

What you say is who you are. How open government data facilitates profiling politicians

Maarten Marx and Arjan Nusselder

ISLA, Informatics Institute, University of Amsterdam
Science Park 107 1098XG Amsterdam, The Netherlands

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.

The high quality of the system is due to extensive data cleaning and transformation which could have been avoided when it had been available in an open machine readable format.

1 Introduction

The Internet is changing from a web of documents into a web of objects. Open and interoperable (linkable) data are crucial for web applications which are build around objects. Examples of objects featuring prominently is (mashup) websites are traditional named entities like persons, products, organizations [6,4], but also events and unique items like e.g. houses.

The success of several mashup sites is simply due to the fact that they provide a different grouping of already (freely) available data. Originally the data could only be grouped by documents; the mashup allows for groupings by objects which are of interest in their specific domain.

Here is an example from the political domain. Suppose one wants to know more about *Herman van Rompuy*, the new EU “president” from Belgium. Being a former member of the Belgium parliament and several governments, an important primary source of information are the parliamentary proceedings. Most states serve these proceedings on the web with some search functionality. But in most states the information is organized in a document-centric way. One can retrieve documents containing the verbatim notes of a complete parliamentary day. It is simply not possible to state the query

return all speeches made by person X on topic Y.

Of course this information is available in the proceedings but most often not in machine readable form. Austria, the EU and the British website theyworkforyou.com show that it is possible and useful to provide entrance to parliamentary information starting from politicians.

Contribution of this paper We build a people search engine using an out of the box search application (Indri <http://www.lemurproject.org/indri>). We did an extensive evaluation using standard information retrieval metrics which showed that its performance compares to the state of the art. We achieved this performance because of an elaborate and very careful data extraction and data cleaning which would not be needed if the data had been available in an open format (as provided by e.g. the EU parliament).

The purpose of this paper is thus to give a concrete example of the power and impact of open, in our case governmental, data. The paper is organized as follows. We first briefly introduce the field of people search. Then we describe the politician search engine that was created. We describe the data and the used retrieval method and end with an extensive evaluation.

2 People search

The field within information retrieval which is concerned with information needs for concrete persons is called *People Search* [1]. The most basic application consists of a Google style search box in which a user can paste text after which a list of references to persons is returned. The list is ranked by relevance of the persons to the topic expressed by the input text. An important application is expertise retrieval: Within an organization, search for experts which can help on a user-provided topic. The matching functionality of dating sites can also be seen as a form of people search.

We now describe in a simplified form the technique behind most expertise retrieval applications participating in the TReC expertise retrieval task []. The first step is data collection. Typically a crawl of some companies' intranet is available, sometimes with personal data as emails etc. Using named entity extraction techniques, occurrences of persons in the text are recognized and reconciliated to items in a given list of persons. Data deduplication [2] is the main bottleneck in this step.

Then the system creates a model of each person based on the textual and structural content of the crawl. Simply said, each person is represented by one long text document consisting of pieces of text strongly related to that person. The problem here is to determine which parts of the texts in the crawl should be connected to which person.

It should be clear that these problems can be simply avoided for parliamentary data if 1) a unique naming convention is used, and 2) what is being said by whom is structurally made clear. An example of an XML data format in which this is the case is given in Figure 1. Note that the text is a small speech given by the chairman, who is called Jerzy Buzek. Using the MPid attribute he is uniquely linked to a concrete person.¹

¹ The person can be found by following this link <http://www.europarl.europa.eu/members/public/geoSearch/view.do?language=EN&partNumber=1&country=PL&id=28269>

```
<speech docno="3-010"
  MPid="28269"
  MPname="Przewodniczcy.">
<p docno="3-010-1">
  <stage-direction>Przewodniczcy. </stage-direction>
  Kolejnym punktem porzdku dziennego s owiadczenia
  Rady i Komisji dotyczce przygotowania posiedzenia
  Rady Europejskiej w dniach 10 i 11 grudnia 2009 r.
</p>
</speech>
```

Fig. 1. Piece of well structured parliamentary proceeding.

3 Matching politicians to news articles

We build a retrieval system which performs the following task:

given a news article, find those members of parliament which have a strong interest in the topic of the article. Rank them by the strength of that interest.

