Voting for Related Entities

Rodrygo L.T. Santos
rodrygo@dcs.gla.ac.uk

Craig Macdonald
craigm@dcs.gla.ac.uk

Iadh Ounis
ounis@dcs.gla.ac.uk

Department of Computing Science
University of Glasgow
G12 8QQ Glasgow, UK

ABSTRACT

Entity search is an emerging research topic in Information Retrieval, where the goal is to rank not documents, but *entities* in response to a given query. A particularly challenging example of this search scenario is when a user's underlying information need is for a list of entities *related* to a given entity, represented in the query. In this paper, we propose to tackle this problem as a voting process, by considering the occurrence of an entity among the top ranked documents for a given query as a vote for the existence of a relationship between this and the entity in the query. Our proposed approach is evaluated using a large Web test collection, in the context of the TREC 2009 Entity track. The results attest the effectiveness of our approach when compared to the top participants at TREC, with unparalleled gains in terms of recall. Moreover, through a comprehensive failure analysis, we uncover important issues to be considered when tackling this new search scenario and draw valuable insights towards achieving an effective related entity search performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*

General Terms

Algorithms, Experimentation

1. INTRODUCTION

Many user queries would be better answered by a ranking of entities rather than a ranking of documents [4]. For example, a user typing in the query '*tennis players*' might be interested in an actual list of athletes instead of any information about the sport. A particularly challenging entity search scenario is when the user is looking for entities related to another entity. For instance, instead of any tennis player, the user might be interested in '*tennis players who have won Wimbledon.*' In this *related entity* search task, the retrieved entities are required to be of a particular type (e.g., people) and to have a specific relationship (e.g., have won) to a given entity (e.g., the Wimbledon championship).

For such *typed* queries, traditional adhoc retrieval techniques may not be the most appropriate, and more refined approaches become necessary. At the very least, the retrieval system should recognise entity occurrences among its indexed documents, and should then select the recognised entities that better answer the initial query. Matters are further complicated when the global Web is considered as the corpus to search for entities. For instance, Web documents lack the structure of document collections traditionally used for research on entity search [8], besides being often adversely affected by spam. Moreover, entities in a Web setting are not always unambiguously defined, nor are they referred to in a standardised format, but instead with possibly many variations expressed in natural language.

In this paper, we propose to tackle the related entity search task in this 'wilder' scenario as a voting process [11]. In doing so, we consider occurrences of an entity in documents retrieved for a query as votes for the existence of a relationship between this entity and the one in the query. Our approach draws a connection to the expert search task, where candidate experts are ranked based on their estimated expertise to the topic of the query. In particular, we generalise a state-of-the-art expert search model in order to search for related entities of multiple types (i.e., not only people) in a large Web corpus. Our approach is evaluated in the context of the TREC 2009 Entity track. The results attest the effectiveness of casting related entity search as a voting process, with an attained performance that compares favourably to that of the best systems at TREC. In addition, we conduct a thorough analysis of the performance of our approach and uncover important issues to be considered in order to deliver an effective related entity search system.

The major contributions of this paper are three-fold:

1. We propose to cast the related entity search task as a voting process, by generalising a state-of-the-art expert search model to this new search scenario.
2. We thoroughly validate our proposed approach within the standard experimentation paradigm provided by the TREC 2009 Entity track.
3. Through a comprehensive failure analysis, we draw valuable insights on the effectiveness of our approach and propose directions for further improvements.

The remainder of this paper is organised as follows. Section 2 overviews previous research on entity search, primar-

RIAO '10, 2010, Paris, France.

ily in the context of information retrieval evaluation forums, and on the related task of expert search. Section 3 describes our approach to related entity search, based on the notion of entity profiles. The construction of such profiles is detailed in Section 4. Sections 5 and 6 describe the experimental setup and the evaluation of our approach, while Section 7 provides a comprehensive failure analysis. Finally, Section 8 presents our conclusions and directions for future work.

2. RELATED WORK

Entity search is an emerging area of research, which has been recently encouraged by forums such as the INitiative for the Evaluation of XML Retrieval (INEX) and the Text REtrieval Conference (TREC). In particular, INEX 2007 introduced the XML Entity Ranking track [7]. Since then, two main tasks have been investigated, in which participants were asked to retrieve a list of entities in response to a query, expressed in natural language, given either the type of the entities to be retrieved or a few example entities of this type [7, 8]. Effective approaches to these tasks typically make use of the structure of the Wikipedia XML collection used in these tasks, including the categories assigned to individual entities, as well as their connections to other entities in the hyperlink graph underlying the collection [15, 19, 21].

