A Lazy Approach for Filtering Parliamentary Documents

Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete(✉)

Departamento de Ciencias de la Computación e Inteligencia Artificial,
ETSI de Informática y de Telcomunicación, CITIC-UGR,
Universidad de Granada, 18071 Granada, Spain
{lci,jmfluna,jhg}@decsai.ugr.es

Abstract. We propose a lazy approach to build a content-based recommender system for parliamentary documents. Given a new document to be recommended, the system will decide what Members of the Parliament could find interesting such a document, in order to deliver it to them. Our approach is lazy because we do not build an elaborated profile of each deputy, but collect all the text of his/her speeches within the parliament debates and generate a document collection where we can search through queries. In this way we transform a recommender system problem into an information retrieval problem. Our proposals are tested using the documents of the regional Parliament of Andalusia at Spain.

Keywords: Content-based recommender systems · Information filtering · Information retrieval · Parliamentary documents

1 Introduction

The increasing application of Information and Communications Technologies (ICT) to our daily life has changed our habits: we are connected everytime and everywhere and producing a vast amount of information which is uploaded to the Web and can be consumed by other people. But this revolution is not only focused on individuals, but also on public and private organizations as well as companies, which are involved in a global competition where ICT are crucial for their development.

Governments all around the world are trying to not miss the boat, adapting their administrative processes to the current technology context. This application of ICT to the government framework is called *e-government* [13], and it refers to the ways in which the public administrations of a territory are adopting ICT to serve better to the citizens.

One of the main concerns of the e-government is to facilitate the citizens the access to the information that these administrations generate, as a way of promoting the participation of people and keeping them informed (service *Government to Citizen* in the taxonomy given in [15]). But the problem is that the amount of available information is huge and increasing exponentially, so it is not easy to find what a citizen would need.

A. Kő and E. Francesconi (Eds.): EGOVIS 2015, LNCS 9265, pp. 364–378, 2015.
DOI: 10.1007/978-3-319-22389-6_26

More specifically, the problem presented in this paper is framed in the context of regional or national parliaments, where, again, the problem is how to move closer all the matters discussed in this assembly to the citizens. The generated volume of data should be more accessible than it is. The communication deficit is not only related to the people to whom they have a duty of explanation, the citizens, but also the Parliament staff and the Members of the Parliament (MP) themselves [4]. Then effective applications are required to save this gap. This paper presents an approach to tackle this problem, focusing on the concept of parliamentary initiatives[1]: Given a new initiative, and in order to filter it, the system must evaluate and decide, among the potential MPs (or staff, media or users in general) to whom we send it. But our approach can also be useful to tackle another task, i.e. to help the citizens to circulate their petitions. In this case, a person can submit a request to the system, using its title and a summary that describes the topic (the most important parts since they would capture the MPs attention), and the system can show him those MPs that can be helpful at solving it.

With these ideas in mind, our research may be contextualized within the field of content-based recommender/filtering systems [2,17], i.e. systems that recommend an item (an initiative) to a user based on a description of the item and a profile of the user's interests. There exist many content-based recommender systems for a variety of domains [7,12], as web pages, news, music, movies, books, emails, scientific literature and digital television, among others. However, we are not aware of any such a system in a parliamentary context.

In this paper, we are going to adopt a lazy approach, where we do not build an elaborated profile of each MP (each user) but collect (in several different ways) all the text of his/her speeches within the parliament debates. We consider this information as a kind of document collection and use an Information Retrieval System (IRS) operating on this collection to retrieve the MPs (the "documents") that are more similar to the document to be recommended (or the citizen's topic of interest), which acts as a query to the system. The underlying assumption is that the speeches of an MP can reveal the topics he/she is interested in, and a new document similar in textual content to these speeches probably will also be of interest. Hence we shall recommend the document to the top ranked MPs.

The proposed system is lazy in the sense that it does not build an explicit model of each MP using for example an automatic classifier (based on training data composed of the documents that the user has previously considered as relevant and irrelevant, see for example [3,9,16], where K-NN, decision trees and Naive Bayes, respectively, are used). Instead of learning a user model, our system compares the new document (or topic of interest) with the speeches of the MPs and recommends it to the MPs which seem to be more similar. It works in a way that loosely resembles to a k-NN algorithm [18], which does not train a model based on relevant and irrelevant documents but simply compares the new document with all the documents previously classified. It should be noticed

[1] An initiative is the literal transcription of the discussion in the parliament of a petition presented by the MPs or groups.

that in our case we do not have previous information about irrelevant documents but only about relevant ones (the speeches of each MP). Thus, from a machine learning perspective, it should be tackled using positive unlabeled learning[2] [11].

