A Search Based Approach to Entity Recognition: Magnetic and IISAS Team at ERD Challenge

Michal Laclavík
Magnetic Media Online
122 West 27th Street, 7th floor
New York, NY 10001
laclavik@magnetic.com

Marek Ciglan
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, Bratislava
marek.ciglan@savba.sk

Alex Dorman
Magnetic Media Online
122 West 27th Street, 7th floor
New York, NY 10001
alex@magnetic.com

Štefan Dlugolinský
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, Bratislava
upsysdlu@savba.sk

Sam Steingold
Magnetic Media Online
122 West 27th Street, 7th floor
New York, NY 10001
sds@magnetic.com

Martin Šeleng
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, Bratislava
martin.seleng@savba.sk

ABSTRACT

ERD 2014 was a research challenge focused on the task of recognition and disambiguation of knowledge base entities in short and long texts. This write-up describes Magnetic-IISAS team's approach to the entity recognition in search queries with which we have participated in ERD 2014 challenge. Our approach combines techniques of information retrieval, gazetteer based annotation and entity link graph analysis to identify and disambiguate candidate entities. We built a search index with multiple structured fields extracted from Wikipedia, Freebase and DBPedia. When processing a query, we first retrieve top matching entities from the index. For all retrieved entities, we gather plausible verbalizations, surface forms, that retrieved entities may be referred to with. We match gathered entity surface forms against the original query to confirm the entity relevance to the query. Finally, we exploit Wikipedia link graph to asses the similarity of candidate entities for the purpose of disambiguation and further candidate filtering. In the paper we discuss successful as well as unsuccessful attempts to improve the quality of system results that we have tried during the course of the challenge.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]; H.3.1 [**Content Analysis and Indexing**]; H.3.3 [**Information Search and Retrieval**]

Keywords

entity search, entity back-mapping, link graph

ERD'14, July 11, 2014, Gold Coast, Queensland, Australia.

ACM 978-1-4503-3023-7/14/07 ...$15.00.
http://dx.doi.org/10.1145/2633211.2634352.

1. INTRODUCTION

Search focused on retrieving entities rather than documents is becoming increasingly popular in the research community as well as in the industry. Several research challenges focusing on linking textual content to knowledge base entities have been organized in past few years. In this paper we describe participation of Magnetic-IISAS team in the ERD 2014[1] challenge [1], focusing on the description of our approach to entity search and recognition. We have participated in the *Short Track* of the challenge, which focused on recognizing mentions of entities in search queries, disambiguating them, and mapping them to the entities in a given knowledge base - a subset of Freebase[2] containing more than 2 million entities. Our system was ranked as the 4$^{\text{th}}$ out of 19, with F1 score of 0.6557[3] on the test data set.

Approach in brief. The main steps of the entity search approach described here are the following:

- We apply typical IR models to Entity Search: Indexing and Full-text search, where entity is modeled as a multi-field document, reusing as much structured data as possible.
- We filter search results by back-mapping possible surface forms of retrieved entities to the original query.
- In addition, we disambiguate and filter search results exploiting Wikipedia link graph.

Team and motivation. Team members come from two organizations: Magnetic (Magnetic Media Online[4]) and IISAS (Institute of Informatics, Slovak academy of Sciences[5]). Magnetic is an online advertising company focusing on search retargeting, where audiences are modeled based on the search queries users conduct on visited websites. Search retargeting focuses on displaying advertisements to users who conducted searches for specific keywords or categories in the past. Magnetic's interest in entity recognition tasks is driven by their need to understand the query intent.
Research group from IISAS focuses on entity search, query understanding and question answering problems by combing

[1] http://web-ngram.research.microsoft.com/erd2014/
[2] http://www.freebase.com/
[3] http://tinyurl.com/ShortTrackERD14
[4] http://www.magnetic.com/
[5] http://ikt.ui.sav.sk/

techniques from information retrieval, semantic web, information extraction and complex networks [3]. IISAS motivation comes also from the VENIS project[6], where we try to enhance the enterprise search with an entity centric approach to retrieval, combining structured (database records) and unstructured data (emails, documents). Participation in the ERD challenge [1] helped us enhance our work related to entity recognition.

