Automatic Construction of Multi-faceted User Profiles using Text Clustering and its Application to Expert Recommendation and Filtering Problems

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete*, Luis Redondo-Expósito

Departamento de Ciencias de la Computación e Inteligencia Artificial, ETSI Informática y de Telecomunicación, CITIC-UGR, Universidad de Granada, 18071, Granada, Spain

Abstract

In the information age we are living in today, not only are we interested in accessing multimedia objects such as documents, videos, etc. but also in searching for professional experts, people or celebrities, possibly for professional needs or just for fun. Information access systems need to be able to extract and exploit various sources of information (usually in text format) about such individuals, and to represent them in a suitable way usually in the form of a profile. In this article, we tackle the problems of profile-based expert recommendation and document filtering from a machine learning perspective by clustering expert textual sources to build profiles and capture the different hidden topics in which the experts are interested. The experts will then be represented by means of multi-faceted profiles. Our experiments show that this is a valid technique to improve the performance of expert finding and document filtering.

Keywords: Clustering, Content-based Recommendation, Expert Finding, Filtering, User Profiling

1. Introduction

The content of the world wide web is incredibly wide and varied and so one common search task is to look for people that can help us with a particular problem. For example, we might search for a doctor to treat a specific illness, a builder to repair a leaking roof, or a politician to discuss a local problem with so that solutions may be found. This type of information search is set in the broader field of expert finding [3] whereby users find experts in a given area. For this task to be successful, it is necessary for experts to be represented in

*Corresponding author. Tel.: +34958243196;

Email addresses: lci@decsai.ugr.es (Luis M. de Campos), jmfluna@decsai.ugr.es (Juan M. Fernández-Luna), jhg@decsai.ugr.es (Juan F. Huete), luisre@decsai.ugr.es (Luis Redondo-Expósito)

some way in the retrieval system. The most specialized and accurate way is to consider experts' profiles as these store the most representative keywords to define their areas of expertise. These profiles would be built by considering the documents that best represent the expert: for example, for scientists, this would be their journal or conference publications; for writers, their published books; for programmers, the source codes they have written; for lawyers, the court cases they have worked on; and for politicians, their interventions in parliamentary sessions.

With all of these documents, a system could automatically build expert profiles by selecting the best keywords for the expert's fields of expertise. This source of information would then be used by the expert finding system to match the user's information requirements represented in the form of a query. There are basically two main problems related to finding relevant people where profiles are used:

- Given a group of experts or professionals, the problem consists in returning the most suitable ones that could fit a need expressed by a user (usually in the form of a short query). In this case, only the top-ranked ones will be recommended. This is considered to be an expert-finding problem or, more broadly speaking, content-based recommendation [36]. In this case, we only need the highest ranked experts as these are most relevant to the query.

- When a new document reaches the system for the first time (a situation modeled with a long query), the aim is to decide which experts should receive the document. This is a filtering problem [22] and here the aim is to find every relevant person irrespective of their ranking.

Although both of these problems might well be regarded as "*the two sides of the same coin*" [4] and tackled with a similar approach, in this paper we shall show that differences do exist between them in terms of how they are both formulated and their solutions.

In this paper, we shall consider that the expert's field of expertise is not normally limited to a single subject: a scientist, for example, although specialized in information retrieval, might also have published papers along different research lines (e.g. retrieval models, personalization, recommender systems, etc.) or a politician might sit on three different parliamentary committees (e.g. agriculture, environment and economy) with interventions connected with these areas. If a single profile were built from all of the experts' documents, all of their topics of interest would be mixed up in it. This might result in more general topics taking precedence over more specialized topics and so the profile would not correctly reflect the expert's interests and might mean that they are not found when a specific topic is searched for. One solution might therefore be to consider that a profile is seen not as a monolithic but as a multi-faceted structure comprising other profiles or subprofiles, with each relating to different topics. In this way, the politician would therefore be represented by three subprofiles.

Along these lines, the authors of this paper have undertaken research to find relevant people in a parliamentary setting. In an initial approach, profiles were built for Members of Parliament (MPs) from their parliamentary speeches which could then be used to find relevant MPs [16]. Their profiles were created by considering all of their interventions to build a monolithic profile for each MP. Since many of the MPs' speeches are from specialized parliamentary committees, in [15] compound profiles were considered whereby each MP could have various subprofiles according to their interventions on the corresponding committees to which she/he belongs. This paper demonstrated that this method of organizing user profiles is much more interesting for the recommendation problem both in terms of profile performance and interpretability.

In this paper, we go a step further because our aim is to determine whether the use of machine learning techniques (and more specifically clustering) could enable the different topics that users are interested in to be automatically discovered and subprofiles to be built on the basis of these. This automatic discovery of topics (groups) would be particularly useful when there is no explicit association of documents or if there is, it is not the best one for optimal performance in recommendation or filtering tasks (topics that should be separated are grouped together in the same subprofile), something which is quite common in a parliamentary context. For example, if we were to consider a parliamentary committee that was created for political reasons to simultaneously cover the three areas of agriculture, livestock and fishery, then all MP interventions on this committee would be included in the same subprofile although they might represent different topics. In addition, the committee structure usually changes with each term of office, and so clustering the MPs' interventions according to these commissions would provide at any given time a topic distribution that depended on organizational political decisions. Finally, the cold start problem at the beginning of a term of office, whereby no committees exist yet, would be reduced by considering the clustered topics learnt from the previous term.

In this paper we show how clustering is a suitable technique for discovering hidden topics from documents and creating compound profiles to represent user interests. Our experimental results also show how clustering techniques may be successfully applied to expert recommendation and filtering problems to build multi-faceted profiles, where each subprofile is obtained from the documents that are relevant to a user and which are grouped together. These two problems can be solved from a unified perspective because conceptually in both contexts, given a query, the result is a ranking of expert users to be recommended or to recommend to. We have also investigated two ways of applying clustering to the set of documents: a global approach, where clustering is carried out by considering all the experts' documents; and a local one, where clustering is only performed with each expert's documents.

In order to describe how clustering is applied to these problems and its performance, this paper is organized as follows: Section 2 presents introductory information about user profiling and clustering in order to contextualize the rest of the paper; Section 3 contains the core of the article and describes the clustering proposal for building subprofiles; Section 4 describes the experimental

design and the corresponding results and discusses the main findings; Section 5 reviews the state of the art, presents similar approaches and examines the differences between these and our proposal by highlighting our contributions; and finally, the last section outlines our main conclusions and future lines of research.

2. Preliminaries

Given that the context of this paper is to combine the construction and use of profiles for information access and the application of clustering methods to more accurately organize such profiles, in this section we shall present some concepts and techniques related to these two topics and their combination. Section 5 will present a detailed review of the state of the art.

2.1. User Profiling

A profile could be defined as a representation of a user model, storing the user's basic information (e.g. age, gender or location), knowledge, background and skills, behaviour and interaction, contextual information, interests or preferences and intentions [54, 18]. The process of learning a profile is known as user profiling and is based on collecting information explicitly (users express their interests or preferences unequivocally [19]) or implicitly (a system is in charge of automatically detecting the information items of interest to the user by basically analyzing browsing data).

This paper focuses on profiles that mainly express interests so an adequate method is needed to represent them both efficiently and effectively. Gauch et al. in [19] consider that profiles could generally be represented by keywords, semantic networks or concepts. Intelligent techniques based on machine learning and data mining, meanwhile, are also applied to represent user models [54]. Focusing on keyword-based profiles, they store a list of relevant words extracted from the sources used to build them (documents, web pages, textual descriptions of any type of items, etc). These keywords or terms are weighted in order to reflect their importance for the user and usually modeled as weighted vectors (e.g. by using a TF-IDF weighting scheme [36]). Interests may also be expressed as abstract concepts rather than keywords. More elaborate profile representations that are built by combining different elements (e.g. topics and keywords) will be discussed in Section 5. Although knowledge-based profiles can be obtained (possibly as a human readable representation of user interests), they are not successful for recommendation or filtering problems particularly when it comes to documents that represent speeches and oral discussions.

Profiles are considered basic tools for user adaptation in a wide range of fields in computer science [18] and more specifically, [54] indicate various domains relating to information access. Taking into account the context of this paper, these include personalized information retrieval [20], recommender systems [6] and expert finding [35].

2.2. Clustering

From a general point of view, the main purpose of cluster analysis is to attempt to find a common structure over the instances of an unlabeled data set in order to split them into groups (clusters) with similar characteristics [31].

Of all the various existing clustering techniques [50, 55, 17], we should highlight two main families. The first of these is connectivity-based clustering or hierarchical clustering [31, 51, 24, 64]. This builds a distance tree (or dendogram) to represent the fact that items in the same branch are more similar than items in other branches according to how close they are. This first family is divided into two different categories according to how the dendogram is built: the agglomerative approach [53], where each instance belongs to an independent cluster at the beginning and pairs of similar clusters are combined recursively in the same way as the agglomerative nesting algorithm (AGNES, [31]), and the divisive approach [27], where all the instances start in a unique cluster which is separated recursively into two different groups according to similarity as in divisive analysis clustering (e.g. DIANA [31]).

The second family is centroid-based clustering. In it, the different clusters are shaped around a middle point which is not necessarily an instance of the data set and each item is assigned to the cluster whose middle point is nearby [37, 48]. We shall focus on two different methods to compare the behaviour of the data in different approaches. The K-Means [42, 61] algorithm works by splitting n instances into k different groups and assigning each instance to the group with the nearest mean iteratively and recalculating the group mean point after each iteration. The PAM [31, 41] algorithm function is also similar to the previous one although the middle point of the clusters in PAM is an instance which represents the cluster median.

In addition to this set of classic clustering methods, we may find in the literature other techniques that, not being exactly clustering algorithms, try to capture the underlying semantic of the data and can be adapted or applied to this problem. A first example, in the context of text document collections is Latent Dirichlet Allocation (LDA) [5, 8], which is an algorithm that is mainly used in natural language processing. LDA is a three-level hierarchical Bayesian model. It finds the latent topics from a document collection and assigns a probability distribution of topics to each document and also a probability distribution of terms to each topic. Other example is the Self-Organizing Maps (SOM), which are an effective tool that provides a data visualization of high dimensional space by reducing the dimensions of the data to a low (typically two) dimensional map. SOM implements an artificial neural network that is trained with an unsupervised data set with the objective of condensing all the information of the train set, while the most important topological and metric relations among data are preserved, creating some kind of abstraction of the input space [32]. Table 1 summarizes the main characteristics of the six algorithms considered. For a more detailed study of many clustering algorithms, including their advantages and drawbacks, see [60].

In data clustering analysis, an important problem is to establish the number of clusters and how to calculate it. There are many ways to estimate the number

Table 1: Main characteristics of the algorithms considered.

Algorithm	Family	Type
AGNES	hierarchical	agglomerative
DIANA	hierarchical	divisive
K-MEANS	centroid-based	based on mean
PAM	centroid-based	based on median
LDA	topic model	Bayesian network
SOM	neural network	competitive learning

of clusters that best fit the data set. It is highly common in well-known problems to naturally determine the number of clusters in order to obtain a number of well-defined groups but in other cases this is very difficult because there are no clues about this number. More specifically in text databases, an alternative approach to determine the number of cluster is to consider the values of n (the total number of documents), m (the total number of terms) and t (the number of non-zero entries in the respective document-term matrix). The number of clusters k is then defined as $k = mn/t$ [9]. Another outstanding approach to determine the value of this parameter is to calculate it with the general and effective method $\sqrt{n/2}$ [31].