If we are going to transform our recommender system problem into an information retrieval problem, we must address two main questions: which are the documents that compose our document collection and which are the queries submitted to the IRS against this collection (and how to process them).

The rest of the paper is organized in the following way: in Sect. 2 we consider several proposals to build our document collection from the MP's speeches within the parliament debates. Section 3 studies different alternatives to transform the new document to be recommended (or the topic of interest for a citizen) into a query to the IRS. In Sect. 4 we describe the experiments designed to validate our proposals and the obtained results. Finally, Sect. 5 explains our conclusions and proposals for future work.

2 Document Collection

As we have already mentioned, the source of information about the interests of the MPs will be the textual transcriptions of their speeches within parliamentary debates. These debates are organized around the concept of *parliamentary initiative*, whereby an action taken by an MP or political party is discussed in a plenary or specific area committee session. The transcription of the discussion of an initiative is identified by means of a code, and contains a title (short description of the matters being discussed), a general information section (e.g. type of session, term of office, date and presidency), a summary that includes a detailed description of the agenda (proposer and the list of MPs participating), created once the session has finished. And finally the transcriptions of all the speeches are included and set out like a script for a play or a film (see Table 1).

In this paper we are going to consider and analyze three different configuration alternatives of the document collection. In any case, these documents will be indexed by a search engine and used to find the requested relevant information. In our particular case, the objective is to know either who are the MPs that should receive the new document to be recommended or (for the other possible use of our system) who MPs should be contacted in order to discuss or seek advice for some particular point. Therefore, in both cases the output of our system will be a ranked list of MPs.

2.1 Initiative-Based Collection

This is the simplest approach since initiatives correspond to the original format of the parliamentary documents, i.e. each document represents a whole initiative (Table 1 presents an XML example of an initiative). For that reason this approach

[2] A type of binary classification problem where we have a set of positive examples and another larger set of unlabeled examples, but there is no set of negative examples.

Table 1. Example of the documents' initiative view in the collection encoded in XML, in this case the document represents the initiative with ID *ini1234*.

<initiative>
<iniID> 1234 </iniID>
<summary> a short summary representing the initiative.</summary>
<intervention>
<deputy> Mr. XXX </deputy>
<speech> the first intervention of Mr. XXX in the initiative 1234 </speech>
</intervention>
<intervention>
<deputy>Mr. YYY </deputy>
<speech> reply to Mr. XXX proposals </speech>
</intervention>
<intervention>
<deputy>Mrs. ZZZ </deputy>
<speech> new request to the government </speech>
</intervention>
<intervention>
<deputy>Mr. XXX </deputy>
<speech> this is the answer to Mrs. ZZZ</speech>
</intervention>
<intervention>
<deputy>Mr. WWW </deputy>
<speech> ..... </speech>
</intervention>
</initiative>

will be considered the baseline. In this paper we shall denote this collection *c_INI*, acronym of collection of initiatives.

In this case, the retrieval system will match the query (i.e. the new initiative to be recommended or the citizens' request) to the set of initiatives previously discussed in the parliament, finding the most similar ones. Thus, when the system returns an initiative as relevant, then we shall assume that all the MPs participating in this initiative are the right MPs we are looking for, because they took part in an initiative similar to the new document or to the citizen's information need (for the initiative with code 1234 in Table 1, these MPs are Mr. XXX, Mr. YYY, Mrs. ZZZ and Mr. WWW).

2.2 Profile-Based Collection

Since MPs usually participate in several initiatives (depending on the specific MP, it can be only a few or many initiatives), we thought that an alternative representation could be to construct a document collection of MP profiles, where we store the interests of the MPs. In this paper, we shall explore a lazy way of representing the MPs profiles. Particularly, in this case we propose to collect, for each MP, the text of his/her speeches within all the initiatives in only one document, thus obtaining a document collection with as many documents as MPs.

