Supporting the Negation Operator in the Hermes Graphical Query Language for Document Ranking

Arnout Verheij
308057av@student.eur.nl

Allard Kleijn
303118ak@student.eur.nl

Flavius Frasincar
frasincar@ese.eur.nl

Damir Vandic
vandic@ese.eur.nl

Frederik Hogenboom
fhogenboom@ese.eur.nl

Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

ABSTRACT

Hermes is a Web-based framework designed to build personalized news services using Semantic Web technologies. Users of Hermes query news items using the Hermes Graphical Query Language (HGQL), which makes it possible to create complex queries without any knowledge of specific query languages. In this paper, we propose a document ranking model that can be used with HGQL and that supports the use of the negation operator when constructing the queries. For this purpose we adopt the p-norm Extended Boolean model and compare it favourably against TF.IDF-based approaches.

1. INTRODUCTION

Nowadays, the user is confronted with the problem of information overflow on the Web. This calls for a way to filter the non-relevant information in order to fulfil the user's wishes. Information searching can be achieved in many ways, of which keyword matching is probably the most used. This approach aims for matching user-entered words (related to desired information) with stored key words, and ranking matching information in order of relevance. However, this poses a problem: a search engine cannot cope with the multiple meanings a keyword can have. For example, the word 'turkey' can mean the animal but also the country. Present search engines are not able to find the meaning that the user is using for his/her query.

In order to deal with this problem, we propose to use techniques from the Semantic Web to define the meaning of domain concepts. This way, we do not longer suffer from the previously identified keyword-based problem. The Hermes framework [1] uses this approach to serve personalized news to users. Users of the Hermes framework can use a graphical query language, called the Hermes Graphical Query language (HGQL), to perform their news queries. This allows them to create fairly complex structured queries in an intuitive way with little understanding of query languages.

As HGQL supports negation, in this paper we propose a ranking algorithm that is able to deal with negations. We choose to adapt the p-norm Extended Boolean model in order to devise a ranking algorithm that supports the negation operator in queries. The ranking algorithm we propose is evaluated extensively using several approaches for determining the term weights. The extended version of the paper is to be presented at the 27th ACM Symposium on Applied Computing (SAC 2012) [6].

DIR 2012 February 23-24, 2011, Ghent, Belgium.

2. RELATED WORK

One of the earliest ranking models that can be used for text retrieval is the Vector Space model (VSM) [5]. In the VSM, documents and queries are represented by vectors, where each dimension corresponds to a weight for a single term from the vocabulary. Many weighting models have been developed in the past 25 years [2]. Ranking in the VSM is done by using a computed similarity between document and query vectors. A commonly used similarity for this purpose is the cosine of the angle between two vectors, i.e., the cosine similarity. The VSM approach implicitly performs disjunctive queries, as any document that has at least one of the terms in the query is ranked with a score larger than zero.

The authors of [5] showed how the VSM can be transformed into a p-norm Extended Boolean model, such that it supports conjunctive queries as well. Consider the case where there are only 2 terms in the vocabulary space and all document and query vectors are normalized to unit length. If the query is $[1,1]$, then document $[1,1]$ has the highest relevance if the query is conjunctive, while document $[0,0]$ has the lowest relevance if the query is disjunctive. Therefore, if $\mathbf{q} = [1,1]$, the similarity between a document $\mathbf{d}$ and the disjunctive query $\mathbf{q}$ is defined as the Euclidean distance between the 'zero vector' (all zeros) and $\mathbf{d}$. For a conjunctive query $\mathbf{q} = [1,1]$, this similarity is the complement of the distance between the 'ones vector' (all ones) and $\mathbf{d}$. Generalizing these formulas to m dimensions, any query $\mathbf{q} = [q_0, q_1, \ldots, q_{m-1}]$, and making the term weights depend on p, we obtain the p-norm Extended Boolean model [5]:

$$\begin{aligned} \text{score}\left(\mathbf{d}, \mathbf{q}\ \text{OR}_{(p)}\right) &= \left(\frac{\sum_{k=1}^{m}(q_k)^p(d_k)^p}{\sum_{k=1}^{m}(q_k)^p}\right)^{1/p} \\ \text{score}\left(\mathbf{d}, \mathbf{q}\ \text{AND}_{(p)}\right) &= 1 - \left(\frac{\sum_{k=1}^{m}(q_k)^p(1-d_k)^p}{\sum_{k=1}^{m}(q_k)^p}\right)^{1/p} \end{aligned} \quad (1)$$

The p-norm has several interesting properties. We obtain the regular VSM for $p = 1$ and the strict Boolean model for conjunctive and disjunctive queries for $p = \infty$. By choosing p to be somewhere between 1 and ∞, we can let the ranking results resemble from a VSM ranking to a strict Boolean ranking.

The VSM and the p-norm Extended Boolean model both work with the assumption that all terms appear in the documents and in the query. The authors of [3] mention the use of '-1' for negated query terms, but this approach has no theoretical nor empirical backing so far.