In 2009, TREC introduced the Entity track to stimulate research on entity search on the Web, initially focusing on a *related entity search task* [4]. The goal of this task is to retrieve a list of entities of a target type, which should be related to a given entity, according to a predetermined criterion, as specified in the query. Differently from its INEX counterparts, this task is based on a large Web corpus, which poses several additional challenges, not only for entity search systems, but also for their evaluation. In fact, as this task does not require entities to have a Wikipedia entry, virtually every entity with a uniquely identifiable homepage in the collection might be of interest. Successful approaches to this task explored the occurrence of named entities among the documents retrieved for the initial query or those in the neighbourhood of the query entity in the hyperlink structure underlying the collection. For instance, the title and anchor text of retrieved documents [22], or those linked to from the homepage of the query entity [18], or even documents highly ranked by a commercial search engine [9, 20] were used as sources for finding entities related to the one in the query.

A closely related task to entity search is expert search [3], in which the goal is to retrieve people (e.g., enterprise employees) with relevant expertise for a given query. In fact, this task can be regarded as a special case of entity search, in which the entities of interest are of a single type (i.e., people), and the topic of the query represents an underlying information need for the desired expertise [4]. Inspired by this connection, in this work, we extend a state-of-the-art approach to expert search for the task of retrieving entities of multiple types (i.e., not only people). The Voting Model [11] was initially proposed to rank people in response to an expertise search request, by representing candidate experts as aggregates of their associated documents in the target corpus. Differently from most of the approaches deployed at the TREC 2009 Entity track, which tackled the related entity search task with specially devised heuristics [4], we generalise previous research on expert search in order to deliver a more principled approach to searching for related entities, as described in the next section.

3. RELATED ENTITY SEARCH AS A VOTING PROCESS

In this section, we describe our novel approach to related entity search, which aims to aggregate evidence from an initial ranking of documents as votes for ranking entities with respect to the query entity. In particular, we propose a generalisation of the Voting Model [11], in order to account for entities of multiple types (i.e., not only people). Figure 1 illustrates our extended model for related entity search.

Initially, we are given a query Q, which describes the relationship of interest (e.g., have-won) between a given entity (e.g., Wimbledon), and the entities to be retrieved, which should be of a particular type (e.g., tennis players). In order to search for related entities of the type of interest, we start by searching for documents that mention the query entity and the desired relationship, by using any document retrieval approach. The ranking of documents produced for the query Q is then converted into a ranking of the entities that occur in these documents, by aggregating such occurrences as votes for the relevance of each entity. Finally, in order to ensure that the retrieved entities comply with the type of interest for the query, we introduce a new type detection component in the retrieval flow. The remainder of this section describes our choices for each of these steps. The profiling of entities, which associates entities with the documents where they occur, is detailed in Section 4.

3.1 Ranking Entities

The Voting Model defines many *voting techniques*, which convert a ranking of documents into a ranking of aggregates, given a set of *profiles*. The profile of an aggregate or, in our case, an entity, comprises all the documents associated to this entity. In our extension, documents retrieved for a given query that belong to the profile of an entity are considered as votes for the entity to be relevant to the query. In this paper, we employ two of the Voting Model's most effective voting techniques, namely, expCombSUM and expCombMNZ [12].

The expCombSUM voting technique takes into account the scores of the documents in the profile of an aggregate, which are also retrieved for the query Q, transformed by an exponential function. Applying the exponential function has two effects: it removes the logarithm present in many document weighting models and, in doing so, it places more emphasis on highly scored documents. It is defined as:

$$score_{expCombSUM}(e,Q) = \sum_{d \in R(Q) \cap profile(e)} \exp(score(d,Q)) \quad (1)$$

where $R(Q)$ corresponds to the set of documents retrieved for the query Q, $profile(e)$ corresponds to the profile of the entity e, and $score(d,Q)$ corresponds to the score of document d with respect to the query Q.

The expCombMNZ voting technique is similar to expCombSUM. However, it also considers the number of voting documents associated to an aggregate. It is defined as:

$$score_{expCombMNZ}(e,Q) = |R(Q) \cap profile(e)| \times \sum_{d \in R(Q) \cap profile(e)} \exp(score(d,Q)) \quad (2)$$

where $R(Q)$, $profile(e)$, and $score(d,Q)$ are as defined for the expCombSUM voting technique.

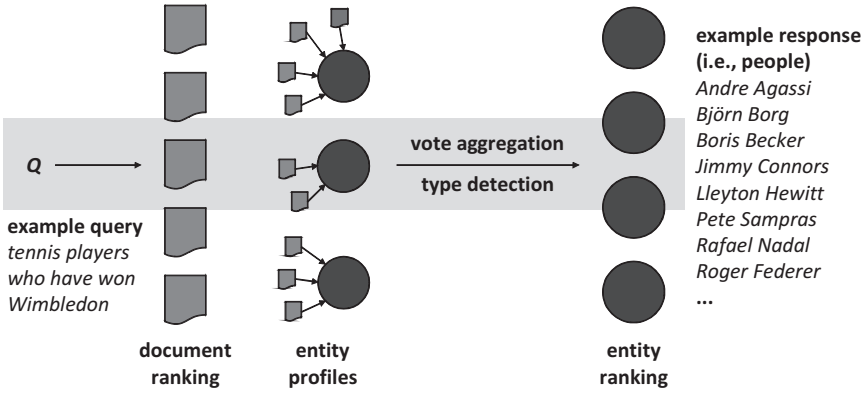

Figure 1: Retrieval flow within our proposed approach.