Paper overview. The paper is structured as follows: section 2 briefly describes the approach and its main features. Each of the three important features is then described in a subsequent section, followed by an additional section summarizing remaining features influencing results quality. The section 7 provides evaluation on a very small beta TREC dataset, which gives some insight on how different features influence the results. The last section concludes and summarizes the results.

2. OVERVIEW OF THE APPROACH

In this section we describe the main steps of our algorithm on examples provided in the ERD guidelines. Our system was designed for the short track of the ERD challenge, where the task was to annotate short textual queries with the query mentions of entities.

Our solution relies on the information retrieval model to identify candidate entities. We use the original query as a search query against the index of entity documents. The top k retrieved entities are considered as candidates for the query annotation. It is rarely the case that the top scoring entity documents contain only the correct entities suitable for the annotation of the given query. The search results usually contain a mix of target entities and non-target concepts/entities related to the query. To overcome this difficulty, we introduced an additional step to filter non-target entities - for each entity, we construct a set of surface forms (verbalizations, string representations that are likely to be used to denote the target entity in a text) and we match those surface forms to the original query. Only entities with a surface form that can be mapped to the original query are considered in the subsequent processing. We refer to this step as entity surface form back mapping to the query; or "entity back mapping" for short. The final step of the process is the disambiguation and filtering based on similarity of candidate entities. To this end, we exploit the Wikipedia link graph, i.e., the hyperlinks defined between distinct articles - "entity pages". We select the subgraph of the Wikipedia link graph induced by the vertices equivalent to the search result entities.

To summarize, our solution uses three main steps:

- retrieval the form entity documents index (built from the Wikipedia, Freebase and DBPedia data)
- entity back mapping - mapping entity verbalizations to the query
- disambiguation and filtering based on similarity of entities

To briefly illustrate the process, we discuss the following example. Consider the query "total recall arnold schwarzenegger" and search our index for it to get following results (names correspond to Wikipedia page titles, items with Freebase identifiers - starting with `/m/` - are the entities eligible for annotation):

1. Arnold Schwarzenegger filmography
2. **Arnold Schwarzenegger** (`/m/0tc7`)
3. Political career of Arnold Schwarzenegger
4. **Total Recall (2012 film)** (`/m/0gvrws1`)
5. **Total Recall (1990 film)** (`/m/0fd4x`)
6. California gubernatorial recall election
7. List of awards received by Arnold Schwarzenegger
8. Patrick Schwarzenegger (`/m/0gttlcb`)

   ...

Please note that the first search result is quite relevant to query, but it is not a part of entities eligible for the task (only a defined subset of freebase entities were defined as eligible). Only four of the listed results have eligible ERD Freebase IDs. As search results indicate, we search the entire space of Wikipedia concepts rather than the subset of entities eligible for the ERD challenge. This is because additional relevant concepts, such as the first one, have proved to be instrumental for the disambiguation and identification of result similarity.

In the second step, "entity back mapping", we are mapping surface forms of all search results back to the query. The items in bold were successfully mapped back to the query by their alternative names, i.e., surface forms.

In the third step, we examine the entity with the top score, which is successfully mapped to the query, *Arnold Schwarzenegger* in this case. We compute the similarity with the remaining mapped items, the two *Total Recall* movies in this case. The "2012 movie" has a higher search score, but, when computing the link similarity, the "1990 film" has higher similarity with the actor. The first search result, *Arnold Schwarzenegger filmography*, contributes to the link similarity computation, for example, it links to both relevant concepts, "Arnold Schwarzenegger" and the "1990 movie".

We apply additional techniques which are described further in the paper, but these main 3 steps are the core of our algorithm.

3. ENTITY MODELLING

In this section, we describe how we model entities using data from Wikipedia, Freebase and DBPedia.