To match politicians to news we used the people search approach described above. We used the parliamentary proceedings to build a profile of each politician. A description of the data is given in section 4. The resulting system can be seen as answering the question: “Given the words spoken in parliament by a politician, how well does she match a given text?”

Our approach to the retrieval of politicians is based largely on work done by Balog [1]. We used his *Model 1*, which describes the idea of representing experts –politicians in our case– as single documents. This model itself is based on language modelling techniques [7][5].

4 Data

We created a language model of each sitting member of the Dutch parliament. As textual input data we took the parliamentary proceedings which record everything being said in parliament. Through the PoliticalMashup project [3], 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.

5 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 $Q : \text{Word} \rightarrow \text{Wordcount}$ as the function over the words in the query, and $P : \text{Word} \rightarrow \text{Wordcount}$ 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).

$$KL(Q|P) = \sum_i Q(i) \log \frac{Q(i)}{P(i)} \quad (1)$$

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://zookma.science.uva.nl/politiciensearch/search.php>

For the actual implementation, the Lemur Toolkit was used.² The important Lemur parameters are *Simple KL* as retrieval model, and for smoothing a *Dirichlet prior* set at the total number of word types.

Some additional ideas focusing more on the presentation of the results have been implemented. It is possible to not only collect texts on a per person basis, but also split the aggregations on a temporal or party level. Using a log-likelihood comparison, politicians can then be described as opposed to other politicians, or in a specific time-frame. Extensions like these could improve the usefulness of a system, but are left for future evaluation.

6 Evaluation

To see how well our approach performs, an experimental evaluation similar to the TREC 2005 W3C enterprise search task was devised.³ The Dutch parliament has 23 committees, each focused on a policy topic, roughly corresponding to the existing ministries⁴. Each committee consists of about eight to twenty-five members, and an equal or smaller number of reserve members. For each committee its name, a short description and its members (all MP's) are known. We used the both the committee names and their descriptions as topics. A result (i.e., a politician) is correct (“relevant”) on a topic iff it is an active member of the committee described by that topic (reserve members were not counted). The

² See: <http://www.lemurproject.org>

³ See: <http://trec.nist.gov/>

⁴ See: <http://www.tweedekamer.nl/kamerleden/commissies/index.jsp>

total number of candidates is 150, which is the number of current members of parliament.

Thus we do two evaluation runs, one with the names of the committees as topics, and one with the descriptions of the committees. Committee names consist of 1 to 5 words (excluding stopwords); descriptions are between 500 and 1000 words. For instance, the description for the finance committee is 638 words (including stopwords).⁵ Table 1 gives two examples of committee names; Table 2 contains a part of the description of committee with topic id 6.

These longer descriptions match the purpose of our recommendation system more closely.

6	Commissie voor de Verzoekschriften en de Burgerinitiatieven
8	Financien

Table 1. Names of topics 6 and 8, as they were used as query-text for the evaluation.

Commissie voor de Verzoekschriften en de Burgerinitiatieven Commissie Verzoekschriften en Burgerinitiatieven De commissie voor de Verzoekschriften en de Burgerinitiatieven heeft twee taken: het voorbereiden van een beslissing van de Kamer over een individuele aangelegenheid (waar een burger in een verzoekschrift om heeft gevraagd) en het voorbereiden van een beslissing van de Kamer over de ontvankelijkheid van een burgerinitiatief. . .

Table 2. Beginning of the description of topic 6.

Results. We measured the mean average precision (MAP) and precision at 10 (P@10) over two times 23 topics. The results are in Table 3.

Precision at ten is taken as an appropriate measure for two reasons. First, some committees have little more than ten members, which would make precision over ten difficult to evaluate. Second, the intended use of the application foresees a human-readable result set. Figure 2 shows the P@10 for each topic for both evaluation runs (full description and the committee-name only), with the topics ordered by their P@10 for the description run. Figure 3 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 figure 2, 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

⁵ The description can be found at <http://www.tweedekamer.nl/kamerleden/commissies/FIN/sub/index.jsp>.

	MAP	P@10
committee names	.38	.48
committee descriptions	.44	.56

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

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.

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://zookma.science.uva.nl/politiciansearch/search.php>.

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

Acknowledgements Maarten Marx acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the FET-Open grant agreement FOX, number FP7-ICT-233599.

