A Domain-Independent Approach to Finding Related Entities

Olga Vechtomova*

Department of Management Sciences, University of Waterloo

200 University Avenue West, Waterloo, Ontario, N2L 3GE, Canada

Tel: +1 519 888 4567 ext. 32675; Fax: +1 519 746 7252

Email: ovechtom@uwaterloo.ca

Stephen E. Robertson

Microsoft Research Cambridge

7 J J Thomson Avenue

Cambridge, CB3 0FB, UK

Email: stephenerobertson@hotmail.co.uk

Abstract

We propose an approach to the retrieval of entities that have a specific relationship with the entity given in a query. Our research goal is to investigate whether related entity finding problem can be addressed by combining a measure of relatedness of candidate answer entities to the query, and likelihood that the candidate answer entity belongs to the target entity category specified in the query. An initial list of candidate entities, extracted from top ranked documents retrieved for the query, is refined using a number of statistical and linguistic methods. The proposed method extracts the category of the target entity from the query, identifies instances of this category as seed entities, and computes similarity between candidate and seed entities. The evaluation was conducted on the Related Entity Finding task of the Entity Track of TREC 2010, as well as the QA list questions from TREC 2005 and 2006. Evaluation results demonstrate that the proposed methods are effective in finding related entities.

Keywords

Information retrieval, entity retrieval, related entity finding, distributional similarity of words

1. Introduction

Most information retrieval (IR) systems, including commercial search engines, respond to the user's query by retrieving documents. If a user is looking for entities that have a specific relationship to an entity already known to him, he has to manually find them in the documents retrieved by an IR system. Arguably, users with such information needs may prefer to use an IR system that retrieves entities, rather than documents, as this would eliminate the time-consuming process of reading through large amounts of text in order to find the relevant entities. For instance, if a user wishes to know which awards a particular film received,

* corresponding author

then it is likely that he would prefer to get a list of such awards from an IR system directly, rather than look for them in the retrieved documents.

Entity finding is a growing field in Information Retrieval and Computational Linguistics research communities as evidenced by the emergence of several evaluation frameworks within the last few years. For instance, the Question Answering (QA) track of the Text Retrieval Conference (TREC) 2005, 2006 and 2007 (Dang et al., 2007) included in the main task the so-called "list" questions, where the correct response for a query was a list of entities, rather than a single answer. An example of a list question in the 2007 QA track is: "What singers made recordings with Ella Fitzgerald?". The Enterprise track of TRECs 2005-2008 (Balog et al., 2008) included an expert search task, where the objective was to retrieve a list of experts in an area specified in the topic. An example of a query from the 2006 task is "Find individuals with expertise regarding ontology engineering." In 2007-2009 INEX (INitiative for the Evaluation of XML retrieval) organised Entity Ranking track (Demartini et al., 2009), where the task was to find entities in Wikipedia in response to a query. An example of a 2009 query is: "I want a list of movies, fully or partly shot in Venice." The latest entity finding initiative is the Related Entity Finding (REF) task of the Entity track of TREC 2009 (Balog et al., 2009) and TREC 2010 (Balog et al., 2010). The task consists of finding entities that have a specific relationship to the entity given in the query. An example topic is given in Figure 1.

```
<num>23</num>
<entity_name>The Kingston Trio</entity_name>
<entity_URL>clueweb09-en0009-81-29533</entity_URL>
<target_entity>organization</target_entity>
<narrative>What recording companies now sell the Kingston Trio's songs? </narrative>
```

Figure 1. Example of the Entity track topic (TREC 2010).

The components of the Entity track topic include the name of the entity, which is the focus of the topic (henceforth referred to as "topic entity"), the document ID of its homepage, the type of the sought (target) entities, which in the 2010 track could be an "organization", "person", "location" or "product", and a one-sentence free-form narrative, specifying the relationship that must exist between the given topic entity and the target entities. The retrieval result is the ranked list of homepages of the found entities. The collection used in the 2010 track is the English portion of ClueWeb09 Category A, containing approximately 504 million web pages. The dataset consists of 50 topics and their relevance judgements.

In this paper we propose a method for retrieving and ranking entities related to the entity given in the query by a specific relationship, and use the Related Entity Finding task of the Entity track 2010 as the main dataset for evaluation. We also evaluate our method using the Entity track evaluation methodology on the "list" questions of the QA track of TREC 2005 and 2006. The parameters for the methods were tuned on the 20 topics from the Entity track of TREC 2009.

The main goal of our research is to determine whether correct answer entities can be identified in an unsupervised manner by combining the following two characteristics:

- Relatedness of the candidate answer entity to the query topic.
- Likelihood that the candidate answer entity belongs to the sought (target) entity category as specified in the query.

The high-level research questions that we aim to answer are:

RQ1: How to measure the relatedness of the candidate answer entity to the query topic?
RQ2: How to estimate the likelihood that the candidate answer entity belongs to the target entity category specified in the query?
RQ3: Is the likelihood that the candidate answer entity belongs to the target entity category useful in identifying correct answer entities?

Our approach to identifying candidate entities related to the query is first to retrieve documents in response to the initial query, extract an initial set of candidate entities from the text of the documents, and then rank them by relatedness to the query topic. As ranking methods we compare Pointwise Mutual Information, Pearson's χ^2, and TF*IDF measures.

In order to estimate the likelihood that the candidate answer entity belongs to the target entity category specified in the query we propose a method whereby we automatically construct a set of seed entities, which represent hyponyms of the target entity category specified in the narrative, and then rank candidate entities based on their similarity to the seeds. For example, the target entity category in the topic given in Figure 1 is "recording companies". The proposed method automatically extracts the category name from the free-text narrative, finds seed entities belonging to this category (e.g., "decca records"), and computes the similarity of candidate entities to these seeds. Below is the summary of the main steps:

1) Retrieve an initial set of documents for the query from the Web;
2) Retain sentences containing query terms plus one preceding/following sentence;
3) Perform named-entity (NE) tagging of these sentences and extract candidate entities;
4) Extract category name from the narrative;
5) Generate a set of seed entities representing hyponyms of the target entity category;
6) Compute distributional similarity between each candidate entity from Step 3 and a seed entity from Step 5;
7) Rank candidate entities by distributional similarity to all seed entities

The proposed approach is unsupervised and domain-independent, extracting entities from the texts of documents retrieved for the user's query. Our goal is to minimise the reliance on knowledge bases in the process. We only use Wikipedia in the query processing stage to determine boundaries of noun phrases in the narrative. This task could alternatively be handled without Wikipedia, by using a Noun Phrase chunker.

The main contributions of the paper are summarised below:

- A novel method for calculating distributional similarity of words using BM25 document ranking function (Robertson et al., 1995) is proposed. This method is compared to Lin's distributional similarity method (Lin, 1998), and shows improvements on the Entity track REF dataset of TREC 2010.
- The proposed multi-step approach to estimating likelihood that the candidate answer entity belongs to the target entity category specified in the query. The evaluation results demonstrate that this likelihood is a useful factor in ranking candidate answers.

The paper is organised as follows: Section 2 provides an overview of related work, Section 3 gives a detailed description of the proposed methods, Section 4 presents evaluation of the methods, in Section 5 we analyse performance on sample topics, and conclude the paper in Section 6.

2. Related Work

Previous approaches to finding relationships between entities can be divided into two groups based on the type of the problem they address: (1) pre-defined relationship finding, where the target relationship is given and the goal is to find pairs of entities, for which this relationship is true (e.g., Agichtein and Gravano 2000; Riloff and Jones, 1998, Brin, 1999); (2) open relationship finding, where the task is to discover interesting/pertinent relationships between entities in a corpus, or between a given entity and any other entities (e.g., Hasegawa et al., 2004; Davidov et al., 2007, Banko and Etzioni, 2008).

Approaches to the pre-defined relationship finding are usually supervised or semi-supervised. The former take as input a training set of manually labeled entity pairs or an annotated corpus and attempt to classify new pairs of entities into those that have or do not have a given relationship (Miller et al., 2000; Zelenko et al., 2002; Culotta and Sorensen, 2004; Kambhatla, 2004; Zhou et al., 2005). Semi-supervised approaches start with a small set of seeds and find new relationship instances by a bootstrapping process (Riloff and

Jones, 1998; Agichtein and Gravano 2000; Ravichandran and Hovy, 2002, Zhang, 2004; Chen et al., 2006, Rosenfeld and Feldman, 2007). For example, the Snowball system (Agichtein and Gravano 2000) takes a set of seed entity pairs, and iteratively finds new extraction patterns and entity pairs. A system by Ravichandran and Hovy (2002) was developed for factoid question answering. It requires a small seed set of entity pairs (question entity and correct answer) for each question type (e.g., birth year, discoverer), and learns a set of patterns that are most likely to contain the correct answer. Rosenfeld and Feldman (2006) proposed a system for relationship finding given only a general relationship template and a small number of keywords representing the relationship, such as (*Acquisition(ProperNP, ProperNP)ordered keywords={"acquired" "acquisition"}*). The system finds seeds from the Web by using a small set of patterns generated from the relationship keywords, and then uses the found seed pairs to discover other entity pairs that have the given relationship.