3.1 Wikipedia concept modeling

Wikipedia is a collaboratively created and maintained ("crowdsourced") compendium of human knowledge. It contains a variety of human-maintained information on many topics. Each Wikipedia page describes a concept or an entity. In addition to the Freebase data, Wikipedia contains a variety of hypertext links among documents as well as anchor texts of links or section headers, which are related to described entities.

The ERD challenge covers a subset of 2 million of Freebase entities, which are contained in Wikipedia as well; we have used all entities of Wikipedia pages, since it helps improve the disambiguation and entity similarity computations.

Each Wikipedia page, in addition to the human-readable text, contains much semi-structured information; e.g., incoming and outgoing links to other concepts, anchor text of links, alternative names (redirect pages), section headers or categories. Similarly to the approach reported in [4], we aim at representing Wikipedia concepts by a richer feature set than a plain textual description. In order to prepare a knowledge base, we have to parse and index Wikipedia, Free-

[6] `http://www.venis-project.eu/`

base and DBPedia. For Wikipedia parsing we have adapted the Wikipedia Miner Toolkit[7].

The following information has been extracted from Freebase, DBPedia and Wikipedia for each Wikipedia concept:

- *title*: title of the Wikipedia concept (article),
- *alt*: alternative names of articles (redirects),
- *anchor*: anchor text of links pointing to an article,
- *text*: text of the article,
- *sentence*: first sentence of article,
- *abstract*: abstract of concept,
- *section*: section headers of article,
- *links*: titles of articles linked from the current page,
- *template*: names of templates used on the article,
- *category*: Wikipedia categories of the article,
- *db_category*: DBPedia categories for article,
- *fb_category*: Freebase categories for article.
- *fb_name*: Freebase title for article.
- *fb_alias*: Freebase alternative names for article.
- *fb_id*: Freebase id for article.
- *fb_erd_id*: Freebase id for articles provided Freebase subset by ERD.
- *alt_disambig*: Names of disambiguation pages linking to this entity.

For parsing of Freebase, Wikipedia and DBPedia, we use several Map-Reduce jobs on Hadoop. The final JSON output, where each entity is modelled by the above fields is indexed by Lucene[8] toolkit. Data related to a single Wikipedia page is stored as a single document in the index. Different types of extracted information (e.g. *title*, *text*, *anchors*) are modeled as distinct fields in the document representing a Wikipedia/Freebase concept. Fields are stored in the index because we are using one set of fields to search in the index, and another set of fields is used to model surface forms for entities.

3.2 Entity Retrieval

Entities were retrieved by querying the Lucene index with the input query. Original query could be modified by spelling correction, as is described in section 6.2. We use only a subset of extracted fields for full-text search, specifically: *title*, *alt*, *anchor*, *text*, *section*, *link* and *category*. The other fields did not contribute to search results quality and were not used in retrieval. However, some of the remaining fields were used either for entity back-mapping (section 4) or for detecting entity categories for better similarity computation (section 5).

For searching, the default Lucene scoring model, TF-IDF based, delivered the best results when compared with other retrieval models available in the Lucene toolkit (with default parameters). However, based on our previous experience [2], we have experimented with combining of several retrieval models results. On the training set of the ERD challenge, we have achieved the highest scores with a combination of default Lucene scoring, BM25 [5] and a language model based on the Jelinek-Mercer smoothing [6]. The combination of different scoring functions was done by averaging normalized scores for distinct retrieved items.

Modeling entities as multi-fields documents for IR models and entity back-mapping helped us retrieve the most relevant entity(s) for a given query in the majority of cases. However, it often failed to deliver additional mentions of entities valid for query. This is why we have also introduced a technique of striping of the top entity mention and executing a shorter, rewritten query to retrieve an additional set of entities. For example for the query "chris brown nicki minaj right by my side download" we received *Nicki Minaj* as the top entity but did not receive any entity matching the "chris brown" mention. We also executed an additional query where the top mention was removed, in this case new query was "chris brown , right by my side download". We replaced mention with a comma to avoid possible matching conflicts. Entities retrieved by the top entity mention striping were further validated as candidates by entity similarity computation (discussed in section 5).