Table 2. An XML profile view of the documents in the collection. In this case, one profile is build for each MP in the parliament.

```
<profile>
<deputy>Mr. XXX </deputy>
<initiative>
   <iniID> 1234 </iniID>
   <summary> a short summary.</summary>
   <intervention>
      <speech> the first intervention of Mr. XXX in the initiative 1234 </speech>
   </intervention>
   <intervention>
      <speech> this is the answer to Mrs. ZZZ</speech>
   </intervention>
</initiative>
<initiative>
   <iniID> 5678 </iniID>
   <summary> the summary.</summary>
   <intervention>
      <speech> the speech in this initiative </speech>
   </intervention>
   <intervention>
      <speech> another speech </speech>
   </intervention>
</initiative>
<initiative>
   ....
</initiative>
....
</profile>
```

In this case we have few documents but rather large, as Table 2 illustrates. We shall denote this collection as *c_PRF*, acronym of collection of profiles.

2.3 Discourse-Based Collection

After looking at the MP profiles, we found out that some MPs used to participate in a range of initiatives that are related to different areas of interest, for example, agriculture, fishery, economy, food, and so on. Putting all this information together in a common user profile could bias the profile towards the most frequent areas, diminishing the importance of the uncommon ones. So that, we propose a different way (but still lazy) to represent the MP's profiles. Particularly, we divide the document representing an initiative in different documents containing the text of the speeches of each MP who participated in the initiative (in other words, we collect, for each MP, the text of its speeches within each initiative in a different document). As a consequence we have a larger set of documents but with shorter length, as it is illustrated in Table 3, where we show the discourses of Mr. XXX, Mr. YYY and Mrs. ZZZ extracted from our

Table 3. A discourse view of the collection, in this case the discourses of each MP in an initiative are considered as isolated documents. In this example, the original initiative with ID 1234 can be split into four different documents, representing each one of the MP's interventions.

XXX_ini1234.xml	YYY_ini1234.xml
<**discourse**> <iniID> 1234 </iniID> <deputy> Mr. XXX </deputy> <summary> initiative short summary.</summary> <intervention> <speech> the first intervention of Mr XXX ... </speech> </intervention> <intervention> <speech> this is the answer to Mrs. ZZZ</speech> </intervention> </**discourse**>	<**discourse**> <iniID> 1234 </iniID> <deputy>Mr. YYY </deputy> <summary> initiative short summary.</summary> <intervention> <speech> reply to Mr. XXX </speech> </intervention> </**discourse**>
ZZZ_ini1234.xml	**WWW_ini1234.xml**
<**discourse**> <iniID> 1234 </iniID> <deputy>Mrs. ZZZ </deputy> <summary> initiative short summary.</summary> <intervention> <speech> new request to the government</speech> </intervention> </**discourse**>	<**discourse**> <iniID> 1234 </iniID> <deputy>Mrs. WWW </deputy> <summary> initiative short summary.</summary> <intervention> <speech> ... </speech> </intervention> </**discourse**>

toy initiative with ID 1234. The collection obtained following this approach will be denoted *c_DIS*, acronym of collection of discourses.

2.4 Transforming the IRS Output

In case of using the *c_PRF* collection, when we formulate a query (associated either to the document to be recommended or to a citizen information need) against this document collection, we obtain a ranking of documents, each one representing an MP, i.e. we directly obtain a ranking of MPs whose profiles are more similar to the query.

However, in the other two cases, *c_INI* and *c_DIS*, the IRS does not directly return a ranking of MPs but a ranking of complete initiatives (each one associated to several MPs) and discourses (each one associated to only one MP), respectively. Then, to obtain a ranking of MPs, we substitute a discourse in the ranking by its associated MP, and an initiative by the set of MPs who participated in it (preserving the scores). In both cases the problem is that, as the same MP can appear in different initiatives or discourses related to the query, the obtained ranking of MPs can contain duplicate MPs (each duplicate having a different score). Therefore, we must use some strategy for aggregating the scores associated to the same MP into a single score, in order to re-rank the list of (non duplicate) MPs according to this score. The best strategy found in our preliminary experiments was to use the maximum as the aggregation operator.