With respect to the evaluation of the quality of the clustering process, typical evaluation measures try to maximise the intra-cluster similarity, i.e., documents placed in the same cluster must be very similar among them, and minimise the inter-cluster similarity, i.e. documents placed in different clusters must be very dissimilar. This is the case of the well known Silhouette index [57], which computes the average distance of a given object with the objects of the nearest cluster and subtracts the average distance of an object with respect to the elements from its own cluster (averaged for all the objects). Other example is the Davies-Bouldin index [14], which is the ratio between the within cluster distances and the between cluster distances (averaged as well). It identifies how compact and well separated are the clusters. These are known as internal validity measures because they are computed only with the information of the dataset and the resulting clustering. The other alternative is to perform an external evaluation that depends on the application domain. In those cases where clustering is only part of the system being built, it is important to evaluate how the clustering algorithm affects the global behaviour of the system [13]. As this is our case, the clustering quality will be indirectly measured through the quality of the obtained recommendations using standard measures in this Information Retrieval (IR) field (see Section 4).

3. Building multi-faceted profiles by clustering documents

As we mentioned in the introduction to this paper, since a user might be interested in a number of different topics and their profile consists of a set

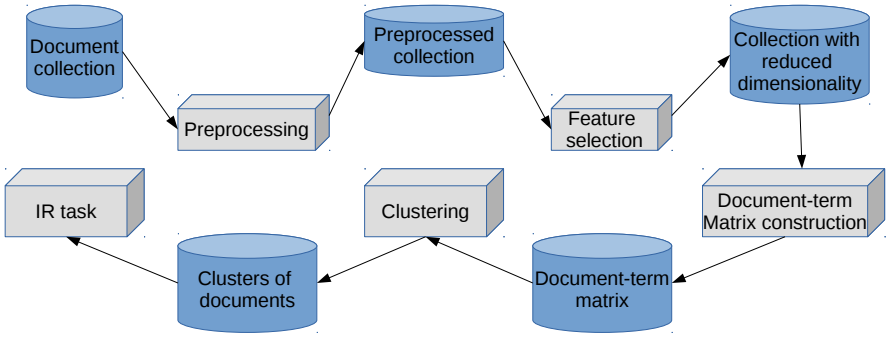

Figure 1: Steps in the text clustering process.

of concepts or topics comprising weighted terms, we could in turn say that the profile is multi-faceted since it attempts to capture the different facets contained in the set of documents associated to a user. In this paper, each facet or concept comprising a profile will be called the subprofile. These multi-faceted profiles are the opposite of monolithic profiles where the underlying topics are not made explicit.

In most situations, the concepts are hidden, i.e. they are implicit in the set of documents. This means that a process to automatically extract or learn them is required. In our case, we have applied clustering analysis. The idea is to cluster the sets of documents to obtain k groups of documents.

3.1. Document Clustering

When the objects to be clustered are texts, as it is our case, this process is called Document Clustering. The first time this machine learning technique was used in IR was more than 40 years ago, with the aim of improving the efficiency of the retrieval process, originating the *Cluster-based retrieval model* [25]. Once documents are clustered and related documents are placed in the same group, given a query submitted by a user, this is confronted to the representatives of the clusters and the system would return the documents belonging to those clusters whose representatives are the closest to the query. The fundamental assumption to apply this cluster-based retrieval model is the *cluster hypothesis*, stated as "closely associated documents tend to be relevant to the same requests" [49].

Figure 1 shows the general process of clustering applied to IR. Given a collection of documents, where clustering is going to be performed on, the first step is its preprocessing, which may consist of tokenization (extracting the tokens or terms, typically splitting at non-letter characters), stop word removal (removing the most common words in the collection, as function words) and stemming (removing word suffixes and leaving the words in their lexical stems). The next step could be the reduction of dimensionality of features (the terms), because we are dealing with a high-dimensionality problem [62], typically removing very

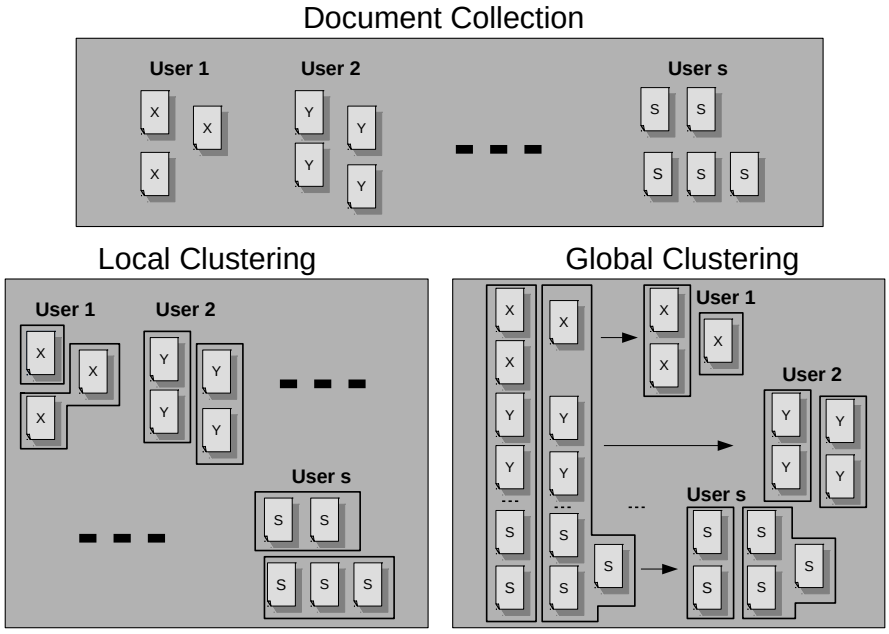

Figure 2: Local versus global clustering.

infrequent terms (those appearing in less than a given percentage of the documents). The construction of the document-term matrix is the next process of the pipeline. The rows corresponds to the documents in the collection and the columns to the terms. The documents are therefore represented by vectors containing the different terms in the collection for the columns. If a document contains a term, then in the corresponding cell there will be a weight that reflects the importance of this term in that text (typically using the TF-IDF scheme) and 0.0 otherwise. This matrix, which is usually very sparse, will be the input of the clustering algorithm, as well as the number of clusters to generate. As output it will offer a partition of the corpus in such number of clusters. and within each one, there is high similarity between all the documents (we could say that all the documents in the clusters deal with the same topic) but low similarity with the documents from other clusters. These clusters could be applied in lots of IR tasks [58], for example, document organization and browsing, text summarization, document retrieval, etc.

3.2. Global and Local Approaches to Clustering

For the purposes of creating user profiles based on the content of their documents, we could consider two approaches for clustering their documents. The first is a local approach and finds the underlying document groups locally for each user, i.e. by only considering their documents. The alternative option is global because it performs the clustering process with all the documents from

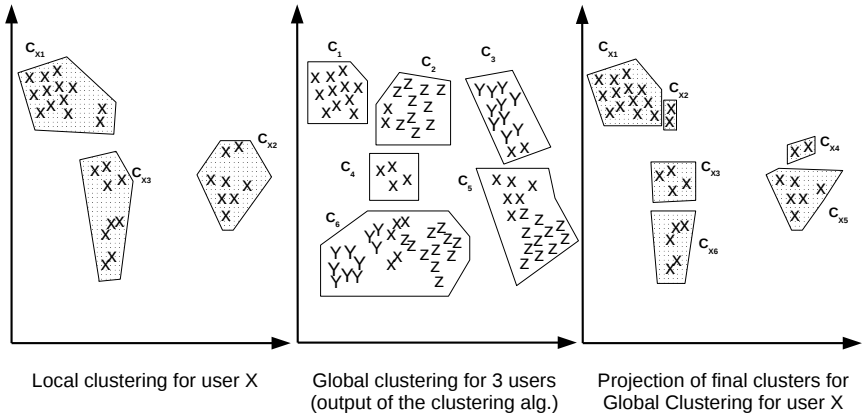

Figure 3: Example of local and global clustering.

every user. The first approach captures a specific user's topics whereas the second attempts to find common concepts that are usually shared by every user. This means that in local clustering, the learned groups are exclusive for each user and therefore only contain documents for that user. In global clustering, the clusters will contain documents from different users. A specific cluster for each user will therefore be obtained from this global clustering by grouping the documents within each global cluster that belong to the given user. Figure 2 illustrates these two approaches.

It should be noticed that in the local approach, for a given user the number of instances is equal to the number of documents associated to her/him. The clustering process is repeated for each user in the system so all the users will obtain their own clusters. In the global method, on the other hand, the number of instances is the number of documents in the system and the clustering algorithm is executed just once.

The left-hand side of the graph in Figure 3 shows the arrangement of all of user X's documents and how they are grouped into local clusters of similar documents. From this aggregation, three subprofiles will be built for the user. In terms of the global approach where the documents of all the users (X, Y and Z) are incorporated into the clustering algorithm, the central graph shows the hypothetical groups found. The clusters c_2, c_3, c_5 and c_6 are heterogeneous in the sense that they integrate documents from different users. If we again focus on user X, the number of profiles to be built following this global approach will depend on the number of clusters the documents belong to. On the right-hand side of the graph, we can see that new clusters are considered for X and so the final number of clusters for X is 6, and this will therefore be the associated number of subprofiles for this user.

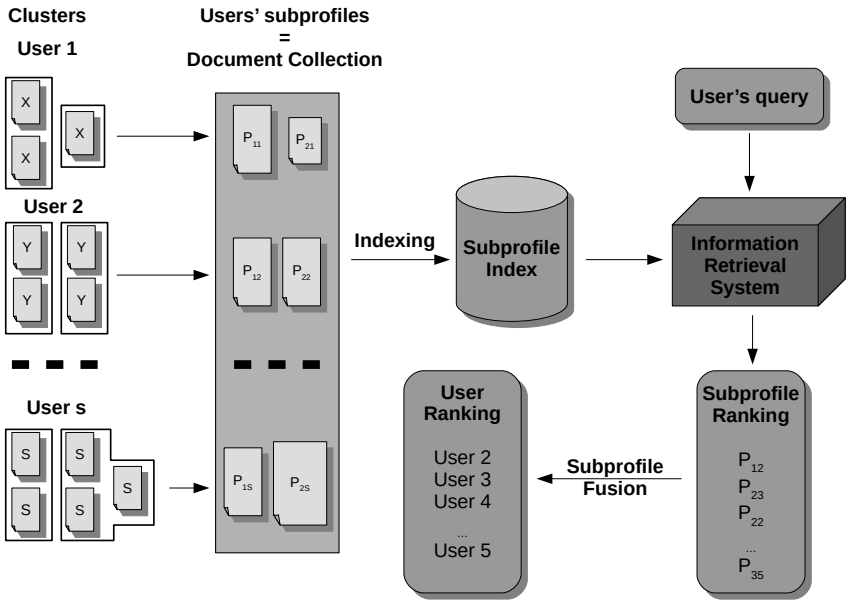

Figure 4: Building subprofiles from the clusters and feeding the IRS.

3.3. Building Subprofiles from the Clusters

In both the local and the global approaches, the final output of the non-supervised process is an association of each document from the given user to a cluster. All the documents of a user which are grouped together in the same cluster are supposed to deal with the same concept. We are then going to build a subprofile from each of the clusters associated to the user. In order to do so, for each user and for a given cluster, a "macro-document" is created by compiling all the documents included in that same cluster. This document will correspond to a subprofile. A new document collection is generated containing all the subprofile documents from all the users. This will be indexed for use by an information retrieval system (IRS). When a query is submitted to the system, it returns a ranking where different subprofiles from the same user may be distributed across it. As we are recommending experts, the final ranking must be composed of users, so it becomes necessary to use some fusion strategy to compute a final score for each user, considering all their different subprofiles in the ranking. Figure 4 illustrates this process.

4. Evaluation

This paper addresses the general problem of finding people but our evaluation will focus on a parliamentary setting. The basic objective is to find relevant

Members of Parliament (MPs) given a query formulated by a citizen or to determine which MPs might be interested in reading a new document received by the system. In order to do so, we have opted to represent MP interests by means of a profile that will be constructed on the basis of their interventions in the political initiatives presented at the public parliamentary sessions. More specifically, let us take into account the fact that an MP might sit on a number of committees and that these are smaller in terms of the number of MPs involved and cover more specific topics. Since the user might be interested in several political topics (e.g. agriculture, education, economy, etc.), the aim is to create subprofiles for each MP to represent the MP's interest in these different topics.