3. HGQL RANKING MODEL

The ranking algorithm that we propose is based on the p-norm Extended Boolean model. This model, like many other ranking models, requires a document and query term weight calculation model. The methods that are considered for this purpose are the Extended Boolean model (Rank ($e\mathbb{B}$)), which uses a simple binary weight, the traditional TF.IDF weights (Rank (tfc.tfc)), the TF.IDF model with logarithmic weights (Rank (lxc.ltc)), and the TF.IDF model with a combination of the document length and the average document length for length normalization (Rank (Lnu.ltu)).

In order to be able to compute the relevance scores of documents based on structured queries consisting of disjunctive, conjunctive, and negation operators, we need to adjust the formulas of the original p-norm Extended Boolean model. This is necessary as the original p-norm Extended Boolean model does not provide support for negation operators. In order to support negation, we make an adjustment in the way term weights are assigned. Term weights in document vectors for terms that occur in the document are not computed differently than before, while terms that do not occur in the document are given weight -1 instead of 0. In query vectors, the term weights that are not part of the query are set to be 0, and the term weights that are used in the query and are not negated are computed with the four different methods described above. The term weights in a query vector that are negated are computed as formerly mentioned, however now they are multiplied with -1.

Because we adapt the term weighting procedure, the original ranking model of the p-norm Extended Boolean model needs to be adapted (Equation 1). Consider the vector space consisting of solely two terms. Then the disjunctive query part in the original model is based on the distance to the worst case $d = [0, 0]$. If we allow for negation, the worst case document becomes $[-q_a, -q_b]$. In general, the worst case is $-\mathbf{q}$. The conjunctive query part in the original model is based on the best document $d = [1, 1]$. Again, if we allow for negation operators, the best document changes to $[q_a, q_b]$, i.e., in general the best case is $\mathbf{q}$. With these new worst and best cases, the upper bound of the nominator in the original formulas is $\sum_{k=1}^{m}(2 * q_k)^p$. This means that we have to change the normalization factor in order to have the similarity score range between 0 and 1. We obtain the following updated p-norm Extended Boolean model:

$$\begin{aligned} score\left(d, q\ \mathrm{OR}_{(p)}\right) &= \left(\frac{\sum_{k=1}^{m}(q_k)^p(d_k + q_k)^p}{\sum_{k=1}^{m}(2 * q_k)^p}\right)^{1/p} \\ score\left(d, q\ \mathrm{AND}_{(p)}\right) &= 1 - \left(\frac{\sum_{k=1}^{m}(q_k)^p(q_k - d_k)^p}{\sum_{k=1}^{m}(2 * q_k)^p}\right)^{1/p} \end{aligned} \quad (2)$$

4. EVALUATION

For the evaluation of the proposed ranking algorithm we consider two different measures. The first measure is the precision for the first ten documents in our results list, i.e., 'Mean Precision @ 10' (MP@10). The second measure is the Mean Average Precision (MAP). We evaluate the ranking algorithm based on a results list retrieved for the same set of queries from 5 test users. The test set used is a database of 927 news items about various subjects.

Our study showed that our adapted p-norm Extended Boolean model is performing best with a Mean Precision at 10 (MP@10) of 0.85 and a Mean Average Precision (MAP) of 0.874. The lxc.ltc algorithm provides the second best results with an MP@10 of 0.73 and a MAP of 0.694. The Lnu.ltu algorithm and the tfc.tfc algorithm perform relatively poor with an MP@10 of respectively 0.48 and 0.47 and a MAP of 0.572 and 0.467, respectively.

Our results are in line with the findings presented in [4], where the authors show that it is better to map the document and query vectors differently in the vector space. We can observe that both Rank(lxc.ltc) and Rank(Lnu.ltu) perform better than the basic TF.IDF model Rank(tfc.tfc), where the query and document vectors are mapped similarly.

5. CONCLUSIONS

We presented a ranking model for HGQL, a query language that allows users to build complex structured queries, without the knowledge of any specific query languages. The ranking algorithm of HGQL is based on an adapted version of the p-norm Extended Boolean model. The main contribution of this paper is that our proposed ranking model supports the negation operator in queries. Additionally, the evaluation results have shown that our proposed approach performs better than the traditional TD.IDF approach and its variations.

In the future we would like to extend the ranking algorithm by employing user-defined weights for query concepts and evaluate its performance with respect to the user-specified interests. Another research direction that we would like to pursue is to perform a user-based evaluation with respect to the ease of use of the query language.

6. REFERENCES

[1] F. Frasincar, J. Borsje, and L. Levering. A Semantic Web-Based Approach for Building Personalized News Services. *International Journal of E-Business Research*, 5(3):35–53, 2009.

[2] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.

[3] J. Hynek. Document Classification in a Digital Library. Technical report, University of West Bohemia in Pilsen, 2002.

[4] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523, 1988.

[5] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[6] A. Verheij, A. Kleijn, F. Frasincar, D. Vandic, and F. Hogenboom. Querying and Ranking News Items in the Hermes Framework. In *27th Symposium on Applied Computing (SAC 2012)*. ACM, 2012.