3.2 Detecting Entity Types

Our extension to the Voting Model allows for entities of multiple types to be handled in the retrieval flow. As different queries usually require entities of a single target type, as exemplified in Figure 1, we must be able to classify each retrieved entity accordingly. In this work, we assume that a short descriptive summary is available for every retrieved entity. This could be, for instance, a collection of snippets extracted from the documents where the entity occurs, or from its identified homepage in the collection, or even from an external resource, such as the entity's Wikipedia page.

To enable this, we introduce a type detection component for the Voting Model, which estimates the likelihood of a retrieved entity given the target type of the query and the entity's descriptive summary. In particular, instead of classifying entities into a broad target type (e.g., people), we make use of the more refined type information present in the query (e.g., tennis players). Accordingly, we define the t_{match} type detection mechanism as follows:

$$t_{match}(e, Q) = score(e_{desc}, Q) \tag{3}$$

where e_{desc} represents the descriptive summary associated to the entity e, and Q is the initial query. In this paper, we make use of the categories associated to DBPedia[1] entities as these entities' descriptive summary. In order to apply the t_{match} type detection mechanism, we integrate its score with that given by a voting technique as a simple product:

$$score(e, Q) = t_{match}(e, Q) \times score_{VM}(e, Q) \tag{4}$$

where $score_{VM}(e, Q)$ is given by a voting technique, such as those in Equations (1) or (2).

In the next section, we describe the construction of entity profiles, which support the deployment of our proposed approach to related entity search.

4. ENTITY PROFILING

Our proposed approach to related entity search is based on the concept of *entity profiles*, which comprise the documents in which entities occur. Quality profiles have been shown to play a critical role in expert search performance [11]. To produce quality entity profiles, with a comprehensive coverage of the entities on the Web, we resort to DBPedia, a structured version of Wikipedia, as a source of entity names. DBPedia provides several datasets covering most entries on Wikipedia, including information extracted from each entry's Wikipedia page, such as its title, abstract, categories, official (non-Wikipedia) homepages, incoming and outgoing hyperlinks, redirection pages, etc. In our approach, we consider every unique DBPedia entry as a candidate entity. Additionally, as not every entity of interest on the Web has a Wikipedia page—which is particularly true for people—we complement the entities gathered from DBPedia with proper names obtained from the 1990 US Census.[2] In particular, we generate complementary proper names by combining the 5k most common first and 88k most common last names in the US. We then estimate the probability of each generated combination, based on the individual probabilities of first and last names occurring in the population (assuming first and last names to be independent). Finally, names already obtained from DBPedia or those with an estimated probability lower than 0.001 are discarded.

4.1 Recognising and Filtering Entities

To produce an initial entity profile, we employ an effective and efficient dictionary-based named entity recognition approach. In particular, we build a large dictionary of all entities extracted from DBPedia and the US Census data. For DBPedia entities, their entries in the dictionary are mapped to both their official name (as given by the corresponding Wikipedia title) and their possible aliases (as given by the title of the Wikipedia pages that redirect to each entity's). By doing so, a given entity can be identified by different patterns (e.g, documents mentioning 'Barack Obama' or '44th President of the United States' should all be associated to the unique entity 'Barack Obama'). Using this large dictionary as input, we apply the Aho-Corasick algorithm [1] to identify occurrences of dictionary entities in the target corpus (e.g., a Web collection). This algorithm initially constructs a suffix tree in linear time with the number of entries in the dictionary (i.e., entity names and aliases). Once this structure is built, the algorithm can be applied linearly with the length of the documents in the corpus and the number of matched dictionary entries in each document.

The entities recognised from the target corpus are further categorised into one of the three target types considered

[1] `http://www.dbpedia.org`

[2] `http://www.census.gov`

target type	clue terms
organisations	agencies, bands, clubs, companies, federations, franchise, governing, bodies, institutions, manufacturers, organisations
people	people
products	albums, awards, books, brands, devices, films, products, singles, software, vehicles

Table 1: Example terms matched against DBPedia category descriptors for entity type detection.

in the TREC 2009 Entity track (as detailed in Section 5): *organisations*, *people*, and *products*.[3] DBPedia entities are classified into one of these types by a simple filtering mechanism, denoted t_{filter}.[4] This mechanism is based on the occurrence of clue words in each entity's associated categories, obtained from DBPedia. It is defined as:

$$t_{filter}(e, Q) = \begin{cases} 1 & \text{if } e_{desc} \cap W_{Q_{type}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where e_{desc} are the categories (from DBPedia) associated to an entity e retrieved for the query Q, and $W_{Q_{type}}$ is a set of clue words associated to the target type of the query.