The general objective of this evaluation is to determine whether text clustering is a good tool for automatically identifying the different topics of interest to a user and whether it might be useful to recommend experts to them and filter information for them. In order to achieve this, we propose that the following specific research questions be answered by means of the evaluation described in this section:

- RQ1: Is text clustering an appropriate technique to automatically extract the topics in which a person is interested by considering the particular features of the parliamentary context?
- RQ2: Do filtering and recommendation tasks benefit from clustering-based subprofiles?
- RQ3: Is there any difference between building the clusters locally and globally?
- RQ4: Is the number of clusters relevant for recommendation quality?
- RQ5: What are the best clustering algorithms for these tasks?

In this section, we shall therefore describe the experimental design and also the results of the experiments that have been conducted in this evaluation stage.

4.1. Test Collection

The dataset that we have used for the experiments is the collection of Records of Parliamentary Proceedings from the Andalusian Parliament in Spain and more specifically those that belong to 8th Term of Office[1]. This has been organized around the initiatives discussed in committee and plenary sessions containing a total of 5258 records with 12633 interventions. There are 26 different committees and a total of 132 spokespersons. For experimental purposes, we have selected only those MPs with at least 10 interventions.

[1] Available from `http://irutai2.ugr.es/ColeccionPA/legislatura8.tgz`

4.2. Overview of the Recommender and Filtering System

In order to recommend MPs given a citizen's query or a document to be filtered, we have used the open source Apache Lucene Library[2], implementing the well-known BM25 model as a retrieval model [29]. For each MP_i, the input of the indexer is the set of their subprofiles. For example, the documents to be indexed for MP_5 are the subprofiles for three clusters, c_1, c_2 and c_3 (therefore 3 in total, called $MP_5\_c_1$, $MP_5\_c_2$ and $MP_5\_c_3$, respectively). The terms contained in these are filtered, the stop words removed, and reduced to their roots using the stemmer implemented in the Lucene Spanish Analyzer. Any term occurring in fewer than 1% of the interventions is then removed. Given a query, a ranking of MP subprofiles is given as output. However, as the final objective is to rank MPs according to their relevance to the query, the original ranking is filtered by considering the $CombLgDCS$ method presented in [15]. This strategy calculates a single score for each MP_i by aggregating the different scores of their subprofiles but logarithmically devalued according to their positions in the ranking. The formula is the following:

$$score(MP_i, q) = \sum_{MP_i\_c_j} \frac{s(MP_i\_c_j)}{\log_2(rank(MP_i\_c_j) + 1)}, \quad (1)$$

where MP_i is an MP, $MP_i\_c_j$ is a subprofile in the ranking of this politician, $s(MP_i\_c_j)$ denotes its score value (similarity between the profile and the query q) and $rank(MP_i\_c_j)$ is the position of the $MP_i\_c_j$ subprofile in the ranking.

Once the scores have been computed for every MP, they are ranked accordingly.

4.3. Clustering Algorithms

In our experiments, we have tested the R implementations of the following clustering algorithms: AGNES and DIANA as hierarchical methods (agglomerative and divisive, respectively), K-Means and PAM as centroid-based methods, and finally latent Dirichlet allocation (LDA) and Self Organizing Maps (SOM) as generative statistical model-based and artificial neural networks-based methods, respectively. These algorithms have been selected due to the fact that they are state-of-the-art clustering methods or have been used in the clustering process.

Both centroid-based and hierarchical methods use cosine dissimilarity to compute the distance between individuals. In terms of the LDA algorithm [5] used for clustering, once the algorithm has found the distribution of topics for all the documents, each document is assigned to the cluster associated to its most probable topic. With respect to SOM, it can also be used in order to group similar data together. Once the SOM output is obtained, and each document is associated to a neuron, there is a set of weights vectors which represent the position of the neurons in the discretized space of the data and those vectors

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

are grouped in function of their similarity using any clustering method, thus creating clusters of similar instances of the real data that are attached to the clustered neurons. In our case we have used SOM in combination with the K-Means algorithm (noted as SOM-KM) as it has been found as a state-of-the-art association in general clustering tasks [44, 43, 30].

4.4. Selecting the Number of Clusters

As we have already mentioned, the number of clusters, k, given as the output is an important issue in any problem where clustering is applied. The ideal situation is the automatic selection of the best possible value but this is not easy.

In our experimentation, we have tried different approaches, where k is fixed or is computed automatically by taking into account some collection-dependent data. More specifically, we have conducted experiments with the following alternatives:

- $k = \#Com \Rightarrow$ For global clustering, this represents the number of committees in the eighth Term of Office of the Andalusian Parliament, i.e. 26. For local clustering, this number is specific for each MP, and is the number of committees in which each MP has participated: 6.02 committees on average with a standard deviation of 4.52. The objective of setting this value to k is to determine the degree to which the clustering algorithms are able to reproduce the groups of parliamentary initiatives given by the official committees, which is considered as the ground truth.

- $k = m * n/t \Rightarrow m =$ number of terms in the Andalusian Parliament collection; $n =$ number of interventions in the collection; and $t =$ number of non-zero entries in the document-term matrix. This is applicable to both clustering approaches, although the values of m, n and t will depend on the corresponding type. In the case of global clustering, m is 4208; the total number of MP interventions (n) is 10025 (80% of the total number of interventions (the training partition) and $t = 1,702,296$. For local clustering, these numbers vary because they depend on the number of each MP's interventions, but on average, $m = 3427.45 \pm 2056.15$, $n = 58.11 \pm 58.55$ and $t = 12106.66 \pm 12064.64$. The final value for k for the global approach is $k = 24$, and for the local approach, the average is 15.85 ± 9.67.

- $k = \sqrt{n/2} \Rightarrow$ For global clustering, this value is 70, computed by considering $n = 10025$ (80% of the total number of interventions –the training partition), while for the local one, as the number of interventions of a given MP is specific for each politician, the mean value is 4.25 ± 2.60.

4.5. Experimental context

The set of initiatives is randomly partitioned into a training set (80%) and a test set (20%). The training set is used to build MP subprofiles starting with the

clusters obtained and the test set is used for evaluation purposes. This process is repeated five times, and in this paper the reported results are the average values. In other words, we use the repeated holdout resampling method.

We shall use the content of the initiative (full text) as the query for the filtering process (in this case, our aim is to distribute an initiative to any MP that might be interested), and the initiative title for the case of the MP recommendation approach (the aim is to find an MP to talk to, for example, so we might want to obtain the highest ranked relevant MPs). In both cases, and focusing on relevant judgments, since the objective is to find MPs who might be familiar with the topic, the ground truth for each query will only comprise those MPs who participate in its corresponding initiative. Since it is quite reasonable to assume that an initiative will also be relevant and of interest to other MPs who may not have participated in it, we could say that this is a rather conservative assumption to evaluate, particularly for the filtering task.

Given a query, the search engine will return an MP ranking. In order, therefore, to measure the quality, we will use the well-known precision and recall metrics, focusing on the top 10 results (p@10 and r@10, respectively). We will also consider normalized discounted cumulative gain [26] (ndcg@10) in order to consider the ranking position of the relevant documents.

In order to ascertain whether learning the subprofiles is a good approach for representing the MP profile, we have opted to compare the results with three different baselines:

- A single profile for each MP (monolithic profile). From all of the MPs' interventions on all of their different initiatives, only one profile is built for them. This profile will contain all the topics in which they are interested. We could say that this is the case where $k = 1$.

- Several subprofiles are built for each MP according to the committees they are involved with (committee-based subprofiles). Each MP will have different associated subprofiles by considering their difference committee interventions. The committee interventions will be the input for building the corresponding subprofile. From a practical point of view, if a given MP has participated on k committees, their profile will comprise k subprofiles.

- One subprofile for every initiative in which an MP has participated (intervention-based subprofiles). This is the extreme case where each MP's interventions on an initiative will comprise its own subprofile. The number of subprofiles associated with an MP will therefore be the same as the number of her/his initiative interventions.

The underlying idea behind these baselines is to have two extreme situations (i.e. one profile for each MP or as many as the number of their interventions) and an intermediate one, where the number of subprofiles is established by the committees on which they participate. The expected situation would be that the MP recommendation and filtering tasks would perform better with clustering-based subprofiles than those obtained by the baselines.

4.6. Results

In the following sections, we shall present the results of our experiments and answer the following research questions.

4.6.1. RQ1: Is text clustering a suitable technique to automatically extract topics in a parliamentary context?

In order to answer this first research question, we shall show how clusters cover the political topics discussed in the sessions considering both, a qualitative analysis focused on a particular MP and a broad qualitative analysis, focusing on committees.

Individual Qualitative Analysis. This analysis considers an MP from the `Izquierda Unida` party. We selected him because he is a prolific MP (during the 8th Term he spoke in 172 different sessions) covering a wide range of topics (besides 97 interventions in plenary sessions he also participated in 14 specialized committees or working groups, his remaining 75 interventions). So, which are the topics that the MP is 'truly' interest in? We can say that they are related to the committees he participated, but it is common that some topics have more strength than others in the MP's interests. To quantify this idea we can see the second column in Table 2, where we show the size (in terms of percentage of terms) of his different interventions (note that half of the weight is located into Plenary Sessions, where several topics might be discussed). Note that from these data we can see that he is focused on Equality and Social Welfare, Culture and Health (representing the 70% of his interventions in committees, i.e. without considering Plenary Sessions).

Let us consider firstly those situations in which we do not perform any clustering algorithm, i.e., monolithic and committee based-profiles. Focusing on the monolithic profile, we found that it is dominated by terms related to the parliamentary procedures being difficult to identify the topics the MP is interested in, as the word cloud on the left hand side in Figure 5 shows. On the other hand, if we consider committee-based subprofiles, see for example the right word cloud in Figure 5 obtained from the "Gender Equality and Social Welfare Committee", those terms related to the committee dominate the cluster, although common terms in the parliament appear, but with less frequency. Focusing on the large number of interventions in plenary sessions, we do not have any previous association to a given topic, and therefore they are joined in a big profile, exhibiting the same pattern than monolithic-based profiles.

Now, we will focus on the results obtained after applying a clustering algorithm, particularly Global K-Means, being the value of K equal to 26. In this case, all the interventions of the MP (including plenary sessions) are distributed among 14 of the 26 candidate clusters. The size of each cluster (in terms of percentage of terms) is shown in the last column of Table 2. In order to identify the dominant topic of each cluster, a logical approach is to see the most common terms in the cluster, those which have the highest contribution to it, and assign the cluster with the topic they suggest, appearing different situations:

Table 2: Distribution (in terms of profile's size) of the MP interventions in the term of office. The second column shows the 'true' distribution considering the real sessions in the parliament. The third column shows the distribution considering the learned clusters.

Real Distribution		Clustering
Plenary Sessions	0.500	
Committees		
Gender Equality and Social Welfare	0.128	0.286
Culture	0.121	0.151
Health	0.103	0.144
Presidency	0.046	
Tourism and Business	0.018	0.021
European Affairs	0.015	0.052
Public Work and Housing	0.011	0.013
Public Work and Transports	0.010	0.030
Technology, Science and Business	0.009	0.063
Trade, Technology and Science	0.009	
Governance	0.008	
Justice	0.008	
Radio and Television	0.007	0.016
Environment	0.005	
Economy topic		0.139
Gender violence topic		0.066
Labour movement topic		0.007
Education topic		0.007
Young people topic		0.006

Figure 5: Word Cloud representation for the different profiles: left graph shows Monolithic (all the interventions forms a unique profile) and right graph shows a Committee-based profile obtained using the data from the "Gender Equality and Social Welfare Committee".