The ad-hoc related entity finding task, such as the one formulated in the Entity track of TREC, is particularly challenging, since the input to the system only consists of the query entity, the target entity type and a free-form one-sentence narrative. Supervised approaches are not suitable for this task as they require a large number of manually labeled examples. Although semi-supervised methods are generally more attractive since they require only a small set of seed pairs, they are poorly suited for this task, as the seed pairs are not given.

In our work we address the problem of the missing seeds by generating them automatically from the narrative. Specifically the goal of the seed generation stage is to find a small high-precision set of entities that have a hyponymy relationship with the target entity category stated in the narrative. For example, if the category of target entities is "operating systems", our goal is to identify instances of operating systems as seeds. Our next stage is to compute the similarity between candidate entities and seeds. We then rank candidate entities, such that those that are most similar to all seeds are ranked higher. Pairwise similarity between a candidate and a seed entity is calculated based on the distributional similarity principle. Each entity is represented as a vector of weighted grammatical dependency triples, co-occurring with it in a corpus. We developed a new method for calculating distributional similarity between candidate and seed entities using BM25 with query weights (Robertson et al., 1995). This is a new application for BM25 matching function, which has not been used for computing word similarity before. As a second method of computing candidate-seed similarity we adapted Lin's (Lin, 1998) distributional word similarity method.

There is a large number of methods developed for calculating distributional similarity between words. A comprehensive review of different approaches is given by Weeds and Weir (2006). According to the distributional similarity principle, words that occur in similar contexts, are likely to mean similar things. Distributional similarity methods differ by what linguistic units they use as context. For example, Pantel et al. (2009) use noun phrase chunks to the left and right of a term as its context, while Lin (1998), Weeds and Weir (2006), Kotlerman et al. (2009) use grammatical dependency relations as features. As discussed in (Kilgariff and Yallop, 2000), the use of grammatical relations as features leads to the identification of "tighter" relationships between words, whereas the use of document- and sentence-level word co-occurrences would lead to the identification of "looser" relationships. So, while the former are better for identifying words belonging to the same semantic class, the latter are more appropriate for grouping words into subject categories. Since our goal is to find entities having a very specific relationship to the query entity, we use grammatical dependency relations as features.

2.1 Approaches to related entity finding in TREC

In this section we review some of the approaches to related entity finding in the Entity track of TREC 2009. Most of the methods developed by participants of the Entity track of TREC 2009 start with the retrieval of some units of information (documents, passages, sentences) in response to the queries generated from the topic. The retrieved units are then used for extracting candidate entities. Below we discuss various

approaches based on: (a) how the queries are constructed, (b) what units are retrieved (e.g., documents, passages, sentences), (c) how candidate entities are extracted and ranked.

2.1.1 Query construction

As an alternative to using "entity name" and "narrative" sections of topics as queries directly, some query structuring and expansion methods are explored. Vydiswaran et al. (2009) model the information need as a triple (topic entity; relationship type; target entity), of which the first two are known. The word(s) denoting the relationship are extracted from the narrative and expanded with WordNet synonyms. Fang et al. (2009) expand the entities from narratives with their acronyms identified using a dictionary-based approach.

2.1.2 Retrieval units

Most approaches start with the retrieval of documents by using experimental IR systems and/or web search engines. For example, Vydiswaran et al. (2009) use documents retrieved by the Indri IR system, from which they select snippets containing the query terms. McCreadie et al. (2009) use the Divergence from Randomness (DFR) model, a term proximity-based model, and the number of incoming links to the documents. They also experiment with community-based document ranking. Zhai et al. (2009) use BM25, Fang et al. (2009) use Google results filtered by the ClueWeb09 Category B documents, and Wu and Kashioka (2009) compare the use of Indri and Google.

2.1.3 Candidate entity extraction and ranking

Vydiswaran et al. (2009) extract entities from the document snippets containing query terms, apply a NER tagger, apply a number of heuristics to prune the list of candidate entities, and rank entities by a combination of the frequency of candidate entities in the retrieved snippets, and their co-occurrence with the entity given in the topic.

McCreadie et al. (2009) use DBPedia and US Census data to build detailed representations of entities found in the ClueWeb09 Cat. B collection. The representation of each entity include alternative names (DBPedia aliases), DBPedia categories and documents in ClueWeb09 Cat. B containing the entity. They propose a voting model to rank entities.

Zhai et al. (2009) use titles and anchor texts in the retrieved documents as candidate entities. For each candidate entity they build a pseudo-document, consisting of top 100 sentences containing this entity. They experiment with ranking entities based on the similarity of their pseudo-documents and the pseudo-documents of the entity given in the topic.

Wu and Kashioka (2009) use Wikipedia's hyperlink structure, to reduce the list of candidate entities. Entity scores are calculated based on the presence of hyperlinks between the candidate entity's Wikipedia page and the Wikipedia page of the entity given in the topic. They then retrieve snippets containing each candidate entity, and calculate a similarity score between the set of snippets and the query entity, experimenting with a language modelling approach and Support Vector Machines.

Kaptein et al. (2009) calculate similarity between the entity given in the topic and each candidate entity based on the co-citation information from the hyperlink graph constructed for the ClueWeb09 Cat. B collection. They also propose a method that extracts candidate entities from Wikipedia.

Fang et al. (2009) combine a number of approaches for ranking candidate entities, such as extracting entities from tables and lists in the retrieved web documents and using proximity in retrieved documents between a candidate entity and the entity given in the topic. They experiment with hand-crafted templates, such as "<candidate entity> is <narrative>". One other method consists of extracting the first term from the narrative, which usually represents the category of the sought entities, and checking for each candidate entity if it occurs in the body or categories of its Wikipedia page.

3. Methodology

In the following subsections we describe our approach to entity finding in detail. Figure 2 provides an overview of the main components of our method.

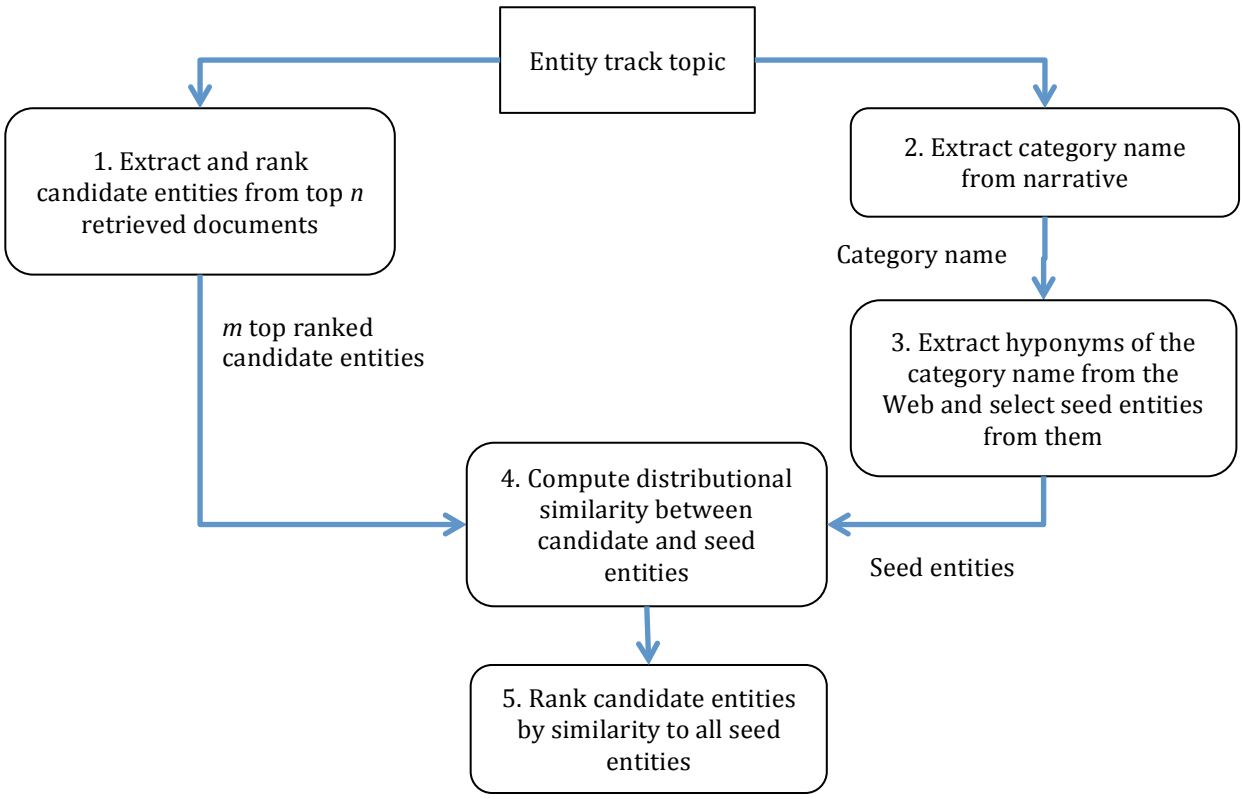

Figure 2. Components of the proposed method.

In the first stage (rectangle 1 in Figure 2), the system retrieves an initial set of documents for the query from the Web. Only the sentences containing query terms plus one preceding/following sentence are retained. Named Entity tagging is applied to these sentences, and candidate entities are extracted and ranked. In the second stage, the target category name is automatically identified from the topic narrative; and in stage 3 the system finds hyponyms of this category name, and selects seed entities from the hyponyms. In stage 4, the entities (candidates and seeds) are represented as vectors of weighted grammatical dependency triples, and pairwise (candidate-seed) similarity is calculated. In stage 5, candidate entities are ranked by similarity to all seeds. Stage 1 is described in Sections 3.1 and 3.2, while stages 2-5 are presented in Section 3.3.

