jackendoff, X-Bar Syntax, The MIT Press, Cambridge, MA, 1977.

[17] N. Chomsky, Barriers, MIT Press, 1986.

[18] N. Fukui, M. Speas, Specifiers and projection, in: MIT Working papers in Linguistics, Vol. 8, No. 128, 1986.

[19] R. Saha Roy, N. Ganguly, M. Choudhury, S. Laxman, An IR-based Evaluation Framework for Web Search Query Segmentation, in: SIGIR'12, ACM, 2012, pp. 881–890.

[20] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1986.

[21] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Company, Inc., Boston, MA, USA, 1999.

[22] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423.

[23] G. Salton, The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

[24] P. Koehn, Europarl: A parallel corpus for statistical machine translation, in: MT Summit, vol. 5, 2005, pp. 79–86.

[25] J. Huang, J. Gao, J. Miao, X. Li, K. Wang, F. Behr, C.L. Giles, Exploring Web scale language models for search query processing, in: Proceedings of the 19th International Conference on World Wide Web, WWW'10, ACM, New York, NY, USA, 2010, pp. 451–460. http://dx.doi.org/10.1145/1772690.1772737.

[26] R. Saha Roy, M. Choudhury, K. Bali, Are Web search queries an evolving protolanguage? in: Proceedings of the 9th International Conference on the Evolution of Language, Evolang 9, World Scientific Publishing Co., Singapore, 2012, pp. 304–311.

[27] S. Bergsma, Q.I. Wang, Learning noun phrase query segmentation, in: EMNLP-CoNLL'07, 2007, pp. 819–826.

[28] M. Manshadi, X. Li, Semantic tagging of Web search queries, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, ACL'09, Association for Computational Linguistics, Stroudsburg, PA, USA, 2009, pp. 861–869. URL http://dl.acm.org/citation.cfm?id=1690219.1690267.

[29] M. Hagen, M. Potthast, B. Stein, C. Bräutigam, Query segmentation revisited, in: Proceedings of the 20th International Conference on World Wide Web, WWW'11, ACM, New York, NY, USA, 2011, pp. 97–106. http://dx.doi.org/10.1145/1963405.1963423.

[30] R. Saha Roy, N. Ganguly, M. Choudhury, N.K. Singh, Complex network analysis reveals kernel-periphery structure in Web search queries, in: QRU'11, ACM, New York, NY, USA, 2011, pp. 5–8.

[31] E. Guichard, L'Internet: mesures des appropriations d'une technique intellectuelle, These, Ecole des hautes études en sciences sociales, October 2002. URL http://tel.archives-ouvertes.fr/tel-00294711/en/.

[32] A. Spink, D. Wolfram, M.B.J. Jansen, T. Saracevic, Searching the Web: the public and their queries, J. Am. Soc. Inf. Sci. Technol. 52 (2001) 226–234. URL http://dl.acm.org/citation.cfm?id=362968.362979.

[33] C. Barr, R. Jones, M. Regelson, The linguistic structure of english Web-search queries, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP'08, Association for Computational Linguistics, Stroudsburg, PA, USA, 2008, pp. 1021–1030. URL http://dl.acm.org/citation.cfm?id=1613715.1613848.

[34] G. Pass, A. Chowdhury, C. Torgeson, A picture of search, in: Proceedings of the 1st International Conference on Scalable Information Systems, InfoScale'06, ACM, New York, NY, USA, 2006, pp. 1–7. http://dx.doi.org/10.1145/1146847.1146848.

[35] Y. Li, B.-J.P. Hsu, C. Zhai, K. Wang, Unsupervised query segmentation using clickthrough for information retrieval, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11, ACM, New York, NY, USA, 2011, pp. 285–294. http://dx.doi.org/10.1145/2009916.2009957.

[36] M. Paşca, B. Van Durme, What you seek is what you get: extraction of class attributes from query logs, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 2832–2837. URL http://dl.acm.org/citation.cfm?id=1625275.1625731.

[37] R. Jones, K.L. Klinkner, Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM'08, ACM, New York, NY, USA, 2008, pp. 699–708. http://dx.doi.org/10.1145/1458082.1458176.