- It is possible to find a 1-to-1 match between the documents in the cluster and a given committee, as is illustrated in the left hand side of Figure 6 with red words suggesting that the cluster is related to `culture`.
- Also, a committee can be split into different topics, 1-to-n. For example, clustering was able to discover *"gender violence"* as a new topic, as the graph in the right-hand side of Figure 6 shows. The interventions in this cluster are highly related to "Gender Equality and Social Welfare" committee, but clustering was able to distinguish among *"gender violence"* and *"social welfare"*.
- Joining two different committees in one cluster, 2-to-1: The interventions of two highly related committees as "Technology, Science and Business" and "Trade, Technology and Science"[3], are grouped in the same cluster.
- Discover transversal topics, n-to-1: there exist clusters having interventions from several committees, as is the case of the topic of *"economy"*, representing a transversal interest for the MP. This topic includes interventions from plenary sessions and a large number of committees. This reflects that `economy` is a multidisciplinary topic shared by all the political activities, although it has not been stated explicitly.
- In other cases, a global cluster includes only one intervention of this MP, so it can be considered they represent a marginal topic for the MP interest (last three rows in Table 2).

[3]The reasons for the existence of different committees with highly overlapping topics are political. These committees do not overlap in time but any governmental restructuring also causes modifications to the committees associated with certain areas

Figure 6: Word Cloud representation for two different learned profiles using Global K-Means as clustering approach.

Therefore, we can say that clustering was able to identify the topics of interest of the MP, beyond the committees he is a member. Moreover, it can help to distribute his interventions in plenary sessions among the respective topics.

Overall Qualitative Analysis. In this analysis the objective is to graphically determine the degree to which each output of the clustering algorithms is close to or far from the ground truth, in our case, given by the official Parliamentary committees which represent a political division of the subjects discussed by the committees. Note that in this case we include all the MPs and all the committees, so clustering were therefore executed with $k = \#Com$, i.e. 26. In the coloured matrices represented in Figures 7 and 8, we have arranged the 26 clusters into columns, and the committees into rows. The shade of each cell provides information about the percentages of the committees included in each cluster (the higher, the darker). We have selected K-Means and DIANA to represent the behaviour of the other algorithms.

Although Figure 7 represents the Global K-Means, a very similar pattern is displayed by the Global LDA algorithm and we can observe that there is a considerable degree of matching between the different committees and the clusters, i.e. most of the initiatives from the same committee have been assigned to the same single cluster. As mentioned before, in many cases, the clusters have been able to capture the essence of the topics associated to a given committee. This is the case, for example, of Clusters 12 (almost entirely comprising documents from the Health Committee), 2 (associated with the Culture Committee) or 4 (the Education Committee). Also, clustering is able to group together the documents belonging to different committees, which essentially deal with the same topics. This is the case, for example, of the Committee for Public Works and Transport and the Committee for Public Works and Housing, which are mostly included in Cluster 18. In other cases, the same committee is divided into different clusters (for example, see Clusters 23, 25 and 26) related to econ-

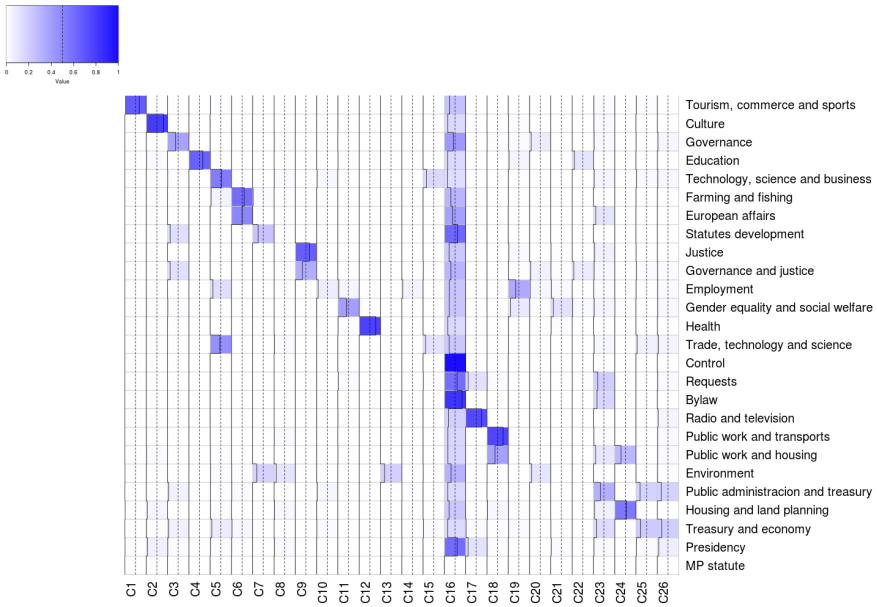

Figure 7: Distribution of committees in clusters given by the Global K-Means algorithm

omy and public administration. Therefore, Global K-Means and LDA capture the subjects from committees with relatively high precision.

Another detected pattern is the one presented by Global DIANA. Figure 8 shows a very different behaviour: considering the committee arrangement given by the K-Means clustering, in order to establish a common comparison point, the diagonal presented in Figure 7 does not occur in the same noticeable way and the distribution of committees in a single cluster does not occur, so initiatives from the same committee are split into different clusters. For example, documents from the Committee for health are distributed in 10 clusters, as happens with many other committees.

We could conclude that the clustering of official Parliamentary documents better reflects the different topics present in the initiatives than the sometimes artificial political division, and this is probably why the profiles resulting from these clustering processes behave better than those originating directly from the committees, as shown in Tables 6 and 7.

4.6.2. RQ2: Do filtering and recommendation tasks benefit from clustering-based subprofiles?

In order to answer this question, and once we have performed the evaluation described in Section 4.5, we present the raw results in Tables 6 and 7 in the Appendix. The first contains the results of the filtering task and the second shows the results of the recommendation task, respectively. In both tables, the first column indicates the type of clustering (T (global or local), the second

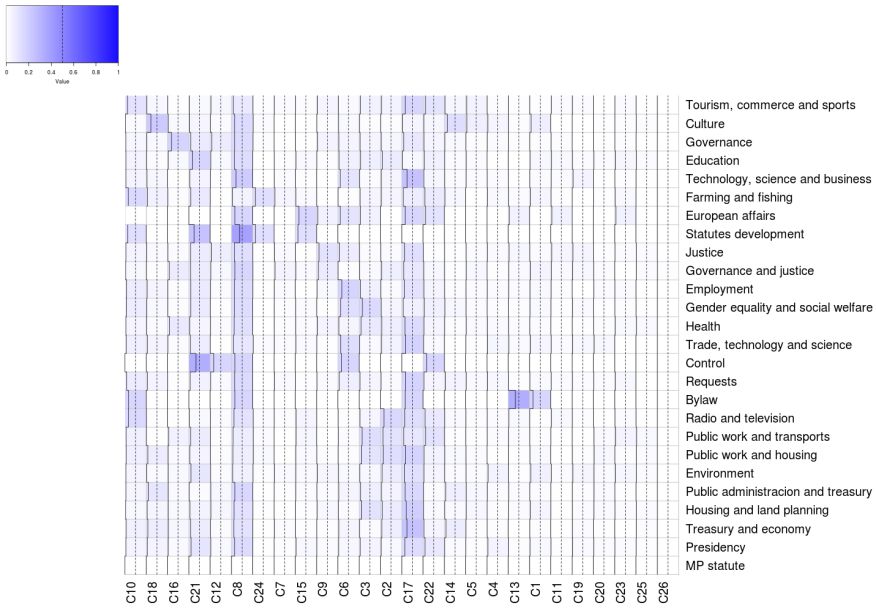

Figure 8: Distribution of committees in clusters given by the Global Diana algorithm.

(*Alg.*) contains the name of the clustering method, and the third the method for computing k, and its value (*#Clusters*[4]). Below the four columns, we have also included the baselines: monolithic profiles (M-Prof) and committee and intervention-based subprofiles (C-SubP and I-SubP), respectively. The columns labeled with *r@10*, *p@10* and *ndcg@10* contain the values of the Precision, Recall and NDCG metrics for the top 10 documents. We have also computed the position of each clustering method and baselines in the ranking resulting from each metric (in the tables, columns *P-r*, *P-p* and *P-ndcg*, respectively).

As considering different measures gives different rankings of methods and baselines, we have attempted to find a way to show a final ranking that would unify these three metrics with a clear idea of the overall performance of the compared methods. We have therefore used Reciprocal Rank Fusion as presented in [12] and which is originally a method for combining rankings from different IR systems to offer a single ranking. In Table 3 we show, for both the filtering and recommendation tasks the obtained RRF values, and the clustering approaches and the baselines are ranked in decreasing order according to this last value. We believe this fairer way of presenting the results facilitates analysis and enables conclusions to be drawn.

In order to illustrate the main trends with respect to the different clustering

[4]For local clustering, as this #Clusters depends on each MP, we show the mean and standard deviation of every MP

Table 3: Values of the Reciprocal Rank Fusion for profiles based on clusters and baselines for filtering and recommendation (Labels of columns: T = Type of clustering; (L)ocal or (G)lobal; k = method for computing the number of clusters; RRF = Reciprocal Rank Fusion value).

Filtering				**Recommendation**			
T	**Algorithm**	**k**	**RRF**	**T**	**Algorithm**	**k**	**RRF**
G	AGNES	$\sqrt{n/2}$	0.0489	L	LDA	$m*n/t$	0.0489
G	AGNES	$\#Com$	0.0464	L	KMEANS	$m*n/t$	0.0484
G	PAM	$\sqrt{n/2}$	0.0460	L	SOM-KM	$m*n/t$	0.0476
G	AGNES	$m*n/t$	0.0457	L	PAM	$m*n/t$	0.0471
L	DIANA	$\sqrt{n/2}$	0.0447	G	AGNES	$\sqrt{n/2}$	0.0450
L	AGNES	$\sqrt{n/2}$	0.0445	G	DIANA	$\sqrt{n/2}$	0.0450
G	DIANA	$\#Com$	0.0430	G	SOM-KM	$\sqrt{n/2}$	0.0448
L	KMEANS	$\sqrt{n/2}$	0.0429	G	LDA	$\sqrt{n/2}$	0.0437
L	SOM-KM	$\sqrt{n/2}$	0.0422	L	DIANA	$m*n/t$	0.0433
G	PAM	$m*n/t$	0.0422	G	DIANA	$\#Com$	0.0418
L	LDA	$\sqrt{n/2}$	0.0417	G	DIANA	$m*n/t$	0.0418
G	DIANA	$m*n/t$	0.0415	G	KMEANS	$\sqrt{n/2}$	0.0413
G	PAM	$\#Com$	0.0413	L	KMEANS	$\sqrt{n/2}$	0.0409
G	LDA	$\sqrt{n/2}$	0.0412	G	LDA	$m*n/t$	0.0400
G	LDA	$\#Com$	0.0404	L	LDA	$\#Com$	0.0392
L	AGNES	$\#Com$	0.0395	L	AGNES	$m*n/t$	0.0392
G	DIANA	$\sqrt{n/2}$	0.0392	L	LDA	$\sqrt{n/2}$	0.0387
M-Prof			0.0390	G	LDA	$\#Com$	0.0378
G	KMEANS	$\sqrt{n/2}$	0.0387	L	SOM-KM	$\sqrt{n/2}$	0.0377
L	AGNES	$m*n/t$	0.0385	L	SOM-KM	$\#Com$	0.0367
G	LDA	$m*n/t$	0.0367	L	KMEANS	$\#Com$	0.0365
G	SOM-KM	$\#Com$	0.0365	**C-SubP**			0.0363
L	PAM	$\sqrt{n/2}$	0.0364	G	SOM-KM	$m*n/t$	0.0360
L	KMEANS	$m*n/t$	0.0363	L	DIANA	$\sqrt{n/2}$	0.0358
G	SOM-KM	$\sqrt{n/2}$	0.0361	G	KMEANS	$m*n/t$	0.0357
G	SOM-KM	$m*n/t$	0.0358	G	SOM-KM	$\#Com$	0.0352
I-SubP			0.0353	L	PAM	$\sqrt{n/2}$	0.0350
L	LDA	$m*n/t$	0.0352	L	DIANA	$\#Com$	0.0349
L	SOM-KM	$m*n/t$	0.0345	**M-Prof**			0.0346
L	DIANA	$m*n/t$	0.0344	G	KMEANS	$\#Com$	0.0344
G	KMEANS	$\#Com$	0.0341	L	AGNES	$\sqrt{n/2}$	0.0335
L	PAM	$m*n/t$	0.0340	G	PAM	$\sqrt{n/2}$	0.0329
G	KMEANS	$m*n/t$	0.0332	**I-SubP**			0.0325
C-SubP			0.0319	L	PAM	$\#Com$	0.0324
L	DIANA	$\#Com$	0.0318	L	AGNES	$\#Com$	0.0321
L	KMEANS	$\#Com$	0.0311	G	AGNES	$m*n/t$	0.0319
L	PAM	$\#Com$	0.0310	G	AGNES	$\#Com$	0.0318
L	SOM-KM	$\#Com$	0.0305	G	PAM	$m*n/t$	0.0311
L	LDA	$\#Com$	0.0304	G	PAM	$\#Com$	0.0308

