Implementation and Evaluation of Query Filtering in a Role Ontology-Enhanced Search Engine

Stijn Vandamme Tim Wauters Thomas Demeester Filip De Turck
Ghent University – IBBT
Gaston Crommenlaan 8, bus 201
B-9050 Gent
Belgium
Tel. +32 9 331 49 75
{Stijn.Vandamme, Tim.Wauters, Thomas.Demeester, Filip.DeTurck}@intec.ugent.be

ABSTRACT
We designed a role ontology-enhanced multimedia search engine where the user can search and subsequently filter news items with queries and filter options describing the roles of the people who appear in the items, specifically politicians. The system makes use of a separate knowledge base with domain information on politics. We demonstrate that when a user fails to recollect the name of a politician, role-based queries combined with filter options tailored to the query and the result set, lead the user fast to both the name he failed to recollect and the intended results in the multimedia database.

Categories and Subject Descriptors
H.3.3 [**information storage and retrieval**]: Information Search and Retrieval – *query formulation, search process*

General Terms
Design, experimentation, human factors

Keywords
Semantic search, search interface, tip-of-the-tongue experience

1. INTRODUCTION
A traditional search engine is keywords-based: it allows the user to formulate a query consisting of one or more keywords. For a multimedia database with annotated items, the results for the queries are traditionally based on textual metadata. However, the conceptual level of what is searched for might differ from the high-levelness of the annotations of the items.

Roles are an important feature of societies worldwide. Roles entail specific rights, obligations, responsibilities, duties, honors or trusts to their holders. Fulfilling a function, a mandate, a position or an office at a given time influences the fulfiller's standing. Roles can be found in different areas of human society: business, organizations, sports, politics (e.g. Secretary of State), etc.

Some roles can only be held by one person at the time: e.g. the role of CEO of a company or business; the role of president of an organization, etc. Other roles can be held by many individuals simultaneously: the role of employee, the role of member of an organization, etc.

DIR 2012 23–24 February 2012, Ghent, Belgium

In earlier work [6][7] we presented CROEQS, a semantically enhanced search engine. CROEQS allows the user to formulate queries with semantic clauses not only on the names of the annotated persons, but also on their roles at the time the multimedia item was broadcast. CROEQS specifically focuses on searching a news database for items with politicians, allowing the user to query the database for politicians of a given party, with a given role or responsible for a given competence using semantic clauses.

In CROEQS, queries with semantic clauses are translated into traditional queries, and the results of that traditional query in a traditional search engine are returned. Basically, the semantic clause is replaced by a long disjunctive ("or") expression of all the politicians which did match the clause and the interval in which they matched it. This translation is based on role knowledge stored in an ontology.

In this paper, we focus on the optimization of automatic support for the user who's looking for an item with a specific politician, but fails to retrieve that politician's name from memory. People occasionally fail to retrieve a name from memory, often combined with partial recall and the feeling that retrieval is imminent: the name is "on the tip of one's tongue". In such a tip-of-the-tongue experience, the user is immediately able to recognize the name, when it is presented to him.

Faced with such a tip-of-the-tongue experience, the user actually faces *two* related information retrieval problems, each with a specific end goal. One problem is the recollection problem: the user wants to recollect the name of the politician, which he fails to remember. The other information retrieval problem is the user's original problem: the user wants to find the items about that politician, possibly in combination with other keywords.

Our approach is to present the user not only with the results of the translated query, but also with the politicians or their characteristics that account for some of the results. Clicking on these names or facets enables the user to filter the result set. In a tip-of-the-tongue experience, when the user recognizes the name in the filter options, simply clicking it leads to the results of the intended query.

In this paper, we first describe the design of our system (Section 2). After characterizing the database we work on, and the traditional search engine whose results are enhanced, we explain how our system translates queries and generates filter options. In Section 3, we evaluate how the filter options, in combination with semantic queries, help the user who wants to formulate a query with a name that is on the tip of his tongue, with both of his information retrieval problems.

