Entity-Based Information Retrieval System: A Graph-Based Document and Query Model

Mohannad ALMASRI[1], Jean-Pierre Chevallet[2], Catherine Berrut[1]

[1] UNIVSERSITÉ JOSEPH FOURIER - GRENOBLE 1, LIG laboratory, MRIM group, Grenoble, France
mohannad.almasri@imag.fr catherine.berrut@imag.fr

[2] UNIVERSITÉ PIERRE-MENDÈS-FRANCE - GRENOBLE 2, LIG laboratory, MRIM group, Grenoble, France
jean-pierre.chevallet@imag.fr

Abstract : Named Entity has been playing an important role in information seeking and retrieval. Many queries in web search are related to entities. Several knowledge bases contain valuable information about named entities and their relations, such as Wikipedia, Freebase, DBpedia and YAGO. Most existing works, about entity-based search, propose exploiting knowledge about entities and their relationships for expanding or reformulating a user query. Query expansion and reformulation yield effective retrieval performance on average, but results a performance inferior to that of using the original query for many information needs. In this paper, we propose to differently investigate knowledge about named entities and their relations. Therefore, we first present an entity-based document and query model. Then, we suggest a retrieval model based on language models to match between document and query models.

Keywords: Knowledge Base, Named Entity, Information Retrieval, Language Models.

1 Introduction

Named Entity plays an important role in information seeking and retrieval. According to some statistics, about 71% of queries in web search contain named entities (Guo *et al.*, 2009). knowledge bases like Wikipedia, Freebase, DBpedia and YAGO, contain valuable information about entities and their relations. These knowledge bases are essentially used for identifying entities in a user query. In an entity-based search scenario, retrieval systems exploit relationships between entities in order to expand or reformulate a named entity query. (Liu *et al.*, 2014; Dalton *et al.*, 2014; Audeh *et al.*, 2014; ALMasri *et al.*, 2013; Guo *et al.*, 2009; Xu *et al.*, 2008).

Query expansion yields effective retrieval performance on average, but results a performance inferior to that of using the original query for many information needs[1] (Zighelnic & Kurland, 2008). We think that one of the reasons that yields to this problem in entity-based search that the related entities are integrated as keywords into the original query. For example, the query «Silent Film» which searches for documents on history of silent film, actors and directors. A document talks about «Charlie Chaplin», for instance, is a relevant document to this query. However, in a classical query expansion system, the term «Charlie Chaplin» is added to the original query «Silent Film» as two keywords: «Charlie» and «Chaplin». As a result, a classical retrieval model retrieves documents contain «Charlie», «Chaplin», and «Charlie Chaplin» without respecting that «Charlie Chaplin» represents in total one entity.

[1] This robustness issue is called the query drift problem.

Our main contributions, in this study, are three fold:

- Identifying named entities in documents as well as in queries using a knowledge-based for entity detection.
- Proposing an entity-based a document and a query model starting from the identified entities and their relationships.
- Proposing a retrieval model based on language models framework (Ponte & Croft, 1998) for matching between our document and query models, and takes into account entity relations.

An entity is something that has a distinct, separate existence, which can be a person, a place, an organization or miscellaneous. The information associated with an entity is more abundance and less ambiguous for retrieval task than query keywords. Thus, it is our intuition that retrieval performance can benefit from passing into an entity-based retrieval system.

We use a knowledge-based approach for entity detection in documents and queries, where we propose to use Wikipedia as a knowledge repository about named entities. Wikipedia is a free online encyclopedia, it records one article for a real world entity, with information focuses on this entity. Wikipedia contains a huge number of linked articles about named entities. It is in fact a large manually edited repository of entities. Its large volume of structured data, and high quality content make it a convenient and perfect knowledge source which could play an important role in named entity detection.

2 Entity-Based Search System

Three essential components are existed in an information retrieval system: a document model, a query model, and a retrieval model that matches document and query model. Given a query and a document, we identify entities which are mentioned within them based on Wikipedia. Then, we build an entity-based graph representation for a query and a document. Finally, We adapt language models to achieve the matching between document and query graphs. We detail these steps in the following.