[38] E. Agichtein, R.W. White, S.T. Dumais, P.N. Bennet, Search, interrupted: Understanding and predicting search task continuation, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'12, ACM, New York, NY, USA, 2012, pp. 315–324. http://dx.doi.org/10.1145/2348283.2348328.

[39] S. Debnath, N. Ganguly, P. Mitra, Feature weighting in content based recommendation system using social network analysis, in: Proceedings of the 17th International Conference on World Wide Web, WWW'08, ACM, New York, NY, USA, 2008, pp. 1041–1042.

[40] J. Guo, G. Xu, X. Cheng, H. Li, Named entity recognition in query, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'09, ACM, New York, NY, USA, 2009, pp. 267–274. http://dx.doi.org/10.1145/1571941.1571989.

[41] J. Cohen, A coefficient of agreement for nominal scales, Educ. Psychol. Meas. 20 (1) (1960) 37–46. http://dx.doi.org/10.1177/001316446002000104.

[42] M. Porter, An algorithm for suffix stripping, Program 14 (1980) 130.

[43] H. Yu, F. Ren, Role-explicit query identification and intent role annotation, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM'12, ACM, New York, NY, USA, 2012, pp. 1163–1172. http://dx.doi.org/10.1145/2396761.2398416.

[44] X. Li, Understanding the semantic structure of noun phrase queries, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL'10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 1337–1345. URL http://dl.acm.org/citation.cfm?id=1858681.1858817.

[45] X. Yin, S. Shah, Building taxonomy of Web search intents for name entity queries, in: Proceedings of the 19th International Conference on World Wide Web, WWW'10, ACM, New York, NY, USA, 2010, pp. 1001–1010.

[46] X. Yin, W. Tan, X. Li, Y. Tu, Automatic extraction of clickable structured Web contents for name entity queries, in: Proceedings of the 19th International World Wide Web Conference, WWW'10, ACM, New York, NY, USA, 2010, pp. 991–1000. http://dx.doi.org/10.1145/1772690.1772791.

[47] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (2002) 422–446.

[48] M. Paşca, B. Van Durme, Weakly-supervised acquisition of open-domain classes and class attributes from Web documents and query logs, in: Proceedings of ACL-08: HLT, Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 19–27. URL http://www.aclweb.org/anthology/P/P08/P08-1003.

[49] M. Paşca, Outclassing Wikipedia in open-domain information extraction: weakly-supervised acquisition of attributes over conceptual hierarchies, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2009, pp. 639–647.

[50] J. Reisinger, M. Paşca, Low-cost supervision for multiple-source attribute extraction, in: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 382–393.

[51] E. Alfonseca, M. Paşca, E. Robledo-Arnuncio, Acquisition of instance attributes via labeled and related instances, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'10, 2010, pp. 58–65.

[52] M. Paşca, Attribute extraction from synthetic Web search queries, in: IJCNLP'11, 2011, pp. 401–409.

[53] T. Lin, P. Pantel, M. Gamon, A. Kannan, A. Fuxman, Active objects: actions for entity-centric search, in: Proceedings of the 21st International Conference on World Wide Web, WWW'12, ACM, New York, NY, USA, 2012, pp. 589–598. http://dx.doi.org/10.1145/2187836.2187916.

[54] A. Jain, M. Pennacchiotti, Open entity extraction from Web search query logs, in: Proceedings of the 23rd International Conference on Computational Linguistics, COLING'10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 510–518. URL http://dl.acm.org/citation.cfm?id=1873781.1873839.

[55] R. Baeza-Yates, A. Tiberi, Extracting semantic relations from query logs, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 76–85.

[56] B. Van Durme, M. Paşca, Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction, in: Twenty-Third AAAI Conference on Artificial Intelligence, AAAI'08, 2008, pp. 1243–1248.

[57] V.B. Nguyen, M.Y. Kan, Functional faceted Web query analysis, in: Proceedings of the Workshop on Query Log Analysis: Social and Technological Challenges, WWW'07, 2007, pp. 1–8.

[58] C. González-Caro, R. Baeza-Yates, A multi-faceted approach to query intent classification, in: R. Grossi, F. Sebastiani, F. Silvestri (Eds.), SPIRE'11, in: Lecture Notes in Computer Science, vol. 7024, Springer, Berlin, Heidelberg, 2011, pp. 368–379.