ADoCS: Automatic Designer of Conference Schedules

Diego Vallejo[1], **Paulina Morillo**[2], **Cèsar Ferri**[3]
[1]Universidad San Francisco de Quito, Department of Mathematics, Quito, Ecuador
dvallejoh@asig.com.ec
[2]Universidad Politécnica Salesiana, Research Group IDEIAGEOCA, Quito, Ecuador
pmorillo@ups.edu.ec
[3]Universitat Politècnica de València, DSIC, València, Spain
cferri@dsic.upv.es

Abstract

Distributing papers into sessions in scientific conferences is a task consisting in grouping papers with common topics and considering the size restrictions imposed by the conference schedule. This problem can be seen as a semi-supervised clustering of scientific papers based on their features. This paper presents a web tool called ADoCS that solves the problem of configuring conference schedules by an automatic clustering of articles by similarity using a new algorithm considering size constraints.

1 Introduction

Cluster analysis has the objective of dividing data objects into groups, so that objects within the same group are very similar to each other and different from objects in other groups (Tan et al., 2005). Semi-supervised clustering methods try to increase the performance of unsupervised clustering algorithms by using limited amounts of supervision in the form of labelled data or constraints (Basu et al., 2004). These constrains are usually restrictions of size or relations of belonging of objects to the clusters. These membership restrictions have been incorporated into the clustering process by different works (Zhu et al., 2010; Zhang et al., 2014; Ganganath et al., 2014; Grossi et al., 2015) showing that in this context, semi-supervised clustering methods obtain groupings that satisfy the initial restrictions.

On the other hand, in generic form, document clustering should be conceived as the partitioning of a documents collection into several groups according to their content (Hu et al., 2008). A scientific article is a research paper published in specialised journals and conferences. Conferences are usually formed of various sessions of fixed size where the authors present their selected papers. These sessions are usually thematic and are arranged by the conference chair in a manual and tedious work, specially when the number of papers is high. Organising the sessions of a conference can be seen as a problem of document clustering with size constraints.

In this work we present a web application called ADoCS for the automatic configuration of sessions in scientific conferences. The system applies a new semi-supervised clustering algorithm for grouping documents with size constraints. Recently, a similar approach has been described by (Škvorc et al., 2016). In this case the authors also use information from reviews to build the groups, however, this information is not always available.

2 Methodology

In this section we summarise the semi-supervised clustering algorithm of ADoCS system.

The information about the papers in the conference is uploaded to the system by means of a simple csv file. This csv represents a paper per row, and each row must contain, at least, three columns: Title, Keywords and Abstract. For the text preprocessing, NLP techniques and information retrieval techniques are applied to obtain a dissimilarity matrix. We used a classical scheme for data pre-processing in documents: tokenization, stopwords removal and stemming (Jha, 2015).

To structure the dissimilarity matrix of titles and keywords, Jaccard coefficient is applied, since these two elements usually have a small number of tokens. In the case of abstracts a vector model with a cosine similarity index on TF-IDF weighting matrix is used. ADoCS web tool has these default settings, although these parameters can be directly adjusted by the user.

Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017, pages 41–44

The bag-of-words, or vector model representation derived from the texts, configure a Euclidean space where several distances can be applied in order to estimate similarities between elements. Nevertheless, in some cases, we need to average several criteria and unify them to obtain a single metric that quantifies dissimilarities, and with this, we can derive a distance matrix. This is the case we address here, since for each paper we have three different features: title, abstract and keywords. In these situations we cannot directly apply clustering methods based on centroids, such as K-Means, since there is not a Euclidean space defined for the elements. One way to solve this type of problems is to apply algorithms that, for the clustering process, use only the dissimilarity or distance matrix as input. ADoCS works with a new algorithm called CSCLP (Clustering algorithm with Size Constraints and Linear Programming) (Vallejo, 2016) that only uses as inputs: the size constrains of the sessions and the dissimilarity/distance matrix.

Clustering algorithms obtain better results if a proper selection of initial points method is computed (Fayyad et al., 1998). For this reason, the initial points in our clustering algorithm is chosen using a popular method: Buckshot algorithm (Cutting et al., 1992). In CSCLP, the initial points are used as pairwise constraints (as cannot-link constraints in semi-supervised clustering terminology) for the formation of the clusters, and with binary integer linear programming (BILP) the membership and assignment of the instances to the clusters is determined, satisfying the size constrains of the sessions. In this way, the original clustering problem with size constraints becomes an optimisation problem. Details of the algorithm can be found in (Vallejo, 2016).

