Co-Bidding Graphs for Constrained Paper Clustering

Tadej Škvorc[1], Nada Lavrač[2], and Marko Robnik-Šikonja[3]

1 University of Ljubljana, Faculty of Computer and Information Science,
Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si

2 Jožef Stefan Institute, Ljubljana, Slovenia; and
University of Nova Gorica, Nova Gorica, Slovenia
nada.lavrac@ijs.si

3 University of Ljubljana, Faculty of Computer and Information Science,
Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si

Abstract

The information for many important problems can be found in various formats and modalities. Besides standard tabular form, these include also text and graphs. To solve such problems fusion of different data sources is required. We demonstrate a methodology which is capable to enrich textual information with graph based data and utilize both in an innovative machine learning application of clustering. The proposed solution is helpful in organization of academic conferences and automates one of its time consuming tasks. Conference organizers can currently use a small number of software tools that allow managing of the paper review process with no/little support for automated conference scheduling. We present a two-tier constrained clustering method for automatic conference scheduling that can automatically assign paper presentations into predefined schedule slots instead of requiring the program chairs to assign them manually. The method uses clustering algorithms to group papers into clusters based on similarities between papers. We use two types of similarities: text similarities (paper similarity with respect to their abstract and title), together with graph similarity based on reviewers' co-bidding information collected during the conference reviewing phase. In this way reviewers' preferences serve as a proxy for preferences of conference attendees. As a result of the proposed two-tier clustering process similar papers are assigned to predefined conference schedule slots. We show that using graph based information in addition to text based similarity increases clustering performance. The source code of the solution is freely available.

1998 ACM Subject Classification I.2.8 Problem Solving, Control Methods, and Search

Keywords and phrases Text mining, data fusion, scheduling, constrained clustering, conference

Digital Object Identifier 10.4230/OASIcs.SLATE.2016.1

1 Introduction

In many real world situations data can be present in various different formats and can therefore be difficult to understand. In order to efficiently use such data it must first be converted into a common format. Combining data present in different forms is known as data fusion and is a useful technique, particularly in machine learning, where data must be present in the form of feature vectors. One field where data fusion can be useful is conference organization.

5th Symposium on Languages, Applications and Technologies (SLATE'16).
Editors: Marjan Mernik, José Paulo Leal, and Hugo Gonçalo Oliveira; Article No. 1; pp. 1:1–1:13
OASICS Open Access Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Scientific conferences, which allow scientists to share new discoveries, are a key to progress in science. Several conferences are very large. Organizing them can be difficult and time consuming. Various tools have been developed to help conference organizers deal with this problem by automating some of conference management tasks. However, scheduling paper presentations is currently still performed manually to a large extent. This can be time consuming, as large conferences often have a lot of presentations that must be grouped together based on similarities and differences between them.

A conference schedule usually consists of multiple time slots which contain semantically similar papers. We propose automated paper scheduling using text mining to find similar papers, grouping similar papers using clustering and assigning them into schedule time slots. Conference organizers may have access to additional metadata describing the papers. Such data is sometimes present in a graph form which can be vectorized and used in clustering.

In general, text documents can be linked within a graph where two documents are connected if they share the same author, if they are published in the same publication or based on citations in the paper's list of references. Such graphs may contain a large amount of information, which is mostly ignored by conventional text mining methods. To make graph data suitable for use with standard machine learning algorithms, a feasible approach is to convert the graphs into a feature vector format. To this end, we have adapted the methodology proposed by Grčar et. al. [6], which converts the information from graphs to feature vectors using the Personalized PageRank (PPR) algorithm [11] and then combines these vectors with text representation in bag-of-words (BOW) vectors. The combined feature vectors can be used by a wide variety of machine learning algorithms for different tasks, such as classification model construction.

We describe how this method can be adapted to categorize and cluster similar conference papers based on the textual content of their abstracts and titles in combination with the reviewers' bidding information collected during the conference evaluation period, where the reviewers select papers they would like to review and the ones they would rather not. This information is used to create a graph, where two papers are connected if the same reviewer expressed the wish to review them. We use this information in combination with the textual information to cluster papers.

The novelty of the proposed methodology is due to combining three previously unrelated approaches: (a) paper clustering using text-based similarity of BOW vectors, (b) paper similarity computation using the co-bidding graph to compute PageRank-based instance weighting, (c) constrained clustering for matching paper clusters to appropriate conference slots, and finally, (d) a user friendly web interface for conference organizers. The utility of the proposed approach was show-cased on the AIME 2013 conference (14th Conference on Artificial Intelligence in Medicine), where the results of automatic approach were nearly as accurate as the manual conference scheduling approach.

The paper is organized into six sections. In Section 2 we present the related work. Section 3 describes the data set we used. In Section 4 we describe the method used to enrich text with metadata present in graphs and how we used this method to group similar papers. In Section 5 we describe a web application used for conference schedule management that supports semi-automated schedule construction. Section 6 concludes the paper.