[7] `http://sourceforge.net/projects/wikipedia-miner/`
[8] `http://lucene.apache.org/core/`

4. ENTITY BACK-MAPPING

After the retrieval of candidate entities from the entity documents index, we filter candidates by the entity back-mapping mechanism. For each candidate entity, we construct a set of possible verbalizations (surface forms) and we consider the entity to be "mapped" if at least one of its surface forms is contained as a substring in the given query. For example the movie *Total Recall* has to have a title/alternative name, lets call it AN, *total recall*, to match the query "*total recall* arnold schwarzenegger" or query "*total recall* movies". In order to capture as many meaningful surface forms as possible, we need to gather, create and modify a variety of alternative names from Wikipedia and Freebase data.

As we have mentioned in the previous section, we are storing parsed values in index fields to retrieve them for the task of entity back mapping. Returning to index fields from the previous section, we have used following index fields to represent ANs:

- *title*: title of the Wikipedia concept (article).
- *alt*: alternative names of articles (redirects).
- *fb_name*: Freebase title for article.
- *fb_alias*: Freebase alternative names for article.
- *alt_disambig*: Name of disambiguation page linking to entity.

In addition to those above, we have also experimented with anchor texts of links. Anchor text often contains alternative names, but it also brought lot of noise to the data, as anchor text can often denote other things, e.g., category, entity property or text relevant only in the article context. Experimental runs where anchor text was included showed a decrease in result quality. Thus we did not use the anchor text, since it would require further analysis.

The best candidates for AN are Wikipage titles and Wikipage alternative names (titles of redirect pages pointing to the article). However, looking at the example of "Total Recall" movie, the Wikipage title is *Total Recall (1990 film)*, which can not be matched to the query. This is why we also used modified titles and redirects to extend ANs, where text in parentheses was removed. Freebase also contains entity names and alternative names, which are often different form Wikipedia titles and redirects. We used them for AN as well.

The last piece of data we have experimented with were names of disambiguation pages, which were added as ANs to all entities linked from Wikipedia disambiguation pages. On a subset set of annotated TREC queries provided by the organizers of the ERD challenge, the inclusion of disambiguation names has increased result quality. For example,

for the "texas *border patrol*" query, we could identify *United States Border Patrol* because of the *Border Patrol* disambiguation page. Similarly, for the query "*kenmore* gas water heater", *Kenmore Appliances* were returned as the top search result, but we could map it back to the query only when using the AN gathered form *Kenmore* disambiguation page. However, when testing on a larger number of queries, disambiguation pages brought additional noise and worsen the results; we did not use it at the end in the final run.

In addition to the stripping of content in parentheses from alternative names, we have used further modifications of all AN strings to extend the set of possible surface forms:

- Converting all strings to lowercase;
- Adding variations where commas or any non alphanumeric character was replaced by space or removed;
- Adding variations to strings containing a number at the end, for example, for *Area Code 719*, we would add *719 area code* as a surface form as well;
- For ANs containing 3 words, we added also the AN which exchanged the first and the last words, for example for *Boston Opera House* we added *opera house boston* AN;
- For ANs containg 4 or 5 words we also added new ANs which omitted one of the middle words, for example, for *Pacific Northwest National Laboratory* we added *pacific northwest laboratory* and *pacific national laboratory* as ANs.

When we gathered all AN variations for the search results, we matched them against the lower-cased original query. Matching ANs where then taken as mention string for the results. Nested matching mentions were removed.

We treated all ANs as equally reliable, and did not consider the search score as a confidence indicator for back-mapping, which could be a part of our future work and further improvements.