2.1 Wikipedia as a Graph

Wikipedia is an encyclopedia that represents a very large, high quality, and valuable knowledge source in natural language. Moreover, Wikipedia is also a hypertext in which each Wikipedia article can refer to other Wikipedia articles using hyperlinks. We consider only *internal links*, i.e. links that target another Wikipedia article.

We represent Wikipedia as a directed graph $G(A, L)$ of articles A, connected by links $L \subseteq A \times A$. Each article $a \in A$ is a description of an object, an entity, an historical fact, etc.. Furthermore, each article contains links to other articles. Relations between articles L are defined on $A \times A$, where (a_1, a_2) means that the article a_1 shows a link to the article a_2. In this article, we define:

$$\begin{array}{rcl} I, O & : & A \rightarrow 2^A \\ I(a) & = & \{x \in A | (x, a) \in L\} \\ O(a) & = & \{x \in A | (a, x) \in L\} \end{array}$$

where $I(a)$ is the set of articles that point to a (*Incoming Links*) , and $O(a)$ is the set of articles that a points to (*Outgoing Links*).

We propose to weight the Wikipedia graph in order to evaluate the strength of links between articles. For that, we consider that two Wikipedia articles are semantically similar, if they share similar links, i.e. if they have similar incoming and outgoing link sets.

Hence, two articles a_1 and a_2 in A are semantically similar if they share articles that point to them, and if they share articles that a_1 and a_2 point to. Then, we propose the following semantic similarity:

$$SIM(a1, a2) = \frac{|I(a_1) \cap I(a_2)| + |O(a_1) \cap O(a_2)|}{|I(a_1) \cup O(a_1)| + |I(a_2) \cup O(a_2)|} \quad (1)$$

where $I(a)$ are incoming links to article a, and $O(a)$ are outgoing links from a.

2.2 Document and Query Models

- **Entity Detection**. Given an n-word query q (w_1, w_2, ..., w_n). Instead of representing the query q by their keywords, we represent the query by entities which are mentioned within. Query annotation establish a link from query sub-phrases to entities in a knowledge base. To do this, we verify for each sub-phrase of q if there is an entity page entitled exactly by this sub-phrase, or it is redirecting to an entity page. Figure 1 shows a sentence that contains four named entities: silent film, film, recorded sound, and dialogue. Similarly, given a document d, we apply the same annotation strategy for each sentence in this document. Finally, we obtain a list of entities for each document or query.

A silent film is a film with no synchronized recorded sound, especially with no spoken dialogue.

Figure 1: Example of a sentence that contains four named entities: silent film, film, recorded sound, and dialogue.

- **Entity Linking**. Following the detection step, where each document or query is represented as a list of entities which are mentioned within. In Wikipedia, each entity page is linked with hyperlinks to a number of other entity pages. We inherit these links into our document and query representation, i.e. we link between two entities in a document or a query representation if there is a link between these two entities in Wikipedia. As a result, documents and queries are represented as a sub-graph of Wikipedia graph. Figure 2 gives an example of document and query representation.

2.3 Retrieval Model

In the previous section, we see that each document and query are represented by a graph of named entities, we look for a retrieval model achieving the following two goals:

– **Non-Matching Entities**. First goal is to deal with unmatched query entity, i.e. query entity which does not appear in a document, e.g. like e_2 in figure 2. In this case, we verify the existence of any link between this entity and other document entities in order to reduce the semantic gap during the matching between a document model d and a query model q (considering dashed links in figure 2).
– **Entity-Centered Documents**. Second goal is to identify from a series of documents that mention a given query entity, those which are entity-centered. Intuitively, we assume that an entity-centered document contains entities linked to this entity. For instance, the case of the query «Silent Film», a document contains the two linked entities «Silent Film» and «Charlie Chaplin», should be ranked before another document contains «Silent Film» with another non-linked entity like «Lionel Richie».