3 Query Approaches

As we have already said, our model can be used for two different, but related tasks. For the first one, which could be considered as a *filtering* task, we have

to tackle the internal needs of the parliament, where it is necessary to filter (possibly by e-mail) new initiatives to those MPs who might be interested in the discussed topics. For this task, the query is an initiative, which implies that its internal structure is known (the query source is the XML file). The second one, which could be considered as a *recommendation* task, is related to an external use in the sense that there exists a citizen who wants to know which are the MPs being concerned about a given topic. In this case, the citizen expresses his/her information need by means of a query which includes those terms that might be related to the topic, and the system will recommend the most appropriate MPs. Thus, in this case, the structure of the query has no relationships with the way in which the topics are discussed in the parliament. Therefore, two different types of queries can be considered:

q_SGL: In this case, there is a single query being just a set of terms that should be given directly by the user (when recommending MPs) or extracted considering all the speeches in an initiative (in the case of filtering). Note that in this last case we shall obtain very large queries, possibly with hundreds of terms, although it does not represent a big handicap since filtering is a task that can be done offline.

q_CMP: One of the problems of the previous approach is that all the interventions are considered as a whole, being unable to differentiate the particularities of the MPs discourses. In order to solve this problem we propose the use of the structure of the initiative to obtain a compound query, and therefore this approach can only be applied for filtering purposes.

Particularly, we shall divide the initiative by grouping all the text associated to the speeches of each MP who participated in the discussion of the initiative (as illustrated by Table 3), thus obtaining as many subqueries as MPs participated. Ideally, in this case each subquery should represent the point of view of an MP in the initiative, since it is focused in her own intervention. Then, these queries can be executed independently, being necessary a method to fuse the different rankings obtained for each query into a final ranking. The rationale for this proposal is that each subquery may possibly more accurately identify the corresponding MP who participated in the initiative. So, we hope that the compound query is more effective than the single one, in those cases where both can be applied. The ranking fusion strategies considered will be discussed in the next Subsection.

3.1 Ranking Fusion

Ranking fusion has to be applied when considering compound queries, where a set of n different (sub)queries, $\{q_1, \ldots, q_n\}$, are used in order to find the relevant MPs. In this case, for each query q_i we have computed a sorted list of MPs, $L_{q_i} =< m_1, m_2, \ldots, m_i >$, being $s_i(m_j)$ the score of the j^{th}-MP for the query q_i. Therefore, it becomes necessary to merge the different ranked lists into a single one which represents the relevance of each MP to the compound query.

In the literature several methods to tackle data fusion problems can be found [19]. They mainly try to estimate the relevance of all the retrieved documents for

a given query via combining these retrieved documents from multiple information retrieval systems into a single list. Although our problem is different, i.e. different queries are launched into one retrieval system, the same techniques for ranking combination can be applied. As we have access to the scores given by the IRS to each result, and it has been reported [1,20] that in these cases is preferable to use a ranking fusion method that takes the scores into account, instead of methods based exclusively on the ordering, we shall use the score information.

Previously to any combination, since each query represents a speech of an MP in an initiative, it is usual that they differ in length. Therefore, in the case that the retrieval system does not give normalized scores as output, it might happen that those (sub)queries with greater length have larger scores. This situation has a negative impact in the retrieval performance because it could bias the output towards the MPs having large speeches, reducing the relevance of those MPs that with less words are able to express their opinion. Therefore, previous to any combination, a score normalization step is indispensable, in order to make the scores, which are obtained from different queries, comparable to each other. Particularly, in this paper we shall normalize them by considering the score of the top MP for a particular query, i.e.

$$s_i(m_j) \leftarrow s_i(m_j)/s_i(m_1)$$

Focusing on the combination strategies, in this paper we shall explore the following alternatives, proposed originally in [6], which have been reported as good methods for ranking fusion in several studies.

- $CombMAX$: choose the maximum of the relevance values, i.e. $s(m_j) = \max_{i=1,\ldots,n} s_i(m_j)$
- $CombSUM$: the sum of the score values in the different rankings, i.e. $s(m_j) = \sum_{i=1}^{n} s_i(m_j)$.
- $CombMNZ$: this methods tries to promote those MPs appearing more frequently and is computed as $s(m_j) = k \sum_{i=1}^{n} s_i(m_j)$, being k the number of lists where the MP appears.

In the previous formulas, in the case an MP m_j does not appear in a list L_{q_i}, then a zero score is used, $s_i(m_j) = 0$.

4 Evaluation Framework and Results

In this section we describe the components of the evaluation framework, as well as the obtained results and conclusions. The evaluation framework is composed by the following components: a document collection formed by all the initiatives (5258) from the eighth term of office of the Parliament of Andalusia at Spain[3], marked up in XML [5]. These initiatives contain a set of 12633 interventions of the MPs (and a total of 28706 different speeches). In order to evaluate the

[3] http://www.parlamentodeandalucia.es.

performance of our proposals, we have used the repeated holdout method [10]: the set of initiatives is randomly partitioned into training (80 %) and test (20 %), and we repeat this process five times (the results presented in the study are the averages over the different rounds). With respect to the initiatives in the test set, we remove the information related to the MPs who participate in their discussion, being totally anonymous.