Table 1 illustrates some of the clue words employed for each of the target types considered in this paper. Interestingly, entities of the type people can be confidently identified with the use of the single clue word 'people'. This is possible since most DBPedia entities of this type belong to categories such as 'living people' or 'people from *some place*'. Although reasonably simple and aimed at providing only a coarse classification of the retrieved entities into a broad type, this mechanism provides an initial baseline for the more sophisticated t_{match} type detection mechanism, introduced in Section 3.2. Additionally, it can be computed offline, during the construction of entity profiles, which also helps filtering out entities that are not of interest (e.g., location names) at an earlier stage.

Table 2 provides statistics for the entities recognised from the 50-million documents Web collection used in our experiments (see Section 5), classified by the t_{filter} mechanism into one of the aforementioned target types. As shown in the table, entities of the type people account for the majority of the considered entities, with roughly ten times the number of unique organisations or products. As evidenced by further breaking down the statistics for people's names, this is mainly due to the addition of non-DBPedia entities, derived from the US Census data. Looking at the size of the profiles constructed for the recognised entities, we can observe a highly skewed distribution, with the presence of a few very common entity names, covering more than 6.2 million documents (12% of the whole collection). On average, organisations and products have roughly the same profile size, which is much larger than the average profile size of people. As expected, DBPedia people have larger average profiles when compared to the intuitively less famous people derived from the US Census.

[3]Although our description focuses on the types defined by the TREC 2009 Entity track, it is worth noting that our approach is general and may be applied for other entity types.
[4]By definition, all entities obtained from the US Census data are classified in the people target type.

target type	#entities	profile size		
		mean	std.dev.	max
organisations	184,545	1,299	22,784	5,931,015
people	1,763,095	107	1,702	749,535
(Census)	1,339,384	35	564	382,509
(DBPedia)	425,877	338	3,313	749,535
products	179,630	1,265	23,383	6,207,142
all types	2,126,930	306	9,641	6,207,142

Table 2: Statistics of the entity profiles derived from the ClueWeb09 corpus, broken down by target type.

target type	page	#entities	profile size		
			mean	std.dev.	max
organisations	home	19,389	1	1	15
	link	45,779	2	2	117
people	home	6,603	1	1	6
	link	44,905	2	1	71
products	home	7,544	1	1	6
	link	24,796	2	1	23

Table 3: Statistics of the homepages derived from DBPedia, broken down by target and page type.

4.2 Finding Entity Homepages

As an alternative to representing the retrieved entities by their name, we investigate a simple mechanism for identifying each retrieved entity's homepage. This provides a further dimension for evaluating the effectiveness of our approach. In particular, differently from entity names, official homepages can be usually unambiguously associated to the entities they represent. Moreover, we treat homepage finding as a post-process with respect to the actual related entity search task. In other words, we first identify entities related to the input entity, and only then determine the correct homepages for each retrieved entity. For such, we propose the following homepage finding strategy. Given a retrieved entity, we choose as its candidate homepages documents in the following order of precedence:

1. the entity's official homepage;
2. the pages linked to from the entity's Wikipedia page;
3. the top ranked documents from the entity profile.

The official homepages and external Wikipedia links for a given entity can be easily obtained from DBPedia, and are very likely to be relevant, as they are usually curated by enthusiastic Wikipedia contributors. In turn, the top ranked documents from the target collection work as a fallback plan for those entities without a DBPedia entry (e.g., entities derived from the US Census data), or whose DBPedia pages are not present in the target corpus. As observed in the statistics shown in Table 3, only a small fraction of the considered entities have a corresponding homepage or external link on DBPedia, which could also be found in the target corpus. In particular, organisations are more likely to have a homepage listed on DBPedia when compared to people and products. As for external Wikipedia links, roughly the same number of organisations and people are covered, which is substantially larger than the number of products. As expected, entities have usually a very few associated homepages and Wikipedia links in the target corpus, although outliers with many of such pages do exist.

5. EXPERIMENTAL SETUP

In the remainder of this paper, we aim to answer the following three research questions:

1. Is our proposed voting approach effective at finding related entities with respect to a given entity?
2. Is a fine-grained type detection approach more effective than a simple type filtering mechanism?
3. Which other aspects have an important impact on our related entity search performance?

The first two research questions are addressed in Section 6, while Section 7 addresses the last question. In the remainder of this section, we describe the experimental setup aimed at supporting these investigations.

5.1 Collection and Topics

Our experiments are conducted in the context of the related entity search task of the TREC 2009 Entity track. The goal of this task is to produce a ranking of entities of a given type, which must also be related to an input entity, as specified by a test topic. In particular, a total of 20 topics were produced for this task. Each topic comprises 5 fields, as exemplified in Figure 2: the topic identifier, the input entity name, its primary homepage in the collection, the target type of the entities to be retrieved, and a narrative describing the relationship of interest between each retrieved entity and the input entity.

```
<query>
  <num>20</num>
  <entity_name>Isle of Islay</entity_name>
  <entity_URL>clueweb09-en0008-96-25389</entity_URL>
  <target_entity>organization</target_entity>
  <narrative>
    Scotch whisky distilleries on the island of Islay.
  </narrative>
</query>
```

Figure 2: TREC 2009 Entity track, topic 20.