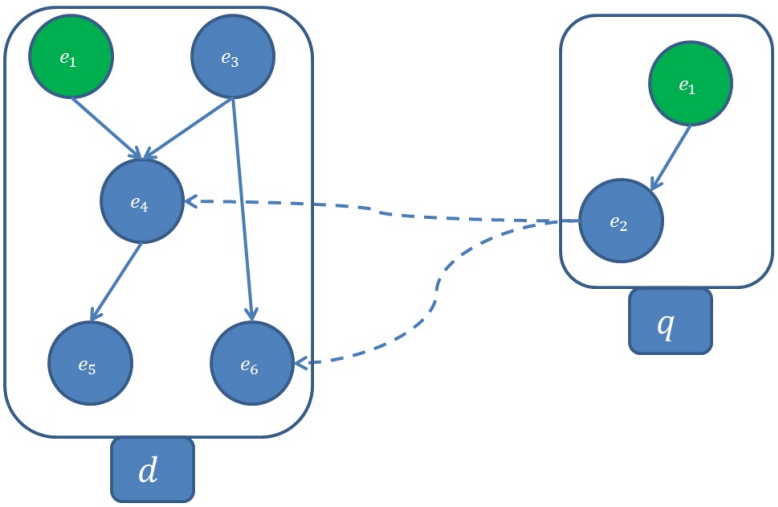

Figure 2: Example of document and query model. The document d contains five entities (e_1,e_3,e_4,e_5,e_6). The query q contains two linked entities (e_1,e_2).

In our approach, we propose to model these two concepts using the language models framework (Ponte & Croft, 1998).

2.3.1 Language Models in Information Retrieval

The basic idea of language models assumes that a query q is generated by a probabilistic model based on a document d. Language models are interested in estimating $P(d|q)$, i.e. the probability that a document d is used to generate a query q. By applying Bayes' formula, we have:

$$P(d|q) \propto P(q|d).P(d) \tag{2}$$

$\propto$ means that the two sides give the same ranking. $P(q|d)$ the query likelihood for a given document d. $P(d)$ is often assumed to be uniform, and thus it is discarded for ranking docu-

ments. We can rewrite $P(q|d)$ the query likelihood after adding the log function as:

$$logP(q|d) = \sum_{e \in V} \#(e;q).logP(e|d) \tag{3}$$

where $\#(e;q)$ is the count of entity e in the query q and V is the vocabulary set. Assuming a multinomial distribution, the simplest way to estimate $P(e|d)$ is the maximum likelihood estimator:

$$P_{ml}(e|d) = \frac{\#(e;d)}{|d|} \tag{4}$$

where $|d|$ is the document length. Due to the data spareness problem, the maximum likelihood estimator directly assign $null$ to the unseen entities in a document. Smoothing is a technique to assign extra probability mass to the unseen entities in order to solve this problem.

Jelinek-Mercer smoothing Zhai & Lafferty (2004) is one of the smoothing technique to add an extra pseudo entity frequency $\lambda P(e|C)$, based on the collection entity collection, to the unseen entity as follows:

$$P_\lambda(e|d) = (1-\lambda)P(e|d) + \lambda P(e|C) \tag{5}$$

We distinguish in the previous equation two main parts: a part related to the document $P(e|d)$, and another part related to the collection $P(e|C)$. In fact, the next section provides our adaption of language models which is based on modifying the way we estimate the document part probability $P(e|C)$ in order to achieve our two goals.

2.3.2 Language Models Adaptation

Our approach to achieve our two goals: **Non-Matching Entities** and **Entity-Centered Documents**, is based on modifying the way we estimate the probability $P(e|d)$ inside the the language models framework.

- **Non-Matching Entities**. We aim to reduce the semantic gap during the matching between a document model d and a query model q. To do this, we propose to modify a document model according to the query and the external knowledge about entity relations.