2. FUNCTIONAL OVERVIEW

2.1 Multimedia database

For this article, we work with a database that contains broadcasted television material from Flemish public and commercial broadcasters. At the moment of writing, the database contains about 15,000 hours of broadcasted television material, for the most part news and news-related items. This corresponds to 202,868 individually annotated television items.

Each item is manually annotated with both keywords from a dictionary and a "free text" description. 63,000 different keywords are used in tags, and 209,500 different words used in the description. Most text is in Dutch. Additional metadata for each multimedia item includes the programme title, the date on which the item was broadcasted, etc. In total, the textual metadata comprises 440 MiB.

2.2 Traditional search engine

We first built a traditional search engine, using Lucene. The engine indexes the textual metadata. The user interface to expose this search capacity is in the form of a web application based on the Jetty web server.

The user interface is very standard: it includes a text field and a "Search" button, allowing the manual entry of a query. When the users enters a query, it is sent using a HTTP POST request, and the HTTP response to a query includes the total number of items matching the query, and the details of the first ten results.

2.3 Semantically enhanced search engine

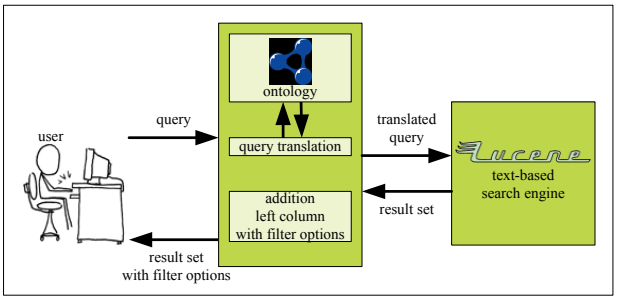

Figure 1. System set-up.

Next, we built a semantically enhanced search engine (Figure 1). This system accepts the same queries as the text-based search engine, but also queries with a semantic clause. With a semantic clause, the user can express the role of a politician. Our system provides 3 kinds of semantic clauses: "politicians belonging to a given party", "politicians responsible for a given competence", and "politicians with a give role", such as Minister, Secretary of State, etc.

2.3.1 Politics ontology

The domain knowledge about Belgian politics is stored independently of the multimedia database and can be maintained independently. The knowledge is stored in an ontology, created with Protégé.

The domain ontology has a core part, which models the generic concept of roles. The core ontology can be used to express that a person holds or has held a specific function, which often is associated with a particular organization, during a given interval. A person can hold multiple roles (even in multiple organizations) at the same time. Additionally, the core ontology can be used to express that the name of the organization can change over time.

This core ontology is extended for the domain of (Belgian) politics. This politics extension defines organizations such as parliaments and the political roles of Minister, Prime Minister and Secretary of State. This politics extension can be used to express that a politician has the role of Minister, responsible for a number of competences, during a given interval.

We populated our politics ontology with RDF triples about all federal ministers and secretaries of state since 1978, all Flemish ministers since 1981, and a number of influential politicians of all Belgian political parties over the last decades.

Using SPARQL, we query the ontology for the politicians that match the clause. In the original query, we then replace the semantic clause with a long disjunctive ("or") expression of all the politicians that match. This query is sent to the traditional search engine. The response sent to the user consists of the traditional search engine's response to the translated query. When the query has at least one semantic clause, a column on the left side with filter options is added. Figure 2 depicts an example result page with left column.

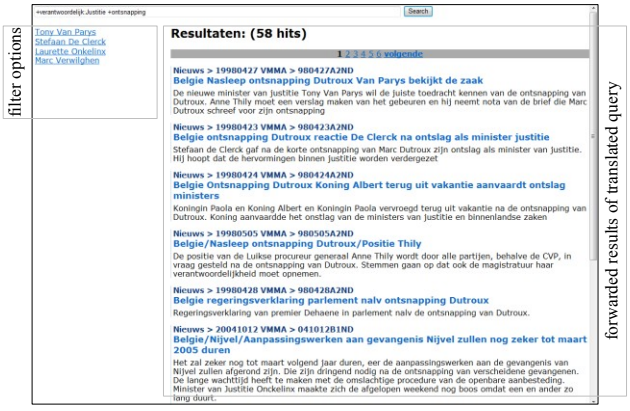

Figure 2. Screenshot of a result page with filter options.