3 ADoCS Tool

In this section we include a description of the ADoCS tool. You can find a web version of the tool in the url: `https://ceferra.shinyapps.io/ADoCS`.

On the left part of the web interface we find a panel where we can upload a *csv* file containing information of the papers to be clustered. In the panel, there are several controls where we can configure some features of this *csv* file. Concretely, the separator of fields (comma by default) and how literals are parsed (single quote by default). Additionally, we find three control bars (Title, Keywords and Abstract) with values between 0 and 1 that establish the weights of each of these factors for the computation of distances between papers. By default, the three bars are set to 0.33 to indicate that the three factors will have the same weight in computing the distances. The values of the weights are normalised in such a way that they always sum 1. The user can also configure whether the TF-IDF transformation is applied or not, as well as the metric that is employed to compute the distance between elements. These controls are responsive, i.e., when the user modifies one of the values, the distance matrix is recomputed, and also all the components that depend on this matrix.

Once the file is correctly uploaded in the system, the application enables the function tabs that give access to the functionality of the web system. There are four application tabs:

- **Papers**: This tab contains information about the dataset. We include here the list of papers. For each paper, we show the number, Title, Keywords and Abstract. In order to improve the visualisation, a check box can be employed to show additional information of the papers.

- **Dendrogram**: In this part, a dendrogram generated from the distance matrix is shown. The distance between papers is computed considering the weights selected by the user and the methodology detailed in Section 2.

- **MDS**: In this tab, a Multidimensional Scaling algorithm is employed over the distance matrix to generate a 2D plot about the similarity of papers. Once the clusters are arranged, the membership of the papers to each cluster is denoted by the colour.

- **Wordmap**: This application tab includes a word map representation for showing the most popular terms extracted from the abstracts of the papers in the dataset.

- **Schedule**: In this part, the user can configure the number and size of the sessions and execute the CSCLP algorithm, described in Section 2, to build the groups according the similarity between papers.

The typical use of this tool starts by loading a *csv* file with the information of the papers. After the data is in the system, we can use the first tab *Papers* to know the number of papers in the conference. We can also observe specific information of all the papers (title, keywords, abstract or other information included in the file). The tabs *Dendrogram*, *MDS* and *Wordmap* can be used for exploring the similarities between papers and for knowing the most common terms in the papers of the conference. Finally, we can execute the semi-supervised algorithm to configure the sessions in the *Schedule* tab. In this tab, we can introduce the number of the desired sessions as well as the distribution of the size of the sessions. For instance, if we have 40 papers and we want 5 sessions of 5 papers, and 5 sessions of 3 papers, we introduce "10" in the *Number of Sessions* text box and "5,5,5,5,5,3,3,3,3,3" in *Size of Sessions* text box. When the user push button *compute*, if the sum of the number of papers in the distribution is correct (i.e. is equal to the total number of papers in the conference), the algorithm of section 2 is executed in order to find clusters that satisfy the size restrictions expressed by the session distribution. As a result of the algorithm, a list of the papers with the assigned cluster is shown in the tab. Finally, we can download the assignments as a *csv* file if we push the *Save csv* button.

4 Technology

The ADoCS tool has been implemented totally in R (R Core Team, 2015). This system is a free software language for statistical computing. There is a plethora of different libraries that makes R a powerful environment for developing software related to data analysis in general, and computational linguistics in particular. Specifically, we have employed the following R libraries:

- **tm**: The *tm* package (Meyer et al., 2008) contains a set of functions for text mining in R. In this project we have use the functionalities related to text transformations (stemming, stopwords, TF-IDF, ...) with the English corpora.

- **wordcloud**: The *wordcloud* package (Fellows, 2014) includes functions to show the most common terms in a vocabulary in form of the popular *wordcloud* plot. An example of this plot for the AAAI-2014 conference is included in Figure 1.

Figure 1: Example of the WordMap of the abstracts in the papers of Twenty-Eighth Conference on Artificial Intelligence (AAAI-2014)

- **proxy**: The *proxy* package (Meyer and Buchta, 2016) allows to compute distances between instances. The package contains implementations of popular distances, such as: Euclidean, Manhattan or Minkowski, etc.