Each retrieved entity is represented by an optional name and Wikipedia page, and at least one (non-Wikipedia) homepage, which serves as a unique identifier for the entity.

The document collection used in the related entity search task of the TREC 2009 Entity track is a subset of the new ClueWeb09 dataset,[5] which comprises 50 million English Web documents—in contrast, the largest test collection used for the related task of expert search had only 300k documents [3]. We index this collection using Terrier[6] [14], with Porter's stemmer and standard English stopwords removal.

5.2 Evaluation Procedure and Metrics

Relevance judgements in the TREC 2009 Entity track were performed in two stages. In the first stage, assessors judged the homepages retrieved for each entity, pooled to a depth 10. In a second stage, names were judged relevant if they were retrieved in the same record as a judged relevant homepage [4]. We argue that this procedure may not be the most appropriate, as it relegates the main goal of the task (i.e., to retrieve named entities) to a secondary status. In order to draw a better understanding of the performance of our approach in these two distinct subtasks (i.e., related entity search and homepage finding), we evaluate it according to three complementary dimensions. These dimensions measure the ability of our approach to retrieve:

(1) the correct entity names,

(2) the correct homepages regardless of the name,[7]

(3) the correct homepages for the correct name.

While the retrieved names were judged in a binary fashion, the assessment of the retrieved homepages was carried out based on a three-point relevance scale:

(0) *non-relevant*,

(1) *relevant*, but not the entity's official homepage,

(2) *primary*, i.e., the entity's official homepage.

target type	#queries	#names	#homepages	
			relevant	primary
organisations	11	209	504	113
people	6	107	288	29
products	3	31	117	25
all types	20	347	909	167

Table 4: Breakdown of relevance assessments for the TREC 2009 Entity track.

Table 4 shows a breakdown of the relevance assessments for this task, according to the different target types and evaluation dimensions considered. The number of judged names and relevant homepages for each target type is roughly proportional to the number of queries requesting entities of that type. Nonetheless, there is a distinctively higher number of judged primary homepages for organisations when compared to the other types, in accordance with the statistics shown in Table 3 for homepages derived from DBPedia.

Our results are reported mainly in terms of the official metrics used in the TREC 2009 Entity track: nDCG@R, the normalised discounted cumulative gain at rank R (the number of judged primary and relevant homepages for each topic), and P@10, the fraction of retrieved entities with a primary homepage among the top 10 retrieved entities. In particular, we report P@10 values for all three aforementioned evaluation dimensions. For the second and third dimensions, which involve the evaluation of our homepage finding performance, we use the subscripts R and P to denote relevant and primary homepages, respectively.

5.3 Parameter Settings

Since our experimental setup comprises a small number of test queries, we do not conduct any training in order to set the only parameter of our proposed approach, namely, the number of documents to be retrieved for the initial query. Instead, we use the recommended setting of 1000 documents, which was suggested in [11] after extensive experiments. In particular, we experiment with two document weighting models, namely, the DPH Divergence From Randomness model [2], and BM25 [17]. DPH is a parameter-free model.

[5] http://boston.lti.cs.cmu.edu/Data/clueweb09/

[6] http://www.terrier.org

[7] Note that this dimension corresponds exactly to the official procedure adopted in the TREC 2009 Entity track.

retrieval techniques			evaluation dimensions							
document ranking	entity ranking	type detection	name	homepage					name+homepage	
			P@10	R	nDCG@R	P@10$_R$	P	P@10$_P$	P@10$_R$	P@10$_P$
BM25	expCombMNZ	t_{filter}	0.2650	326	0.2611	0.4400	74	0.1500	0.2400	0.1150
BM25	expCombMNZ	t_{match}	0.2900	320	0.2645	**0.4500**	**78**	**0.1600**	**0.2600**	0.1250
BM25	expCombSUM	t_{filter}	0.2900	**327**	**0.2665**	0.4450	75	**0.1600**	0.2500	0.1250
BM25	expCombSUM	t_{match}	**0.2950**	323	0.2661	0.4300	**78**	**0.1600**	0.2550	**0.1350**
DPH	expCombMNZ	t_{filter}	0.2350	**344**	0.2510	0.4400	75	0.1050	0.2150	0.0900
DPH	expCombMNZ	t_{match}	**0.2850**	341	0.2518	**0.4550**	74	**0.1250**	**0.2500**	**0.1100**
DPH	expCombSUM	t_{filter}	0.2550	343	0.2483	0.4400	75	0.1100	0.2250	0.0900
DPH	expCombSUM	t_{match}	0.2750	343	**0.2533**	**0.4550**	**77**	**0.1250**	0.2450	0.1050
TREC 2009 Entity track median			–	–	0.0751	–	–	0.0050	–	–

Table 5: Performance of different retrieval techniques with respect to three different evaluation dimensions: names, homepages, and pairs name+homepage. The best value in each column is highlighted in bold.