3.1 Extracting candidate entities

There are multiple sources, from which entities can be extracted, such as knowledge bases, e.g., Wikipedia, or the texts of top documents retrieved in response to the query. In this paper we aim to investigate the effectiveness of entity extraction from the texts of top retrieved documents. In our preliminary experiments we compared the extraction of entities from top documents retrieved from the ClueWeb09 Category B[1] collection using BM25tp (a variant of BM25 incorporating term proximity information) (Büttcher et al., 2006) with the extraction from top documents retrieved from the Web by a major search engine. The effectiveness of the latter approach is higher, and therefore it is used in the methods described in this paper. Fang et al. (2009) also found that the use of documents retrieved by a search engine from the Web leads to

[1] Clueweb Category B, a subset of Category A, consists of 50 million English webpages, and was used in the Entity track of TREC 2009.

better performance in entity finding than the use of an IR system (Indri) on the ClueWeb09 Category B corpus.

As the first step, the queries to retrieve top documents from the Web are generated from the "entity name" and "narrative" sections of the Entity track topics according to the algorithm specified in Figure 3. The objective of the algorithm is to extract named entities and other noun phrases from the topic. For this purpose we use a Part-Of-Speech tagger (Brill, 1995) and a Noun Phrase chunker (Ramshaw and Marcus, 1995), and a list of titles of Wikipedia pages. The resulting queries are then used to retrieve the top 50 documents from a Web search engine. We did not evaluate alternative values for the number of top documents retrieved. Our motivation to use 50 is to keep the number of documents for subsequent in-depth analysis reasonably small, and at the same time have sufficient amount of text to extract entities from.

The retrieved 50 documents are parsed to remove HTML tags, script and style sections, and broken into sentences. We then extract sentences that contain at least one query term. If a query term is a noun, the system attempts to match its singular and plural forms. For each such sentence, we also extract one sentence before and one after. Let this set of sentences be *{S}*. The sentences are then processed by the LBJ-based Named Entity Recognizer (Ratinov and Roth, 2009). The NER tagger only assigns the categories of "Location", "Organization", "Person" and "Miscellaneous". In the Entity track 2010, the target entity type specified in the topic is either "organization", "person", "location" or "product". For topics belonging to the first three category types, we extract all entities tagged with the corresponding category labels. However, for topics of category "Product" we extract entities labelled as "Organization" and "Miscellaneous".

```
For each topic
  Process narrative using a POS tagger and Noun Phrase chunker.
  Get all possible contiguous ngrams from the "entity name" and
  "narrative" sections of the topic.
  Sort ngrams in the descending order of the number of words
  For each ngram n
    If n OR possessive form of n OR singular form of n matches a
    Wikipedia title
        m = form of n matching a Wikipedia title
    Elsif n is a unigram and does not match a Wikipedia title
        m = n
    If m is not a stopword
        Remove "the" or "a" if found at the beginning of m
        If m is extracted from narrative
            If m is part of any NP in the chunked narrative
                If m has not been written to the query before
                    Add m to the query as a phrase (string in
                    quotation marks)
```

Figure 3. Algorithm for processing queries

3.2 Ranking candidate entities by TF*IDF and co-occurrence association measures

Having extracted candidate entities from the top 50 documents retrieved for each topic, we evaluate the following entity ranking methods:

- *TF*IDF*
- Pearson's χ^2 (chi-square)
- Pointwise Mutual Information (PMI)

Co-occurrence association measures (χ^2 and PMI) are calculated between each candidate entity and the topic entity, i.e. the entity given in the "entity name" section of the topic (Figure 1).

*3.2.1 TF*IDF*

The *TF*IDF* score for each candidate entity is calculated using the following equation:

$$TF \times IDF = TF \times \log \frac{N}{n} \quad (1)$$

Where: *TF* – frequency of the entity in the sentence set *{S}* extracted from the top 50 retrieved documents; *N* – number of documents in the ClueWeb09 Category B collection; *n* – number of documents in the collection containing the entity.

3.2.2 Pearson's χ^2

Pearson's Chi-square statistic compares the observed frequencies of two words with their expected frequencies given the independence assumption (Manning and Schütze, 1999). If the difference between the observed and expected frequencies is large, then the null hypothesis of independence can be rejected. The Chi-square statistic described in (Manning and Schütze, 1999) applies to adjacent co-occurrences. In our work, we consider that the candidate entity (*c*) co-occurs with the topic entity (*t*) if *c* appears within the window of 100 words either side of *t*. The reason for choosing the span of 100 is to capture broader contextual associations between words, rather than specific lexico-syntactic relationships. Tables 1 and 2 show how expected and observed frequencies are calculated taking in account the window size greater than one around the topic entity *t*. Table 3 shows how χ^2 is calculated based on Tables 1 and 2. All frequencies are gathered from the ClueWeb09 Category B corpus. *T* is the number of tokens (term instances) in the corpus. The joint frequency of the topic and candidate entities, *f(t,c)*, is calculated by counting the number of times *t* and *c* occur within the span of 100 words in the corpus. To obtain joint frequencies, we use the Wumpus search engine[2] to find the number of shortest substrings containing *t* and *c* within the span of 100. Each shortest substring may only contain one instance of *t* and *c*. The ideal window around *t* is 200, i.e. 100 words either side, however, in reality the window may hit the document boundary or a window around another instance of *t* in a document. Because of this the observed windows can be smaller, therefore the average window size (*v*) around *t* in the corpus must be calculated. In our experiments, it is computationally too expensive to obtain observed window sizes around every instance of *t* in the ClueWeb09 Category B corpus, therefore we approximate *v* as 100.

Table 1. Observed frequencies

	Candidate entity (*c*)	Not candidate entity (¬ *c*)	Total
Topic entity (*t*)	$O_a = f(t,c)$	$O_b = f(t) - f(t,c)$	$f(t)$
Not topic entity (¬ *t*)	$O_c = f(c) - f(t,c)$	$O_d = T - f(t) + f(t,c) - f(c)$	$T - f(t)$
Total	$f(c)$	$T - f(c)$	T

Table 2. Expected frequencies

	Candidate entity (*c*)	Not candidate entity (¬ *c*)	Total
Topic entity (*t*)	$E_a = f(t)f(c)v/T$	$E_b = f(t) - E_a$	$f(t)$
Not topic entity (¬ *t*)	$E_c = f(c) - E_a$	$E_d = T - f(c) - f(t) + E_a$	$T - f(t)$
Total	$f(c)$	$T - f(c)$	T

[2] http://www.wumpus-search.org/

Table 3. χ^2 table

	Candidate entity (c)	Not candidate entity ($\neg c$)
Topic entity (t)	$\frac{(f(t,c)-f(t)f(c)v/T)^2}{f(t)f(c)v/T}$	$\frac{(f(t,c)-f(t)f(c)v/T)^2}{f(t)-f(t)f(c)v/T}$
Not topic entity ($\neg t$)	$\frac{(f(t,c)-f(t)f(c)v/T)^2}{f(c)-f(t)f(c)v/T}$	$\frac{(f(t,c)-f(t)f(c)v/T)^2}{T-f(c)-f(t)+f(t)f(c)v/T}$

The χ^2 is calculated using the following equation:

$$\chi^2 = \sum_{a,b,c,d} \frac{(O_x - E_x)^2}{E_x} = (O-E)^2 \sum_{a,b,c,d} \frac{1}{E_x} \qquad (2)$$

3.2.3 Pointwise Mutual Information

PMI compares the probability of the joint co-occurrence of two terms with the probability that they occur independently. The original PMI as used, for example, in (Church et al., 1991) is applied to adjacent and ordered co-occurrence, i.e. c immediately following t. Since we are calculating co-occurrence of c within 100 words either side of t, we use the modified version of PMI (Vechtomova et al., 2003). Frequencies were obtained in the same way as for χ^2, and the same approximation for v is applied in calculating PMI.

$$I(t,c) = \log_2 \frac{P(t,c)}{P(t)P(c)} = \frac{f(t,c)/vT}{f(t)f(c)/T^2} \qquad (3)$$

3.3 Ranking candidate entities by the similarity to the target entity category

Since the chosen NER tagger can only be used to identify entities of a few broad categories, such as organisations and people, the list of candidate entities can be noisy. This is further compounded by the NER tagger errors. To refine the list of entities, we apply the distributional similarity principle, which is based on the observation that semantically close words occur in similar contexts. If we have a small number of correct seed entities, we can rank the candidate entities by the distributional similarity to them. As discussed in Section 2, many methods reported in the literature that utilise the distributional similarity principle are semi-supervised methods, such as (Thelen and Riloff, 2002). They start with the list of known seed words and then find other words that are distributionally similar, and therefore, likely to be semantically close to the seed words. The problem in our task is that the seed words are not given. However, the topic narratives have descriptions of the categories of entities that are to be retrieved. Our approach is to find seed entities based on the described categories. For example, the narrative of TREC 2010 REF topic #62 is "What cruise lines have cruises originating in Baltimore?" We developed a method to extract the category name from the narrative, i.e. "cruise lines" in this topic, and adapted a method for the automatic acquisition of the hyponymy relation proposed by Hearst (1992) to find entities that belong to this category. Seed entities are then selected from the hyponyms. We also developed a new method for computing the distributional similarity between seeds and candidate entities using BM25 with query weights, and ranking the entities by similarity to all seed entities.