2.3.2 Filter options

The algorithm to determine what these filter options are, is represented in Figure 3.

For each politician that fulfills the semantic clause, the algorithm checks if that politician contributes results to the result set, i.e. whether the result set for the subquery in which the semantic clause is replaced by only that politician's name results in an non-empty result set. If the query contains multiple semantic clauses, only the politicians that fulfill all of the semantic clauses are considered.

If the number of politicians fulfilling the semantic clause who contribute results to the result set is smaller than or equal to 10, we simply list their names as filter options. When the user clicks on such a filter option, the response is the result page for his query where the semantic clause is replaced by the politician's name on displayed in the link of the filter option. By design, that new, filtered query has at least one result.

The filter options are sorted by the number of results they account for, in descending order.

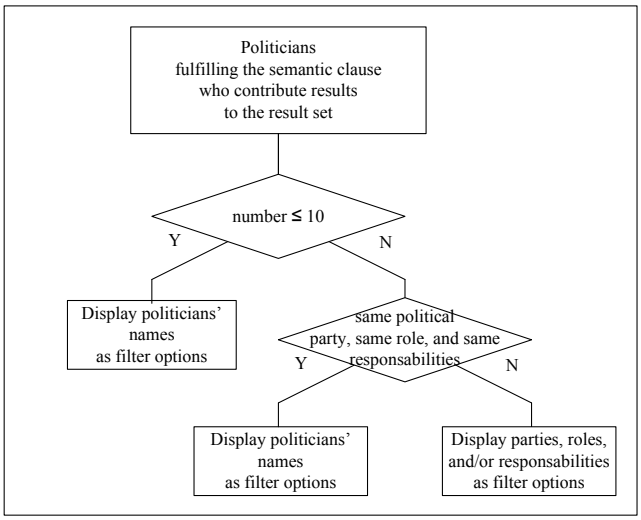

Figure 3. Determination of filter options.

If the list of politicians fulfilling the semantic clause(s) who contribute results to the result set is too extensive, featuring 11 or more politicians, the system first tries to split the list of politicians in sublists. The reason for this is that we don't want to overwhelm the user with long lists. The algorithm tries to split the list according to all the facets that are included in the ontology.

If the list of politicians includes politicians of different parties, their parties are used as filter options. If the list contains politicians that fulfill different roles, their roles are used as filter options. If the list contains politicians with executive roles with different responsibilities, the responsibilities are used as filter options. For the purpose of generating filter options, "no party (independent)", "no formal role" and "no executive responsibility" are possible options in the lists of parties, roles and responsibilities.

Only parties, roles and responsibilities of the politicians that fulfill the semantic clause(s) and contribute results to the query are listed as options. It is possible that the list contains both politicians of different parties and different roles. In that case, the filter options contain different lists of options with different facets. Each facet list contains at least two filter options. A facet displayed in the left column with filter options can never occur in response to a query which includes a semantic clause on that facet.

When the user clicks on such a filter option, the response is the result page for his query where the additional semantic clause is added. In this case too, the new query has at least one result.

The filter options for different facets are not necessarily strict partition of the list of politicians: a politician may be responsible for more than one responsibility. Also, a politician may change role, responsibility, or even party, during his career. However, since the ontology is time-aware, the response to a query with a semantic clause only includes those results that were broadcasted during the time that the politician fulfilled the formulated semantic clause.

If the ontology does not contain any information to make a differentiation between the politicians who fulfill the semantic clause(s) and contribute results to the result set, the system returns the full list of the politicians' names are filter options, even though the list is more extensive.

3. EVALUATION

3.1 Number and nature of the filter options

3.1.1 Methodology

In [5], we evaluated the results set size of CROEQS (without filter options) using 60 queries with one semantic clause and one normal keyword. We used keywords that appear common in political contexts. An example of such a query is "+responsible:Justice +escape".

