Expert Finding of Dutch Politicians

Arjan Nusselder
ISLA, University of Amsterdam
arjannusselder@uva.nl

Maarten Marx
ISLA, University of Amsterdam
KRuilsalaan 403 1098SJ Amsterdam, The Netherlands
maartenmarx@uva.nl

ABSTRACT
A system is proposed and implemented that creates a language model for each member of the Dutch parliament, based on the official transcripts of the meetings of the Dutch Parliament. Using expert finding techniques, the system allows users to retrieve a ranked list of politicians, based on queries like news messages.

1. INTRODUCTION

Motivation for this research. The Dutch House of Representatives (Tweede Kamer) is supplied with information about current and past affairs by its information department, the Dienst Informatievoorziening (DI). The DI often pro-actively collects information about topics and events when they suspect one of the politicians will show a special interest in this topic. The DI also performs recommendations of “hot topics” to politicians likely to show an interest in that topic. The DI asked us to implement a system that automates this recommendation process.

Our approach. To match politicians to topics an approach named expert finding was used. This approach is detailed in section 3 and based on work by Balog [1]. We used the parliamentary proceedings to build a profile of each politician. A description of the data is given in section 3. The DI asked us to implement a system that automates this recommendation process.

The current approach shows that, given well-structured parliamentary proceedings, it is possible to construct a good performing retrieval system using out-of-the-box information retrieval techniques. Our evaluation using committee descriptions suggests that the current approach has merit and could be explored further, incorporating more advanced techniques.

2. RELATED WORK

The current approach to the retrieval of politicians is based largely on work done by Balog [1]. We used his approach named expert finding based largely on work done by Balog [1]. We used his 2. RELATED WORK techniques.

3. DATA

We created a language model of each politician in the Dutch parliament (Tweede Kamer) in the summer of 2008. As textual input data we took the parliamentary proceedings which record everything being said in parliament. Through the PoliticalMashup project [2], this data is available in XML in a format which is excellent for our task: every word is annotated with the name of its speaker, her party and the date.

Besides these primary data sources we used biographical data about our politicians available at www.parlement.com.

4. METHOD

What needs to be expressed somehow, is the chance that a politician is knowledgeable on—or at least interested in—the topic expressed by a query. To do so, each politician must be represented with a profile. We first define such a profile as a document in which all text related to that politician is concatenated. This way, the politician–topic matching problem can be reduced to an instance of ranked document retrieval. To calculate the probabilities and ranking, the query is compared to all politicians, each represented as a language model of the concatenation of the related texts.

The measure used for comparison is the Kullback-Leibler divergence. We take \( Q : Word \rightarrow Wordcount \) as the function over the words in the query, and \( P : Word \rightarrow Wordcount \) as the function over the words in a document representing a politician. The basic formula to calculate the chance of a query given a politician is expressed in equation (1).

\[
KL(Q|P) = \sum Q(i) \log \frac{Q(i)}{P(i)}
\]

The result of a query is a ranked list of document identifiers, corresponding to the politician the texts belong to. To create an accessible and usable interface, the results are embedded in a block of additional information. At the time of writing, an interface is available at http://zoekma.science.uva.nl/politiciansearch/search.php.

For the actual implementation, the Lemur Toolkit was used. The important Lemur parameters are Simple KL as

1See Balog, section 3.2.1.
2See: http://www.lemurproject.org
Table 1: Names of topics 6 and 8, as they were used as query-text for the evaluation.

Table 2: Beginning of the description of topic 6.

Table 3: MAP and P@10 of our experiments.

their P@10 for the description run. Figure 2 additionally shows the MAP score of each topic, ordered by topic id, for the full descriptions topics.

For the majority of topics –or committees– more than 6 from the first ten results were correct when we used the full description. Looking at figure1, some possible problems can be identified. Query 8 shows a large discrepancy between the full description and the name only. This may be due to the fact that the topic –just the single word finance– can be and probably is used in virtually all contexts. The full text of the finance topic is descriptive enough to allow for a match between politicians focused on this area and the committee. The fact that almost all politicians will talk about financial issues however, could make the committee name by itself insufficient. Because the focus of the application lies on a search for more verbose text, this is not necessarily a problem.

Query 6 performs worse both with the full description and only the committee name. Several problems may be the cause of this. First, the committee itself consists –as an exception– of only eight members, which makes it harder to correctly retrieve the correct politicians. Also the topic of the committee is relatively new as compared to others, meaning there is probably less data available to create a profile that acknowledges this specific interest of the members. Third, the topic is pretty vague and seems rather specialized.

6. CONCLUSION

As asked for by the information department of the Dutch parliament, we created a recommendation system which matches current members of parliament to hot topics being described by a piece of text. These are typically news articles. We used an out-of-the-box expert search system based on Model 1 of [1] which showed promising performance using an evaluation similar to that of the TREC 2005 W3C enterprise search task.

A small evaluation (3 topics) which mimics exactly the use-case in mind (finding politicians likely to be interested in a news-story) gave even better results: all topics got a P@10 of .6 or higher. These results can be found at http://zoekma.science.uva.nl/politiciansearch/search.php

Here the reader can also evaluate the system herself. Interesting queries are ‘iek’ (I), “Nederland” (“The Netherlands”) and “vrede” “peace.”

7. REFERENCES

Figure 1: Precision at ten for the full description (desc) and the committee-names (name).

Figure 2: Mean average precision for each full text query.