The methods represented in Figure 2 as rectangles 2-5 are described in this section. Among the ranking methods proposed in Section 3.2 the best results on the training topics are obtained using TF*IDF. Therefore, it was decided to use entities ranked by TF*IDF as the input to subsequent stages. Since this list of entities can be large and may contain a lot of noise, we select the top m ranked entities. In subsequent sections we will refer to this ranked list of entities as "candidate entities". In our experiments we set m to

200. In order to determine the value for *m* we took the entity names ranked by the TFIDF for the training topics, and did a simple pattern matching with the correct answer set of entity names. This showed that of all the correct entities present in the TFIDF-ranked list, 81% were ranked in the top 200. Therefore, 200 seemed to be sufficient. Larger values are likely to introduce more noise terms into the subsequent stages, and only a small number of relevant entities.

3.3.1 Extracting category names from topic narratives

To extract category names, the narratives are first processed using Brill's Part-of-Speech (POS) tagger (Brill, 1995) and a Noun-Phrase chunker (Ramshaw and Marcus, 1995). Then a set of rules is applied to select one of the initial noun phrases (NPs) from the narrative. Generally, the first noun phrase in the narrative is selected as the category name, unless it is a personal pronoun or the noun "searcher". Table 4 lists examples of NP-chunked narratives of the TREC 2010 topics and the extracted category names.

Table 4. Examples of NP-chunked narratives and extracted category names

Topic	NP-chunked narrative	Extracted category name
21	[What/WP] [art/NN galleries/NNS] are/VBP located/VBN in/IN [Bethesda/NNP] ,/, [Maryland/NNP] ?/.	art galleries
22	Find/VB [countries/NNS] [that/WDT] are/VBP [members/NNS] of/IN [OPEC/NNP] (/([the/DT Organization/NNP] of/IN [Petroleum/NNP Exporting/NNP Countries/NNPS])/SYM ./.	countries
23	[What/WP] [recording/NN companies/NNS] now/RB sell/VBP [the/DT Kingston/NNP Trio's/NNP songs/NNS] ?/.	recording companies
30	Find/VB [U.S./NNP states/NNS and/CC Canadian/JJ provinces/NNS] where/WRB [Ocean/NNP Spray/NNP growers/NNS] are/VBP located /VBN ./.	U.S. states [OR] Canadian provinces
38	[Who/WP] are/VBP [the/DT drivers/NNS and/CC crew/NN chiefs/NNS] for/IN [Richard/NNP Petty/NNP Motorsports/NNS] ?/.	drivers [OR] crew chiefs

Other rules included splitting a NP containing conjunction (e.g. "and", "/", "or") into two or more NPs (see topics #30 and #38 in Table 2).

3.3.2 Identifying seed entities

After the category name is extracted from the topic narrative, the next step is to find entities that belong to this category. We adapted the unsupervised hyponymy acquisition method proposed by Hearst (1992). Hearst's method uses six domain- and genre-independent lexico-syntactic templates that indicate a hyponymy relation. The templates and examples of sentences found for the topic "I'd like to find which operating systems I can use on the EEE PC." are shown in Table 5. The extracted category name for this topic is "operating systems".

Table 5. Hearst's patterns for hyponym acquisition

Template	Examples
NP such as {NP,} {(or|and)} NP*	…on some multi-user **operating systems such as** Windows XP, Windows Vista, Mac OS X, OpenSUSE, Ubuntu and Fedora.
such NP as {NP,} {(or|and)} NP*	Experienced in **such operating systems as** MS-DOS, Windows, Windows NT/2K/XP/2K3, Solaris, Linux, FreeBSD.

NP {, NP}{,} or other NP*	If you want to run Windows, Linux, **or other operating systems** on…
NP {, NP}{,} and other NP*	PostScript font installation - Unix, Linux **and other operating systems**.
NP{,} including {NP ,} {or | and} NP*	We will cover each of the major **operating systems, including** DOS, Windows 9x/NT/2000/XP, and UNIX/Linux.
NP {,} especially {NP,} {or | and} NP*	…but the program is used mainly on other **operating systems, especially** Linux.

For each topic, six queries are constructed using the above templates, and the category name is extracted from the topic narrative. For example, the query for the first template in Table 5 is: "operating systems such as". Each query is submitted to a commercial search engine as a phrase (i.e. quote-delimited). If the total number of pages retrieved by all six queries is fewer than 10, the first word in the category name is dropped and the search is repeated. If again it returned fewer than 10 pages, the first two words are dropped, and so on until either 10 or more pages are retrieved, or the remaining category name consists of only a single word, in which case we use whatever number of pages were found. If a category name is a single word, the query includes the topic title in addition to the template, in order to minimise the extraction of unrelated entities.

The documents retrieved for each query are processed to remove HTML tags, and split into sentences. The sentences containing the hyponymy lexico-syntactic patterns are then processed using the LBJ-based NER tagger (Ratinov and Roth, 2009). Depending on the expected position of hyponyms in the lexico-syntactic pattern, NEs either immediately preceding, or following the pattern are extracted. If several NEs are used conjunctively, i.e., separated by a comma, "and" or "or", all of them are extracted. For each topic, we extract only NEs with tags corresponding to the entity type specified in the topic, i.e. NEs tagged with "ORG", "PER", "LOC" and ("MISC"|"ORG") are extracted for topics with entity types "organization", "person" "location" and "product" respectively. Below is an example of the output of the NER tagger for the above topic:

"…on some modern multi-user operating systems such as [MISC Windows XP] , [MISC Windows Vista] , [ORG Mac OS X] , [PER OpenSUSE] , [ORG Ubuntu] and [ORG Fedora] ."

The entity type for this topic is "product", the following entities are extracted as hyponyms of "operating system": "Windows XP", "Windows Vista", "Mac OS X", "Ubuntu" and "Fedora". In this sentence the tagger erroneously tagged "OpenSUSE" as person, therefore it is not extracted.

One problem with using all found hyponyms as seed entities is that they can be unrelated to the topic. Some category names extracted from topic narratives are very broad, for example, "countries" in topic #22 (Table 4). Applying the above hyponym acquisition algorithm based on such high-level hypernym is bound to produce a very large number of topically-unrelated hyponyms. Computing distributional similarity between candidate entities and these hyponyms is likely to be ineffective and possibly detrimental to performance. We therefore must ensure that we use only those hyponyms as seeds, for which there exists some evidence of relationship to the topic. For this purpose, we defined as seeds the intersection of found hyponyms and entities extracted from the top 50 documents retrieved for the initial query as described in Section 3.1. For example, for the above topic, the following hyponyms were identified as seeds: "FreeBSD", "Mac", "Linux", "Ubuntu", "Windows XP", "Microsoft Windows", "Unix", "Microsoft". If only one seed word is identified as a result of this process, we do not perform entity re-ranking on this topic, and keep the original rank order, which is output by the TF*IDF method. We believe that one seed entity is an insufficient representation of the category of the sought entities, and could substantially degrade performance if it is incorrect.

3.3.3 Computing distributional similarity between candidate and seed entities.

Distributional similarity between entities is computed based on the commonality of their contexts of occurrence in text. In their simplest form, contexts could be words extracted from windows around entity occurrences. Alternatively, they could be grammatical dependency relations, with which an entity occurs in text. The use of grammatical dependency relations is more constraining in calculating entity similarity, and allows us to identify tightly related entities, which could be inter-substituted in a sentence without making it illogical and ungrammatical. For example, grammatical dependency relations in Table 6 tend to occur with noun phrases referring to sports people.

Table 6. Examples of grammatical dependency relations.

Grammatical dependency relation	Source sentence
win *V:subj:N* Rubens Barrichello	**Rubens Barrichello won** the Italian Gran Prix.
teammate *N:appositive:N* Rubens Barrichello	His **teammate, Rubens Barrichello**, managed second with a 1:35.681 lap at the end of the session.

In contrast if we only use co-occurring words ("win" and "teammate") in calculating similarity, we would get more loosely related entities, e.g. names of teams, sponsors, etc. As discussed in Section 2, several previous approaches to calculating distributional similarity between words use grammatical dependency relations. Since we are interested in identifying entities that are of the same semantic category as the seed words, we decided to use grammatical dependency relations in calculating entity similarity.

For each seed and candidate entity we retrieve 200 documents from ClueWeb09 Category B using BM25 (Robertson et al., 1995) implemented in the Wumpus search engine. Each document is split into sentences, and sentences containing the entity are parsed using the Minipar[3] syntactic parser (Lin, 1993) to extract grammatical dependency triples. Each dependency triple consists of two words and a grammatical relation that connects the two entities (examples are given in Table 6). These dependency triples are transformed into features representing the context of each candidate and seed entity. To transform a triple into a feature, we replace the entity name in the triple with 'X', e.g., "win V:subj:N Rubens Barrichello" is transformed into "win V:subj:N X". To avoid using features that are specific to only one or few seed entities, only features that occur with at least 50% of all seed entities are used in computing entity similarity. For each seed and candidate entities we build a vector consisting of these features and their frequencies of occurrence with this entity. To compute pairwise similarity between the vectors of seed and candidate entities, we use two approaches: similarity computed using BM25 with query weights (Section 3.3.3.1) and similarity computed using Lin's method based on Mutual Information (Section 3.3.3.2).