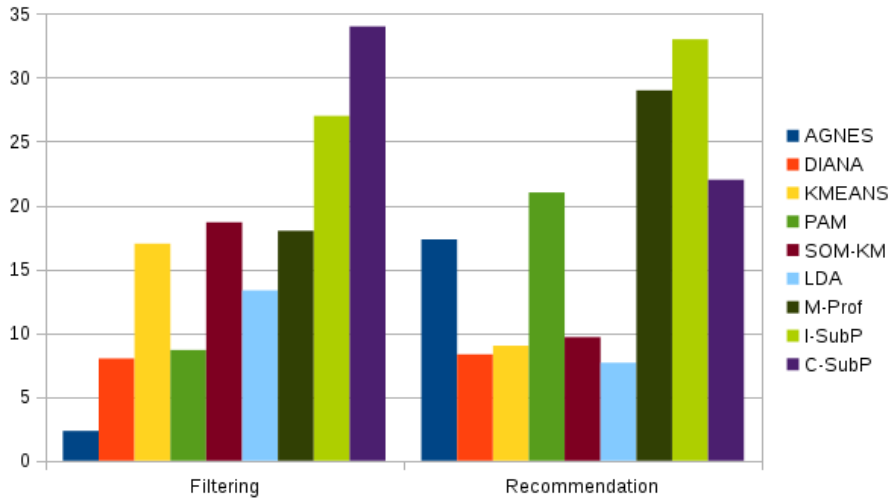

Figure 9: Average position where the best three versions of each cluster algorithm appear (the lower, the better). Positions of the baselines are also included.

algorithms, we have computed the average position where the different versions of each clustering method appear in the ranking of Table 3, but considering only the best three positions (to eliminate the possible bias generated by some poor performing configurations). Figure 9 shows the results (including also the positions of the baselines) for both filtering and recommendation.

First, we shall focus on the performance of baseline methods in both recommendation and filtering tasks, taking into account the aggregated ranking of evaluation measures. The first conclusion is that all of them are placed in the lower half of Table 3. This means that there is a good number of clustering algorithms that outperform them. In terms of performance and in the context of filtering, it is noticeable that monolithic profiles and intervention-based subprofiles are better than committee-based ones. Focusing on the recommendation problem, the best baseline is committee-based subprofiles and the worst is intervention-based subprofiles.

For the filtering problem we consider recall to be the most interesting metric, because we want to send the given document to as many relevant/interested MPs as possible (trying to avoid that an interested reader does not receive the document). Recall, by computing the fraction of relevant MPs that receive the document to the total of relevant MPs, is the appropriate metric to reflect this behavior. If we are able to include for example 6 relevant MPs among the first 10 MPs in the ranking, we have a recall value of 60%, no matter which are the exact positions in the ranking where these 6 MPs are located. However, for the expert finding problem we believe NDCG to be a more valuable metric than recall, because in this case is more important to retrieve relevant experts/MPs in the top positions of the ranking than retrieving many relevant experts. In

the previous example, if the first 4 MPs in the ranking are not relevant (and the following 6 are relevant), from an expert finding perspective the ranking is very bad (the first 4 are not true experts). A much better ranking would be one where the 3 first MPs are relevant and the other 7 are not, although this ranking gets a worse value for recall. This priority towards correct results in the top positions of the ranking is precisely what NDCG emphasizes. In both cases, and focusing on the corresponding metric (see Tables 6 and 7), we notice that monolithic profiles perform worst and the intervention-based subprofiles the best, while committee-based subprofiles are placed between them.

Observing the results we could say that there is quite a good number of clustering algorithms that perform better than the baseline profiles both for recommendation and filtering tasks. In both cases, more than half of the clustering methods outperform the baselines C-SubP and M-Prof (the given political clusters and the option of no clustering at all, respectively), considering the final combined ranking. This number increases to two thirds when we focus on recall for filtering and NDCG for recommendation.

Table 4: Improvement percentages of the best clustering methods for the baselines. $*$ means a statistically significant difference.

	Filtering – Recall	**Recommendation - NDCG**
	Global AGNES $\sqrt{n/2}$	**Local LDA m*n/t**
M-Prof	7.35 % *	14.27 % *
C-SubP	5.06 % *	8.33 % *
I-SubP	2.91 % *	6.72 % *

Table 4 shows the improvement percentage of the best clustering algorithms for filtering and recommendation, considering recall and NDCG, respectively, with respect to the baselines. These percentages are moderate but reflect the fact that clustering is a good alternative for capturing the underlying topics and creating subprofiles. We should highlight that the greatest improvement percentages are achieved for M-Prof, which is good news because it supports the fact that the use of subprofiles by clustering initiatives is better than using a single profile. These percentages are lower when compared with C-SubP but are still relevant, and this supports our hypothesis that political divisions in certain cases may well be somewhat artificial. It is also worth mentioning that the differences between the top clustering methods and baselines are always statistically significant (using a t-test) as occurs with most of the clustering algorithms placed above the baselines.

The general conclusion of this analysis, and to answer the second research question proposed in this section, is that clustering-based subprofiles are a good option for filtering and recommendation tasks since they perform better than baseline approaches, as Figure 9 clearly shows. In our opinion, it is much better and more natural that the fixed committee groups, which are constructed from committees that have in turn been created for political reasons, because they are

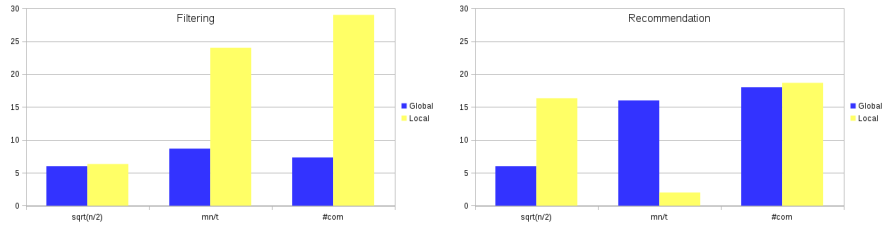

Figure 10: Average position where the best three combinations of global/local clustering and a criterion to select the number of clusters appear, for both filtering and recommendation (the lower, the better).

able to represent the topics that a user is interested in more precisely. Clustering enables the creation of different clusters for topics from the same committee or the combination of facets that might have been artificially separated into two different committees, and it is of course a much better approach than to create a single profile where all the topics are jumbled up.

4.6.3. RQ3: Is there any difference between building the clusters locally or globally?

The next step is to determine whether the best approach is global or local clustering. In the case of filtering new documents, and focusing both on recall and the ranking of the combined metrics, it is remarkable how global clustering is superior to the local alternative: most top clustering algorithms used to build the subprofiles use a global approach, while the local grouping techniques perform worse. This clear distinction in terms of performance is not so evident when we focus on the politician recommendation problem (NDCG and combined metrics). In this case, the global and local clustering algorithms are more mixed throughout the ranking, but it is true that the best clustering algorithms positively employ the local approach. We have computed the average position where the clustering methods using a given combination of global/local clustering and a criterion to select the number of clusters ($\sqrt{n/2}$, mn/t or #Com) appear in the ranking of Table 3, again considering only the best three positions. Figure 10 shows the results, which corroborate our previous findings.

There is one possible explanation for this behaviour: the local approach forces the interventions of a given MP to be distributed among exactly k clusters and this may be a more artificial division in some cases. On the other hand, in a global approach, these MP's documents are probably not assigned to all these k clusters and they could therefore be divided into more cohesive and natural groups. This means that profile sizes in the local approach could be smaller than those built with the global one. In a filtering setting, since the query is the full text of an initiative, it is quite a large query in comparison with the recommendation problem where the query is basically a paragraph with a few lines. Our conjecture is that large queries perform better with large subprofiles as occurs with the global approach in the filtering context. When the query is much shorter (recommendation problem), subprofile lengths are not so

important and so global and local clustering are much more mixed. In terms of the answer to $RQ3$, we would say that the global approach is more interesting for filtering and local somewhat better for recommendation.

4.6.4. RQ4: Is the number of clusters relevant for the recommendation quality?

As we have already mentioned, selecting the number of clusters is the main problem that computer or data scientists must face in the context of clustering. In our case, we have tried three different methods for computing such a value although we do not intend to exhaustively try many methods and find the best but simply to check the sensibility of different values for this parameter. In addition to the well-known $\sqrt{n/2}$ and $m*n/t$, we have also used the number of committees as a kind of baseline for k. In global clustering, the values of $\#Com$ and $m*n/t$ are very close (26 and 24, respectively) and so their results are quite similar, independently of the type of problem and clustering algorithm.

For the filtering problem and with the global strategy, all of these methods combined with any k, including $k = \#Com$, are better than C-SubP for both the recall metric and the combination of metrics and most are better than I-SubP and M-Prof. The best value for k in absolute numbers is $\sqrt{n/2}$. We believe that there is more room for including new subtopics when $k = 70$ than with 26, where these are grouped together in the same clusters, and so this is a more robust value for every clustering algorithm. For the local mode, meanwhile, it is noticeable how the performance of most clustering algorithms is really bad when $k = \#Com$ is applied and even worse than C-SubP. On the other hand, $\sqrt{n/2}$ is again the method that behaves best (see Figure 10). The reason for this may be that in the local case $\sqrt{n/2}$ is the method that generates the lowest mean number of clusters (4.25) and so subprofile sizes are larger and this tendency is positive in the filtering problem.

For recommendation and global clustering, $\sqrt{n/2}$ performs best and is very robust across clustering methods. The performance of $\#Com$ and $n * m/t$ clearly depend on the algorithm but it is generally much worse (for NDCG and the combined ranking). Focusing on local clustering, it seems that $\#Com$ and $\sqrt{n/2}$ do not provide enough space to include the different topics that MPs deal with and more groups are required and so $n * m/t$ is the best alternative, as Figure 10 clearly shows (it performs best and every clustering method achieves the best values with it). Any algorithm combined with $n * m/t$ does in fact outperform C-SubP and the other baselines.

In terms of individual algorithms, AGNES is very robust independently of the k selected for filtering in local and global approaches, and for the recommendation problem, the performance of this clustering clearly varies according to it. For the remaining algorithms, it is not possible to draw such an obvious conclusion since performance varies according to the number of clusters used, the type of clustering and the problem at hand.

By way of conclusion and to answer RQ4, we would say that selecting a good value for the k parameter is important for good performance in the filtering and recommendation problems. We should also mention that we have found k values

that outperform the number of committees. This means that we cannot restrict MPs only to the committees where they intervene.

4.6.5. RQ5: What are the best clustering algorithms for these tasks?

As previous step to answer this question, we have carried out two ANOVA variance tests, with $\alpha = 0.05$, one for the results of recall@10 for filtering and the other for those of ndcg@10 for recommendation, for all the combinations. The conclusion is that there are significant differences among them (p-values of $4,9309E-37$ and $1.1842E-24$, respectively). Therefore, it is important to make a good decision about the clustering method in combination with the local or global approaches and the number of clusters in order to get a good performance in these problems.