2 Related work

Several authors present methods of finding similar academic papers using graph-based metadata. Such methods can produce better results than methods using only text similarity approaches. Huynh et al. [8] and Liang et al. [10] present methods that use graphs constructed from citations to find similar papers. Grčar et al. [6] show how such data can be effectively used in combination with text. They describe a method which fuses graph data with standard bag-of-word vectors into a single feature vector format. These vectors can be used as input to standard machine learning algorithms. Our approach differs from other approaches searching similar papers by using a graph constructed from preferences expressed by reviewers during the conference evaluation period. The information contained in such graphs have yet not been exploited. Additionally, we use this information in constrained clustering with the final aim to construct a useful schedule.

Academic conference recommendation systems help users select talks and presentations they would likely be interested in. Several papers [14, 21, 22] show that information extracted from socially-aware networks can be helpful. Our approach is not oriented towards conference attendees but conference organizers. It applies fusion of text and graph information to conference scheduling. Since a high quality schedule is a prerequisite for conference attendee recommendations, our approach can be viewed as a foundation for user recommendations. Instead of using a socially-aware network constructed during the conference, we construct a graph from the opinions reviewers expressed during the paper review period.

Constrained clustering imposes specific constraints to the results produced by clustering to improve performance. Wagstaff et al. [20] proposes a method of imposing must-link and cannot-link constraints that limit which examples can be in the same cluster. Zhu et al. [23] presents a heuristic method that imposes size constraints to clusters. We use a new approach to limit cluster sizes so they match a predefined conference schedule structure.

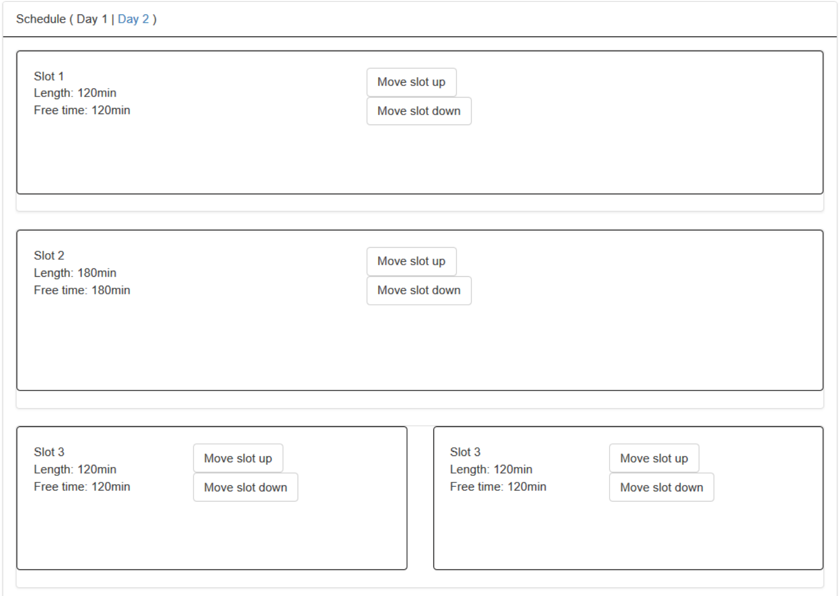
3 Automatic conference scheduling

Automating conference scheduling is a hard task. To solve it we first find semantically similar papers and then group them together according to the conference schedule. In this section we present our approach, which uses text-mining enriched with information obtained from the reviewers' co-bidding graph to find similar papers.

3.1 Problem overview

To automate conference scheduling we must solve two separate problems. The first is finding papers that are semantically similar. This is important because conference schedules are usually composed of several *sessions*. Each session contains paper presentations related to a specific research topic and is usually named with the adequate topic name. In automatic scheduling each of these sessions must be filled with similar papers. We solve this problem using clustering. First, each paper is turned into a Bag-of-Words (BOW) vector, and vector components are weighted with tf-idf (term frequency–inverse document frequency) [16]. This vector is constructed from the abstract and title of the paper. We extend the vector with paper similarity components extracted from conference reviewers' co-bidding preferences. After this the papers are clustered using clustering algorithms.

Other methods can also be used to solve this problem. Topic modeling [2] is commonly used to assign text documents into separate topics and could be used to find similar papers. However, such methods only take into account the text of the paper and not the additional graph-based metadata present in our case. Topic modeling also has to be trained on a large corpus before it can be used to assign documents into different topics. For use in a general conference management tool the training corpus would need to encompass papers from a wide variety of topics.



Schedule (Day 1 | Day 2)

Slot 1
Length: 120min
Free time: 120min

Slot 2
Length: 180min
Free time: 180min

Slot 3
Length: 120min
Free time: 120min

Slot 3
Length: 120min
Free time: 120min

Figure 1 An example of an empty conference schedule. Each slot corresponds to a session that needs to be filled with papers from the same research topic. Sessions in Slots 1 and 2 occur sequentially, while Slot 3 has two parallel sessions. The application allows conference organizers to add new slots, move the slots around and set the duration of the slots. It allows organizers to create a multi-day conference program.