- **Rglpk**: The *Rglpk* package (Theussl and Hornik, 2016) represents an R interface to the GNU Linear Programming Kit. GLPK is a popular open source kit for solving linear programming (LP), mixed integer linear programming (MILP) and other related problems with optimisation.

The ADoCS source code and some datasets about conferences to test the tool can be found in `https://github.com/dievalhu/ADoCS`.

The graphical user interface has been developed by means of the Shiny package (Chang et al., 2016). This package constitutes a framework for building graphical applications in R. Shiny is a powerful package for converting basic R scripts into interactive web applications without requiring programming web knowledge. As a result we obtain an interactive web application that can be executed locally using a web browser, or can be uploaded to a shiny application server where the

tool is available for a general use. In this case, we have uploaded the application to the free server *http://www.shinyapps.io/*.

5 Conclusions and Future Work

Arranging papers to create an appropriate conference schedule with sessions containing papers with common topics is a tedious task, specially when the number of papers is high. Machine learning offers techniques that can automatise this task with the help of NLP methods for extracting features from the papers. In this context, organising a conference schedule can be seen as a semi-supervised clustering. In this paper we have presented the ADoCS system, a web application that is able to create a set of clusters according to the similarity of the documents analysed. The groups are formed following the size distribution configured by the user. Although initially the application is focused on grouping conference papers, other related tasks in clustering documents with restrictions could be addressed thanks to the versatility of the interface (different metrics, TF-IDF transformation).

As future work, we are interested in developing conceptual clustering methods to extract topics from the created clusters.

Acknowledgments

This work has been partially supported by the EU (FEDER) and Spanish MINECO grant TIN2015-69175-C4-1-R, LOBASS, by MINECO in Spain (PCIN-2013-037) and by Generalitat Valenciana PROMETEOII/2015/013.

References

Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD conference*, pages 59–68. ACM.

Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson, 2016. *shiny: Web Application Framework for R*. R package version 0.14.

Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. 1992. Scatter/gather: a cluster-based approach to browsing large document collections. *Proceedings of the 15th annual international ACM SIGIR conference*, pages 318–392.

Usama Fayyad, Cory Reina, and Paul S. Bradley. 1998. Initialization of iterative refinement clustering algorithms. *Proceedings of ACMSIGKDD*, pages 194–198.

Ian Fellows, 2014. *wordcloud: Word Clouds*. R package version 2.5.

Nuwan Ganganath, Chi-Tsun Cheng, and Chi. K Tse. 2014. Data clustering with cluster size constraints using a modified k-means algorithm. *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), International Conference IEEE*, pages 158–161.

Valerio Grossi, Anna Monreale, Mirco Nanni, Dino Pedreschi, and Franco Turini. 2015. Clustering formulation using constraint optimization. *Software Engineering and Formal Methods*, pages 93–107.

Guobiao Hu, Shuigeng Zhou, Jihong Guan, and Xiaohua Hu. 2008. Towards effective document clustering: a constrained k-means based approach. *Information Processing & Management*, 44(4):1397–1409.

Monica Jha. 2015. Document clustering using k-medoids. *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, 4(1):2319–2526.

David Meyer and Christian Buchta, 2016. *proxy: Distance and Similarity Measures*. R package version 0.4-16.

David Meyer, Kurt Hornik, and Ingo Feinerer. 2008. Text mining infrastructure in R. *Journal of statistical software*, 25(5):1–54.

R Core Team, 2015. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Tadej Škvorc, Nada Lavrac, and Marko Robnik-Šikonja. 2016. Co-Bidding Graphs for Constrained Paper Clustering. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, volume 51, pages 1–13.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2005. *Introduction to data mining*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Stefan Theussl and Kurt Hornik, 2016. *Rglpk: R/GNU Linear Programming Kit Interface*. R package version 0.6-2.

Diego F. Vallejo. 2016. Clustering de documentos con restricciones de tamaño. Master's thesis, Higher Technical School of Computer Engineering, Polytechnic University of Valencia - UPV.

Shaohong Zhang, Hau-San Wong, and Dongqing Xie. 2014. Semi-supervised clustering with pairwise and size constraints. *International Joint Conference on Neural Networks (IJCNN), IEEE*, pages 2450–2457.

Shunzhi Zhu, Dingding Wang, and Tao. Li. 2010. Data clustering with size constraints. *Knowledge-Based Systems*, 23(8):883–889.