The clustering techniques that perform best vary according to the problem at hand (see Figure 9). For the filtering problem, hierarchical clustering techniques work quite well and AGNES, in particular, is the best algorithm in its global clustering version in terms of the four evaluation metrics used. For the expert finding problem, although good results are also obtained by AGNES, the best approaches are Local LDA in most of the metrics, followed by the centroid-based algorithm Local K-MEANS, the SOM-KM approach and PAM. In this case, we could observe how recommendation could be performed with quality using a wide variety of clustering techniques. We have performed another ANOVA test with the top 5 combinations of Tables 6 and 7, again with the recall@10 and ndcg@10 values, respectively, and the result is that there are no significant differences among them (p-values of 0.8154 and 0.9691, respectively). This means that any of them could be selected for these tasks with high confidence of doing a good job. However, we should mention that the performance of the clustering algorithms clearly depends on the value of the k parameter, as we have discussed in the previous section.

Finally, we have plotted the recall@10 (for filtering) and ndcg@10 (for recommendation) values of all the clustering combinations in a graph (Figure 11) in order to graphically discover the combinations with better performance in both tasks. In this plot, we have used the different shapes to represent the global (circle) or local (filled plus sign) clustering. Also, different colors to represent the parameter k, i.e., the number of clusters, being $\#Com$, $\sqrt{n/2}$ and $m * n/t$ represented by green, red and blue, respectively. Finally, the type of cluster is represented by the first letter of the name of the technique: Kmeans, Lda, Diana, Agnes, Som, Pam. We have also included the three baselines (MONolithic, COMmittee-based and INTervention-based), represented with a triangle in the graph.

From this graph we can see that using $\#Com$ as the number of clusters is not a good alternative for filtering (where the worse results has been obtained) and also for recommending. This can be considered as an evidence that the number of committees in the parliament does not match properly with the topics discussed, and therefore to obtain the best results is not essential to know a priori this information. This is an interesting result, since our approach could be extended to other problems where such information is not available.

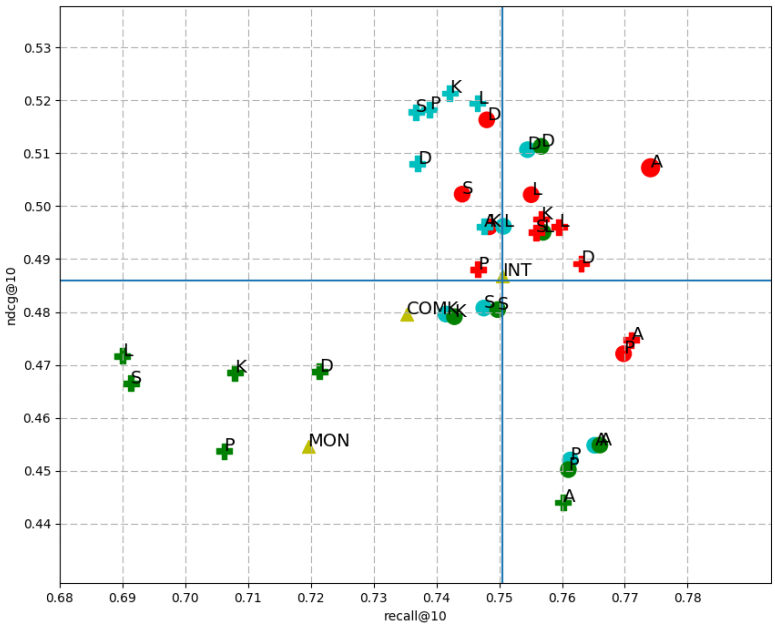

Figure 11: recall@10 vs ndcg@10 of all the clustering combinations plus the baselines.

Also, if we want to find one strategy which fits both filtering and recommendation tasks we propose to focus on those methods which are better than the best baseline (intervention-based) for both problems, i.e., those situated in the top-right area of the graph (delimited by solid lines). In this case, we can see that using a global hierarchical or LDA as clustering algorithm with $\sqrt{n/2}$ as the number of clusters are reasonable alternatives, obtaining the best results with AGNES-Global-$\sqrt{n/2}$. Nevertheless, if we focus only in the problem of recommending, this algorithm seems to be very dependent on the value of the other parameters, being necessary a proper estimation of the parameters. Therefore, if we are looking for an good cluster strategy, suitable for both problems, the most stable algorithms seem to be both LDA and DIANA, particularly using global clustering.

5. Related work

In this section, we shall present relevant published work on the general subject of compound profiles. Although some of these publications do not use clustering, we thought that it would be interesting to include them in this review since we represent multi-faceted profiles in our work and most authors

agree that users would benefit from a richer representation. In order to summarise the papers, we have included Table 5 in the section with the following main features: profile purpose (Purpose); information source used (Inf. Source); clustering algorithm used (Clust. Alg.), if necessary; entities considered (Entity cons.) and their features (Features); and finally, the type of compound profile obtained (Type of profile). We have also included our approach in the bottom row of the table.

A first group of more closely related papers deals with a structured way of representing profiles based on different information sources. In the context of expert finding, [45] creates a structured profile with experts' personal data, expertise and interests, represented by a tree with three sibling nodes, typically containing the terms from each source. A second case of this style is [23], where based on Twitter topical lists, two subprofiles are built for a user: one comprising the tags assigned to the lists to which they belong and the other with the tags from the lists for the user's friends. A third example is presented in [63], where user evidence is represented by different types of objects (Web pages, users, items, queries, etc.), which are clustered in a multi-layered graph, creating cluster connections by applying mutual reinforcement. Finally, in [40], the authors, with the aim of recommending social items, create three subprofiles using as sources the weighted keywords extracted from the user's social items, tags associated to these keywords, and new terms connected by underlying concepts.

Other related papers present approaches where the information coming from one source (typically documents) is organized into different profiles. One first case designed for news recommendation ([21]) proposes a method for representing two faceted-profiles: a long-term subprofile comprising terms and categories from the history of relevant documents; a short-term one, with the same information but created after the first subprofile has been built. A second example is [7], where the profile comprises two subprofiles: the list of terms extracted from positively judged documents, enriched by the terms belonging to the cluster to which the user belongs (after applying K-means to every user) and enriched by Wordnet hypernyms, and the terms from negatively judged documents.

Another type of structured profile is the one based on hierarchies. The paper [52] builds expertise profiles comprising a series of time-based hierarchical profiles, where the nodes are weighted topics. [10] presents a personalization system based on keeping a hierarchy of the user's interests (personal view) from visited web pages. In both cases, the hierarchies are given not learned.

Considering a structure where a profile comprises a list of categories/topics/-concepts which are not interlinked, and each is represented by a set of keywords, we can cite the recommender *Syskill & Webert* [46], the personalizer *Alipes* [59] and the recommender *Webmate* [11]. In these first two cases, the category list is given to the system in one way or another, and not learned automatically as it is in the third system which uses a clustering algorithm. Another example in this category is presented in [33], where category-based subprofiles are created not with documents but with terms from the formulated queries (clustering queries).

The following papers offer a similar profile structure but are automatically learned by clustering: [56] applies a local incremental clustering to generate

topics from clusters for search personalization; *Web Personae* [38] uses a hierarchical clustering on the user's visited set of web pages (local construction) for the same personalization purposes.

In [1], once the terms have been extracted from the information sources, the induced bisecting K-means algorithm is applied to group these terms into semantically related concepts, representing each user (scholars, in this case) with a set of research areas (groups) and characterized by a set of terms. A similar approach is presented in [2], where using a document clustering technique based on community-discovery methods, the authors create groups of tags to represent the users. While this is a local tag-based clustering, we also consider global clustering but with terms as features. [47] essentially presents the same idea as [2], grouping similar documents, but the main difference is that the first does not explicitly represent user interests but uses the structure of clusters to directly recommend scientific articles.

5.1. Differences of our approach with the related work

The first four approaches presented in this review ([45, 23, 40, 63]) in the context of a structured way of representing profiles based on different information sources, differ from our proposal in that they consider various sources of information to build the profiles whereas we only take into account one type and the profile structure is also relatively complex to support this diversity. Another difference is that with the exception of [40], none of the papers is interested in capturing and representing underlying topics as we are. This referenced paper uses concepts that are extracted from an external source and not automatically learnt as we implicitly do. Finally, none of them use clustering to build profiles with the exception of [63].

In the case where information coming from one source (typically documents) is organized into different profiles [21], in our proposal, we do not consider positive and negative documents. In the second reference, [7], the authors apply global clustering at the user level (users are the instances and the terms, the attributes) while our clustering is carried out at the document level and locally (only for the active user).

The main difference with the approaches presented in [52, 10] is obvious as they do not use clustering and we do not use concept hierarchies to represent the profiles. In our case, the concepts are not interlinked.

When focusing on profiles composed of a list of categories, topics or concepts [46, 59, 11, 33], although the profile structure is very similar to the one presented in [33], the main difference is that we use proper clustering algorithms to automatically create the categories (the clusters). These approaches construct the profiles locally. Our proposal also considers the global information of all the users. Additionally, we use the document terms as features whereas in [33] query terms are used.

With respect to papers [56] and [38], which build a similar profile structured by means of clustering, once again, the main difference with our approach is the locality of profile construction. The use of profiles is also another difference

since in these papers, profiles are considered to be personalization tools, whereas in our case, they are used for content-based recommendation. Finally, a third difference is profile selection because these examples use the most relevant profile (only one) while our proposal combines all of these results to find the user to recommend to.

Considering the papers [1, 2, 47], the main difference is that, in our case, the clusters contain documents and we also consider global clustering but with terms as features.

In addition to the differences described in this section, we should mention that our experiments have tested the suitability of different clustering algorithms and different methods to decide on the number of clusters to be used. It is difficult to find any specific published reference as to how the number of clusters should be determined.

6. Conclusions and Further Research

In this paper, we have presented a proposal based on text clustering to automatically build compound profiles of experts to properly reflect the topics in which they are usually interested. Two different but highly related application domains have been considered, namely filtering and expert recommendation. In the first case the task is to decide which experts would be interested in receiving a new document, according to their interests and expertise. In the second case the decision to be made is which experts are more appropriate to satisfy an information need expressed by a user. The specific setting where we have experimentally tested our proposals is political, where the experts considered are Members of Parliament and the information source used to build the profiles are the transcriptions of their interventions when discussing initiatives within the parliamentary debates.

Although these two problems, filtering and recommendation can be formulated in a unified way (given a query, either a document to be filtered or an information need to be satisfied, return a ranked list of experts which are either interested in the document or able to satisfy the information need) and both can be managed using a similar approach, our experimental results suggest that there are some important differences between them. These differences determine that the more appropriate tools for solving these problems within our formulation (type of clustering, local or global, type of clustering algorithm and selection of the number of clusters) are different.

We have proposed two clustering alternatives: a local method and a global method. The local method separately clusters the documents of each expert (i.e. the interventions of each MP), whereas the global method performs a single clustering of the documents of all the experts. In any case we have tested clustering algorithms of very different nature (hierarchical agglomerative and divisive, centroid-based, generative statistical model-based, neural network-based), as well as different techniques to select the number of clusters. Three different baselines have been considered: two extreme cases, a single (monolithic) profile for each MP and one subprofile for each MP intervention, and an

intermediate situation where the subprofiles of an MP are not learned through clustering but each subprofile is extracted from the interventions of the MP in each of the different committees where she participates.

The main conclusions extracted from our experimental results are the following:

- Clustering is generally a good option to discover groups of documents dealing with different topics of interest, even improving situations where these groups are given explicitly and externally.

- Many of the alternatives based on clustering outperform all the three baseline methods for both filtering and recommendation, the differences in performance being statistically significant.

- Concerning the type of clustering, for the filtering problem the choice is clear: global clustering is preferable. However, for the recommendation problem the situation is not so clear, although the four top performing alternatives use local clustering. This different behavior seems to be related to the sizes of the clusters generated by each approach and the fact that in the filtering problem the size of the "query" (the complete document) is normally larger than in the recommendation problem (where the "query" is the information need of a user).