  Classical IR models compute the relevance value between a document d and a query q based on the coordination level, namely $d \cap q$. Instead of that, we here propose to compute the relevance value by considering also the *unmatched entities* of the query $e \in q \setminus d$, where $\setminus$ is the set difference operator. We therefore expand d by the query entities that are not in the document, but they are semantically linked to at least one document entity (like e_2 in figure 2). In this way, we maximize the coordination level between the document and the query. As a result, we maximize the probability of retrieving relevant documents for a given query. To put it more formally, the modified document, denoted by d_q, is calculated as follows:

  $$d_q = d \cup F(q \setminus d, G, d) \tag{6}$$

  where $F(q \setminus d, G, d)$ is the transformation of $q \setminus d$ according to the knowledge graph G and the document d. The knowledge graph G provides a similarity function between entities

$SIM(e, e')$, see formula 1, denoting the strength of the semantic relatedness between the two entities e and e'. For each entity $e \in q \setminus d$, we look for a document entity e^* which is given by:

$$e^* = argmax_{e' \in d} SIM(e, e') \tag{7}$$

e^* is the most similar entity of d for $e \in q \setminus d$. Then, the pseudo frequency of a query entity e in the modified document d_q relies on the frequency of its most similar document entity $\#(e^*; d)$, we define the pseudo frequency of e as follows:

$$\#(e; d_q) = \#(e^*; d).SIM(e, e^*) \tag{8}$$

This pseudo frequency of the entity e is then included into the modified document d_q. Based on this definition, we now define the transformation function F which expands the document.

$$F(q \setminus d, G, d) = \{e | e \in q \setminus d, \exists e^* \in d, e^* = argmax_{e' \in d} SIM(e, e')\} \tag{9}$$

Note that, if e is not related to any document entity, then we do not have a corresponding e^* for e. Therefore, the unmatched entity $e \in q \setminus d$ will not expand d. Now, we replace the the transformation F with its value in the Eq.6 to obtain the modified document as follows:

$$d_q = d \cup \{e | e \in q \setminus d \wedge \exists e^* \in d : e^* = argmax_{e' \in d} SIM(e, e')\} \tag{10}$$

The length of the modified document $|d_q|$ is calculated as follows:

$$|d_q| = |d| + \sum_{e \in q \setminus d} \#(e^*; d).SIM(e, e^*) \tag{11}$$

Now, the modified document d_q replace the original document model d in any smoothing method inside language models. As a result, the language models for a query q will be estimated according to the modified document d_q instead of d. We believe that the probability estimation will be more accurate and more effective than ordinary language models. We estimate therefore the following probability $P(e|d_q)$ instead of $P(e|d)$.

- **Entity-Centered Documents**. As we mentioned, $P(e|d_q)$ is normally estimated using maximum likelihood. We propose instead to combine two probabilities to estimate $P(e|d_q)$: the maximum likelihood $P_{ml}(e|d_q)$, and another probability that promotes entity-centered documents, denoted as $P_{ecd}(e|d_q)$. $P_{ecd}(e|d_q)$ is the probability of having a linked entity for e inside the document. We suppose that $P_{ml}(e|d_q)$ and $P_{ecd}(e|d_q)$ are conditionally independent, and therefore we estimate $P(e|d_q)$ using the following equation:

$$P(e|d_q) = P_{ml}(e|d_q) \times P_{ecd}(e|d_q) \tag{12}$$

For the probability $P_{ecd}(e|d_q)$, we propose to estimate it following equation shows:

$$P_{ecd}(e|d_q) = \frac{\sum_{e_i \in d_q} \#(e_i; d_q) \times SIM(e, e_i)}{|d_q|} \tag{13}$$

where SIM is the similarity defined in the equation 1.

The maximum likelihood $P_{ml}(e|d_q)$ is estimated using the following equation:

$$P_{ml}(e|d_q) = \frac{\#(e; d_q)}{|d_q|} \tag{14}$$

Finally, if we take the example of Jelinek-Mercer smoothing, we simply write the extended Jelinek-Mercer smoothing as follows:

$$P_{\lambda}(e|d_q) = (1 - \lambda)P(e|d_q) + \lambda P(e|C) \tag{15}$$

We note that the collection related part of the model is not affected, whereas, the document related probability is differently estimated to consider our two goals.