It is important to mention that we were treating all Wikipedia entities the same way as the ERD Freebase entity subset. This helped us not to return false matches, in case the real entity mentioned in the query was not a part of the ERD subset. For example, for the query *hp mini 2140*, we have correctly identified the *HP Mini 2140* Wikipage, but it was not a part of the ERD subset, so we returned nothing for this query. If we were to map to query only the entities from the ERD subset, *Mini* car would have been returned as a valid entity. Also, by experimenting on the ERD queries we found out that the strategy to map all Wikipedia entities is valid. However, the final dataset introduced some problems with this strategy. E.g., in the query "*harry potter spells*", the *Harry Potter* entity has not been identified, because we have detected an entity which covered the whole query, the *List of spells in Harry Potter* Wikipage, and the nested entity was removed; subsequently the empty result set was returned. Similarly, in a query "*microsoft office 2010*" the empty result set was returned, since *Microsoft Office 2010* was not in the ERD subset, while *MS Office* was, but has not been considered.

This would be easy to improve in the future, where we would not remove the nested concept in case the top concept is not in the target subset, only if there is no link similarity between the top matching concept and the nested one. So, while *Mini* car has no similarity with *HP Mini 2140*, *MS Office* has a high similarity with *Microsoft Office 2010*. (Link similarity and how it was used is described in next section.)

5. COMPUTING SIMILARITY BETWEEN ENTITIES

Modeling entities as multi-fields documents for IR models and entity back-mapping provides us with a set of candidate entities for a given query. The first two steps deliver a set of candidate entities often containing false positives - entities unrelated to the intent of a query. Consider, for example, a query "*chris brown nicki minaj right by my side download*". The query contains names of two artists; however, we retrieve several different entities which can be back-mapped to the surface form "chris brown". The first question we address in this section is how to disambiguate the correct entity(s) based on the similarity; the second question is when is it better not to consider similarity between entities at all.

Informally, our approach to the disambiguation was to compute a similarity score of a candidate entity to entities that we are already quite confident are good annotations for the query. If the similarity score with one of the entities from an intermediate result exceeds a given threshold, the examined entity is added to the intermediate result set. We compute the similarity by exploiting the Wikipedia link graph. We select vertices from the Wikipedia link graph which are related to the entities retrieved in the first step (IR-based retrieval) and we use the subgraph of the link graph induced by those vertices to compute the topological similarity between vertices. We are dealing with quite small graphs, the number of verices is limited by the number of top retrieved documents from the IR step. We derived the similarity by simply counting the intersection of vertex multisets defined by the out-links of given vertices. E.g., given verices a and b, let A be the multisetset of neighbours of the vertex a and B be the set of neighbours of the vertex b in the induced subgraph of the Wikipedia link graph. The similarity s is defined as $s(a, b) = |A \cap B|$.

From the induced subgraph, we construct the entity similarity graph. This derived structure contains all the nodes from the previous graph, and two vertices are connected by an edge in the similarity graph if their similarity exceeds a given threshold and if they are mapped to different surface forms in the query. From the first two steps (IR and entity back-mapping) we have a set of candidate entities together with the surface forms that appear in the query. We select the surface form that belongs to the retrieved entity with the highest score from the retrieval step. For all entities mapped to this surface form we traverse the entity similarity graph and construct sets of possible interpretations for the query.

Let us illustrate the process on the example query "*chris brown nicki minaj right by my side download*". After the retrieval and back-mapping step we would have two sets of entities mapped to surface forms "chris brown" and "nicki minaj". We select the subgraph of the Wikipedia link graph induced by the entities belonging to those two sets. From there, we form the similarity graph. The highest score from the IR step is associated with a (single) entity linked to the surface form "nicki minaj". We start the depth-first traversal of the similarity graph from that entity. Although we have received multiple entities mapped to the surface form "chris brown", there is just a single edge from the entity "nicki minaj" to one entity mapped to "chris brown" in the similarity graph. The depth-first traversal of the similarity graph has just one branch, which will be returned as an interpretation of the query.

Although computing the link-based similarity of entities is very useful in a number of cases for disambiguation purposes, it can have negative effects for some entity types. E.g., let us consider a query "billy idol bratislava", which contains two entities - a music artist and a location. Although the two entities have little in common when examining the link graph or even the textual descriptions, the query makes sense. We have concluded that some entity types, such as locations or websites, most often just describe the geolocation context. In our system, when comparing similarity of entities from which at least one is of the location type, we return a default similarity value.