- If we focus on the selection of the number of clusters, we again find differences between filtering and recommendation: for filtering the best method to select the number of clusters is $\sqrt{n/2}$. For recommendation, however, $n * m/t$ performs best.

- Concerning the specific clustering algorithms being used, hierarchical methods (in particular the agglomerative one) work quite well for the filtering problem, whereas LDA, centroid-based methods and SOM are preferable for the recommendation problem. However, it seems that the decision about which clustering algorithm to use is not critical, because we have not found statistically significant differences among the five best performing methods within each problem.

By way of future research, we plan to tackle the problem of how recommendation and filtering problems would be affected when experts are represented by temporary profiles. In this case, short and long profiles would be built for them using clustering techniques. Another line that we would like to explore is the potential capacity of LDA and other topic models [28, 34, 39] for creating good (sub)profiles but by exploiting the semantic perspective in which these algorithms specialize.

Table 5: Summary of the related works on compound profiles.

Reference	Purpose	Inf. Source	Clust. Alg.	Entity Cons.	Features	Type of profile
[45]	Expert finding	Heterogeneous documents	–	Documents	Keywords	Tree with nodes containing keywords
[23]	User modeling	Lists in Twitter	–	–	Tags assigned to lists	Intentional (weighted tags in lists the user follows) and Extensional profiles (weighted tags in lists the friends follows)
[63]	User modeling	Web heterogeneous objects	Probabilistic clustering	Web Objects	Keywords	Multi-layered graph with nodes in different layers representing different type of objects
[40]	Content-based recommendation	Social items from Facebook & Instagram	–	–	Keywords & Concepts	Wikipedia Concepts & Extended keywords
[7]	Content-based recommendation	News	Variation of K-Means	Users	Keywords	Terms in positive documents (and in clusters they belong to), in negative documents, plus WordNet hypernyms
[21]	Content-based recommendation	News	–	News	Keywords	Keywords in observed News, keywords in concepts
[52]	IR Personalization	Web pages	–	User's interest	Keywords	Hierarchy of web pages
[10]	IR Personalization	Web pages	–	Web pages	Keywords	Hierarchy of the user's interests
[46]	Content-based recommendation	Web pages & their categories	–	–	Keywords & Categories	List of categories and the associated terms
[59]	News personalization	News	–	–	Keywords and categories	List of categories comprising three lists of keywords, respectively
[11]	Content-based recommendation	Web pages	–	Keywords	–	List of categories comprising keywords
[33]	Collaborative tagging	Documents	Community Discovery-based	Documents	Keywords	Subprofiles comprising keywords
[56]	IR Personalization	Web pages	Incremental clustering	Web pages	Keywords	List of topics
[38]	IR Personalization	Web pages	Hierarchical clustering	Web pages	Keywords	List of cluster centroids
[47]	Content-based recommendation	Scientific articles	–	Scientific articles	Keywords	Clusters of articles
[2]	Collaborative tagging	Documents	Community discovery technique	Documents	Keywords	Subprofiles comprising tags (extracted from clustered documents)
[1]	Expert finding	Heterogeneous sources	Bisecting K-Means	Different sources	Keywords	Subprofiles of research areas
de Campos et al. (2018)	Content-based recommendation & filtering	Parliament initiatives	AGNES, DIANA, LDA, K-Means, PAM, SOM	Initiatives	Keywords	List of subprofiles, containing weighted keywords

Acknowledgements

This work has been funded by the Spanish Ministerio de Economía y Competitividad under project TIN2016-77902-C3-2-P, and the European Regional Development Fund (ERDF-FEDER).

Appendix

Table 6: Values of the evaluation metrics for profiles based on clusters and baselines for filtering (Labels of columns: T = Type of clustering; (L)ocal or (G)lobal; k = method for computing the number of clusters; $\#Clusters$ = number of clusters; r@10 = recall at 10; P-r = Position in the recall ranking; p@10 = precision at 10; P-p = Position in the precision ranking; ndcg@10 = NDCG at 10; P-ndcg = Position in the NDCG ranking).

T	Alg.	k	#Clusters	r@10	P-r	p@10	P-p	ndcg@10	P-ndcg
G	AGNES	$\sqrt{n/2}$	70	0.7724	1	0.1779	1	0.6549	2
G	AGNES	$\#Com$	26	0.7660	4	0.1754	7	0.6547	3
G	PAM	$\sqrt{n/2}$	70	0.7698	3	0.1767	3	0.6391	10
G	AGNES	$m*n/t$	24	0.7652	5	0.1752	8	0.6543	4
L	DIANA	$\sqrt{n/2}$	4.25 ± 2.60	0.7630	6	0.1766	4	0.6379	12
L	AGNES	$\sqrt{n/2}$	4.25 ± 2.60	0.7710	2	0.1775	2	0.6308	22
G	DIANA	$\#Com$	26	0.7567	12	0.1744	13	0.6509	5
L	KMEANS	$\sqrt{n/2}$	4.25 ± 2.60	0.7567	13	0.1749	9	0.6408	8
L	SOM-KM	$\sqrt{n/2}$	4.25 ± 2.60	0.7559	14	0.1754	6	0.6377	14
G	PAM	$m*n/t$	24	0.7613	7	0.1748	11	0.6366	16
L	LDA	$\sqrt{n/2}$	4.25 ± 2.60	0.7595	10	0.1761	5	0.6303	23
G	DIANA	$m*n/t$	24	0.7545	16	0.1739	16	0.6470	6
G	PAM	$\#Com$	26	0.7610	8	0.1747	12	0.6353	19
G	LDA	$\sqrt{n/2}$	70	0.7551	15	0.1740	15	0.6400	9
G	LDA	$\#Com$	26	0.7570	11	0.1742	14	0.6354	18
L	AGNES	$\#Com$	6.02 ± 4.52	0.7602	9	0.1748	10	0.6164	33
G	DIANA	$\sqrt{n/2}$	70	0.7480	21	0.1723	24	0.6452	7
M-Prof				0.7195	35	0.1724	23	0.6577	1
G	KMEANS	$\sqrt{n/2}$	70	0.7484	20	0.1729	20	0.6377	13
L	AGNES	$m*n/t$	15.85 ± 9.67	0.7476	22	0.1725	22	0.6380	11
G	LDA	$m*n/t$	24	0.7507	17	0.1727	21	0.6269	28
G	SOM-KM	$\#Com$	26	0.7497	19	0.1730	18	0.6246	31
L	PAM	$\sqrt{n/2}$	4.25 ± 2.60	0.7466	24	0.1730	17	0.6277	27
L	KMEANS	$m*n/t$	15.85 ± 9.67	0.7421	28	0.1722	26	0.6377	15
G	SOM-KM	$\sqrt{n/2}$	70	0.7441	26	0.1720	27	0.6365	17
G	SOM-KM	$m*n/t$	24	0.7475	23	0.1730	19	0.6254	30
I-SubP				0.7505	18	0.1687	33	0.6283	25
L	LDA	$m*n/t$	15.85 ± 9.67	0.7465	25	0.1722	25	0.6280	26
L	SOM-KM	$m*n/t$	15.85 ± 9.67	0.7367	32	0.1711	29	0.6339	21
L	DIANA	$m*n/t$	15.85 ± 9.67	0.7370	31	0.1702	32	0.6345	20
G	KMEANS	$\#Com$	26	0.7429	27	0.1712	28	0.6265	29
L	PAM	$m*n/t$	15.85 ± 9.67	0.7389	30	0.1707	31	0.6290	24
G	KMEANS	$m*n/t$	24	0.7415	29	0.1710	30	0.6218	32
C-SubP				0.7352	33	0.1655	34	0.6108	35
L	DIANA	$\#Com$	6.02 ± 4.52	0.7214	34	0.1652	35	0.6131	34
L	KMEANS	$\#Com$	6.02 ± 4.52	0.7079	36	0.1622	37	0.5996	36
L	PAM	$\#Com$	6.02 ± 4.52	0.7062	37	0.1628	36	0.5861	37
L	SOM-KM	$\#Com$	6.02 ± 4.52	0.6913	38	0.1583	38	0.5826	39
L	LDA	$\#Com$	6.02 ± 4.52	0.6900	39	0.1576	39	0.5842	38

Table 7: Values of the evaluation metrics for profiles based on clusters and baselines for recommendation (Labels of columns: T = Type of clustering; (L)ocal or (G)lobal; k = method for computing the number of clusters; $\#Clusters$ = number of clusters; r@10 = recall at 10; P-r = Position in the recall ranking; p@10 = precision at 10; P-p = Position in the precision ranking; ndcg@10 = NDCG at 10; P-ndcg = Position in the NDCG ranking).

T	Alg.	k	#Clusters	r@10	P-r	p@10	P-p	ndcg@10	P-ndcg
L	LDA	$m * n/t$	15.85 ± 9.67	0.6529	1	0.1486	1	0.5195	2
L	KMEANS	$m * n/t$	15.85 ± 9.67	0.6502	2	0.1482	3	0.5214	1
L	SOM-KM	$m * n/t$	15.85 ± 9.67	0.6498	3	0.1482	2	0.5178	4
L	PAM	$m * n/t$	15.85 ± 9.67	0.6481	4	0.1475	4	0.5183	3
G	AGNES	$\sqrt{n/2}$	70	0.6438	5	0.1465	6	0.5065	9
G	DIANA	$\sqrt{n/2}$	70	0.6408	7	0.1459	8	0.5163	5
G	SOM-KM	$\sqrt{n/2}$	70	0.6437	6	0.1470	5	0.5023	10
G	LDA	$\sqrt{n/2}$	70	0.6398	8	0.1459	7	0.5022	11
L	DIANA	$m * n/t$	15.85 ± 9.67	0.6385	9	0.1453	11	0.5080	8
G	DIANA	$\#Com$	26	0.6357	13	0.1445	17	0.5113	6
G	DIANA	$m * n/t$	24	0.6364	11	0.1445	18	0.5107	7
G	KMEANS	$\sqrt{n/2}$	70	0.6380	10	0.1452	12	0.4961	16
L	KMEANS	$\sqrt{n/2}$	4.25 ± 2.60	0.6350	15	0.1450	13	0.4976	12
G	LDA	$m * n/t$	24	0.6342	16	0.1446	16	0.4962	13
L	LDA	$\#Com$	6.02 ± 4.52	0.6352	14	0.1457	9	0.4717	29
L	AGNES	$m * n/t$	15.85 ± 9.67	0.6322	17	0.1443	19	0.4962	14
L	LDA	$\sqrt{n/2}$	4.25 ± 2.60	0.6320	18	0.1442	20	0.4961	15
G	LDA	$\#Com$	26	0.6316	19	0.1440	22	0.4951	17
L	SOM-KM	$\sqrt{n/2}$	4.25 ± 2.60	0.6313	20	0.1441	21	0.4950	18
L	SOM-KM	$\#Com$	6.02 ± 4.52	0.6295	21	0.1449	14	0.4665	32
L	KMEANS	$\#Com$	6.02 ± 4.52	0.6284	22	0.1447	15	0.4685	31
C-SubP				0.6048	38	0.1457	10	0.4795	25
G	SOM-KM	$m * n/t$	24	0.6281	23	0.1435	25	0.4807	22
L	DIANA	$\sqrt{n/2}$	4.25 ± 2.60	0.6262	26	0.1430	27	0.4892	19
G	KMEANS	$m * n/t$	24	0.6278	24	0.1437	24	0.4796	24
G	SOM-KM	$\#Com$	26	0.6254	27	0.1432	26	0.4804	23
L	PAM	$\sqrt{n/2}$	4.25 ± 2.60	0.6247	29	0.1426	29	0.4880	20
L	DIANA	$\#Com$	6.02 ± 4.52	0.6262	25	0.1440	23	0.4687	30
M-Prof				0.6358	12	0.1357	38	0.4546	35
G	KMEANS	$\#Com$	26	0.6248	28	0.1429	28	0.4791	26
L	AGNES	$\sqrt{n/2}$	4.25 ± 2.60	0.6215	30	0.1422	32	0.4747	27
G	PAM	$\sqrt{n/2}$	70	0.6151	33	0.1398	33	0.4721	28
I-SubP				0.5959	39	0.1355	39	0.4868	21
L	PAM	$\#Com$	6.02 ± 4.52	0.6173	31	0.1423	31	0.4537	36
L	AGNES	$\#Com$	6.02 ± 4.52	0.6172	32	0.1423	30	0.4440	39
G	AGNES	$m * n/t$	24	0.6123	34	0.1392	34	0.4548	34
G	AGNES	$\#Com$	26	0.6119	35	0.1389	35	0.4548	33
G	PAM	$m * n/t$	24	0.6097	36	0.1387	36	0.4520	37
G	PAM	$\#Com$	26	0.6089	37	0.1384	37	0.4502	38

References

[1] Amini, B., Ibrahim, R., Othman, M. S., and Selamat, A. (2014). Capturing scholar's knowledge from heterogeneous resources for profiling in recommender systems. *Expert Systems with Applications*, 41(17):7945 – 7957.