3 Conclusion

We propose, in this paper, a graph-based model for representing documents and queries. First, we identify entities which are mentioned in a query or a document using Wikipedia. Then, we link between those identified entities based on the structure of Wikipedia, i.e. two entities are linked in a document or a query if there is already a link between them in Wikipedia. Finally, we adapt language models for information retrieval in order to match between document and query graphs. The proposed adaption for language model could be easily applied for any smoothing method like: Dirichlet or Jelinek-Mercer (Zhai & Lafferty, 2004).

For future work, we find many campaigns focusing on entity retrieval evaluation, as a result, many test collection are available for testing our proposed approach. We find among them: Cultural Heritage collections CHIC (Petras *et al.*, 2013, 2012), Entity Retrieval track studies entity retrieval in Wikipedia (Vries *et al.*, 2008; Demartini *et al.*, 2010), the TREC Entity track which defines the related entity finding task (Balog *et al.*, 2009, 2012).

References

ALMASRI M., BERRUT C. & CHEVALLET J.-P. (2013). Wikipedia-based semantic query enrichment. In *Proceedings of the Sixth International Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '13, p. 5–8, New York, NY, USA: ACM.

AUDEH B., BEAUNE P. & BEIGBEDER M. (2014). Exploring query reformulation for named entity expansion in information retrieval. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, p. 929–930, New York, NY, USA: ACM.

BALOG K., DE VRIES A. P., SERDYUKOV P., THOMAS P. & WESTERVELD T. (2009). Overview of the trec 2009 entity track. In *TREC 2009 Working Notes*: NIST.

BALOG K., SERDYUKOV P. & DE VRIES A. P. (2012). Overview of the TREC 2011 entity track. In *Proceedings of the Twentieth Text REtrieval Conference (TREC 2011)*: NIST.

DALTON J., DIETZ L. & ALLAN J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, p. 365–374, New York, NY, USA: ACM.

DEMARTINI G., IOFCIU T. & DE VRIES A. P. (2010). Overview of the inex 2009 entity ranking track. In *Proceedings of the Focused Retrieval and Evaluation, and 8th International Conference on Initiative for the Evaluation of XML Retrieval*, INEX'09, p. 254–264, Berlin, Heidelberg: Springer-Verlag.

GUO J., XU G., CHENG X. & LI H. (2009). Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, p. 267–274, New York, NY, USA: ACM.

LIU X., YANG P. & FANG H. (2014). Entexpo: An interactive search system for entity-bearing queries. In M. DE RIJKE, T. KENTER, A. DE VRIES, C. ZHAI, F. DE JONG, K. RADINSKY & K. HOFMANN, Eds., *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, p. 784–788. Springer International Publishing.

PETRAS V., BOGERS T., TOMS E., HALL M., SAVOY J., MALAK P., PAWŁOWSKI A., FERRO N. & MASIERO I. (2013). Cultural heritage in clef (chic) 2013. In P. FORNER, H. MÜLLER, R. PARDES, P. ROSSO & B. STEIN, Eds., *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, volume 8138 of *Lecture Notes in Computer Science*, p. 192–211. Springer Berlin Heidelberg.

PETRAS V., FERRO N., GÄDE M., ISAAC A., KLEINEBERG M., MASIERO I., NICCHIO M. & STILLER J. (2012). Cultural heritage in clef (chic) overview 2012.

PONTE J. M. & CROFT W. B. (1998). A language modeling approach to information retrieval. SIGIR '98, p. 275–281: ACM.

VRIES A. P., VERCOUSTRE A.-M., THOM J. A., CRASWELL N. & LALMAS M. (2008). Focused access to xml documents. chapter Overview of the INEX 2007 Entity Ranking Track, p. 245–251. Berlin, Heidelberg: Springer-Verlag.

XU Y., DING F. & WANG B. (2008). Entity-based query reformulation using wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, p. 1441–1442, New York, NY, USA: ACM.

ZHAI C. & LAFFERTY J. (2004). A study of smoothing methods for language models applied to information retrieval. **22**(2), 179–214.

ZIGHELNIC L. & KURLAND O. (2008). Query-drift prevention for robust query expansion. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, p. 825–826, New York, NY, USA: ACM.