We have carried out experiments with both, filtering and recommending problems (see Sect. 3):

1. *Filtering* task is simple, since we can use either all the text in the initiative to build a simple query (q_SGL), or formulate a compound query (q_CMP), using the different discourses of the MPs. In the last case we shall experiment with three different ranking fusion methods (MAX, SUM and MNZ).
2. *Recommending MPs* implies that there exists a user who expresses his/her information need by means of a set of terms. In this experimentation we shall not use real users, but simulated ones instead. Particularly, we shall use as (q_SGL) queries the titles of the initiatives (typically one or two lines of text[4]) which are hand-made brief descriptions of their contents (we shall denote this type of queries as hm). We shall also experiment with automatic summaries of the initiatives which, in order to be able to capture the initiative topics, have been constructed by selecting the best 25 and 50 terms[5] (these queries will be denoted as $au25$ and $au50$, respectively). Note that these numbers are under the limits of words in the summary of a citizen's request, which for example is set to 75 words in some US states, as indicate the National Conference of State Legislature (www.ncsl.org).

Thus, independently on the query, the purpose of our model is to predict those MPs who would be relevant to its content and, as a consequence the output has to be a ranking of MPs. In this experimentation we have considered, as the ground truth that an initiative, i, will be relevant only to those MPs that participate in it, its number being denoted as ni. This is a rather conservative assumption, because it is quite reasonable to think that an initiative can also be relevant to other MPs.

In order to evaluate the accuracy of our approach, we shall measure the quality of the ranking at fixed low levels of retrieved results, particularly we present the recall values considering the top 10 MPs, rec@10. This metric measures how many among the relevant MPs appear in the top positions of the ranking, measuring the capability of the system at finding these MPs. We also show the Normalized Discounted Cumulative Gain [8] over the top 10 positions, NDCG@10, which measures the ranking quality. Moreover, considering that the number of relevant MPs varies with the initiative (on the average there are 2.4 interventions per initiative) we shall also measure the accuracy on the top ni positions using MAP@ni (Mean Average Precision) and the R-precision

[4] An example of the title of an initiative is *"Non-legislative proposal on social and employment situation of women in Andalusia"*.

[5] Using the *MoreLikeThis* facility in Lucene.

(that represents the precision over the ni top MPs). These results were obtained when considering three different information retrieval models, namely BM25, Language Model (LM) and vectorial (VECT), using the implementation in the search engine library Lucene[6]. All the results obtained from our experiments for the different metrics are displayed in Tables 4 and 5. More specifically, the former contains the values of the metrics for the filtering task, while in the latter, the results shown are related to the recommending task.

4.1 Analyzing Filtering Results

The baseline filtering approach to compare our proposals is to use the initiative-based collection, c_INI, and the single queries, q_SGL (i.e. the whole content of the test initiatives to be filtered). That is to say, we use the documents as they are already stored in the parliament, without any processing for both collection and queries. In Table 4 we present the obtained results, where we highlight in bold and underlined fonts the results obtained with the baseline and best approaches, respectively.

From these results we can obtain several interesting conclusions:

1. All the metrics point in the same direction, i.e. what is better from the perspective of one metric is also better from the perspective of any other metric. This means that the differences in performance between different configurations are consistent across the different metrics.
2. With respect to the collection, the best results are obtained for the profile-based collection, c_PRF, followed by the discourse-based one, c_DIS (the worst collection is c_INI), and this happens almost independently on the type of query being selected.
3. For the queries, the best alternative is to submit a compound query, q_CMP, using the different discourses, instead of using a single query, q_SGL, and this is also independent on the type of collection being used.
4. In the case of compound queries, how we aggregate the rankings matters, being in general CombMax the strategy with the best results (except when we consider the initiative-based collection, where it is better to aggregate the results using the other strategies). The results obtained using the CombMNZ and CombSUM strategies are very similar. We think this is because the different subqueries tend to get the same set of MPs[7] (although in different orderings). This may be because all the subqueries are about the same topic (the one discussed in the initiative). Although it has been established [14] that CombMNZ provides best results in the general case, here the situation is different, as CombMax is clearly the best ranking fusion method. This would deserve further analysis. Nevertheless, it should be noticed that our ranking fusion problem is different from the usual one: we have different queries submitted to a single retrieval system, whereas the usual problem is the converse, with a single query launched into several retrieval systems.