In our approach, clustering returns several groups of papers where each group corresponds to a different research topic. The second problem we need to solve is to construct a conference schedule from these groups. Some aspects of the schedule, such as keynote presentations and lunch breaks, are independent of paper presentations. Because of this we let conference organizers manually construct the conference structure. The structure consists of different blocks with prespecified duration. The blocks can occur either sequentially or in parallel. After the schedule structure is constructed, conference organizers decide which of the blocks will be automatically filled with thematically similar papers and which should be used for other purposes, such as keynote presentations. To integrate the automatic scheduling with manual schedule structure construction we built a user friendly web application. The application allows conference organizers to first manually construct the schedule structure and then automatically place papers into the schedule based on similarities between them. An example of an empty schedule constructed with this application is presented in Figure 1. An overview of the entire process is presented in Figure 2.

3.2 Data set

We tested our approach on a real-world example. Below we give a short description of the data we used. The data set consists of papers from a specific conference. The papers were described by abstracts and titles. We used this data to create feature vectors describing each paper. The BOW vectors were obtained by first removing stop words from the abstracts and titles and then weighting the vector components with tf-idf [16]. The data also included a list of reviewer preferences. For each paper, the reviewers selected one of the following opinions: *"I want to review this paper"*, *"I can review it"*, *"I prefer not to review it"* and *"I have a conflict of interest"*.

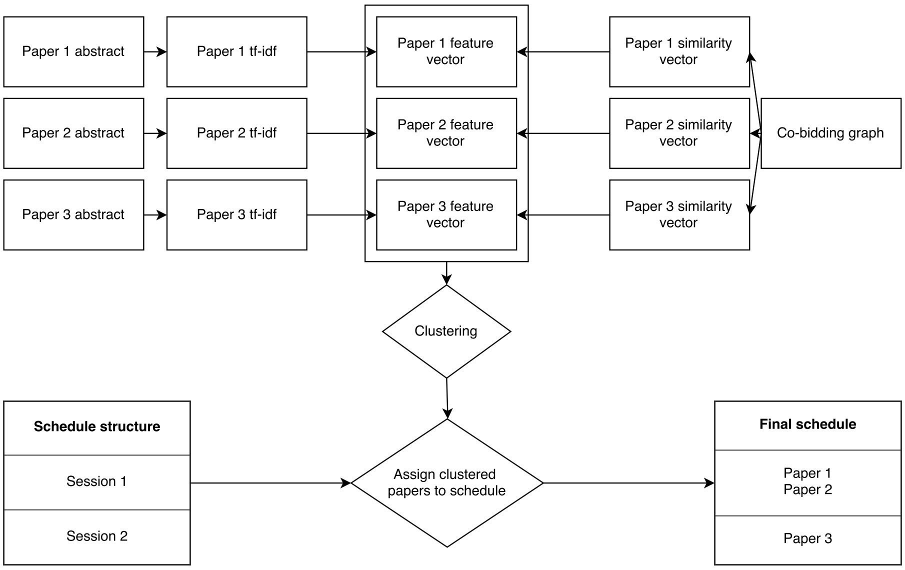


Figure 2 A summary of the entire scheduling algorithm. Each paper is described by the tf-idf vector of its abstract and similarity to other papers, which is obtained from the co-bidding graph. The papers are clustered and assigned to a preconstructed conference schedule.

From these preferences we constructed a graph in which two papers are connected if the same reviewer expressed a wish to review them. Additionally, each connection is weighted. Connections between two papers where the reviewer expressed the opinion *"I want to review this paper"* for both papers were assigned larger weights (we have chosen the weight of 4) than the connections between two papers with opinions *"I want to review this paper"* and *"I can review it"*, which were weighted with the weight of 2. Two abstracts, both with the opinion *"I can review it"*, were weighted even lower, with the weight of 1. Papers with other combinations of expressed preferences were not connected. We created such a graph for each reviewer and combined them into a single co-bidding graph by summing the edge weights from all the reviewers' graphs.

This presents a way of modeling the experts' opinion on the similarity of the articles. Since most conference reviewers tend to focus on a limited number of research fields and they usually review papers from these fields, papers they review tend to be semantically similar. By weighting the graph we can assign larger weights to links between nodes (papers) for which reviewers express stronger preferences, as this indicates that they are likely to be similar and from the same topic. The papers connected by the same reviewer are also likely to be interesting to the same group of conference attendees, of whom the reviewers are an excellent and valuable sample. Even if the reviewers tend to follow more than one topic this is not necessarily bad for the connected papers. First, they might be logically connected and second, the effect of a single reviewer is limited as co-bidding information is merged with textual similarity.