	retrieval techniques			evaluation dimensions							
target type	document ranking	entity ranking	type detection	name	homepage					name+homepage	
				P@10	R	nDCG@R	P@10$_R$	P	P@10$_P$	P@10$_R$	P@10$_P$
organisations	BM25	expCombSUM	t_{match}	**0.3000**	137	**0.2305**	0.4000	**54**	**0.1818**	**0.2364**	**0.1727**
	DPH	expCombMNZ	t_{match}	0.2909	**153**	0.2186	**0.4091**	50	0.1364	0.2273	0.1364
people	BM25	expCombSUM	t_{match}	**0.4333**	**139**	**0.3455**	**0.5167**	**15**	**0.1333**	**0.4167**	**0.1333**
	DPH	expCombMNZ	t_{match}	0.4167	135	0.3177	**0.5167**	**15**	**0.1333**	**0.4167**	0.1167
products	BM25	expCombSUM	t_{match}	0.0000	47	0.2375	0.4000	**9**	**0.1333**	0.0000	0.0000
	DPH	expCombMNZ	t_{match}	0.0000	**53**	**0.2418**	**0.5000**	**9**	0.0667	0.0000	0.0000

Table 6: Per-target type performance breakdown of selected retrieval techniques from Table 5 with respect to the aforementioned evaluation dimensions. The best value in each column is highlighted in bold.

In other words, it requires no parameter tuning, as all variables in its formula can be directly obtained from the query or the collection statistics. As for BM25, we use the often suggested parameter settings of $k_1 = 1.2$ and $b = 0.75$ [16].

6. EXPERIMENTAL VALIDATION

In this section, we thoroughly validate our proposed approach to related entity search in the context of the TREC 2009 Entity track. We begin by investigating our first two research questions, concerning the effectiveness of our approach as a whole and that of its fine-tuned type detection mechanism. Table 5 shows the results of the evaluation of our approach under different settings, and according to different evaluation dimensions. In particular, we consider two different approaches for each of the document ranking (BM25 and DPH), entity ranking (expCombSUM and expCombMNZ), and type detection (t_{filter} and t_{match}) components. As detailed in the previous section, the evaluation is conducted with respect to the performance of our approach in finding the correct names, the correct homepages, and the correct homepages for each of the correct names.

From Table 5, we observe that our approach markedly outperforms the median of the TREC 2009 Entity track participants in all settings. In fact, our approach using BM25 would have ranked second among the submitted TREC systems in terms of nDCG@R and third in terms of P@10$_P$, the official metrics considered in this task [4]. As for recall, our approach outperforms the top performing system at TREC by a very large margin (159% for relevant homepages and 29% for primary homepages). This is remarkable, particularly if we consider that the best performing TREC system relied on a commercial search engine for retrieving an initial set of candidate entities [9]. In contrast, our approach is entirely based on standard document retrieval approaches with no further parameter tuning whatsoever. Additionally, our performance in terms of finding the correct names is markedly higher than that of finding primary homepages, as well as the correct name+homepage pairs. This suggests that our approach is effective at ranking relevant entities, although a more refined homepage finding approach would be beneficial. Nevertheless, the results attest the effectiveness of tackling the related entity search task as a voting process, and answer our first research question.

As for the deployed retrieval techniques, BM25 performs generally better than DPH, particularly on the early precision measures based on names, homepages, and the combination name+homepage. As for the entity ranking component, there is no clear difference between the performance of expCombSUM and expCombMNZ. Finally, we observe that the t_{match} type detection mechanism brings further improvements in almost all cases when compared to the simple t_{filter} mechanism, which answers our second research question.

Finally, Table 6 shows the performance of our best settings from Table 5 for each of the considered document retrieval approaches (i.e., BM25+expCombSUM+t_{match} and DPH+expCombMNZ+t_{match}), however broken down based on the three target types used in our evaluation. From Table 6, we first note that the highest performance for both settings is obtained for the people type, followed by the organisations type. In particular, our best performance is attained at ranking relevant pages for people, and primary pages for organisations. Of some concern, however, is the fact that our approach could not retrieve relevant product names early in the ranking, even though relevant and primary product homepages could be retrieved. A further investigation is conducted in the next section. In particular, after validating our approach under the standard experimentation paradigm provided by the TREC 2009 Entity track, in Section 7, we provide a comprehensive failure analysis.

7. FAILURE ANALYSIS

Given the small number of available test queries in the first edition of the TREC Entity track, we conduct a failure analysis of our approach in an attempt to derive a bet-

target type	#queries	name						name+homepage			
		P@20	e-1	e-2	e-3	e-4	e-5	$P@20_R$	h-1	h-2	h-3
organisations	11	0.2182	0.27	**7.27**	5.09	2.09	0.00	0.1591	0.00	**1.18**	0.73
people	6	**0.2750**	0.00	**9.83**	1.67	0.67	0.00	**0.2667**	**0.50**	**0.50**	0.33
products	3	0.0000	0.67	2.67	5.33	**11.33**	0.00	0.0000	0.00	0.00	0.00
all types	20	0.2025	0.25	**7.35**	4.10	3.05	0.00	0.1675	0.15	**0.80**	0.50

Table 7: Per-target type average error incidence rates. The highest value in each row is highlighted in bold.

ter understanding of its performance [6]. By answering our third research question, we uncover possible limitations of the currently deployed retrieval techniques within our approach. Additionally, we draw insights towards improving the performance of these techniques and, consequently, of our approach as a whole.