[6] https://lucene.apache.org.

[7] Note that we have set to 200 the number of documents returned by the search engine.

Table 4. Accuracy metrics for filtering with the selected retrieval models. The baseline approach is the one obtained processing neither collection nor queries, i.e. $c\_INI \times q\_SGL$ (in bold fonts for each retrieval model). The best results for each retrieval model have been highlighted using underlined values.

Coll.	Query	Comb	rec@10	NDCG	MAP	R-prec
Filtering Task: BM25						
c_INI	**q_SGL**	—	**0.7563**	**0.5790**	**0.3367**	**0.3996**
c_INI	q_CMP	MAX	0.7507	0.5674	0.3184	0.3774
c_INI	q_CMP	SUM	0.7575	0.5968	0.3610	0.4152
c_INI	q_CMP	MNZ	0.7574	0.5968	0.3610	0.4152
c_DIS	q_SGL	—	0.7537	0.6253	0.3929	0.4427
c_DIS	q_CMP	MAX	0.8237	0.7114	0.5147	0.5703
c_DIS	q_CMP	SUM	0.7832	0.6646	0.4430	0.4917
c_DIS	q_CMP	MNZ	0.7816	0.6637	0.4428	0.4915
c_PRF	q_SGL	—	0.7770	0.6778	0.4568	0.4959
c_PRF	q_CMP	MAX	0.8648	0.7786	0.6190	0.6677
c_PRF	q_CMP	SUM	0.7983	0.7017	0.4917	0.5339
c_PRF	q_CMP	MNZ	0.7983	0.7017	0.4917	0.5339
Filtering Task: LM						
c_INI	**q_SGL**	—	**0.7627**	**0.5750**	**0.3270**	**0.3915**
c_INI	q_CMP	MAX	0.7595	0.5689	0.3207	0.3801
c_INI	q_CMP	SUM	0.7519	0.5953	0.3581	0.4121
c_INI	q_CMP	MNZ	0.7519	0.5954	0.3581	0.4121
c_DIS	q_SGL	—	0.7386	0.6081	0.3726	0.4235
c_DIS	q_CMP	MAX	0.8057	0.6777	0.4637	0.5253
c_DIS	q_CMP	SUM	0.7752	0.6438	0.4112	0.4643
c_DIS	q_CMP	MNZ	0.7730	0.6427	0.4110	0.4640
c_PRF	q_SGL	–	0.7421	0.6176	0.3867	0.4329
c_PRF	q_CMP	MAX	0.8421	0.7171	0.5110	0.5706
c_PRF	q_CMP	SUM	0.7780	0.6384	0.4024	0.4544
c_PRF	q_CMP	MNZ	0.7773	0.6382	0.4025	0.4544
Filtering Task: VECT						
c_INI	**q_SGL**	—	**0.6814**	**0.5277**	**0.3110**	**0.3685**
c_INI	q_CMP	MAX	0.6837	0.5237	0.2993	0.3543
c_INI	q_CMP	SUM	0.6980	0.5552	0.3384	0.3915
c_INI	q_CMP	MNZ	0.6979	0.5552	0.3384	0.3915
c_DIS	q_SGL	—	0.6906	0.5844	0.3716	0.4170
c_DIS	q_CMP	MAX	0.7814	0.6735	0.4744	0.5315
c_DIS	q_CMP	SUM	0.7241	0.6183	0.4085	0.4562
c_DIS	q_CMP	MNZ	0.7186	0.6157	0.4081	0.4554
c_PRF	q_SGL	—	0.7971	0.6876	0.4584	0.5030
c_PRF	q_CMP	MAX	0.8757	0.7737	0.5911	0.6462
c_PRF	q_CMP	SUM	0.8222	0.7089	0.4853	0.5319
c_PRF	q_CMP	MNZ	0.8222	0.7089	0.4853	0.5319

5. It is also interesting to note that all these results are also independent on the information retrieval model being used, which implies that our lazy approach could be used independently on the particular search engine implemented in a given parliament[8].

Focusing on the best results, i.e. the combination of c_PRF &q_CMP &CombMAX, we obtain significant improvements with respect to the baseline for those metrics which consider the top 10 retrieval results: 14 %, 10 % and 29 % for the recall@10 and 34 %, 25 % and 47 % for the NDCG@10, considering the BM25, LM and VECT retrieval models, respectively. Similarly, if we consider those metrics that focus on the *ni* top positions, the improvements are even more significant. Particularly we obtain improvements of 67 %, 46 % and 75 % for the R-precision and 84 %, 56 % and 90 % for MAP@*ni*, considering the BM25, LM and VECT retrieval models, respectively. This implies that this approach not only finds out more relevant MPs, but also in better positions.