Past research [21] shows that recommendations from other conference participants can be useful for finding similar and interesting papers. Such recommendations cannot be obtained before the start of the conference. With our approach, recommendations from reviewers are used as a substitute, since they are already familiar with some of papers that appear in the conference. The goal of our automatic scheduling is to construct a program schedule that pleases as many attendees as possible. An important aspect of this is placing paper presentations from the same topic that groups of people would find interesting in the same slot and not to overlap paper presentations with similar audience. It is a reasonable assumption that reviewers want to review papers they find interesting, therefore the constructed co-bidding graph links paper presentations that should be placed close to each other. The co-bidding graph is therefore useful in automatic scheduling.

3.3 Enriching text with co-bidding preferences

The co-bidding graph needs to be converted into a form suitable for clustering algorithms. To this end, we used the Personalized PageRank (PPR) algorithm [11]. This algorithm was designed to assess importance of graphs consisting of web pages. For a given starting page it computes importance of all pages relative to it using a random-walker model. The PageRank R' For a given set of pages can be calculated using the following equation:

$$R'(u) = c \times \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u)$$

where B_u is a set of all pages linking to u, N_v is the number of links from v and c is a normalization factor, which ensures that the L_1 norm of R' is equal to 1 and must be maximized. $E(u)$ represents the PageRank source and is defined as follows.

$$E(u) = \begin{cases} 1 & : \text{u is starting page} \\ 0 & : \text{otherwise} \end{cases}$$

The algorithm can be used to convert information from graph structure into a BOW vector that can be used together with the information obtained by text mining [6]. If we assign a unique word to each web page (a node in the graph) and the user randomly navigating the graph writes down that word each time he/she visits the node, we get a textual document describing our graph. Such a document contains a higher frequency of words that are strongly linked to the starting node and can be naturally combined with textual data contained in the node. By using the Personalized PageRank algorithm we essentially get a normalized BOW vector (a vector holding the relative frequencies of individual words) describing such a document. This is useful as we get a description of the graph in the same form as the BOW vectors we constructed from abstracts and titles, and both forms can therefore be merged.

We use this approach to extract information from biddings expressed by the reviewers. In our bidding graph, two papers are connected if a reviewer expressed a wish to review both of them. Two connected papers are likely to contain similar topics. Reviewers are not picking papers to review at random, but rather because they are from the field they are interested in. Consequently, relevant semantically similar papers will be connected. Additionally, since the connections are weighted, reviewers' stronger opinions will have stronger connections. Applying the Personalized PageRank algorithm to our graph, starting from a specific paper, the algorithm will return larger probabilities for papers that are semantically close to that paper. In this way we obtain vectors containing probabilities of other papers being similar to a given paper. The PPR and the BOW vector can be merged and treated with the same approaches in the processing pipeline. Note also that the bidding graph contains semantic similarity information captured from human experts, which is an important added value of our approach.

Figure 3 shows an example graph obtained from reviewers' preferences. If we apply PPR to this graph starting from paper 1, we get the PPR vector $[0.27, 0.20, 0.33, 0.09, 0.06, 0.05]$.

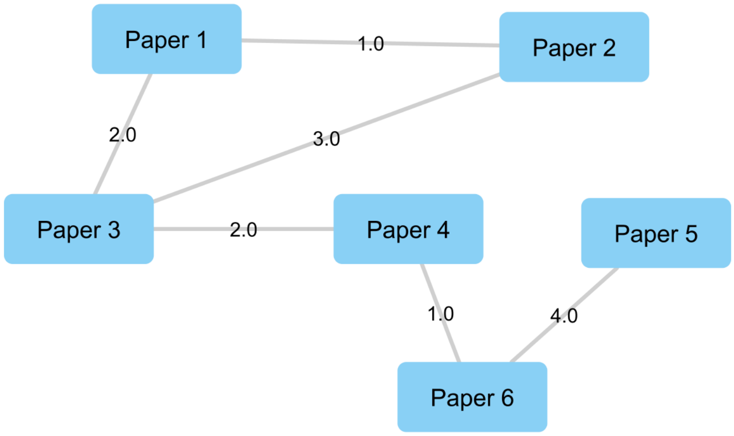


Figure 3 An example graph obtained from reviewers' preferences. By running the Personalized PageRank algorithm starting at a specific node in the graph, we obtain a vector describing how important other papers are for this specific paper.

This vector contains high values for papers 2 and 3, which are closely connected to paper 1. Running the algorithm starting from paper 6 returns vector $[0.03, 0.05, 0.10, 0.10, 0.31, 0.41]$, showing that this paper is closely connected to papers 5 and 4. In our graph edge weights correspond to the number of reviewer's that wanted to review both papers in the paper pair and also indicates the strength of their preferences.

The two vectors (the tf-idf vector of the abstract and title and the PPR vector of the co-bidding graph) are combined into a single feature vector. Since the PPR vectors are normalized, we can treat them as a separate tf-idf vector. We combine the two vectors by multiplying each one by 0.5 and concatenating them. It is possible to weight the vectors differently, which would place more weight on one of the vectors. The result is a single normalized feature vector. This final feature vector is clustered with different algorithms to group similar papers based both on their textual content and on the preferences expressed by the reviewers. This entire process is summarized in Figure 4. The described method combining both graph- and text-based information could also be used in classification.