For future refinements, the link similarity should be modified to account for large intersections (high similarity) dwarfed by even larger differences between the link sets (which imply lower similarity) by either scaling: $s(a,b) = |A \cap B|/|A \cup B|$ or using the symmetric difference as the measure of dissimilariry $d(a,b) = |A \Delta B|$.

6. ADDITIONAL FEATURES

In this section we describe additional features which helped improve result quality:

- **Link based extension of search results:** search results were extended with entities with large number of incoming links from the original search result entities
- **Spell-checking:** Spell-check was applied to queries containing terms with 0 frequency in Wikipedia titles and alternative names
- **Location detection:** additional, gazeteer-based detection of Countries, US states and large websites names

6.1 Entities Related by Wikipage Links

In this section we discuss search result extension with entities related by Wikipage links (RWL) to the original search results.

We have started to evaluate this approach after discovering a drawback of the retrieval mechanics used in our system. Mentioned drawback was the following: often, our retrieval process failed to identify highly central entities, such as countries; e.g. for a query '*new zealand* derm' the retrieval model delivered results related to the country of New Zealand, but non of them represented the entity 'New Zealand' directly. This was mainly caused by the inclusion of several entity document fields in the query retrieval - namely fields containing section headers of Wiki article and Wikipedia categories. However, exclusion of those fields led to significant decrease in the capability of retrieving non-central entities. To summarize, multi-field search index with section headers, categories, links or anchor text has been instrumental in retrieving relevant non-central entities and disambiguating them by context given by a variety of search fields, but it was often failing to detect central entities.

Thus we introduced the RWL approach, where, based on Wikipedia links of search results, we have added to the original search results the top k entities linked by at least 2 search results. We treated those additional entities as additional search results, and we have applied the same back-mapping procedure as we did for the original search results. We have assigned them the score of 95% of the last regular search result's score. In the future work, we will aim at improving the score calculation based on links from search results in a more principled way. This would help us better identify the top related entities and influence the further similarity calculation.

We did not gain any improvement by the RWL technique on the TREC dataset, but on the testing dataset it helped enhance a decent number of queries. For example, for a query "*youtube* music videos *elvis* live", we were able to identify *YouTube* in the query, but we have discovered *Elvis Presley* only after applying the RLW technique.

6.2 Spell-check

As typos and spelling mistakes where not uncommon in the ERD data sets, we have introduce a simple spell-checking mechanism to deal with evident typos. We were checking frequency of query terms in the data set containing Wikipedia titles (regular and redirect pages as well). If a term had 0 frequency we have attempted to correct the spelling of the term. Spell checking algorithm available as a part of the Lucene toolkit was used. This spell checking mechanism allowed us to correct few evident typos (e.g. 'florids' has been rewritten to 'florida'). Wrongly spelled words with a non-zero frequency have not been covered.

6.3 Dirty Hacks

Because of the problem with the retrieval of some central entities (discussed in Sect. 6.1), we have introduced a post-processing step that was designed to capture names of countries and states that have not been detected. After the regular entity resolution procedure was finished, we have stripped detected surface forms from the original query and we have checked for names of states / countries as well as top website names in the remaining content of the query. If one was found we would add it to the result set. This step was a 'dirty hack' to compensate for the downside of the retrieval procedure.

7. EVALUATION ON TREC DATA

In this section we discuss results on a sample of TREC data - the data with annotations posted as the beta version of annotations[9] by the ERD organizers. This data set is very limited, and many of described approaches behaved differently on the TREC dataset and on the 500 queries which were used in the ERD training phase. Since the TREC dataset is the only publicly available one, we provide numbers for this dataset.