7.1 Error Classes

In order to carry out a failure analysis of our approach, we define two broad classes of errors, aimed at explaining the retrieval of irrelevant entities or irrelevant homepages for relevant entities, respectively.

7.1.1 Related Entity Search

This class encompasses errors related to the task of finding the correct entity names for a given query. In particular, each query is manually assessed up to depth 20, with errors categorised into one of the following subclasses:

e-1. *not an entity:* the retrieved entity corresponds to a broad concept rather than a named entity (e.g., 'Corporation', 'African people');

e-2. *semantically close entity:* the retrieved entity is of a very similar type as relevant entities (e.g., a computer science researcher, when winners of a particular computer science award are requested);

e-3. *semantically distant entity:* the retrieved entity is of a less similar type as relevant entities (e.g., a mobile phone manufacturer, when a carrier is requested);

e-4. *'stopword' entity:* an entity completely unrelated to relevant entities, which was only retrieved because it has a frequent name (e.g., 'Yahoo', 'Facebook', 'Microsoft') or alias (e.g., 'GPS', 'PRE', and 'CPR' are aliases for the organisations 'GPS (band)', 'PRE (band)', and 'Communist Party Reconstructed', respectively);

e-5. *misclassified entity:* the retrieved entity is not of the requested target type.

7.1.2 Homepage Finding

This class includes errors with respect to ranking the appropriate homepages given that a relevant entity was found. Analogously to the entity search analysis, errors are further categorised into the following subclasses:

h-1. *homepage not in DBPedia:* the retrieved entity has no associated page in DBPedia;

h-2. *homepage not in ClueWeb09 B:* the retrieved entity has associated pages in DBPedia, but none of these is included in the subset B of the ClueWeb09 corpus;

h-3. *homepage not relevant:* the retrieved entity has associated pages in DBPedia and the target corpus, but these pages were judged irrelevant.

7.2 Error Incidence

Based on the previously defined error classes, Table 7 shows the average error incidence rates of our approach under the best performing setting reported in Section 6, BM25+expCombSUM+t_{match}. Its average performance in terms of P@20 for the considered target types is also shown.

From Table 7, we first observe that e-2 (*semantically close entity*) is the most incident error class, when the average over all types is considered, being the most common error also for the organisations and people types. Typical examples of this error come from documents comprising lists of closely related entities, but of which only a subset is relevant to the relationship specified in the query. In such cases, telling apart relevant from irrelevant entities is particularly challenging, as they differ mostly with respect to their relationship to the input entity. For instance, for the query '*students of Claire Cardie*', several entities are promoted based on a list of Claire Cardie's publications, which happen to be her co-authors, but not necessarily her students. Another example is the query '*airlines that Air Canada has code share flights with*', for which several airlines are retrieved, not only the ones with the desired relationship with respect to the query entity. For tackling this error, a promising direction is to refine the treatment of lists, in order to ensure that only entities that have the required relationship to the query entity are retrieved. Possible approaches include looking for evidence of such a relationship beyond documents, e.g., in the hyperlink structure underlying the collection.

The second most common error, particularly common for the organisations target type, is e-3 (*semantically distant entity*). This error is similar to e-2, but with irrelevant entities being of a different type from the specific type of interest. Examples of this error include sports federations being retrieved for the query '*professional sports teams in Philadelphia*'. Although of the same broad type (i.e., organisations), they do not comply with the precise type specification in the query. Another example is the query '*carriers that Blackberry makes phones for*', for which mobile phone manufacturers are retrieved when only carriers are expected. These are likely promoted by online shopping lists, which often mix entities of these two organisation types together. Differently from the e-2 error class, errors in the e-3 class are more likely to be attenuated by a more refined type detection mechanism. This could be based on additional descriptive evidence for every retrieved entity (e.g., summary descriptions derived from the entity's Wikipedia page or based on snippets retrieved by Web search engines), or on more sophisticated topic modelling techniques (e.g., [5]).

As pointed out in the previous section, when the products target type is considered, our approach could not retrieve relevant names among the top results. Upon inspection, we have found that the relevant entities for the corresponding queries are not contemplated in our dictionary, as they do not have a Wikipedia page. Besides this problem, we ob-

serve that queries of the target type products are the most affected by the e-4 error (*'stopword' entities*). A typical example is the query '*CDs released by the King's Singers*', for which somehow related (though irrelevant) entities are retrieved, which have very common names, such as 'Audio', 'Classical', and 'Spanish'. A promising direction for tackling this problem is to account for the size of each entity's profile. This could be done, for instance, by applying a profile length normalisation so as to penalise very common entities [13].