3.3.3.1 BM25 based similarity method

In order to compute the similarity between the vectors of seed and candidate entities, we adapted the BM25 with query weights formula. For each seed and candidate entity we calculate a Query Adjusted Combined Weight (QACW) by (Spärck Jones et al., 2000). In the QACW formula, the vector of the seed entity is treated as the query and the vector of the candidate as the document:

$$QACW_{c,s} = \sum_{f=1}^{F} \frac{TF(k_1+1)}{K+TF} \cdot QTF \cdot IDF_f \qquad (4)$$

Where: F – the number of features that a candidate entity c and a seed entity s have in common; TF – frequency of feature f in the vector of candidate entity; QTF – frequency of feature f in the vector of the seed entity; $K = k_1*((1-b)+b*DL/AVDL)$; k_1 – feature frequency normalisation factor; b – document length

[3] http://webdocs.cs.ualberta.ca/~lindek/minipar.htm

normalisation factor; *DL* – number of features in the vector of the candidate entity; *AVDL* – average number of features in all candidate entities.

We evaluated different combinations of values of b and k_1 on the 20 topics from the Entity track of TREC 2009, with the best results in NDCG@R obtained with b=0.8 and k_1=0.8.

In order to calculate the *IDF* of a feature, we need to have access to a large syntactically parsed corpus, such as the ClueWeb09 Category B collection. Since we do not have such a resource, and it is computationally demanding to produce one, we approximate *IDF* of a feature with the *IDF* of its component word. For example, for the feature "win V:subj:N X" we calculate the *IDF* of "win" by using its document frequency in the ClueWeb09 Category B collection.

Arguably, when calculating the similarity of candidate entities to seed entities, we should take into account how strongly each seed entity is associated with the original TREC topic. Candidate entities similar to those seed entities, which have weak association with the topic, should be downweighted compared to those candidate entities, which are similar to seed entities strongly associated with the topic. We propose to quantify this association by using entity weights calculated by one of the entity ranking methods presented in Section 3.2. In our experiments we use entity weights calculated using the *TF*IDF* method (Section 3.2.1), which gave the best results on the training data. Thus, the candidate entity matching score with all seed entities is calculated according to:

$$EntitySeedBM25_c = \sum_{s=1}^{S} w_s QACW_{c,s} \qquad (5)$$

Where: w_s – weight of the seed entity s calculated using the TF*IDF entity ranking method.

Only those candidate entities that have EntitySeedBM25 as greater than zero are retained. The final ranking of entities is achieved through a linear combination of TF*IDF and EntitySeedBM25 according to the following equation:

$$TFIDFEntitySeedBM25 = \alpha \times \log(TFIDF) + (1-\alpha) \times \log(EntitySeedBM25) \qquad (6)$$

Values from 0.1 to 1 at 0.1 intervals were evaluated for α on the 20 topics from the Entity track of TREC 2009, with the best results in NDCG@R obtained with α=0.5.

3.3.3.2 Lin's similarity method

As our second approach for computing pairwise similarity between a seed and a candidate entity, we adapted Lin's distributional similarity method (Lin, 1998). This is a well-known distributional similarity method that uses grammatical dependency relations as features, weighted using Mutual Information. As in the method above, each feature in a vector of a word is a dependency triple (w, r, w'), where w and w' are words and r is a grammatical relation, for example, (Rubens Barrichello N:subj:V win). Each feature is weighted by using Mutual Information as follows:

$$I(w,r,w') = \log \frac{f(w,r,w') \times f(*,r,*)}{f(w,r,*) \times f(*,r,w')} \qquad (7)$$

Where: f(w, r, w') – the frequency of occurrence of the triple (w, r, w'), * – wildcard, for example, f(*, r, w') means the sum of frequencies of all dependency triples matching the pattern where the grammatical relation is r and the second word is w'. For example, if the entity is "Rubens Barrichello" and the feature is "win V:subj:N Rubens Barrichello", the Mutual Information between "Rubens Barrichello" and the feature will be calculated as follows:

$$I(\text{Rubens Barichello,V}:\text{subj}:\text{N, win}) = \log \frac{f(\text{Rubens Barichello,V}:\text{subj}:\text{N, win}) \times f(*,\text{V}:\text{subj}:\text{N},*)}{f(\text{Rubens Barichello,V}:\text{subj}:\text{N},*) \times f(*,\text{V}:\text{subj}:\text{N, win})}$$

In Lin's experiments (Lin, 1998), frequencies are computed from a newswire corpus of 64 million words, however in our case it would have been infeasible to parse an entire ClueWeb09 Category B collection, therefore we only use top 200 documents retrieved for each candidate entity using BM25 as outlined in Section 3.3.3.

Each entity (candidate or seed) is represented as a vector T, consisting of pairs (r, w) with $I > 0$. Similarity between a seed entity s and a candidate entity c is calculated as follows:

$$sim(c,s) = \frac{\sum_{(r,w)\in T(c)\cap T(s)} (I(c,r,w) + I(s,r,w))}{\sum_{(r,w)\in T(c)} I(c,r,w) + \sum_{(r,w)\in T(s)} I(s,r,w)} \quad (8)$$

After calculating the similarity score, the candidate entity's matching score EntitySeedLin is calculated in the same way as EntitySeedBM25 (Eq. 5). Only those candidates which have EntitySeedLin greater than zero are retained. The final ranking (TFIDFEntitySeedLin) is achieved through linear combination of EntitySeedLin with TF*IDF in the same manner as was done for TFIDFEntitySeedBM25 (Eq. 6). Again, values from 0.1 to 1 at 0.1 intervals were evaluated for α on the 20 topics from the Entity track of TREC 2009, with the best results in NDCG@R obtained with α=1. The fact that α=1 gives the best performance in TFIDFEntitySeedLin means that Lin's score itself is not useful for candidate entity ranking, but only for filtering out non-relevant entities, with the ranking being done by TFIDF. This contrasts with the TFIDFEntitySeedBM25 method, where the best α is 0.5, meaning that TFIDF and EntitySeedBM25 components contribute equally to the final entity score.

4. Evaluation

Our methods were evaluated on three datasets:

1) The dataset of the Related Entity Finding task of the Entity track of TREC 2010 (Balog et al., 2010).
2) List questions from the Question Answering (QA) track of TREC 2005;
3) List questions from the QA track of TREC 2006;

All parameters were tuned on the 20 Related Entity Finding topics from the Entity track of TREC 2009.

4.1 Evaluation on the Related Entity Finding task of the Entity track of TREC 2010

The requirement in the Entity track is to retrieve a ranked list of up to 100 entities for each topic. For each retrieved entity, the systems are required to retrieve one homepage, which must be represented as the ClueWeb09 Category A document ID. In fact, a working definition of an "entity" in the Entity track is something that has a homepage.

Relevance judgements of entity homepages were done on a 3-point scale: 2 – primary page (i.e. homepage of the correct entity), 1 – descriptive page related to the correct entity, and 0 – all other pages.

The two official evaluation measures are nDCG@R – normalised discounted cumulative gain at R, where R is the number of primary and relevant homepages for that topic, and P@10 – fraction of primary homepages among the documents retrieved for the top 10 entities. The Mean Average Precision (MAP) and Precision at R were also calculated for TREC 2010 topics[4].

[4] The evaluation script provided in TREC 2009 only calculates NDCG@R and P@10.

We developed a simple homepage finding algorithm, which consists of retrieving the top 10 webpages for each entity from a commercial Web search engine, filtering out a small number of common URLs, such as "dictionary.com", "facebook.com", "linkedin.com", "wikipedia.org" and using as homepage the top ranked page that also exists in the ClueWeb09 Category A collection. The evaluation procedure was the same for both training and test topics. The evaluation results on the 20 training topics are given in Table 7. The "TFIDF" run is described in Section 3.2.1, "Chi" in 3.2.2, PMI in 3.2.3, TFIDFEntitySeedBM25 in 3.3.3.1 and TFIDFEntitySeedLin in 3.3.3.2.

Table 7. Evaluation results on 20 training topics.

Run	nDCG@R	P@10	Rel. retr.	Prim. Retr.
PMI	0.1396	0.0900	100	61
Chi	0.1448	0.0850	101	63
TFIDF	0.1712	0.1450	86	63
TFIDFEntitySeedBM25 (b=0.8; k=0.8; α=0.5)	0.1705	0.1700	85	62
TFIDFEntitySeedLin (α=1)	0.1725	0.1550	85	62

Since the "TFIDFEntitySeedBM25" and "TFIDFEntitySeedLin" runs re-rank the top 200 entities in "TFIDF", the latter is used as the baseline for evaluating the performances of these runs. All statistical tests reported in this paper are done using the 2-tail paired t-test. The results of the runs on the 50 test topics are shown in Table 8. All runs have parameters that showed the best NDCG@R values on the 20 training topics (Table 7).

Table 8. Evaluation results on 50 test topics.