In this paper, we evaluate the usage of filter options using the same queries. For each of the queries, we determine whether the filter options are names or items in facet lists, and we count the number of filter options. We compare this to the number of politicians fulfilling the semantic clause.

When the filter options are names, a user with a tip-of-the-tongue experience will recognize the name of the politician in the non-overwhelming list. Clicking on the item will result in the result page for his original query. When the filter options are not names, he will need to filter using the presented facets as an additional step.

3.1.2 Results details

For 55 of these 60 queries, the filter options were politicians' names. In Figure 4, the number of names in the filter option list is displayed for these 55 queries in function of the number of politicians fulfilling the semantic clause.

We expected this for queries where the semantic clause was of the type "politicians responsible for a given competence", or when the semantic clause expresses that the politician must be a prime minister or a secretary of state. For instance, there have been only 13 different ministers of Justice in Belgium since 1978. It is no surprise that the number of ministers of Justice that appear in the database in connection with (prison) escapes is smaller than 10: in fact, there are 4, as depicted in Figure 2.

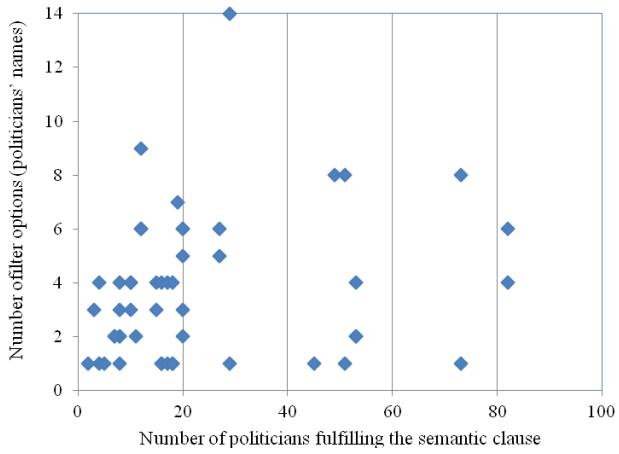

Figure 4. The number of filter options compared to the number of politicians fulfilling the semantic clause

It is therefore no surprise that all of the 6 queries with more than 10 politicians contributing results have a semantic clause is of the type "politicians belonging to party", with a large political party specified. However, even for semantic clauses that specify political parties, the filter options might be just a limited set of names. Often, the same prominent politician has a certain

expertise and appears in the news as the sole voice representing his party's view on his area of expertise. 13 of the 60 queries return the name of only one politician: he is the only politician fulfilling the semantic clause whose name in combination with the rest of the query yields results.

The 5 queries that return facet lists, have also in common that they involve either a core issue or a controversial issue (at one time) for that political party, such as "foreigners' voting rights" for the Flemish liberal party.

For these 5 queries, there are between 14 and 30 politicians that contributed results in the result set, compared to between 29 and 82 prominent politicians of that party mentioned in the ontology.

In 1 query, the filter options list consists of 14 politicians' names, despite the fact that 14 is larger than 10, because the party has never been in power, and hence none of its members have specific roles or executive responsibilities defined in the ontology. The other 5 queries return 2 to 4 roles and 5 to 9 responsibilities as filter options.

3.2 Filtering efficiency of the filter options

For each of the filter options, we determine the number of results given in response to the filtered query, and divide that by the number of results of the original query. This is the percentage of the results that remain after clicking the filter option.

Filter options are considered useful if clicking them considerably reduces the result set. The top filtering option reduces the least.

Obviously, for the 13 queries with only a single filtering option, that option accounts for 100 % of the results.

For the other 47 queries, the top filtering option accounts on average for 61,3 % of the results (σ = 16,0 %). The second filtering option accounts results on average in 30,3 % of the results (σ = 10,9 %). The filtering option that filters the most accounts, leaves on average 9,4 % (σ = 11,3 %) of the results in the result set.