Finally, if we focus on the time needed to perform the queries, the time for compound queries is 68 % greater than for single queries. But this is due to the fact that we have executed each single subquery sequentially. Taking into account that the different subqueries should be executed in parallel, we could also expect to obtain significant improvements (around 30 %) with respect to the baseline, because of the smaller size of the single subqueries.

4.2 Analyzing Recommending Results

In this case (see Table 5, where the best results have been underlined), the results are worse than those obtained for the filtering task, although they are not directly comparable. Its counterpart in the filtering task might be the case where we perform a single query, *q_SGL*, which includes all the content in an initiative. Nevertheless, we shall distinguish between hand-made and automatic summaries, being considerably worse the hand-made queries. One reason for this behavior is that hand-made summaries have a relative large proportion of terms which are common to many other initiatives in the collection (they are used to place the initiative within the parliamentary workflow), whereas automatic terms have been selected according to their retrieval capabilities. Another reason is that the number of terms in the hand-made summaries tends to be considerably smaller than those in the automatic summaries. In fact, we can observe a clear tendency towards obtaining better results as the number of terms used in the queries increases (from the hand-made short summary, to 25 terms, to 50 terms, to all the terms in the initiative). This suggests that users should be rather wordy when expressing their main points in the petition's proposals.

In general, it seems that the best results are obtained using the initiative-based collection when the number of terms in the query decreases (hand-made and au25). This may be explained because it can be guaranteed that all the selected terms belong to the initiative, whereas this is not the case when querying

[8] Although it is not important for our purposes, the best performing model is BM25.

Table 5. Accuracy metrics for recommending with the selected Retrieval Models. The best results for each retrieval model have been highlighted using underlined values.

Query	Coll.	rec@10	NDCG	MAP	R-prec
Recommending Task: BM25					
hm	*c_INI*	0.5515	0.3871	0.1942	0.2377
hm	*c_DIS*	0.5434	0.3818	0.1831	0.2266
hm	*c_PRF*	0.4849	0.3522	0.1763	0.2119
au50	*c_INI*	0.7733	0.5955	0.3491	0.4126
au50	*c_DIS*	0.7521	0.6148	0.3734	0.4250
au50	*c_PRF*	0.7532	0.6299	0.3978	0.4455
au25	*c_INI*	0.7755	0.5988	0.3561	0.4218
au25	*c_DIS*	0.7350	0.5886	0.3471	0.3994
au25	*c_PRF*	0.7232	0.5882	0.3518	0.4000
Recommending Task: LM					
hm	*c_INI*	0.5331	0.3620	0.1689	0.2085
hm	*c_DIS*	0.5020	0.3560	0.1709	0.2117
hm	*c_PRF*	0.4766	0.3414	0.1674	0.2025
au50	*c_INI*	0.7598	0.5734	0.3247	0.3884
au50	*c_DIS*	0.7331	0.5948	0.3575	0.4112
au50	*c_PRF*	0.7129	0.5739	0.3416	0.3915
au25	*c_INI*	0.7575	0.5708	0.3229	0.3880
au25	*c_DIS*	0.7185	0.5762	0.3350	0.3876
au25	*c_PRF*	0.6892	0.5553	0.3290	0.3800
Recommending Task: VECT					
hm	*c_INI*	0.6390	0.4707	0.2590	0.3137
hm	*c_DIS*	0.5882	0.4273	0.2129	0.2610
hm	*c_PRF*	0.6046	0.4633	0.2474	0.2950
au50	*c_INI*	0.7373	0.5906	0.3491	0.4126
au50	*c_DIS*	0.7218	0.6154	0.3810	0.4304
au50	*c_PRF*	0.7795	0.6383	0.3962	0.4462
au25	*c_INI*	0.7719	0.6028	0.3652	0.4308
au25	*c_DIS*	0.7434	0.5997	0.3594	0.4108
au25	*c_PRF*	0.7467	0.6073	0.3638	0.4154

against the different speeches isolately. However, as we increase the number of terms (au50 and all the terms in the initiative), the profile-based collection becomes preferable.