Run	nDCG@R	P@10	MAP	R-prec	Rel. retr.	Prim. Retr.
PMI	0.0765	0.0255	0.0263	0.0308	87	149
Chi	0.0901	0.0426	0.0369	0.0472	90	149
TFIDF	0.1226	0.0936	0.0588	0.1006	89	152
TFIDFEntitySeedBM25 (b=0.8; k=0.8; α=0.5)	0.1400‡	0.1043	0.0722‡	0.1140	91	157
TFIDFEntitySeedLin (α=1)	0.1264	0.0957	0.0632	0.1092	91	155

4.2 Evaluation on the list questions from the QA track of TREC 2005 and 2006

QA list questions are formulated slightly differently from the Entity track Related Entity Finding (REF) topics. For each topic a target entity is specified, which is similar to the entity_name part of Entity track topics. Each topic has one or two list questions, formulated in a similar way as the narrative section of the Entity track topics. A major difference of QA list questions from the Entity track REF topics is that target entity types are not given. Also, some list questions are looking for answers of types other than "Person", "Organization", "Location" and "Product". Examples of such questions are: What are the names of the three Great Pyramids? Name unusual flavors created by Ben & Jerry's.

For our evaluation we only selected questions seeking entities of the above four types, as other types do not necessarily fall under the definition of an entity accepted in the Entity track, i.e. something that has a homepage. We also manually added target entity types (i.e., "Location", "Product", "Person" or "Organization") to make the questions conform to the Entity track topic format. The statistics of the

‡ statistically significant improvement over the baseline (TFIDF) at 0.01 level

questions used in our evaluation are given in Table 9. The total number of all list questions in the QA 2005 and 2006 datasets is 93 and 89 respectively.

Table 9. Number of questions with specific target entity types.

Entity Type	QA 2005	QA 2006
Person	34	31
Organization	8	14
Product	16	16
Location	16	19
Total	74	80

The evaluation methodology for the list questions in the QA track required the participating sites to submit an unordered set of answer–documentID pairs, where answer is the entity string and documentID is the ID of a document supporting the entity as an answer. The document collection in the official QA track evaluation was AQUAINT. The official evaluation measure was an F-measure, computed as F=(2*IP*IR)/(IP+IR), where Instance Recall (IR) is the number of distinct instances (entities) judged correct and supported by a document out of the total number of known correct and supported instances, and Instance Precision (IP) is the number of distinct instances judged correct and supported by a document out of the total number of instances returned.

Unfortunately, it is not possible to use this evaluation methodology post-TREC since the number of judged supporting documents is very limited. Track organisers released sets of patterns representing the correct answer strings extracted from the answer pool, in order to allow researchers to perform post-TREC evaluations. The set contains only one pattern representing each correct answer, and takes the form of a regular expression, such as "(Holland|Netherlands)". Two major limitations of this pattern set are: it only contains correct answers from the pool, and may therefore be incomplete for some topics, and, secondly, the patterns themselves may not exhaustively cover all spelling and lexical variations of answers.

The evaluation reported in this section was performed using these patterns. F-measure as well as standard evaluation measures used in the Entity track of TREC 2010 were calculated. Since supporting documents are not used, Instance Recall is re-defined as the number of distinct instances that match patterns out of the total number of patterns for the question, and Instance Precision as the number of distinct instances that match patterns out of the total number of instances returned. Each pattern can only be matched once, in other words, any repeated matches on the same pattern are ignored. The evaluation results are shown in Tables 10 and 11.

Table 10. Evaluation results on 74 QA 2005 topics.

Run	nDCG@R	P@10	MAP	R-prec	Rel. retr.	F-measure
PMI	0.0377	0.0351	0.0368	0.0365	225	0.0638
Chi	0.0856	0.0824	0.0695	0.0781	297	0.0707
TFIDF	0.1469	0.1432	0.1241	0.1362	349	0.0831
TFIDFEntitySeedBM25 (b=0.8; k=0.8; ß=0.5)	0.1561	0.1473	0.1299	0.1440	359	0.0854
TFIDFEntitySeedLin (ß=1)	0.1542	0.1436	0.1280	0.1436	355	0.0845

Table 11. Evaluation results on 80 QA 2006 topics.

Run	nDCG@R	P@10	MAP	R-prec	Rel. retr.	F-measure
PMI	0.0529	0.0450	0.0432	0.0515	229	0.0561
Chi	0.0716	0.0588	0.0654	0.0640	295	0.0667
TFIDF	0.1469	0.1312	0.1196	0.1390	321	0.0726
TFIDFEntitySeedBM25 (b=0.8; k=0.8; ß=0.5)	0.1598	0.1512‡	0.1363	0.1562	342	0.0774
TFIDFEntitySeedLin (ß=1)	0.1684	0.1500‡	0.1393‡	0.1618	313	0.0781

5. Discussion

The best performance on the TREC 2010 REF topics was obtained by the run "TFIDFEntitySeedBM25", which re-ranks and filters the top 200 entities in "TFIDF". Figures 4 and 5 show differences in performance by topic in nDCG@R and P@10 respectively on the 50 TREC 2010 topics.

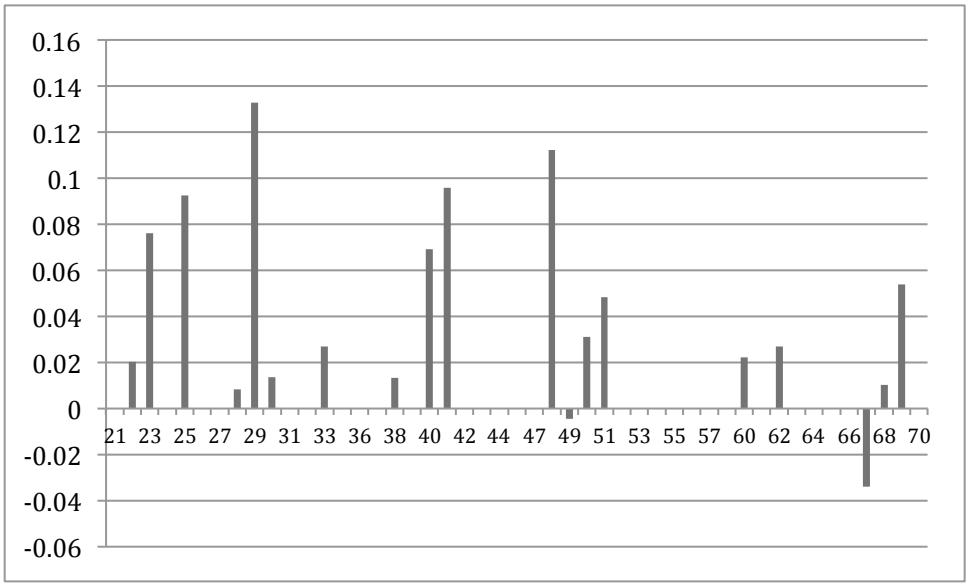

Figure 4. The difference of TFIDFEntitySeedBM25 from TFIDF in nDCG@R by topic (TREC 2010 topics).

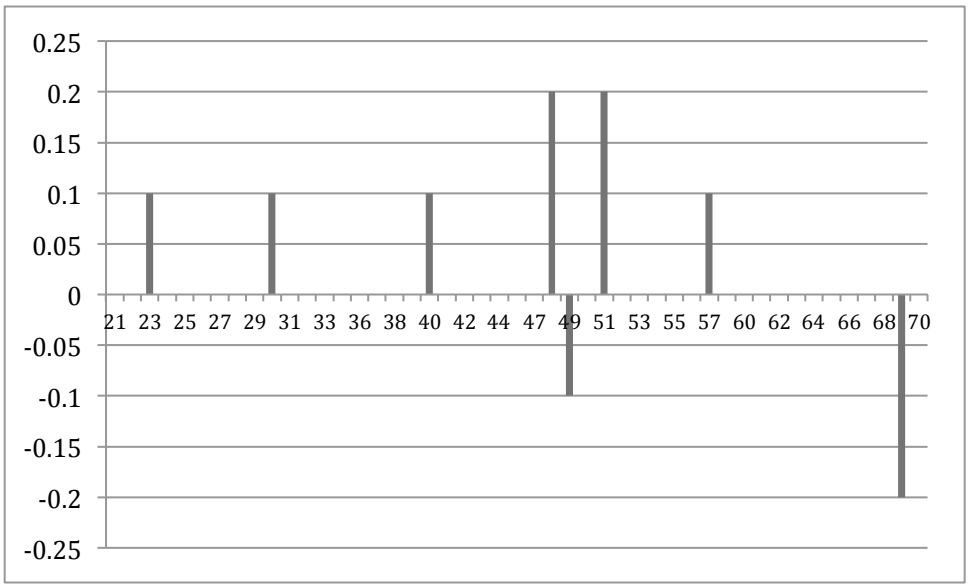

Figure 5. The difference of TFIDFEntitySeedBM25 from TFIDF in P@10 by topic (TREC 2010 topics).