In the ERD Challenge [1], the evaluation focused solely on F1, because it was easier to identify borderline cases with zero retrieved or annotated entities for a query. However, we wanted to get an idea about Precision and Recall while developing the system, so we have calculated Macro Precision and Macro Recall, where there was no problem with borderline cases. In addition to these, we have also calculated Macro F1 and two types of Micro F1. Micro F1 calculated in the same way as defined by the ERD organizers, which we refer to as *Micro F1 Set* and *Micro F1*, which considered each returned entity as correct or incorrect independently of the defined interpretation sets. As one can see, there is always a few percent gap between Micro F1 and Micro F1 set. The results for these five measures are summarized in Table 1, evaluated on the TREC dataset.

In the Table 1, we list evaluations for several features. The first row represents the best run of the system. The last row

[9] `http://tinyurl.com/DatasetERD2014`

Table 1: Results on beta TREC data

	Macro Precision	Macro Recall	Macro F1	Micro F1	Micro F1 Set
best	0.7222	0.7761	0.7482	0.7968	0.7674
mW off	0.7027	0.7761	0.7376	0.7826	0.7461
fLx off	0.6892	0.7612	0.7234	0.7791	0.7426
no AN fix	0.6250	0.7463	0.6803	0.7699	0.7213
SB off	0.7143	0.6716	0.6923	0.7401	0.7248
ID	0.5644	0.8507	0.6786	0.7113	0.6816

with the name *ID* represents the run where we searched only for the entities with the ERD Freebase ID assigned. As one can see, the Recall is high but Precision and F1 have dropped significantly. This validates the idea to have a broader knowledge base to cover wider range of topics. Other results presented in Table 1 are related to switching off a single feature to see its impact on the result quality. The *mW* feature represents the entity back-mapping also for entities with no ERD Freebase IDs. The *fLx* feature represents string operations described in section 4: removing the middle words in ANs or changing the word order in some longer names. In addition, the *no AN fix* row shows results when only the raw lower-cased ANs were used for entity back-mapping. The *SB* feature is related to detecting of US states, country names or top websites in queries as a last step.

There are additional features which contributed to result quality such as combination of retrieval models, entity similarity computation, RWL or spell-check, but had no impact on queries from the limited TREC dataset.

8. CONCLUSIONS

In this paper we have described our approach to the entity recognition which we have used in the ERD 2014 challenge. We have combined IR techniques of indexing and search, with semantics in the form of annotations and types, as well as semantics encoded in the network topology of interconnected entities. Our algorithm first searches for entity candidates in the index and then filters them based on the back-mapping of possible entity verbalizations to the query and based on entity similarity calculations, where we exploit the link graph of Wikipedia.

The solution relies on proven information retrieval models, while reusing the human knowledge encoded in Wikipedia, Freebase and DBPedia to filter search results. Filtering is done on a small subset of search result entities instead of a computation on entities from the whole knowledge base. Thus we have created a search-based solution with low memory consumption and low query response time, applicable to any language or domain where enough data on entity description is available. No language dependent NLP approaches were used.

The described system has been ranked as the 4^{th} out of 19 systems, with F1 score of 0.6557.

9. ACKNOWLEDGMENTS

This work is supported by Magnetic, Inc. In addition, it is supported by project VENIS FP7-284984, VEGA 2/0185/13 and CLAN APVV-0809-11. We would like to thank Tom Comerford for editing the paper.

10. REFERENCES

[1] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang. ERD 2014: Entity recognition and disambiguation challenge. *SIGIR Forum*, 2014 (forthcoming).

[2] M. Ciglan, K. Nørvåg, and L. Hluchý. The semsets model for ad-hoc semantic list search. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 131–140. ACM, 2012.

[3] M. Laclavík and M. Ciglan. Towards entity search: Research roadmap. In *Proceedings of WIKT 2013*, WIKT Workshops, pages 161–166, 2013.

[4] R. Neumayer, K. Balog, and K. Nørvåg. On the modeling of entities for ad-hoc entity search in the web of data. In *Proceedings of ECIR*, ECIR'12, pages 133–145, Berlin, Heidelberg, 2012. Springer-Verlag.

[5] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, Apr. 2009.

[6] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342. ACM, 2001.