Nevertheless, we can say that using the right number of terms we can recommend a good set of top 10 MPs (the recall@10 values are almost equivalent to their counterparts using all the terms in the initiative) but with a worse ranking (as the other metrics indicate).

5 Conclusions

In this work we have proposed a system to either filtering parliamentary documents to MPs that could be interested in reading them, or recommending those

MPs that could be more involved in any given topics of interest to citizens. We followed a lazy approach that avoids learning an elaborated profile of each MP. We simply collect all the text of his/her speeches within the parliamentary debates and build an information retrieval system that returns a ranked list of MPs as a response to a query, which is formed from either the document to be filtered or the citizen's topics of interest.

We have carried out experiments with a collection of documents from the Parliament of Andalusia to test several alternative proposals. These proposals are relative to what document collection should be used by the IRS and how the document to be filtered (or the topics of interest of the citizen) should be transformed into a query against the IRS. Our experiments confirm that some of our proposals obtain significant improvements in performance with respect to a baseline approach.

The best results are obtained when we build a lazy profile for each MP, consisting of collecting all his/her speeches into a single large document. For future research we would like to study more elaborated (not so lazy) ways of building the MPs' profiles. We are also planning to tackle the problem from a machine learning perspective, using positive unlabeled learning methods to deal with the problem of having only positive examples to train the model of each MP (the own speeches of this MP). We shall also try to deploy the filtering/recommendation system within the Andalusian Parliament.

Acknowledgements. Paper supported by the Spanish "Ministerio de Ciencia e Innovación" and "Ministerio de Economía y Competitividad" under the projects TIN2011-28538-C02-02 and TIN2013-42741-P.

References

1. Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings. of the 24th Annual International ACM SIGIR Conference, pp. 24–37 (2003)
2. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM **35**, 29–38 (1992)
3. Billsus, D., Pazzani, M., Chen, J.: A learning agent for wireless news access. In: Proceedings of the International Conference on Intelligent User Interfaces, pp. 33–36 (2002)
4. Busby, A., Belkacem, K.: Coping with the Information Overload: An Exploration of Assistants' Backstage Role in the Everyday Practice of European Parliament Politics. European Integration online Papers. vol. 17 (2013)
5. de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Martin-Dancausa, C.J., Tur-Vigil, C., Tagua, A.: An integrated system for managing the andalusian parliament's digital library. Program Electron. Libr. Inf. Syst. **43**, 121–139 (2009)
6. Fox, E.A., Shaw, J.A.: Combination of multiple searches. In: Proccedings of the Second Text REtrieval Conference (TREC-2), pp. 243–252 (1994)
7. Hanani, U., Shapira, B., Shoval, P.: Information filtering: overview of issues, research and systems. User Model. User-Adap. Inter. **11**, 203–259 (2001)
8. Jarvelin, K., Kekalainen, J.: Cumulative gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**, 422–446 (2002)

9. Kim, J., Lee, B., Shaw, M., Chang, H., Nelson, W.: Application of decision-tree induction techniques to personalized advertisements on internet storefronts. Int. J. Electron. Commer. **5**, 45–62 (2001)
10. Lantz, B.: Machine Learning with R. Packt Publishing Ltd, Birmingham (2013)
11. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 587–592 (2003)
12. Lops, P., de Gemmis, M., Semerano, G.: Content-based recommender systems: state of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 73–105. Springer, New York (2011)
13. Marchionini, G., Samet, H., Brandt, L.: Digital government. Commun. ACM **46**, 25–27 (2003)
14. Montague, M., Aslam, J.A.: Relevance score normalization for metasearch. In: Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, pp. 427–433 (2001)
15. Palvia, S.C.J., Sharma, S.S.: E-government and e-governance: definitions/domain framework and status around the world wide web. foundations of e-government. In: 5th International Conference on E-Governance, pp. 1–12 (2007)
16. Pazzani, M., Billsus, D.: Learning and revising user profiles: the identification of interesting web sites. Mach. Learn. **27**, 313–331 (1997)
17. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
18. Soucy, P., Mineau, G.W.: A simple KNN algorithm for text categorization. In: Proceedings of the IEEE International Conference on Data Mining, pp. 647–648 (2001)
19. Wu, S.: Data Fusion in Information Retrieval. Adaptation, Learning, and Optimization, vol. 13. Springer, Heidelberg (2012)
20. Wu, S., Crestani, F.: Data fusion with estimated weights. In: Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, pp. 648–651 (2002)