3.4 Constrained clustering

The final step is to assign similar papers from the same cluster into one of several predefined schedule time slots. To do this we impose constraints on the clustering, as unconstrained clustering returns groups that do not match the schedule structure. The constraints are implemented iteratively, by matching and filling in the largest empty time slot with papers from the largest cluster until it is full. If a cluster large enough to fill the empty time slot does not exist we rerun the clustering with arguments that produce larger clusters. An overview of our approach is presented in Figure 5. We tested the method using *Affinity propagation* [4], *DBSCAN* [3], *Agglomerative clustering* [9], *K-means* [7] and *Mean Shift* clustering [5], as described in Section 5.

4 The conference scheduling application

We implemented the described methodology in a web application that supports program chairs in conference scheduling by automatically grouping similar articles into predefined

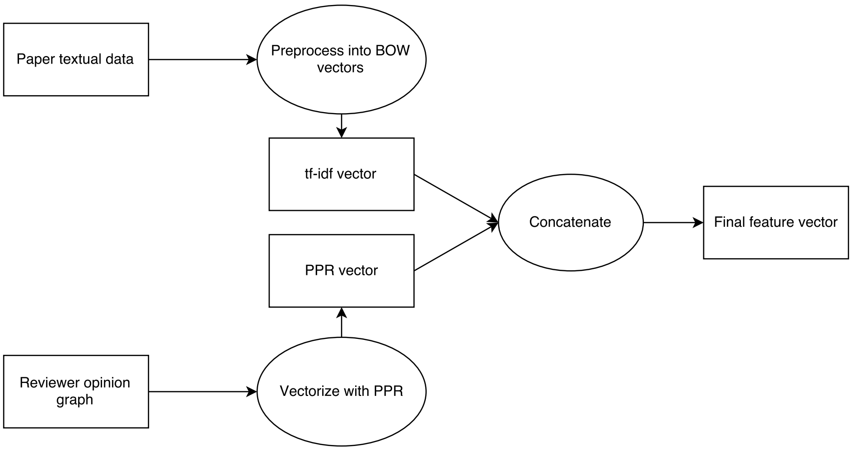


Figure 4 The process to obtain feature vectors. When vectorizing the co-bidding graph with PPR, the starting node of the PPR algorithm corresponds to each paper in turn. This produces a unique PPR vector for each paper. Since both the PPR and tf-idf vectors are normalized, the final feature vector is also normalized.

schedule time slots and also allows manual improvements. The source code of the application is available at `https://github.com/TadejSk/conference-scheduler`. The program chairs first construct the structure of the presentation schedule or import one of the stored schedules. They import papers from a database or manually add them into the application. The application automatically constructs a schedule which is presented to the program chair and allows editing. The application allows chairs to work on multiple conferences at the same time and automatically saves the schedules on the web server running the application. Figure 7 shows an example how paper clusters are visualized by our application and Figure 6 shows the user interface for schedule construction.

5 Evaluation

We evaluated several aspects of our approach. We compared several clustering algorithms to determine which one produces the best results. We tested both the quality of the results as well as the running time of the algorithms. To determine the effect of the co-bidding graph on the final results we compared the results of standard text-mining methods with the results obtained when those methods were fused with the information retrieved from the co-bidding graph. We compared the results by expert analysis and using the silhouette score [15].

5.1 Comparing different clustering methods

As described in Section 3.3, the combined feature vectors produced by our method can be applied to most clustering algorithms. We tested the method on a number of different clustering algorithms implemented in scikit-learn [12] to determine which of them is the most suitable for this type of data. The algorithms we tested were *Affinity propagation*, *DBSCAN*, *Agglomerative clustering*, *K-means* and *Mean Shift* clustering.

For some clusterings examining the visualization of the results was enough to determine that they are not suitable. *Mean Shift clustering* and *DBSCAN* produced clusters that had no clear structure and appeared random on our data set. They required finely tuned parameters to return more than one cluster. A visualization of results returned by Mean Shift is presented