7 Conclusion

The high precision scores obtained by a baseline implementation show the importance of well presented data. Making data available in an open and linkable format using unique identifiers makes it much easier to build robust systems.

References

1. K. Balog. *People Search in the Enterprise*. PhD thesis, University van Amsterdam, September 2008.
2. X. Dong, A. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *Proc. SIGMOD*, pages 85–96, 2005.
3. T. Gielissen and M. Marx. Exemelification of parliamentary debates. In *Proc. DIR*, 2009.
4. M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

5. D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, University of Twente, 2001.
6. C. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
7. J. Ponte and W. Croft. A language modelling approach to information retrieval. *Proc. SIGIR '98*, 1998.

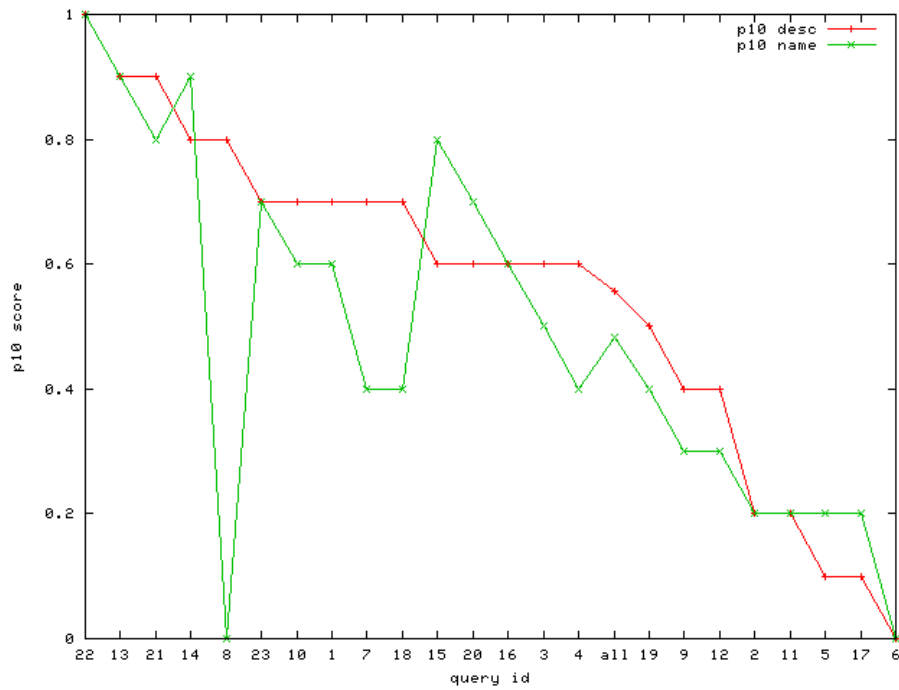


Fig. 2. Precision at ten for the full description (desc) and the committee-names (name).

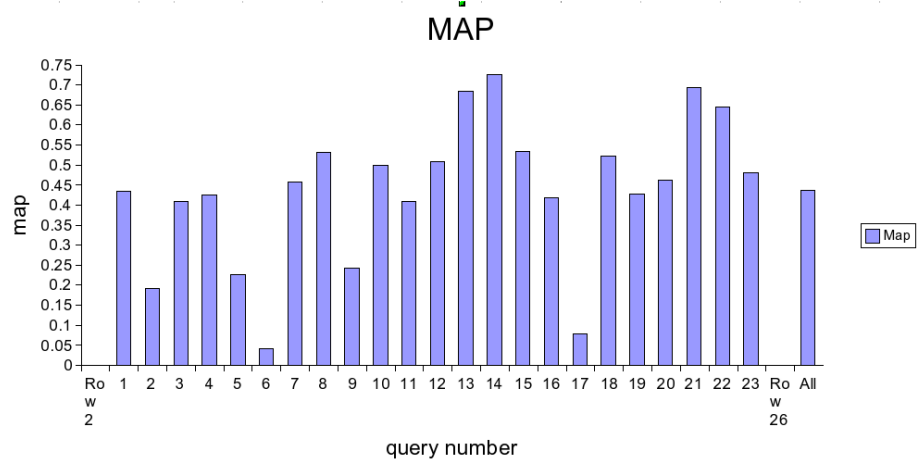


Fig. 3. Mean average precision for each full text query.