As can be seen from the figures, re-ranking entities by the similarity to automatically extracted seed words overall has a positive effect on performance. Entity re-ranking is performed only if more than one seed

entity is found (Section 3.3.2). Out of 50 TREC 2010 topics, 39 have more than one seed entity. For the remaining 11, the TFIDF entity ranking is kept. The average number of seeds for all 50 topics is 16.6. We identified the following reasons for finding zero seeds for some topics:

a) A small number of hyponyms found. For example, in topic #47 (Who are the balloon manufacturers associated with the Albuquerque International Balloon Fiesta?) the category name is correctly identified as "balloon manufacturers". Fourteen sentences were found that matched three of the six hyponymy templates. While the sentences contained the names of 22 balloon manufacturers, only two hyponyms were extracted. Such low recall is primarily due to errors in NE tagging, i.e., either wrong tags were assigned, or a correct entity was not tagged as NE. Out of 22 entities, only 8 were assigned the correct tag "ORG", six of which our method failed to extract because they followed/preceded an incorrectly tagged NE. For example, in the following sentence: "… [MISC Aerostar] , [PER Cameron] , [ORG Thunder & Colt] , [ORG Sky , Ultra Magic] , [PER Adams] , and other balloon manufacturers" the two NEs tagged with ORG are not extracted since the NE immediately preceding the hyponymy pattern "and other balloon manufacturers" is tagged as "PER". In the end, out of the two identified hyponyms, neither were found in the candidate entities list, and therefore not used as seeds.

b) Errors in category name extraction (only two cases):

- Topic #26: "Who has installed (taken delivery of) a Cray XT computer?" The category name is incorrectly identified as "delivery".
- Topic #63: "What institutions Drew Gilpin Faust been affiliated with, for example as a trustee, board member, president, etc?" The identified category name is "institutions Drew Gilpin Faust".

Below we analyse in detail one of the topics that is improved, and one that is degraded as a result of re-ranking.

Positive example: TREC 2010 REF topic #23 ("What recording companies now sell the Kingston Trio's songs?"). TFIDFEntitySeedBM25 has improvements over TFIDF in all measures, specifically nDCG@R increased from 0.225 to 0.301, and P@10 from 0.1 to 0.2. The known correct answers are "capitol records", "decca records" and "vanguard records". The category word extracted from the narrative is "recording companies", and the seed entities automatically identified using the method described in Section 3.3.2 are: "warner bros", "decca", "columbia records", "capitol records", "bear family records"[5]. While all belong to the category "recording companies", only two out of five represent the correct answers. Out of all extracted grammatical dependency triples (features) that co-occur with the seed entities, 24 co-occur with at least 50% of all seed entities, and are therefore used in computing similarity. The top features ranked by the number of seed entities co-occurring with them are listed in Table 12.

Table 12. Top 10 features ranked by the number of co-occurring seed entities (TREC 2010 REF topic #23).

Feature	Number of seeds
release V:subj:N X	5
album N:nn:N X	4
X N:nn:N label	4
release N:nn:N X	4
X N:nn:N /	4
label N:subj:N X	4
label N:nn:N X	4
edit V:obj:N X	4
X N:mod:A record	4
compilation N:nn:N X	3

[5] All entity names are converted to lower case by our system

Table 13 shows the top 10 entities ranked by TFIDF and TFIDFEntitySeedBM25.

Table 13. Top 10 entities ranked by TFIDF and TFIDFEntitySeedBM25 (TREC 2010 REF Topic #23).

TFIDF	TFIDFEntitySeedBM25
kingston trio	kingston trio
kingston trio on record	capitol
new kingston trio	amazon.com
purple onion	decca records
folk kingston trio	capitol records
amazon.com	columbia records
capitol	beach boys
capitol records	fleetwood mac
the kingston	elektra records
kingston trio story	new christy minstrels

TFIDF method found only one name of a recording company (“capitol records), which is also a correct answer, while TFIDFEntitySeedBM25 found four recording company names (“columbia records”, “elektra records”, “capitol records” and “decca records”), with the last two being the correct answers. This demonstrates how re-ranking of the entities by their similarity to seeds promotes entities of the correct category to the top.

Negative example: QA 2005 topic #97.6 (“Counting Crows. List the Crows' band members.”). After re-ranking, performance dropped from 0.5896 to 0.1909 in nDCG@R and from 0.6 to 0.1 in P@10. The known correct entities for the topic are: "adam duritz", "matt malley", "david bryson", "dan vickrey", "ben mize" and "charlie gillingham". The extracted category name is “band members”. The hyponymy finding method extracted 124 entities, with only the following six qualifying as the seed entities: “tom petty”, “morrison”, “david”, “dave”, “simon” and “bass”. While, “tom petty” refers unambiguously to a musician, three other entities (“morrison” and, to a greater extent, “dave”, david” and “simon”) are ambiguous. The last entity (“bass”) has been erroneously tagged by the NE tagger as “person”. Given the type of seed entities found, it is hardly surprising that the resulting entity ranking is poor. Refinement of the automatic seed finding algorithm is one of our top priorities for the future work. The top 10 features ranked by the number of co-occurring seed entities are given in Table 14, and the top 10 entities ranked by TFIDF and TFIDFEntitySeedBM25 are shown in Table 15.

Table 14. Top 10 features ranked by the number of co-occurring seed entities (QA 2005 topic #97.6).

Feature	Number of seeds
start V:subj:N X	6
play V:subj:N X	6
cd N:nn:N X	5
X N:nn:N lyric	5
video N:nn:N X	5
remain V:subj:N X	5
receive V:subj:N X	5
produce V:subj:N X	5
play V:obj:N X	5
offer V:obj1:N X	5

The TFIDF method found all known relevant entities among the top 10 entities retrieved, while TFIDFEntitySeedBM25 found only one. Interestingly, the latter promoted to the top unrelated entities, but which, nonetheless, represent musicians, such as "joni mitchell", "bruce springsteen" and "bob dylan". One thing they have in common compared to the correct answers to this topic is that they all occur more frequently in ClueWeb09 Category B corpus. This means that they co-occur with a larger number of features and, consequently, are likely to have more features in common with the seeds, thereby receiving higher scores. So, it appears that while re-ranking fulfils its objective, i.e. promotes entities of the correct category to the top, more needs to be done to improve ranking by the strength of association of such entities with the topic.

Table 15. Top 10 entities ranked by TFIDF and TFIDFEntitySeedBM25 (QA 2005 Topic #97.6).

TFIDF	TFIDFEntitySeedBM25
duritz	duritz
adam duritz	**adam duritz**
matt malley	van morrison
david bryson	hard candy
dan vickrey	jones
hard candy	geffen
van morrison	bob dylan
ben mize	john mayer
jim bogios	bruce springsteen
charlie gillingham	joni mitchell

Generally, it appears that the number of features with above-zero frequency per seed entity is positively correlated with performance. The Pearson coefficient between the average number of such features per seed per topic and nDCG@R is 0.3. There is also a positive correlation (0.4) between nDCG@R and the number of seeds per topic. Both were calculated on 43 QA 2005 topics, which have more than one seed. It is also interesting to see which grammatical relationships are most representative in the features of both seed and candidate entities. Table 16 contains eight most frequent grammatical relationships in the feature vectors of candidates and seeds from TREC 2010 topics.

Table 16. Most frequent grammatical relations in the feature vectors of entities.

Grammatical relationship	Frequency	Example
Noun – Noun modifier	1463	Columbia Records label
Verb – subject	887	Columbia Records released
Verb – object	238	join Columbia Records
Apposition	185	was signed to a major label, Columbia Records
Number	135	US sales: 4,000,000 (Columbia Records)
Conjunction	130	Motown Records, Columbia Records
Genitive	114	Columbia Records' roster
Noun – Adjective Modifier	73	famous Columbia Records

6. Conclusions

In this paper we propose an approach to finding related entities which relies primarily on statistical and linguistic methods. The approach was evaluated using the Entity track dataset of TREC 2010, as well as the QA track list questions from TREC 2005 and 2006. Below we summarize our findings and possible directions for future work with respect to each research question formulated in Section 1.

RQ1: How to measure the relatedness of the candidate answer entity to the query?

The candidate answer entities in our experiments were extracted from the sentences containing at least one query term plus one preceding and one following sentences in the top 50 documents retrieved for the query. Three measures were evaluated for ranking these entities: Pointwise Mutual Information, Pearson's χ^2, and TF*IDF. Of the three, TF*IDF was the most effective. It may be possible to achieve further improvements by investigating other methods of (a) finding the initial set of candidate answer entities and (b) calculating their association with the query. For instance other researchers (see Section 2.1) made use of the HTML document structure, extracting anchor text and table elements.

RQ2: How to estimate the likelihood that the candidate answer entity belongs to the target entity category specified in the query?

Our approach was to identify target entity category in the narrative, find its hyponyms to be used as seeds, and compute distributional similarity of candidate answers to the seeds. The method for identifying entity category names works well. Only 2 topics out of 50 in TREC 2010 REF dataset have incorrectly extracted category names. The seed finding method, however, needs more work. Out of 50 topics in the same dataset, 11 have one or zero seeds. Our analysis indicates that the number of seed entities is positively correlated with the performance of the entity finding system, therefore we need to investigate ways of increasing the number of seed entities, while still maintaining high confidence that they are representative of the target entity category. In future, other hyponymy finding methods need to be evaluated for this purpose. Finally, we developed a novel method for calculating distributional similarity between words using an IR model (BM25). The method showed improvements over Lin's distributional similarity method on two datasets. Furthermore, Lin's method was only useful for filtering out non-relevant entities, but not for ranking (α=1 in Eq. 6 gave best results), whereas BM25 was useful in ranking candidate entities (α=0.5 gave best results).