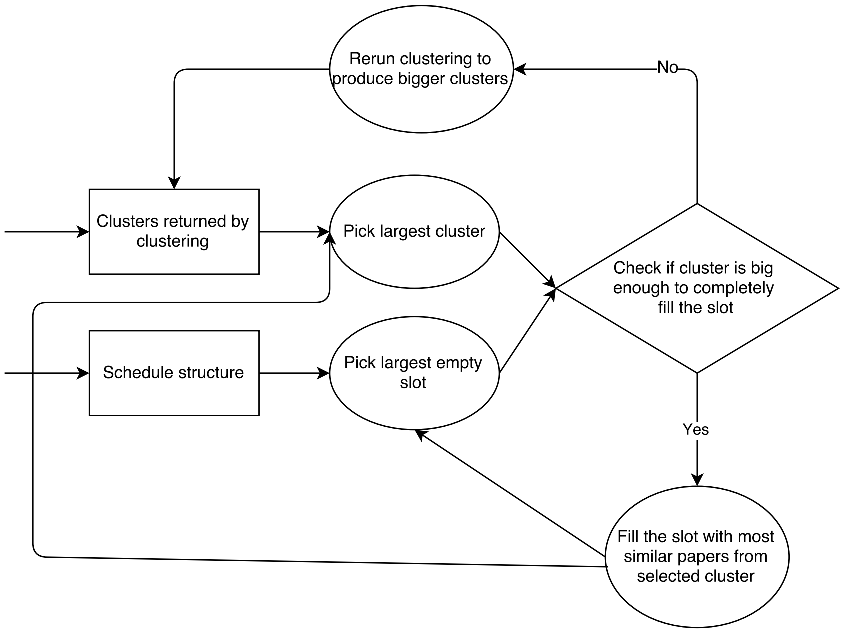

Rerun clustering to produce bigger clusters
No
Clusters returned by clustering
Pick largest cluster
Check if cluster is big enough to completely fill the slot
Schedule structure
Pick largest empty slot
Yes
Fill the slot with most similar papers from selected cluster

Figure 5 The iterative algorithm that fills an empty conference schedule with clustered papers.

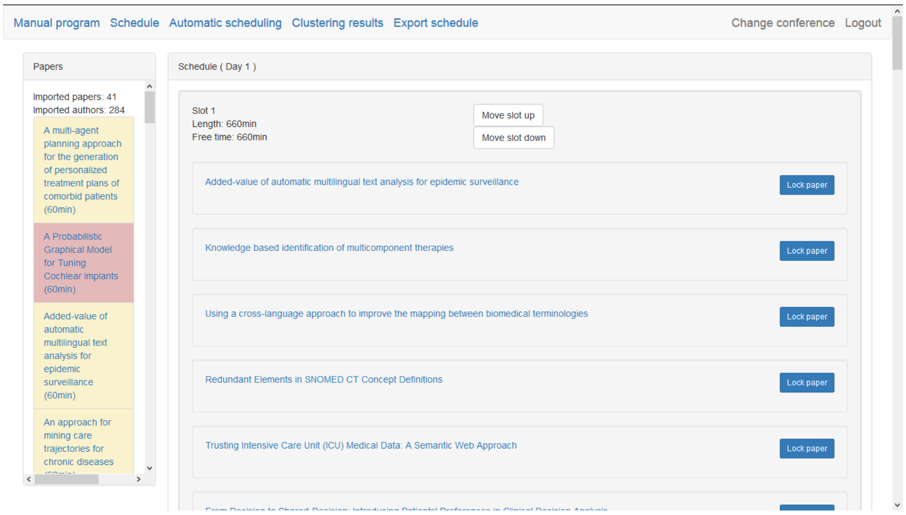
in Figure 8. Visually the best results were obtained by *K-means clustering*, which consistently returned well structured clusters. We also tested the modifications by Sculley [17], which aims to improve the performance of K-means clustering for web applications. The modifications returned similar results while improving the performance. *Agglomerative clustering* returned as visually appealing results as K-means clustering. *Affinity propagation* also returned good results, but does not allow to explicitly set the number of clusters. In practice this is problematic, since the number of conference sessions and the number of clusters should be close.

We also compared the effect of clustering algorithms on the execution time of automatic scheduling. The choice of algorithm had little effect on the overall execution time. We tested the execution time by running the entire automatic scheduling process on an example data set containing 41 papers, using a 2.4 GHz processor. The fastest algorithm was Agglomerative clustering, which finished in 4.31 seconds. Mean Shift was the slowest algorithm and finished in 4.96 seconds. The difference between the fastest and the slowest algorithm was 12%.

5.2 Effect of the co-bidding graph

We tested our approach on papers from the AIME 2013 conference (14th Conference on Artificial Intelligence in Medicine) [13]. The conference schedule consisted of 43 paper presentations scheduled in 9 sessions. We evaluated the approach by comparing the results returned by our method with the actual schedule that was used in the AIME 2013 conference. We also tested the effect of the additional graph data.

We first tested the effect of the co-bidding graph using the silhouette score [15]. The silhouette score is an internal clustering validity measure and measures the quality of produced clusters. It does so by comparing the dissimilarity of an object with other objects in the same cluster to its dissimilarity with objects from a different cluster. If we define $a(i)$ as the average dissimilarity of i to all other objects in its cluster and $b(i)$ as the smallest average dissimilarity of i to all other objects in another cluster, we can compute the silhouette score



Papers

Imported papers: 41
Imported authors: 284

A multi-agent planning approach for the generation of personalized treatment plans of comorbid patients (60min)

A Probabilistic Graphical Model for Tuning Cochlear Implants (60min)

Added-value of automatic multilingual text analysis for epidemic surveillance (60min)

An approach for mining care trajectories for chronic diseases (60min)

Schedule (Day 1)