[2] Au Yeung, C., Gibbins, N., and Shadbolt, N. (2009). Multiple interests of users in collaborative tagging systems. In King, I. and Baeza-Yates, R., editors, *Weaving Services and People on the World Wide Web*, pages 255–274. Springer Berlin Heidelberg, Berlin, Heidelberg.

[3] Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., and Si, L. (2012). Expertise retrieval. *Found. Trends Inf. Retr.*, 6(2):127–256.

[4] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38.

[5] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[6] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Know.-Based Syst.*, 46:109–132.

[7] Bouras, C. and Tsogkas, V. (2017). Improving news articles recommendations via user clustering. *International Journal of Machine Learning and Cybernetics*, 8(1):223–237.

[8] Bui, Q. V., Sayadi, K., Amor, S. B., and Bui, M. (2017). Combining latent dirichlet allocation and k-means for documents clustering: Effect of probabilistic based distance measures. In Nguyen, N. T., Tojo, S., Nguyen, L. M., and Trawiński, B., editors, *Intelligent Information and Database Systems*, pages 248–257, Cham. Springer International Publishing.

[9] Can, F. and Ozkarahan, E. A. (1990). Concepts and effectiveness of the cover-coefficient-based clustering methodology for text databases. *ACM Trans. Database Syst.*, 15(4):483–517.

[10] Chen, C., Chen, M., and Sun, Y. (2001). A web document personalization user model and system. In *Proceedings of the Information Retrieval and User Modelling Conference*.

[11] Chen, L. and Sycara, K. (1998). Webmate: A personal agent for browsing and searching. In *Proceedings of the Second International Conference on Autonomous Agents*, AGENTS '98, pages 132–139, New York, NY, USA. ACM.

[12] Cormack, G. V., Clarke, C. L. A., and Buettcher, S. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 758–759, New York, NY, USA. ACM.

[13] Croft, B., Metzler, D., and Strohman, T. (2009). *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, USA, 1st edition.

[14] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

[15] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2017a). Committee-based profiles for politician finding. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(25):21–36.

[16] de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2017b). Profile-based recommendation: A case study in a parliamentary context. *Journal of Information Science*, 43(5):665–682.

[17] Deng, T., Ye, D., Ma, R., Fujita, H., and Xiong, L. (2020). Low-rank local tangent space embedding for subspace clustering. *Information Sciences*, 508:1–21.

[18] Gao, M., Liu, K., and Wu, Z. (2010). Personalisation in web computing and informatics: Theories, techniques, applications, and future research. *Information Systems Frontiers*, 12(5):607–629.

[19] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). User profiles for personalized information access. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 54–89. Springer Berlin Heidelberg, Berlin, Heidelberg.

[20] Ghorab, M. R., Zhou, D., O'Connor, A., and Wade, V. (2013). Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443.

[21] Gulla, J. A., Fidjestøl, A. D., Su, X., and Castejon, H. (2014). Implicit user profiling in news recommender systems. In *Proceedings of the 10th International Conference on Web Information Systems and Technologies - Volume 1: WEBIST,*, pages 185–192. INSTICC, ScitePress.

[22] Hanani, U., Shapira, B., and Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259.

[23] Hannon, J., McCarthy, K., O'Mahony, M. P., and Smyth, B. (2012). A multi-faceted user model for twitter. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'12, pages 303–309, Berlin, Heidelberg. Springer-Verlag.

[24] Jaiswal, A. and Janwe, N. (2011). Hierarchical document clustering: A review. *IJCA Proceedings on 2nd National Conference on Information and Communication Technology*, NCICT(3):37–41.

[25] Jardine, N. and van Rijsbergen, C. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217 – 240.

[26] Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transaction on Information System*, 20(4):422–446.

[27] Jayaprada, S., Aswani, A., and Gayathri, G. (2014). Hierarchical divisive clustering with multi view-point based similarity measure. In Satapathy, S. C., Udgata, S. K., and Biswal, B. N., editors, *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013*, pages 483–491, Cham. Springer International Publishing.

[28] Jin, M., Luo, X., Zhu, H., and Zhuo, H. H. (2018). Combining deep learning and topic modeling for review understanding in context-aware recommendation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1605–1614. Association for Computational Linguistics.

[29] Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: Development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808.

[30] Juntunen, P., Liukkonen, M., Lehtola, M. J., and Hiltunen, Y. (2013). Cluster analysis by self-organizing maps: An application to the modelling of water quality in a treatment process. *Applied Soft Computing*, 13(7):3191–3196.

[31] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

[32] Kohonen, T. (2001). *Self Organizing Maps*. Springer series in information sciences, 30. Springer, 3rd edition.

[33] Kook, H. J. (2005). Profiling multiple domains of user interests and using them for personalized web support. In Huang, D.-S., Zhang, X.-P., and Huang, G.-B., editors, *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part II*, pages 512–520. Springer Berlin Heidelberg, Berlin, Heidelberg.

[34] Larochelle, H. and Lauly, S. (2012). A neural autoregressive topic model. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2708–2716. Curran Associates, Inc.

[35] Lin, S., Hong, W., Wang, D., and Li, T. (2017). A survey on expert finding techniques. *Journal of Intelligent Information Systems*, 49(2):255–279.

[36] Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 73–105. Springer US, Boston, MA.

[37] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif. University of California Press.

[38] McGowan, J. P., Kushmerick, N., and Smyth, B. (2002). Who do you want to be today? web personae for personalised information access. In De Bra, P., Brusilovsky, P., and Conejo, R., editors, *Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference, AH 2002 Málaga, Spain, May 29–31, 2002 Proceedings*, pages 514–517. Springer Berlin Heidelberg, Berlin, Heidelberg.

[39] Miao, Y., Grefenstette, E., and Blunsom, P. (2017). Discovering discrete latent topics with neural variational inference. In *ICML*, volume 70 of *Proceedings of the 34th Machine Learning Conference*, pages 2410–2419. PMLR.

[40] Narducci, F., Musto, C., Semeraro, G., Lops, P., and de Gemmis, M. (2013). Exploiting big data for enhanced representations in content-based recommender systems. In Huemer, C. and Lops, P., editors, *E-Commerce and Web Technologies: 14th International Conference, EC-Web 2013, Prague, Czech Republic, August 27-28, 2013. Proceedings*, pages 182–193. Springer Berlin Heidelberg, Berlin, Heidelberg.

[41] Nguyen, P. T., Eckert, K., Ragone, A., and Di Noia, T. (2017). Modification to k-medoids and clara for effective document clustering. In Kryszkiewicz, M., Appice, A., Slkezak, D., Rybinski, H., Skowron, A., and Raś, Z. W., editors, *Foundations of Intelligent Systems*, pages 481–491, Cham. Springer International Publishing.

[42] P. Lloyd, S. (1982). Least squares quantization in pcm's. *IEEE Transactions on Information Theory*, 28:129–136.

[43] Pacella, M., Grieco, A., and Blaco, M. (2016). On the use of self-organizing map for text clustering in engineering change process analysis: a case study. *Computational Intelligence and Neuroscience*, 2016:Article n.7.

[44] Palamara, F., Piglione, F., and Piccinini, N. (2011). Self-organizing map and clustering algorithms for the analysis of occupational accident databases. *Safety Science*, 48:1215–1230.

[45] Pavan, M. and Luca, E. W. D. (2015). Semantic-based expert search in textbook research archives. In Risse, T., Predoiu, L., Nürnberger, A., and Ross, S., editors, *Proceedings of the 5th International Workshop on Semantic Digital Archives co-located with 19th International Conference on Theory and*

Practice of Digital Libraries (TPDL 2015), Poznan, Poland, September 18, 2015., volume 1529 of *CEUR Workshop Proceedings*, pages 18–29. CEUR-WS.org.

[46] Pazzani, M., Muramatsu, J., and Billsus, D. (1996). Syskill & webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 54–61. AAAI Press.

[47] Pon, R. K., Cardenas, A. F., Buttler, D., and Critchlow, T. (2007). Tracking multiple topics for finding interesting articles. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 560–569, New York, NY, USA. ACM.

[48] Rakesh, C., Sarma, C., and Jha, M. (2013). Document clustering using k-means and k-medoids. *International Journal of knowledge-based Computer Systems*, 1(1):7–13.

[49] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

[50] Rokach, L. and Maimon, O. (2005). Clustering methods. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, pages 321–352. Springer US, Boston, MA.

[51] Roux, M. (2018). A comparative study of divisive and agglomerative hierarchical clustering algorithms. *Journal of Classification*, 35(2):345–366.

[52] Rybak, J., Balog, K., and Nørvåg, K. (2014). Temporal expertise profiling. In de Rijke, M., Kenter, T., de Vries, A. P., Zhai, C., de Jong, F., Radinsky, K., and Hofmann, K., editors, *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings*, pages 540–546. Springer International Publishing, Cham.

[53] Saad, F., Mohamed, O., and Al-Qutaish, R. (2013). Comparison of hierarchical agglomerative algorithms for clustering medical documents. *International Journal of Software Engineering and Applications*, 3:1–15.

[54] Schiaffino, S. and Amandi, A. (2009). Intelligent user profiling. In Bramer, M., editor, *Artificial Intelligence*, pages 193–216. Springer-Verlag, Berlin, Heidelberg.

[55] Shah, N. and Mahajan, S. (2012). Document clustering: A detailed review. *International Journal of Applied Information Systems*, 4(5):30–38.

[56] Somlo, G. L. and Howe, A. E. (2001). Incremental clustering for profile maintenance in information gathering web agents. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 262–269, New York, NY, USA. ACM.

[57] Starczewski, A. and Krzyżak, A. (2015). Performance evaluation of the silhouette index. In Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L. A., and Zurada, J. M., editors, *Artificial Intelligence and Soft Computing*, pages 49–58, Cham. Springer International Publishing.

[58] Subhashini, R. and Kumar, V. S. (2011). A roadmap to integrate document clustering in information retrieval. *Int. J. Inf. Retr. Res.*, 1(1):31–44.

[59] Widyantoro, D. H., Yin, J., Seif, M., Nasr, E., Yang, L., Zacchi, A., and Yen, J. (1999). Alipes: A swift messenger in cyberspace. In *Proceedings of AAAI Spring Symposium on Intelligent Agents in Cyberspace*, pages 62–67.

[60] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2:165–193.

[61] Yu, H., Inoue, K., Hara, K., and Urahama, K. (2018). A robust k-means for document clustering. *Journal of the Institute of Industrial Applications Engineers*, 6:60–65.

[62] Zamora, J. (2017). *Recent Advances in High-Dimensional Clustering for Text Data*, pages 323–337. Springer International Publishing, Cham.

[63] Zeng, H.-J., Chen, Z., and Ma, W.-Y. (2002). A unified framework for clustering heterogeneous web objects. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, WISE '02, pages 161–172, Washington, DC, USA. IEEE Computer Society.

[64] Zhao, Y., Karypis, G., and Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data Min. Knowl. Discov.*, 10(2):141–168.