PRMS for AOR

Alberto Tonon, Gianluca Demartini, Philippe Cudré-Maouroux

March 24, 2015

1 Introduction

In this technical report we describe how we used the Probabilistic Retrieval Model for Semistructured Data (PRMS) proposed by Kim et al. [1] for the AOR task using the BTC09 corpus. The model was originally designed for XML documents with a fixed number of elements containing text. For each type of XML element a language model on all the text of the corpus contained in elements of the selected type is created. The generated models are then used to rewrite the input query by weighting the occurrence of a query term in a given element type proportionally to the probability of seeing the term in an element of the selected type.

2 Implementation

The easiest way to adapt PRMS to RDF data is to map XML elements to RDF predicates. Ideally, we should create a language model for each property in our dataset, however, we limit ourselves to a list of elements based on the ten most used datatype properties connecting the entity to descriptive text (that is, we discarded all properties connecting to only numbers or URLs expressed as literals). Notice that with this approach we have a number of language models comparable to that used by the authors [1]. We had to make this choice because the dataset we use contains approximatively 15 thousands properties and taking all of them into account would have led to long pre-processing and execution time of the queries.

We implemented the model similarly to what suggested by the original PRMS paper. Starting from the RDF data we create a document for each entity *e* containing the following fields:

label: the literals coming from 20 properties selected because they often lead to a short textual description of the entity, including `http://www.w3.org/2000/01/rdf-schema#label`, and `http://xmlns.com/foaf/0.1/name`.

nick: the text obtained by following the `http://xmlns.com/foaf/0.1/nick` property

title: the text obtained by following the `http://xmlns.com/foaf/0.1/nick` property

member_name: the text obtained by following the `http://xmlns.com/foaf/0.1/member_name` property

Table 1: Effectiveness of the RDF adaptation of PRMS with the available relevance judgments.

SemSearch 2010			SemSearch 2011		
MAP	Prec@10	NDCG	MAP	Prec@10	NDCG
0.0368	0.1098	0.2859	0.0629	0.0960	0.3546

rss_title: the text obtained by following the `http://purl.org/rss/1.0/title` property

tagline: the text obtained by following the `http://xmlns.com/foaf/0.1/tagLine` property

description: the text obtained by following the `http://purl.org/rss/1.0/description` property

encoded: the text obtained by following the `http://purl.org/rss/1.0/modules/content/encoded` property

comment: the text obtained by following the `http://www.w3.org/2000/01/rdf-schema#comment` property

everything: the text obtained by following all the datatype properties.

The documents are then indexed by using the Indri search engine[1]. The statistics for creating the language models are extracted from the generated inverted index and by using a Pig script. The queries we use to test the system are those used in the SemSearch challenge 2010[2] and SemSearch 2011[3] competitions. The modified queries actually issued to the index and their produced results can be found in `http://exascale.info/aor/prms.zip`.

3 Discussion

Table 1 shows the effectiveness of the runs generated by using the method described above in terms of Mean Average Precision (MAP), Precision at 10 (Prec@10), and Normalized Discounted Cumulative Gain (NDCG). The metrics are computed with the available relevance judgments, that is, we did not obtain new relevance judgments in order to cover the top-10 retrieved documents of all queries.

As can be seen the value of the metrics are low. This is due to the presence of many unjudged documents among the top-retrieved results: 758 results among the top-10 results of the SemSearch 2010 topics, and 380 for the 2011 dataset. We manually inspected a small sample of the unjudged documents, concluding that many of them are not relevant, thus, we believe that obtaining additional judgments would not improve drastically the performance of our implementation.

[1] `http://www.lemurproject.org/indri/`

[2] `http://km.aifb.kit.edu/ws/semsearch10/`

[3] `http://km.aifb.kit.edu/ws/semsearch11/`

References

[1] Jinyoung Kim, Xiaobing Xue, and W. Bruce Croft. A probabilistic retrieval model for semistructured data. In *Lecture Notes in Computer Science*, volume 5478 LNCS, pages 228–239, 2009.