Related entity search error classes with a lower incidence rate include e-1 (*not an entity*), and e-5 (*misclassified entity*). Errors in the e-1 class are due to the misclassification of broad concepts into one of the target types of interest, such as 'Corporation' and 'Inn', which are classified as organisations, or 'Bone', classified as a product. Misclassification of target types (the e-5 error class), however, was not observed, which attests the effectiveness of our simple filtering mechanism for broad type detection.

Finally, of the homepage finding error classes considered, we observe that h-1 (*homepage not in DBPedia*) is more frequent for entities of the type people. Along with organisations, entities of this type suffer from pages not being found in the ClueWeb09 corpus (error h-2). This suggests that a more refined homepage finding approach could be beneficial for when no homepage is available from this rich resource. A related problem is observed for the h-3 error class (*homepage not relevant*). This problem affects both organisations and people, and suggests that even the homepages available from DBPedia might require a proper ranking strategy.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have introduced a novel approach to related entity search, by generalising a state-of-the-art expert search model in order to find entities of multiple types. By aggregating evidence for the existence of a relationship between the retrieved entities and the one expressed in the query, we tackle this new search scenario as a voting process. Experiments in the context of the related entity search task of the TREC 2009 Entity track attest the effectiveness of our approach. Moreover, by refining our proposed type detection mechanism, we have shown that further improvements are attained. Finally, by performing a comprehensive failure analysis of our best performing variant, we have uncovered several important aspects to be considered in order to attain an effective related entity search performance.

As future work, we plan to carry on the improvements suggested by our failure analysis. In particular, different named entity recognition techniques could be investigated, in order to provide our approach with a higher coverage of the entities present in the ClueWeb09 corpus. Another promising area of improvement is towards exploiting entity relationships both at the document level, based on extended bigram models, as well as at the level of the network structure underlying this corpus. In this front, an interesting direction is to analyse this network in order to uncover its structural organisation into communities of related entities [10].

9. REFERENCES

[1] A. V. Aho and M. J. Corasick. Efficient string matching: an aid to bibliographic search. *CACM*, 18(6):333–340, 1975.

[2] G. Amati, E. Ambrosi, M. Bianchi, C. Gaibisso, and G. Gambosi. FUB, IASI-CNR and University of Tor Vergata at TREC 2007 Blog track. In *TREC*, 2007.

[3] P. Bailey, N. Craswell, A. P. de Vries, and I. Soboroff. Overview of the TREC-2007 Enterprise track. In *TREC*, 2007.

[4] K. Balog, A. P. de Vries, P. Serdyukov, P. Thomas, and T. Westerveld. Overview of the TREC 2009 Entity track. In *TREC*, 2009.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

[6] C. Buckley. Why current IR engines fail. In *SIGIR*, pages 584–585, 2004.

[7] A. P. de Vries, A.-M. Vercoustre, J. A. Thom, N. Craswell, and M. Lalmas. Overview of the INEX 2007 Entity Ranking track. In *INEX*, pages 245–251, 2007.

[8] G. Demartini, A. P. de Vries, T. Iofciu, and J. Zhu. Overview of the INEX 2008 Entity Ranking track. In *INEX*, pages 243–252, 2008.

[9] Y. Fang, L. Si, Z. Yu, Y. Xian, and Y. Xu. Entity retrieval with hierarchical relevance model. In *TREC*, 2009.

[10] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[11] C. Macdonald. *The Voting Model for People Search*. PhD thesis, Univ. of Glasgow, 2009.

[12] C. Macdonald and I. Ounis. Voting for candidates: adapting data fusion techniques for an expert search task. In *CIKM*, pages 387–396, 2006.

[13] C. Macdonald and I. Ounis. Searching for expertise: experiments with the Voting Model. *Computer*, 2008.

[14] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: a high performance and scalable information retrieval platform. In *OSIR at SIGIR*, 2006.

[15] J. Pehcevski, A.-M. Vercoustre, and J. A. Thom. Exploiting locality of Wikipedia links in entity ranking. In *ECIR*, pages 258–269, 2008.

[16] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *TREC*, 1995.

[17] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *TREC*, 1992.

[18] P. Serdyukov and A. de Vries. Delft University at the TREC 2009 Entity track: ranking Wikipedia entities. In *TREC*, 2009.

[19] A.-M. Vercoustre, J. A. Thom, and J. Pehcevski. Entity ranking in Wikipedia. In *SAC*, pages 1101–1106, 2008.

[20] Y. Wu and H. Kashioka. NiCT at TREC 2009: employing three models for Entity ranking track. In *TREC*, 2009.

[21] H. Zaragoza, H. Rode, P. Mika, J. Atserias, M. Ciaramita, and G. Attardi. Ranking very many typed entities on Wikipedia. In *CIKM*, pages 1015–1018, 2007.

[22] H. Zhai, X. Cheng, J. Guo, H. Xu, and Y. Liu. A novel framework for related entities finding: ICT-NS at TREC 2009 Entity track. In *TREC*, 2009.