Slot 1
Length: 660min
Free time: 660min

Added-value of automatic multilingual text analysis for epidemic surveillance

Knowledge based identification of multicomponent therapies

Using a cross-language approach to improve the mapping between biomedical terminologies

Redundant Elements in SNOMED CT Concept Definitions

Trusting Intensive Care Unit (ICU) Medical Data: A Semantic Web Approach

From Decision to Shared Decision: Introducing Patients' Preferences in Clinical Decision Analysis

Figure 6 The main user interface of the application. Users can manually assign papers into conference slots, or they can use the application to automatically schedule them.

using the following equation:

$$s(i) = \frac{b(i) - a(i)}{max[a(i), b(i)]} \ .$$

A high silhouette score indicates that papers within a cluster are more similar among themselves that to papers in other clusters. The method works best on well defined clusters with large distances between clusters. The clustering methods we tested do not return such well defined clusters, since the BOW vectors used in clusterings are both highly dimensional and similar between each other, which leads to clusters that are blurred. Nevertheless, the silhouette score can still be used to compare the produced clusterings. We compared clustering papers by only using BOW vectors and by using BOW vectors fused with graph data. In both cases we repeated the clustering 50 times and averaged the silhouette score. On average, using graph data increased the silhouette score by 4.5%. This shows that additional graph based data is helpful.

5.3 Scheduling evaluation

Evaluating the quality of the automatically constructed schedule with external clustering validity measures such as adjusted Rand index or variation of information [1] can be unreliable. For each conference there exist multiple ways to construct a good schedule. This means that using the actual conference schedule as a ground truth and comparing the automatically constructed schedule with it will not necessary be objective. An automatic schedule that correctly groups together similar papers will be evaluated as inadequate if it groups them in a different way than they were grouped in the actual conference. As ground truth does not exist, similarly to other text mining tasks, e.g., automatic summarization or machine translation, we use subjective opinion of domain experts to measure the quality of (automatically) constructed schedule. Using blind expert evaluation similarity of papers scheduled to the same time slots was evaluated and papers with matching topics were counted. The process was repeated for two schedules: the actually used timetable produced manually by the

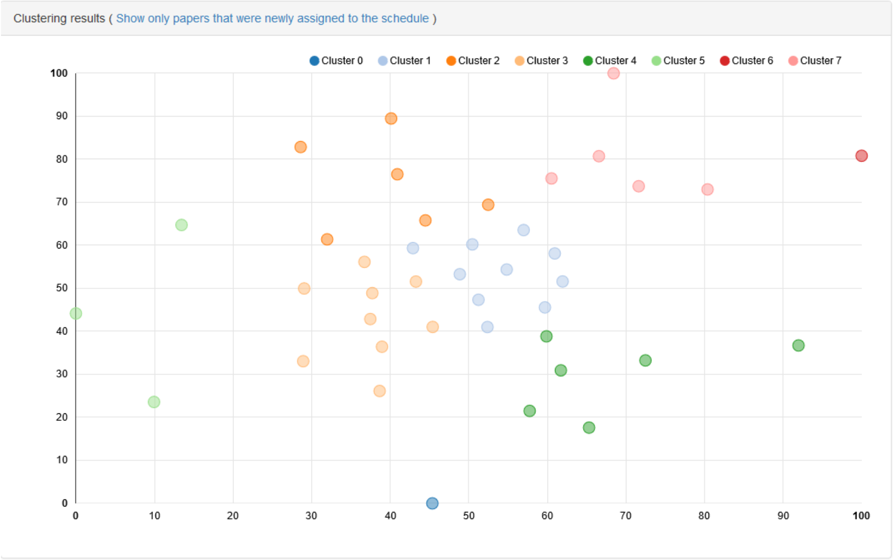

Clustering results (Show only papers that were newly assigned to the schedule)

Figure 7 A visualization of clustering papers in our web application. Each cluster contains semantically similar papers and is shown with a unique color. The distance between papers is preserved in the 2-D graph using t-SNE [19]. The user can move the mouse cursor over the dots to view the papers' titles.

conference chair and the automatically constructed schedule. We compared the number of similar papers assigned to the same slot by our method to the number of similar papers in the actual conference schedule. On average, the percentage of similar papers in schedule time slots by our method was 72%, while the actual schedule had a similarity score percentage of 82%. Our method returned useful results and will be used in further improvements.

6 Conclusion

We demonstrate a methodology which is capable to enrich textual information with graph based data and utilize both in an innovative machine learning application of clustering. The proposed solution is helpful in organization of academic conferences and presents a step towards automating one of its time consuming tasks.

We implemented a method that uses data from text documents enriched with additional information extracted from reviewers' co-bidding graphs to group similar documents and assign them to a predefined conference paper presentation schedule. We converted information from reviewers' co-bidding graphs into a BOW-compatible vector using the Personalized PageRank algorithm. We fused this vector with the BOW vector describing the textual data of abstracts and titles to get the final feature vector. We used this vector with various clustering algorithms to get clusters of similar articles. The clusters were assigned to the schedule with iterative clustering implementing predefined constraints and filling in time slots of the conference program. The method is implemented as an open source web application which supports conference chairs in creating the structure of the timetable and allows automatic conference schedule construction. The web application is freely available at `https://github.com/TadejSk/conference-scheduler`.

We evaluated our method using objective and subjective evaluation. Using silhouette score we determined that using additional graph based data increased clustering performance.

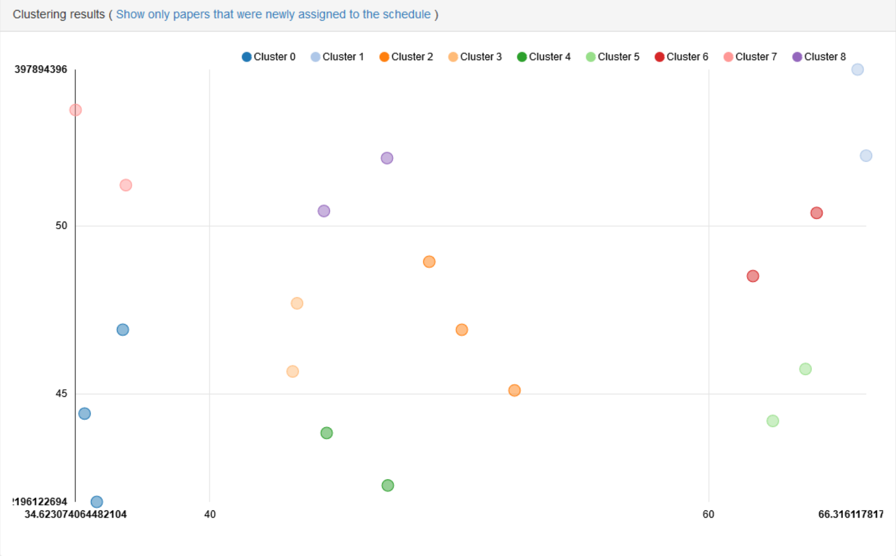


Figure 8 An example of results returned by Mean Shift clustering. There is no clear structure present in the results, and the number of returned clusters is high, with every cluster containing only three or less papers. Compared to Figure 7 the results in this figure are noticeably worse.

Subjective evaluation using an expert's opinion showed that the schedule slots returned by our method contained similar presentations.

Our approach can be further improved. The text analysis could benefit from term extraction and domain specific word weighting. Additionally, domain ontologies have been shown to be useful in semantic text mining [18] and could be used to more effectively find similar papers. The iterative constrained clustering algorithm could an also be improved with additional information extracted from whole documents and their semantic similarity. The approach also needs to be tested on larger conferences.

References

1 Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

2 David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

3 Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

4 Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

5 Keinosuke Fukunaga and Larry D Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

6 Miha Grčar, Nejc Trdin, and Nada Lavrač. A methodology for mining document-enriched heterogeneous information networks. *The Computer Journal*, 2012.

7 John A Hartigan and Manchek A Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied Statistics*, pages 100–108, 1979.

8 Tin Huynh, Kiem Hoang, Loc Do, Huong Tran, Hiep Luong, and Susan Gauch. Scientific publication recommendations based on collaborative citation networks. In *International Conference on Collaboration Technologies and Systems (CTS)*, pages 316–321. IEEE, 2012.

9 Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

10 Yicong Liang, Qing Li, and Tieyun Qian. Finding relevant papers based on citation relations. In *Web-age Information Management*, pages 403–414. Springer, 2011.

11 Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, Stanford, CA, November 1999.

12 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

13 Niels Peek, Roque Marin Morales, and Mor Peleg, editors. *Artificial Intelligence in Medicine: 14th Conference on Artificial Intelligence in Medicine, AIME 2013, Murcia, Spain*, volume 7885 of *Lecture Notes in Artificial Intelligence*. Springer, 2013.

14 Manh Cuong Pham, Dejan Kovachev, Yiwei Cao, Ghislain Manib Mbogos, and Ralf Klamma. Enhancing academic event participation with context-aware and social recommendations. In *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 464–471. IEEE, 2012.

15 Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

16 Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

17 David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1177–1178. ACM, 2010.

18 Irena Spasic, Sophia Ananiadou, John McNaught, and Anand Kumar. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in bioinformatics*, 6(3):239–251, 2005.

19 Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

20 Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proceedings of the International Conference on Machine Learning*, volume 1, pages 577–584, 2001.

21 Feng Xia, Nana Yaw Asabere, Haifeng Liu, Nakema Deonauth, and Fengqi Li. Folksonomy based socially-aware recommendation of scholarly papers for conference participants. In *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, pages 781–786. International World Wide Web Conferences Steering Committee, 2014.

22 Feng Xia, Nana Yaw Asabere, Joel JPC Rodrigues, Filippo Basso, Nakema Deonauth, and Wei Wang. Socially-aware venue recommendation for conference participants. In *Proceedings of the 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 134–141. IEEE, 2013.

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