4. RELATED WORK

Information retrieval problems are not always document-centric, aiming to return the most relevant textual documents from a body of documents. They can also more data-centric, retrieving information from structured or unstructured information, with structure that can be simple or very rich. The Initiative for the Evaluation of XML Retrieval (INEX) has a data-centric track, with retrieval tasks over strongly structured collections such as IMDB. This paper does a bit of both: we return documents with little structure (other than broadcasting data) but filtered and tailored based on additional knowledge stored in a structured political role ontology.

In recent years, INEX and TREC also had a Entity Ranking track, where the task focuses on returning relevant entities described in natural language text. Named entities, such as places and famous persons, are often assumed to have their own Wikipedia page, describing the entity.

The technique we use when the number of politicians fulfilling the semantic clause is entensive, is called faceted search [5]. Faceted search allows accessing information organized according to a faceted classification system, allowing searching and browsing in a collection of similar-style items by applying multiple filters. A faceted classification system classifies each information element along multiple explicit dimensions.

An early faceted search prototype was the Flamenco [1]. Hahn et al. [2] implemented an engine based on the semantic knowledge from DBPedia. Li et al. [4] implemented a faceted retrieval system for information discovery in Wikipedia with dynamic facet generation. In our system, the semantic knowledge empowers not only (limited) faceted search, but also query translation.

Kaptein and Marx [3] have presented a case-study on focused retrieval and result aggregation in the political domain. Their work was not done on political news items, but on the official transcripts of the Dutch parliamentary meetings. These documents are highly structured, and consist of multiple discussions, each with multiple speakers, making the natural unit of retrieval much smaller than the meeting's document, unlike the individual news items in our multimedia database, which are atomic enough. Their system also took into account that a parliamentary discussion can be spread over multiple meetings, summarizing debates even with interdependencies between the meetings' documents.

5. CONCLUSION

In the system we presented, users can express additional role-based constraints to the queries that cannot be expressed in traditional text-based search engine using semantic clauses and filter options for further narrowing. In our experiments, a user faced with a tip-of-the-tongue experience regarding a politician's name is in 55 out of 60 cases presented with both the name he failed to recollect in a limited list of names in just one query with a semantic clause, and with the results of his original intended query in just one additional click on the presented name.

6. REFERENCES

[1] English, J., Hearst, M., Sinha, R., Swearingen, K., and Yee, K.P. 2002. *Flexible search and navigation using faceted metadata*. Technical report. University of Berkeley.

[2] Hahn R., Bizer C., Sahnwaldt C., Herta C., Robinson S., Bürgle M., Düwiger H., and Scheel U. 2010. Faceted Wikipedia Search. In *Proc. of the 13th Int'l Conf. on Business Information Systems* (Berlin, Germany, May 2010).

[3] Kaptein, R., and Marx M. 2010. Focused retrieval and result aggregation with political data. *Inf. Retrieval* 13, 5 (Oct. 2010), 412–433.

[4] Li, C., Yan, N., Roy, S. B., Lisham, L., and Das, G. 2010. FacetedPedia: Dynamic generation of query-dependent faceted interfaces for Wikipedia. In *Proc. of the 19th Int'l Conf. on WWW* (Raleigh, NC, USA, 26–30 April 2010), pp. 651–660.

[5] Tunkelang, D. 2009. *Faceted search. Synthesis lectures on information concepts, retrieval, and services*. Morgan & Claypool.

[6] Vandamme, S., Deleu, J., Wauters, T., Vermeulen, B., and De Turck, F. 2009. CROEQS: Contemporaneous Role Ontology-based Expanded Query Search – Implementation and evaluation. In *Proc. Of ICCSN* (Macau, China, 27–28 February 2009), pp. 448–452.

[7] Vandamme, S., Deleu, J., Wauters, T., Vermeulen, B., and De Turck, F. 2009. CROEQS: Contemporaneous role ontology-based expanded query search – Analysis of the result set size. In *Proc. of WIAMIS* (London, United Kingdom, 6–8 May 2009), pp. 169–172.