RQ3: Is the likelihood that the candidate answer entity belongs to the target entity category useful in identifying correct answer entities?

Evaluation results show that re-ranking of candidates by their similarity to seeds is effective, with some improvements being statistically significant over the baseline (TF*IDF). A detailed analysis of sample topics in Section 5 reveals that the method is generally good at promoting entities of the correct category to the top ranks, which is its primary purpose. However, more work needs to be done at better integrating this measure with the topic association measure. As demonstrated in one of the examples in Section 5, entities that have strong similarity to seeds, and belong to the correct category, but have weak association with the topic, may be ranked highly. This is especially the case when the correct answers are low-frequency words/phrases, and hence there is little context available for accurately determining their distributional similarity to the seeds.

One aspect of related entity finding that this work does not address is identification of the type of semantic relationship between the topic entity and candidate entities. In some cases, there exists only one or a small number of possible relationships between the topic entity and entities of the correct category type, for instance, "What recording companies now sell the Kingston's Trio's songs?" (TREC 2010 REF topic #23). This could be considered as one of the most typical relationships between a recording company and a musician or band, therefore it is likely that if a musician/band and a recording company name have a strong association in the corpus, they have this relationship. On the other hand, consider a query such as "Name people who have won the Iditarod." (QA 2006, topic #185.5). There could be a number of possible relationships between "people" and "Iditarod", for instance, participants, sponsors, winners. For topics such

as this it may be helpful to make use of the additional information provided in the query to identify entities with the correct relationship.

References

Agichtein E. and Gravano L. (2000) Snowball: Extracting relations from large plain-text collections. In Proceedings of the 5th ACM Conference on Digital Libraries.

Balog K., Soboroff I., Thomas P., Bailey P., Craswell N., de Vries A. (2008) Overview of the TREC 2008 Enterprise Track. In Proceedings of the Seventeenth Text REtrieval Conference, Gaithersburg, MD, November 18-21, 2008.

Balog K., de Vries A., Serdyukov P., Thomas P., Westerveld T. (2009) Overview of the TREC 2009 Entity Track. In Proceedings of the Eighteenth Text REtrieval Conference, Gaithersburg, MD, November 17-20.

Balog K., Serdyukov P., de Vries A.P. (2010) Overview of the TREC 2010 Entity Track. In TREC 2010 Notebook, Gaithersburg, MD.

Banko M. and Etzioni O. (2008) The tradeoffs between open and traditional relation extraction. In Proceedings of the ACL-08, pp. 28-36, Columbus, Ohio, USA, June 2008.

Brill E. (1995) Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. Computational Linguistics, 21(4), pp. 543-565.

Brin S. (1998) Extracting Patterns and Relations from the World Wide Web. In Proceedings of the WebDB Workshop at 6th International Conference on Extending Database Technology, Valencia, Spain.

Büttcher, S., Clarke, C., & Lushman, B. (2006) Term proximity scoring for ad-hoc retrieval on very large text collections. In Proceedings of the 29th ACM conference on research and development in information retrieval (ACM-SIGIR) (pp. 621–622). Seattle, Washington.

Chen J., Ji D., Tan C.L. and Niu Z. (2005) Unsupervised Feature Selection for Relation Extraction. In Proceedings of IJCNLP, Jeju Island, Korea.

Church K., Gale W., Hanks P., Hindle D. (1991) Using statistics in lexical analysis. In: Zernik U., ed. Lexical Acquisition: Using On-line Resources to Build a Lexicon. Englewood Cliffs, NJ, Lawrence Elbraum Associates, 1991. pp. 115-164.

Culotta A. and Sorensen J. (2004) Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July 2004, Barcelona, Spain.

Dang H.T., Kelly D., Lin J., Overview of the TREC 2007 Question Answering Track. In Proceedings of the Sixteenth Text REtrieval Conference, Gaithersburg, MD, November 5-9, 2007

Davidov D., Rappoport A. and Koppel M. (2007) Fully unsupervised discovery of concept-specific relationships by Web mining. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 232-239, Prague, Czech Republic, June 2007.

Demartini G., de Vries A., Iofciu T., Zhu J. (2009) Overview of the INEX 2008 Entity Ranking Track. In Lecture Notes in Computer Science, Vol. 5631/2009, Advances in Focused Retrieval, pp. 243-252.

Fang Y., Si L., Yu Z., Xian Y., Xu Y. (2009) Entity Retrieval with Hierarchical Relevance Model, Exploiting the Structure of Tables and Learning Homepage Classifiers. In Proceedings of Text Retrieval Conference, Gaithersburg, MD.

Hasegawa T., Sekine S. and Grishman R. (2004) Discovering Relations among Named Entities from Large Corpora. In Proceedings of ACL.

Hearst M. A. (1992) Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th Conference on Computational Linguistics - Volume 2 (Nantes, France, August 23 - 28, 1992), 539-545.

Kambhatla N. (2004) Combining lexical, syntactic and semantic features with Maximum Entropy Models for extracting relations. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July 2004, Barcelona, Spain.

Kaptein R., Koolen M. and Kamps J. (2009) Result Diversity and Entity Ranking Experiments: Anchors, Links, Text and Wikipedia. In Proceedings of Text Retrieval Conference, Gaithersburg, MD.

Kilgarriff A. and Yallop C. (2000) What's in a thesaurus? In Proceedings of the Second International Conference on Language Resources and Evaluation (LREC), Athens, Greece, pp. 1371-1379.

Kotlerman L., Dagan I., Szpektor I., Zhitomirsky-Geffet M. (2009) Directional Distributional Similarity for Lexical Expansion. In Proceedings of ACL-IJCNLP, Singapore, pp. 69-72.

Lin, D. (1993) Principle-Based Parsing Without OverGeneration. In Proceedings of ACL-93. pp. 112-120. Columbus, OH.

Lin, D. (1998) Automatic retrieval and clustering of similar words. In Proceedings of the 17th international Conference on Computational Linguistics - Volume 2 (Montreal, Quebec, Canada, August 10 - 14, 1998), 768-774.

Manning C. and Schütze H. (1999) Foundations of Statistical Natural Language Processing, MIT Press.

Miller S., Fox H., Ramshaw L. and Weischedel R. (2000) A novel use of statistical parsing to extract information from text. In Proceedings of the 6th Applied Natural Language Processing Conference, 2000, Seattle, USA.

McCreadie R., Macdonald C., Ounis I., Peng J., Santos R.L.T. (2009) University of Glasgow at TREC 2009: Experiments with Terrier. In Proceedings of Text Retrieval Conference, Gaithersburg, MD.

Pantel P., Crestan E., Borkovsky A., Popescu A. and Vyas V. (2009) Web-Scale Distributional Similarity and Entity Set Expansion. In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-09). pp. 938-947. Singapore.

Ramshaw L. and Marcus M. (1995) Text Chunking Using Transformation-Based Learning. Proceedings of the Third ACL Workshop on Very Large Corpora, MIT.

Ratinov L. and Roth D. (2009) Design Challenges and Misconceptions in Named Entity Recognition. In Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL).

Ravichandran D. and Hovy E. (2002) Learning Surface Text Patterns for a Question Answering System. In Proceedings of the 40th ACL Conference.

Riloff E. and Jones R. (1999) Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In Proceedings of AAAI.

Robertson S.E., Walker S., Jones S., Hancock-Beaulieu M., Gatford M. (1995) Okapi at TREC-3. In Harman D. (Ed.) Proceedings of the Third Text Retrieval Conference, NIST, Gaithersburg, MD, U.S., pp.109-126.

Rosenfeld B. and Feldman R. (2007) Using Corpus Statistics on Entities to Improve Semi-supervised Relation Extraction from the Web. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic.

Spärck Jones, K., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments. Information Processing and Management, 36(6), 779–808 (Part 1); 809–840 (Part 2).

Thelen, M. and Riloff E. (2002) A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In Proc. of EMNLP 2002.

Vechtomova O., Robertson S. E., Jones S. (2003) Query expansion with long-span collocates. Information Retrieval, 6(2), 251-273.

Vinod Vydiswaran V.G., Ganesan K., Lv Y., He J., Zhai C.X. (2009) Finding Related Entities by Retrieving Relations: UIUC at TREC 2009 Entity Track. In Proceedings of Text Retrieval Conference, 2009, Gaithersburg, MD.

Weeds J. and Weir D. (2006) Co-occurrence retrieval: a flexible framework for lexical distributional similarity. Computational Linguistics, 31(4), 429-475.

Wu Y., Kashioka H. (2009) NiCT at TREC 2009: Employing Three Models for Entity Ranking Track. In Proceedings of Text Retrieval Conference, Gaithersburg, MD.

Zelenko D., Aone C. and Richardella A. (2002) Kernel Methods for Relation Extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, USA.

Zhai H., Cheng X., Guo J., Xu H., Liu Y. (2009) A Novel Framework for Related Entities Finding: ICTNET at TREC 2009 Entity Track. In Proceedings of Text Retrieval Conference, Gaithersburg, MD.

Zhang Z. (2004) Weakly-supervised relation classification for information extraction. In Proceedings of the 13th ACM conference on Information and Knowledge Management (CIKM), Washington, DC, USA.

Zhou G. Jian S., Zhang J. and Zhang M. (2005) Exploring various knowledge in relation extraction. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL).