Finding Support Sentences for Entities

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

Hugo Zaragoza
Yahoo! Research
Barcelona, Spain
hugoz@yahoo-inc.com

ABSTRACT

We study the problem of finding sentences that explain the relationship between a named entity and an ad-hoc query, which we refer to as *entity support sentences*. This is an important sub-problem of entity ranking which, to the best of our knowledge, has not been addressed before. In this paper we give the first formalization of the problem, how it can be evaluated, and present a full evaluation dataset. We propose several methods to rank these sentences, namely retrieval-based, entity-ranking based and position-based. We found that traditional bag-of-words models perform relatively well when there is a match between an entity and a query in a given sentence, but they fail to find a support sentence for a substantial portion of entities. This can be improved by incorporating small windows of context sentences and ranking them appropriately.

Categories and Subject Descriptors

H.3.3 [**Information Storage And Retrieval**]: Information Search and Retrieval—*retrieval models*; H.3.4 [**Information Storage And Retrieval**]: Systems and Software—*performance evaluation (efficiency and effectiveness)*

General Terms

Algorithms, Experimentation, Performance

Keywords

Sentence Retrieval, Entity Ranking

1. INTRODUCTION

Ranking entities with respect to a query has become a standard information retrieval (IR) task, often referred to as *entity ranking*. People and expert search are the best known entity ranking tasks, which have been conveniently evaluated in the Text REtrieval Conference (TREC [27]) in the past years [21, 22, 2]. Recently, the different types of entities and the number of entity search applications has increased: finding the most important dates or events for a query, the most important locations, companies and so on.

However, presenting a list of entities to the user without any explanation is not sufficient: the entity needs to be contextualized for the user to decide its relevance and relationship with the query, very much like *snippets* help users to select documents from a ranked list of query results. In this paper we tackle the problem of retrieving and ranking sentences that explain the interest (or relevance) of an entity with respect to a query; we call these sentences *entity support sentences*. Note that we are not interested in the entity ranking task (choosing which entities are relevant). We are only interested in finding *explanations* for relevant entities; this is in fact a *sentence ranking* task.

As an illustration, we discuss some examples of good entity support sentences. One can observe different types of support sentences, depending on the generality of the query and the entity, their type of relationship, etc. We tried to represent these types in the examples in Table 1. For example, support sentence (5) is a typical *definition*: it defines the entity, and in doing so, it clarifies the relationship with the query. This is perhaps the easiest type of sentence, since it is query independent, and often used in practice (for example, systems that display the first paragraphs of the Wikipedia entry corresponding to the entity, regardless of the query). Nevertheless, definitions are insufficient in many cases. For example, support sentence (1) is not a simple definition of the entity Picasso (e.g. a XXth century painter), it specifically addresses the "peace" aspect of the query. Note that some support sentences have partial or no matches with the query terms. There are several reasons why this might happen: the usual synonymy or anaphora problems in IR (as in (3), where Picasso is referred to as "the author"), or more complex sentences requiring some domain knowledge and inference (as in (5)).

The main contributions of this paper are:

- a formalization of this problem, which to our knowledge has not been addressed by the IR community so far,
- an evaluation framework and an initial evaluation dataset,
- an empirical evaluation of several families of methods, *including bag-of-words, entity-ranking based and position-dependent ranking*, showing that including a small weighted *context* window of surrounding sentences improves performance of sentence retrieval.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.

Table 1: Examples of entities support sentences.

Query: Picasso and peace
Entity: Picasso.
Support Sentence: "In 1944 Picasso joined the French Communist Party, attended an international peace conference in Poland, and in 1950 received the Stalin Peace Prize from the Soviet government." (1)
Entity: 1944
Support Sentence: (same as above)
Entity: Northern Spain
Support Sentence: "Although it was not conceived by the author as a representation of the disasters of war, but the Nazi bombing of Guernica (a town in Northern Spain), it is now considered an iconic representation of the disasters of war." (3)
Query: Spanish Civil War
Entity: International Brigades
Support Sentence: "The International Brigades were Republican military units in the Spanish Civil War, formed of many non-state sponsored volunteers of different countries who traveled to Spain, to fight for the republic in the Spanish Civil War between 1936 and 1939." (4)
Entity: Franco
Support Sentence: "In 1936, Franco participated in a coup d'etat against the elected Popular Front government." (5)
Query: Lennon and religion
Entity: George Harrison
Support Sentence: "During the late 1960s, bandmate George Harrison became interested in Eastern mysticism; Lennon dismisses Harrison 's beloved gurus and Hare Krishna mantra as "pie in the sky". (6)

Undoubtedly this task is closely related to other tasks in IR, such as sentence retrieval, query expansion, and entity ranking, as detailed in Section 2. However, we believe it is important to clearly formalize this task separately, because its particular characteristics make this problem hard and interesting. A fundamental difference is that the number of entities potentially covered by a single query might be in the order of thousands or millions, which prevents to devise a solution consisting of issuing a single query to find the support sentences for every different entity. Instead, it is more appealing to work with the original set of retrieved sentences and perform some re-ranking among them. In fact, we will extend on this idea to develop context-aware ranking methods which outperform single bag-of-words ranking for traditional sentence retrieval.

The remainder of this paper is organized as follows. Section 2 reviews related work on the IR field and compares it to the support sentence retrieval task. Section 3 presents some sentence-entity ranking features, and Section 4 describes more in depth the dataset manually annotated to evaluate support sentences retrieval. Section 5 presents results of different ranking models and a discussion about the importance of the role of context sentences. The paper ends with conclusions and future work.

2. RELATED WORK

Entity search first appeared in corporate search engines such as Microsoft Office SharePoint Server or Autonomy. These applications deal mostly with metadata entities (like author, document type, or other company-defined annotations) but have recently started to include entities directly extracted from sentences (SAS TextMiner for example). Unfortunately, there is no literature publicly available about the methods used in these applications.

More recently, a number of commercial online applications have sprung dealing specifically with entity searching on the Web (either on the entire Web or on high informational online content such as news and blogs). There are too many to list them all here and little is known about their algorithms, some examples are Evri[1], TextMap [2], Yahoo! Correlator[3] or ZoomInfo[4]. They differ greatly on their sources, user interfaces, entity types, search capabilities and features, as well as the modality of the information. However with respect to their entity ranking functionality they are all very similar: they allow users to search and browse entities related to a topic or to another entity. And they all require solving the task discussed here: finding support sentences for entities.

For example Evri proposes an entity search and browsing site which shows relations between entities as well as links to news stories or web pages where the entities are mentioned. Although it does not provide (yet) a free ad-hoc keyword search, it is possible to search for an entity in the context of another. For example a search for the query *Bill Clinton* returns a definition of the entity, but when one then selects the entity *Korea* (in the context of *Bill Clinton*), Evri brings up snippets relating these two entities, instead of a definition of *Korea*. Finding those snippets is a special case of the task discussed here, where the queries have to be entities themselves.

Another example is Yahoo! Correlator where one can type ad-hoc queries and choose a search type (names, locations, events or concepts). For each search type there is an associated user interface which shows entities relevant to the query; when one hovers over the entities, sentences are shown explaining the relationship between the query and the entity. For example, for the query "Picasso and peace" the entity Neruda appears; when one hovers over it one obtains a sentence that does not define Neruda, but rather states the relationship of Neruda to the query[5]. This is another example of the task of entity support sentence ranking.

Academic research became interested in entity ranking recently, and several evaluation campaigns and competitions have been launched, the most important ones being TREC and INEX. So far research has concentrated with the main problem: identification and ranking of entities. To our knowledge there have been no papers published addressing other entity ranking problems such as generation or ranking of sentences.

Named Entities Recognition (NER) has recently attracted

[1]http://www.evri.com/
[2]http://www.textmap.com/
[3]http://sandbox.yahoo.com/Correlator
[4]http://www.zoominfo.com/
[5]"Pablo Picasso arranged his entrance into Paris and Neruda made a surprise appearance there to a stunned World Congress of Peace Forces, the Chilean government meanwhile denying that the poet could have escaped the country."

much research interest in the IR community. For instance, the *1st Entity Workshop* held in 2009 in TREC is composed of three search tasks which involve entity-related search on Web data. The tasks to be addressed are motivated by the expert finding task at the TREC Enterprise track [21, 22, 2]. INEX has also been running constantly since 2006 an *entity track*, where the goal is to match a textual query to a predefined set of entities that are usually mentioned in text documents and/or described explicitly [11, 28, 5].

The task of finding support sentences can be seen as a specialized form of sentence retrieval, where sentences need to be relevant to both a query and the entity being supported. There is an important body of work on sentence retrieval (see the book from Murdock [13], and references therein). Research in sentence retrieval has been driven by two main topics: relevance retrieval and novelty detection, both mainly geared towards text summarization, question answering [3], topic detection and tracking [24] or a combination of any of them [25]. Li and Croft [9] employ named entity recognition techniques to improve novelty detection in sentence retrieval. They filter out sentences which do not contain specific entity-patterns in the result set and re-rank sentences accordingly. Sentence retrieval has also been employed to assist Question Answering systems (QAs); the motivation is to select a small set of sentences which may contain the answer to a given question and employ QA strategies to them instead to whole documents [3]. None of these works deal with the issue of explaining a query-entity relationship. The closest is perhaps [4] which discusses sentence ranking models where the query includes a constraint on a type of entity (e.g. a location, a person). While these models are interesting and could probably be used to speed up some of the entity support sentence rankings, this work is not directly related to ours.

One way to map our task to the problem of sentence retrieval is to merge the query and the entity into a normal query. In that way we can obtain sentences that are relevant both to the entity and the query, obtaining a good candidate entity support sentence. However, this approach is impractical due to its computational complexity: it requires executing one query per retrieved entity (potentially thousands). . This is also the reason precluding snippet generation techniques [8] from being applied directly to this problem. Instead, we explore methods that do not require issuing any subsequent queries to the retrieval system. On the contrary, all the methods introduced in this paper select one or more support sentences for a query and entity by re-ranking the top retrieved sentences for a given query. A problem of this approach is that of exact-match methods: we would only re-rank and retrieve sentences that partially match the original query. The vocabulary mismatch problem is particularly problematic in sentence retrieval [10] because sentences are short pieces of text and their probability of being relevant to a query term that is not mentioned explicitly in them is higher. We address this issue by introducing a relatively small context window of non-matching sentences surrounding matching sentences (section 5.3). Although non standard, this is quite a natural thing to try, and can be integrated in a relatively simple way in most ranking model paradigms. As an example, this is equivalent to smooth locally a sentence language model using surrounding sentences as proposed in [15]. In this paper we propose to use BM25F to integrate context into ranking, and show that it is effective even for the standard TREC sentence retrieval task.

3. FEATURES FOR RANKING SUPPORT SENTENCES

In this section we introduce the notation used in the paper and describe several features for the problem of entity support sentence ranking.

First we assume that we have a collection of documents, which can be segmented into sentences $s \in \mathbf{S}$ (more generally these could be paragraphs or text windows of fixed size). It will also be useful to consider the sentence's *context* C_s; this context can be defined as surrounding sentences, a passage, the document's title or even the entire document.

We further assume that entities in the collection have been annotated in the text (as a result of automatic or manual information extraction). Entities in the collection are denoted by $e \in \mathbf{E}$. We represent the presence of an entity in a sentence via the matrix $\mathbf{G} \in \{0,1\}^{|\mathbf{S}|\times|\mathbf{E}|}$, where $G_{ij} = 1$ if entity j is present (mentioned) in sentence i, and 0 otherwise. Alternatively we can see $\mathbf{G}$ as a bipartite graph connecting each sentence to the entities mentioned in it. Matrix $\mathbf{G}$ is sometimes referred to as an *entity containment graph* [29, 18]. We will sometimes use the shorthand notation $e \in s$ to denote $G_{se} = 1$.

Our goal is to find a good model for ranking entity support sentences ($H_{qe}(s)$), that scores triples (sentence,query,entity) using sentence scores coming from a retrieval function ($F_q(s)$) and entity scores ($E(q,e)$).

We define the top-k relevant sentences for query q as:

$$\mathbf{S}_q = \{s \,|\, \text{rank}_q(s) < k\} \,, \tag{1}$$

where k is a global parameter. One possible way to incorporate the context into the result set is to augment $\mathbf{S}_q$ with:

$$\hat{\mathbf{S}}_q = \{s \,|\, s' \in \mathbf{S}_q, s \in C_{s'}\}. \tag{2}$$

The set of *candidate support sentences*[6] for an entity e is defined as:

$$\mathbf{S}_{qe} = \{s \,|\, s \in \mathbf{S}_q \,,\, G_{se} = 1\} \,, \tag{3}$$

and

$$\hat{\mathbf{S}}_{qe} = \{s \,|\, s \in \hat{\mathbf{S}}_q \,,\, G_{se} = 1\}. \tag{4}$$

The problem of finding entity support sentences can now be formalized as that of assigning a score $H_{qe}(s)$ to the candidate support sentences.

Now let us discuss some features. A first trivial feature is to use the original score of the sentence. The sentence score $F_q(s)$ can be obtained by any ad-hoc ranking method such as TF-IDF, BM25, language models, etc. In this paper we will use BM25 (see Section 5.1). Formally:

$$H_{qe}(s) = F_q(s) \qquad \forall\, s \in \mathbf{S}_{qe} \tag{5}$$

Ranking sentences using Equation (5) trusts a retrieval scoring function for determining the relevance between query and entity. In other words; it is equivalent to order sentences with respect to their relevance score with respect to

[6] In theory, it is possible that a support sentence does not mention e (due to anaphora) but this is rare and not studied it in this paper.

the given query (without considering the entity) and then filtering out all the sentences not containing the entity. We can also take into account the context of a sentence inside the ranking function, replacing $F_q(s)$ by a context-aware model $F_q(s, C_s)$. There are different ways to do this, such as using query expansion, smoothing, or structural ranking functions. In this paper we will use BM25F (see Section 5.1), which allows to score multiple fields; we put C_s in a context field, separate from the sentence field s. Formally:

$$H_{qe}(s) = F_q(s, C_s) \qquad \forall\, s \in \mathbf{S}_{qe} \tag{6}$$

These two features take into account the statistics and distribution of terms in sentences. Instead, we can also look into the statistics of the *entities*. Therefore, a second ranking method could take the scores of entities in the sentence into account ($E(q, e)$, detailed next in section 3.1):

$$H_{qe}(s) = \begin{cases} \sum_{e' \in s} E(q, e'), & \text{if } e \in s \\ 0, & \text{if } e \notin s \end{cases} \qquad \forall\, s \in \hat{\mathbf{S}}_{qe} \tag{7}$$

Note that here we employ a summation but other aggregation functions (such as the *average, max* and *min*) are also possible.

Another interesting feature of sentences is the position in which the entity and the query terms are found. We tried several heuristic position-dependent models; as an example we report the best performing one: the distance between the last match of query and entity, and the length of the sentence:

$$H_{eq}(s) = length(s) - max(position(q), position(e)) \tag{8}$$

where $position(q) = 0$ if none of the query terms are present in s. This can be regarded as a proxy for deeper linguistic features, since important elements of a sentence tend to occur in higher levels of the syntactic dependencies tree; terms that are central to a given sentences tend to appear in lower levels.

We have also explored features derived from the analysis of the entity containment graph topology as discussed in [18], but we could not obtain interesting results. For lack of space we decided to omit their description and results.

3.1 Entity Ranking

In order to find different models for $H_{qe}(s)$ we will take advantage of some entity ranking methods and incorporate them into H_{qe}.

We describe next some simple and efficient methods for ranking entities; all of them are applied in the context of Equation (7) substituting the $E(q, e)$ function.

One of the simplest entity ranking methods is the number of relevant sentences containing it (hereinafter *frequency*):

$$E_{FREQ}(q, e) = |\mathbf{S}_{qe}| \tag{9}$$

To penalize very frequent entities (such as entity descriptors like "person"), we use the entity inverted sentence frequency [29] (hereinafter *rarity*):

$$E_{RARITY}(e, q) = \log \frac{|\mathbf{S}|}{\sum_{s \in \mathbf{S}} G_{se}} \tag{10}$$

This is similar to the traditional inverse document frequency [16].

Combining these two measures we obtain a very accurate entity ranking function [29], which resembles the well-known TF-IDF weighting scheme [7] (hereinafter *combination*)

$$E_{COMB}(q, e) = E_{FREQ}(q, e) \cdot E_{RARITY}(e) \tag{11}$$

In [26] a slightly different measure was presented, inspired in the cross entropy of the query and collection distributions and measured with KL-divergence (hereinafter *KLD*):

$$E_{KLD}(q, e) = P(e|\theta_q) \log \frac{P(e|\theta_q)}{P(e|\theta_S)} \tag{12}$$

where

$$P(e|\theta_q) = \frac{|\mathrm{S}_{qe}|}{|\mathbf{S}_q|}\ , \quad P(e|\theta_S) = \frac{\sum_{s \in S} G_{se}}{|\mathbf{S}|} \tag{13}$$

We note that the query and sentence models θ_q and θ_S can be parametrized to account for smoothed probability estimations, though in Equation (13) we assume simple parameter-free count-based models.

More complex entity ranking methods have been proposed in the literature, but their computational cost is orders of magnitude higher than the techniques just presented, especially when dealing with millions of potential entities. Since the purpose of this paper is not to evaluate entity ranking methods, we have limited ourselves to these methods, which have been proved to be effective and have a very low computational cost and memory footprint. However we remark that the described support ranking methods account for word-based relevance (Equations (5) and (6)) and entity-based relevance (Equation (7) and subsequent definitions of $E(q, e)$).

4. TASK EVALUATION

We can evaluate this task similarly to how standard retrieval ad-hoc tasks are evaluated. First, we ask human subjects to produce queries about topics they know well. We then produce a (large) set of candidate entities and ask the subject to eliminate the entities that are not relevant to the query (this is similar to how entity ranking could be evaluated [21, 29, 26]). Finally, for every entity selected, we produce a number of candidate sentences for each (query,entity) pair and ask the subject to evaluate them as good or bad support sentences *for that query and for that entity.* By definition, we require that the sentence should mention the entity, otherwise it becomes very difficult to judge them.

Human subjects evaluate sentences (for example as bad, adequate or excellent) assigning them grades, (noted $Grade(q, e, s)$). Using these grades we can now compute standard IR performance measures such as Precision and Recall, Mean Reciprocal Rank (MRR), Discounted Cumulative Gain (DCG), etc. These measures can be normalized in two ways: by (entity, query) pair, or first by entity and then by query. In this paper we report scores normalized by (entity,query) pair since we are interested in providing sentences for all entities, regardless of their number.

We will be interested in precision much more than recall, for two reasons. First, there may be many sentences explaining the relationship between a query and an entity, but the user is typically interested in one (except for very specific settings like in forensics or legal applications). Furthermore, entity retrieval applications tend to have very crowded user

interfaces and leave little space for sentences and typically only one or two sentences are displayed.

We now describe how we built our evaluation collection. As corpus, we used the Semantically Annotated Snapshot of the English Wikipedia v.1 (SW1) [1]. The collection contains around 75M sentences (coming from 1.5M documents) and 20.3M unique named entities (with type). From all the annotations we used only the 12 first level Wall Street Journal entity types (e.g. person, location, facility, etc.), removing *DESC* entities (description) within their tag, as it is too hard to provide an accurate evaluation due to its broad semantic generality.

Firstly, we built a manually evaluated dataset of 226 (query, entity) pairs with 45 unique queries (see Table 1 for examples). These queries were entered manually by the assessors and a random selection of relevant entities was considered for evaluation. The total number of obtained relevance judgments was 4814. Judges were asked to assign a $Grade(q, e, s)$ using four levels of relevance: 1 for non-relevant, 2 for fairly relevant, 3 for relevant and 4 for very relevant. A triple (q, e, s) is considered relevant iff $Grade(q, e, s) \geq 3$. We consider that if there is a support sentence for a (query,entity) pair then the entity must be relevant for the query, thus separating entity and sentence ranking evaluations. The collection is available trough Yahoo's! Webscope program, and we have made available the evaluation data[7].

The particular task described through the paper requires to retrieve one or two relevant support sentences per (query, entity) pair; this is why we focus on top-precision retrieval performance metrics: NDCG[8], MAP, P@1 and MRR (which is arguably the best suited one). Because in principle many sentences scores might be the same, the evaluation can be biased: if a number of sentences have the same score the models introduced are not able to decide how to order them. This is potentially problematic not only for some of our models but also for standard ranking methods for sentences. Our solution is to employ tie-aware evaluation [12] which takes into account the fact that score-tied sentences could have been ranked randomly. The final performance value is the average over all possible permutations on the ties, which can be efficiently computed in linear time. As a consequence, all these measures (including high-precision measures like P@1, MRR) might be affected by adding more results from the lower part of the ranked documents, which could introduce ties in the top rank.

5. EXPERIMENTS

In this section we evaluate the new dataset just described in Section 4 using different models for H_{qe}.

5.1 Ranking Models

The support sentences ranking functions H_{qe} described (Equations (5), (6) and (7)) use different models for F and E. The specific models tested are described in this section.

We employ BM25 [19] as a model for $F_q(s)$, BM25F [17] for $F_q(s, C_s)$ and different options for E. BM25 models use a standard parametrization with parameters k_1 and b [17]. In our setting, H_{qe} functions operate on a top-k set for a given query (S_q) that can be augmented with a context C_s.

[7]http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=support_sentence_evaluation_set

[8]gains of $(0, 1, 3, 7)$ and decaying factor of $log(1 + \text{rank})$

The context of a sentence was defined as the surrounding four sentences (two preceding, two following) plus the title of its Wikipedia entry. We represented each sentence in three fields: the first with the sentence s, the second with surrounding sentences, and the third with the title. This resulted in BM25F parameters k_1, (w_1, w_2, w_3) and (b_1, b_2, b_3). As customary in BM25F, and without loss of generality, we set $w_1 = 1$. To reduce the number of parameters we tied all b parameters to a single parameter b. This resulted in the four parameters (k_1, w_2, w_3, b).

In both cases parameters were optimized, for each value of k, by a greedy algorithm (described in [17]) using 2-fold cross validation; results reported are the average over the two test sets[9].

We also experimented with a simpler context model, where surrounding sentences and title are concatenated to the sentence which was ranked with BM25. This led to poor results, and therefore were omitted from this paper.

Another batch of features for sentence ranking are aggregations of entity ranking functions, as stated by Equation (7) which exemplifies the *sum*; we also report on the *average* aggregator. Aggregated functions are the models selected for $E(q, e)$. We include results for *Frequency* (Equation (9)), *Rarity* (Equation (10)), *Combination* (Equation (11)) and *KLD* (Equation (12)).

We experimented with $k = 1000$ and $k = 4000$. We note that in practice, increasing k has a high performance cost. The main reason is that in order to find candidate support sentences for an entity e we need to check at run time if each scored sentence contains e or not. This is in fact a query operation on the entity containment graph $\mathbf{G}$; even with fast-access dedicated structures the cost of this increases linearly with k and it is costly even for k values in the order of a few thousands.

5.2 Results

We will evaluate here the different methods discussed in Section 3 on the SW1 collection. Table 2 summarizes the results.

First we note that BM25 with $k = 1000$ obtains reasonable results in this task; e.g., it achieves a MRR of 0.61. However, context can be exploited to obtain better results. Using the sentence context with BM25F produces a large improvement in the results, obtaining a 16% relative improvement in MRR and 20% in NDCG.

We next investigate the interest of entity-ranking features. First we note that most improve over BM25, but none over BM25F. Recall that these features work re-ranking the extended set $\hat{\mathbf{S}}_q$. It is quite remarkable that some of these features can improve over BM25 given that they are parameter-free. This indicates that these features are very informative for the task. The sum aggregator stands out slightly among

[9]For $k = 1000$ the resulting BM25 parameters for MRR in each run was $k_1 = (1, 1)$ and $b = (0.18, 0.22)$. For this set, the over-fitted maximum (using the entire set) is $k_1 = 1$ and $b = 0.18$, so the trained values are quite close to the optimum. For BM25F, the resulting parameters were: $k_1 = (0.2, 1)$, $b = (0.62, 0.11)$, $w_2 = (0.64, 0.22)$ and $w_3 = (0.10, 0.50)$, being the over-fitted maximum at $k_1 = 0.26$, $b = 0.15$, $w_2 = 0.23$ and $w_3 = 0.23$. In this case there is a higher variance in the parameter selection, although performance on the test sets is not far from the optimal values; this indicates that BM25 and BM25F parameters are quite robust.

Table 2: Performance of Entity Support Ranking Methods (* = statistical significance at $p < 0.05$ using the Wilcoxon signed-ranks test with respect to BM25).

	MRR	NDCG	P@1	MAP
$k = 1000$				
BM25	.61	.59	.57	.45
BM25F	**.71***	**.71***	**.66***	**.53***
Position dependent	.69*	.64*	.65	.51*
Sum Frequency	.67*	.67	.58*	.52*
Sum Rarity	.55	.60	.42	.42
Sum Combination	**.71***	.67*	.60*	**.53***
Sum KLD	.67*	.66*	.59*	.51*
Average Frequency	.62*	.63*	.51	.50*
Average Rarity	.47	.54	.32	.40
Average Combination	.63	.64	.53*	.51*
Average KLD	.63*	.64*	.53	.50*
$k = 4000$				
BM25	.63	.61	.57	.53
BM25F	**.76***	**.75***	**.69***	**.58***

the rest, especially if used together with the *combination* entity ranking scorer. The features could be ranked in the following way: *Combination* > *KLD* > *Frequency* > *Rarity*, and the aggregators as *Sum* > *Average*. We also experimented with other aggregation techniques besides the sum and average but their performances are always inferior and we do not report them.

Nevertheless, these features are working on a candidate set that is much larger than $\mathbf{S}_q$ (as explained in Section 5.3). To make a fair comparison, we also report (bottom of Table 2) results for BM25 and BM25F for $k = 4000$, which yields result sets of size comparable to $\hat{\mathbf{S}}_q$. We see that increasing k improves only slightly BM25; this demonstrates that these features are informative. BM25F with k=4000 improves even more (25% relative in MRR and 27% in NDCG). This is further investigated in the next section.

5.3 The Role of Context

Context plays a crucial role in the successful retrieval of support sentences. In Section 3 we introduced context formally and in Section 5 we saw that features exploiting context improved results significantly. In this section we explore the different roles of context and provide a more in-depth and intuitive introduction of the role of context in this task and in sentence retrieval in general.

First, let us look at the problem, illustrated in figure 1 Given a fixed query q and a fixed entity e, there is a relevant set $R_{q,e}$ of sentences which are *correct support sentences* (i.e., good explanations of the relevance of e to q). Now consider the top retrieved sentences for q, noted $\mathbf{S}_q$. We know it intersects $R_{q,e}$ because otherwise simple models such as BM25 could not perform well. But it is often the case that some support sentences do not contain a match of any of the query terms at all, which are the ones outside of $\mathbf{S}_q$. There are many reasons for this, such as anaphora and synonymy among others. This is a typical problem in IR, but the extremely short length of sentences (compared to documents) exacerbates this problem

We have addressed this problem by considering sentences preceding and following matching sentences. Indeed, we found that good support sentences for an entity are often found in the sentences immediately before or after the sentence that mentions the entity. The rationale behind this is that if a sentence provides a good description to a concept related to a information need, the referred query and entity must appear somehow nearby in a document. Pronouns for example usually refer to entities mentioned in the one or two preceding sentences; beyond pronouns, many complex linguistic relations conspire to make nearby words relevant to each-other. We have followed two alternative paths to take the context into account.

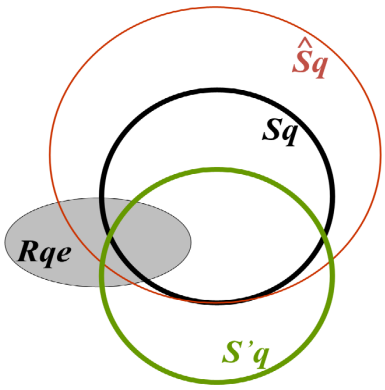

Figure 1: Venn Diagram of sets $\mathbf{R}_{q,e}$,$\mathbf{S}_q$, $\mathbf{S}'_q$ and $\hat{\mathbf{S}}_q$.

Our first method is to score sentences without context first, obtaining $\mathbf{S}_q$, and then extending this set with the context of each of its sentences ($\hat{\mathbf{S}}_q$). Note that by construction $|\mathbf{S}_q| = k$ and $\mathbf{S}_q \subseteq \hat{\mathbf{S}}_q$, and therefore $|\hat{\mathbf{S}}_q| \geq k$. Therefore the top-$k$ sentences in $\mathbf{S}_q$ are necessarily somewhere in $\hat{\mathbf{S}}_q$. Note that *not* all of the new sentences introduced had an initial score of zero; some might have had a small score that did not put them in the top-k. The second method is to score sentences taking into account the context in the ranking function itself. In this case we obtain a different $\mathbf{S}'_q$. It is important to understand that at as $k \to \infty$ we would have $\mathbf{S}'_q = \hat{\mathbf{S}}_q$ and $\mathbf{S}_q \subseteq \mathbf{S}'_q$, but for fixed k this is not the case. Instead $|\mathbf{S}'_q| = k$ and $\hat{\mathbf{S}}_q \neq \mathbf{S}'_q \neq \mathbf{S}_q$. These two methods are fundamentally different although complementary.

Figure 2 shows the sizes of the sets $\mathbf{S}_q$ (BM25 no context) and $\mathbf{S}'_q$ (BM25 context window) with respect to the "percentage of queries answered". By this we mean the number of (query,entity) pairs for which at least one relevant (support) sentence was found. This number provides an upper bound on precision, since it is impossible for a ranking function to find a relevant sentence outside of the $\mathbf{S}$ set scored. We see that without the context, BM25 is not able to find any relevant results for approximately 20% of the queries, whereas including the context this number is less than 1%. The figure also reveals that a pure bag of words approach struggles to cover a high percentage of the queries which require a very large k value in order for BM25 to find any answer for them.

It was mentioned before that BM25F is one way of introducing context into sentence retrieval. In order to demonstrate that BM25F performs well for early-precision metrics in standard sentence retrieval, we experiment with the three TREC Novelty collections. The TREC Novelty Track ran for three years (2002-2004) [6, 23, 20]. The tracks included

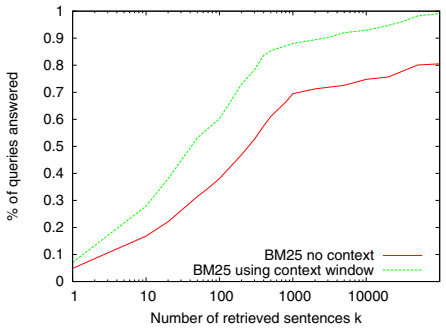

Figure 2: Percentage of support sentences found when varying the number of top retrieved sentences with and without using a context window of size 5.

an ad-hoc sentence retrieval task consisting of 50 topics for each year. We use the title of the topics as queries (as these are more representative of Web search) and do very little pre-processing to the collections (no stemming or stop-words removal). Results are presented in table 5.3.

A common baseline for ranking sentences is the TF-IDF measure [13], that we include for comparison. The standard performance metric is the F measure which only takes into account the proportion of relevant elements over the total number of sentences retrieved; however, as explained before, in our particular approach we are more interested in early precision metrics (like P@X and MRR). We also introduce results using BM25 for comparison. We tuned BM25 and BM25F's parameters using two-fold cross validation (denoted †), as well as using the entire collection (denoted *). Note that we tune for P@10 only and report on every measure for the best P@10 run.

BM25F outperforms BM25 and TF-IDF in almost every case for P@10, MRR and MAP, whereas being inferior in terms of the F measure in two collections. This may be due to the fact that the context introduces many non-relevant low-ranked sentences which affect the F measure but are mostly unlikely to appear in the top results presented to the user. Although a more thorough comparison with other context and query expansion approaches for sentences retrieval is out of the scope of this paper, the performance numbers obtained are better than ones reported in previous work (for example [14] reports a P@10 of 0.11 in the TREC Novelty 2002 collection) and comparable or better than well-tuned query expansion techniques [10] in every collection. These results agree with the findings reported on the Wikipedia collection - adding small context windows is beneficial for sentence retrieval if weighted appropriately.

6. FUTURE WORK AND CONCLUSIONS

In this paper we presented the novel task of finding *support* sentences which explain the relationship between a query q and an entity e. We developed a framework which consisted of a definition, formalization and a thorough evaluation dataset for the problem at hand. For tackling the problem we developed several features embracing different paradigms (entity score-based, position-based, retrieval-based).

We show that the most interesting feature is the context of a sentence which can be effectively exploited using the BM25F ranking algorithm. Other entity-score derived features show promise but further research is required to capitalize them. Furthermore, we experimented with TREC Novelty collections and found out that weighting with BM25F can improve on best published results for high precision metrics.

Table 3: Results on TREC Novelty 2002, 2003, 2004 sentence retrieval tasks. Parameter values are obtained using two-fold cross validation † and the whole test set *.

	P@10	MAP	MRR	F
TREC Novelty 2002				
TF-IDF	0.19	0.12	0.44	0.19
BM25*	*0.19*	*0.12*	*0.45*	*0.20*
BM25F*	*0.25*	*0.15*	*0.49*	*0.12*
BM25†	0.18	0.12	0.39	0.20
BM25F†	0.22	0.14	0.44	0.12
TREC Novelty 2003				
TF-IDF	0.71	0.40	0.84	0.53
BM25*	*0.72*	*0.40*	*0.86*	*0.54*
BM25F*	*0.79*	*0.57*	*0.90*	*0.54*
BM25†	0.69	0.38	0.84	0.54
BM25F†	0.76	0.53	0.87	0.54
TREC Novelty 2004				
TF-IDF	0.46	0.27	0.65	0.37
BM25*	*0.47*	*0.27*	*0.73*	*0.38*
BM25F*	*0.51*	*0.35*	*0.74*	*0.34*
BM25†	0.46	0.27	0.67	0.38
BM25F†	0.48	0.34	0.68	0.34

Some possible extensions to the ranking formulae presented in section 3 could take into account some additional features, mostly related to sentence normalization; we found out that the methods might have a bias for longer sentences, similarly to standard document retrieval. For instance, the sum-based entity ranking measures will score higher if there are many entities inside a sentence, which may be not necessarily more relevant than another that have just a few very highly-ranked entities; averaging the scores for different entities in a sentence might suffer for very entity-crowded sentences, etc. Consequently, these models can be normalized with respect to document length and number of entities using traditional document length normalization functions, for instance BM25's term frequency normalization factor [19], using sentence length, number of entities in a sentence or a weighted mixture of both.

We are interested in pursuing other linguistic features of sentences in the future. For example, it is likely to be important to detect matches in lists or long coordinations, since these are likely to be less relevant. Also, finding the entity and the query matches in a subject/object relation is likely to be relevant. The notion of what determines a proper context for a given candidate support sentence is also subject to variability and tuning.

Finally, if more sophisticated definitions of context are taken into account, or any other features (linguistic, statistic) are to be incorporated into the ranking models, it could be necessary to devise more fine-grained parametrization and tuning.

Acknowledgments: This work is partially funded by FP7 EU Project LivingKnowledge (ICT-231126).

7. REFERENCES

[1] J. Atserias, H. Zaragoza, M. Ciaramita, and G. Attardi. Semantically annotated snapshot of the english wikipedia. In *LREC'08*, 2008.

[2] P. Bailey, A. P. de Vries, N. Craswell, and I. Soboroff. Overview of the trec 2007 enterprise track. In *Proceedings of TREC 2007 the 16th Text REtrieval Conference*, 2007.

[3] C. Cardie, V. Ng, D. Pierce, and C. Buckley. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering system. pages 180–187, 2000.

[4] S. Chakrabarti, K. Puniyani, and S. Das. Optimizing scoring functions and indexes for proximity search in type-annotated corpora. In *WWW '06*, pages 717–726, New York, NY, USA, 2006. ACM Press.

[5] D. Gianluca, A. P. de Vries, T. Iofciu, and J. Zhu. Overview of the inex 2008 entity ranking track. In *7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008 Dagstuhl Castle, Germany*, 2008.

[6] D. Harman. Overview of the trec 2002 novelty track. In *Proceedings of TREC 2002, the 11th text retrieval conference*, 2002.

[7] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[8] T. Kanungo and D. Orr. Predicting the readability of short web summaries. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 202–211. 2009, ACM.

[9] X. Li and W. B. Croft. Improving novelty detection for general topics using sentence level information patterns. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 238–247, New York, NY, USA, 2006, ACM .

[10] D. Losada and R. T. Fernández. Highly frequent terms and sentence retrieval. In *Proceedings of 14th String Processing and Information Retrieval Symposium, SPIRE'07*, pages 217–228, Santiago de Chile, October 2007.

[11] S. Malik, A. Trotman, M. Lalmas, and N. Fuhr. Overview of inex 2006. In *5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany*, pages 1–11, 2006.

[12] F. McSherry and M. Najork. Computing information retrieval performance measures efficiently in the presence of tied scores. In *ECIR 2008, Proceedings of the 30th European Conference on IR Research*, pages 414–421, Glasgow, Scotland, 2008, Springer.

[13] V. Murdock. *Exploring Sentence Retrieval*. VDM Verlag Dr. Mueller e.K., 2008.

[14] V. Murdock and W. B. Croft. A translation model for sentence retrieval. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 684–691, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[15] T. Okamoto, T. Honda, and K. Eguchi. Locally contextualized smoothing of language models for sentiment sentence retrieval. In *TSA '09: Proceeding of the 1st workshop on Topic-sentiment analysis for mass opinion*, pages 73–80, New York, NY, USA, 2009, ACM.

[16] S. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[17] S. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond, foundations and trends in information retrieval. Volume 3, pages 333–389, 2009.

[18] H. Rode. From document to entity retrieval: Improving precision and performance of focused text search. PhD thesis, University of Twente, CTIT. 2008.

[19] S. Robertson and S. Walker. Some simple effective approximations to the 2 poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. ACM/Springer Verlag.

[20] I. Soboroff. Overview of the trec 2004 novelty track. In *Proceedings of TREC 2004, the 13th text retrieval conference*, 2004.

[21] I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the trec 2005 enterprise track. In *Proceedings of TREC 2005 the 14th Text REtrieval Conference*, 2005.

[22] I. Soboroff, A. P. de Vries, and N. Craswell. Overview of the trec 2006 enterprise track. In *Proceedings of TREC 2006 the Fifteenth Text REtrieval Conference*, 2006.

[23] I. Soboroff and D. Harman. Overview of the trec 2003 novelty track. In *Proceedings of TREC 2003, the 12th text retrieval conference*, 2003.

[24] N. Stokes and J. Carthy. First story detection using a composite document representation. In *Proceedings of HTL01, the Human Language Technology Conference, San Diego, USA*, 2001.

[25] S. Sweeney, F. Crestani, and D. Losada. Show me more: incremental length summarisation using novelty detection. *Information Processing and Management*, 44(2):663–686, 2008.

[26] D. Vallet and H. Zaragoza. Inferring the most important types of a query: a semantic approach. In *SIGIR '08*, pages 857–858, New York, NY, USA, 2008. ACM.

[27] E. M. Voorhees and D. K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, 2005.

[28] A. P. Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas. Overview of the inex 2007 entity ranking track. In *6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007 Dagstuhl Castle, Germany*, pages 245–251. Springer-Verlag, 2008.

[29] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on wikipedia. In *CIKM '07*. ACM Press, 2007.