POST: A Machine Learning Based Paper Organization and Scheduling Tool

Nathan Moore
James Madison University
Harrisonburg, VA, USA
moorena@dukes.jmu.edu

Sven Mayer
Carnegie Mellon University
Pittsburgh, PA, USA
svenmayer@cmu.edu

Kevin Molloy
James Madison University
Harrisonburg, VA, USA
molloykp@.jmu.edu

Paweł W. Woźniak
Utrecht University
Utrecht, The Netherlands
p.w.wozniak@uu.nl

William Lovo
James Madison University
Harrisonburg, VA, USA
lovowh@dukes.jmu.edu

Michael Stewart
James Madison University
Harrisonburg, VA, USA
stewarmc@jmu.edu

Companion of the 2020 ACM International Conference on Supporting Group Work, January 6–8, 2020, Sanibel Island, FL, USA
ACM 978-1-4503-6767-7/20/01.
https://doi.org/10.1145/3323994.3369892

Abstract
Organizing and assigning papers into sessions within a large conference is a formidable challenge. Some conference organizers, who are typically volunteers, have utilized event planning software to ensure simple constraints, such as two people can not be scheduled to talk at the same time. In this work, we proposed utilizing natural language processing to find the topics within a corpus of conference submissions and then cluster them together into sessions. As a preliminary evaluation of this technique, we compare session assignments from previous conferences to ones generated with our proposed techniques.

Introduction
Automated solutions for difficult problems have been at the forefront of Computer Science development since the beginning of the field. Every year, thousands of academic papers are submitted, reviewed and presented in conferences all over the world. Once accepted, the papers must be scheduled into sessions to be presented alongside similar works.

The task of scheduling the often hundreds of papers into sessions can be a daunting and tedious task, and must be undertaken by the volunteer members of the organizing committees of the host conference. These volunteers must be experts in at least the umbrella domain (if not in the

subdomains as well), in order to have reasonably related papers presented in the same session. The whole process takes a significant amount of experts' time.

This initial work employs a machine learning approach using text mining and grouping techniques to automatically classify the submitted papers into sessions. Our research question is whether we can generate an acceptable conference papers schedule based purely on the title, abstracts and author keywords.

Related Work

The current work builds on a long tradition of Participatory Design, Autobiographical Research, and Research Through Design within the ACM SIGCHI community. For several years beginning with CHI'13, Juho Kim, et al. [6] worked on "Cobi", a project to improve the process of scheduling paper presentations at technical conferences by leveraging real-time collaborative web applications and crowdsourcing [1, 2, 4–6, 10].

This work supported a relatively small group of volunteer conference organizers who shared the monumental task of scheduling technical presentations at conferences into "coherent" sessions [1].

The actual facilitation was in providing visual feedback to the user including:

- warnings when they scheduled an author to be in two places at once (an "author conflict")
- indicators of which other papers the "authorsourcing" suggests may belong with those in the session under construction [6]
- real-time collaborative web application such that multiple volunteers could work on assigning papers to sessions simultaneously

Much of this prior work is still used by many of the same technical conferences (e.g. CHI) via the Cobi web application. While the work has proven invaluable in supporting the rotating volunteers tasked with producing the schedule, it still requires that the volunteers make the paper-to-session assignments *manually*.

The current work complements the existing Cobi system, but employs unsupervised Machine Learning to help draft the paper-to-session assignments.

Machine Learning

Topic modeling, an active area of research in the field of natural language processing (NLP), accepts a corpus and utilizing unsupervised learning, clusters individual papers together by topics. This is a multi-membership clustering, where a single paper p can be assign to more than one topic $t \in T$. Given an a priori determined number of topics T, Latent Dirichlet Allocation (LDA) employs a generative probabilistic model to perform topic modeling [3]. LDA produces two products: the distribution of words used by each identified topic t, and the distribution of the topics T for each document/paper (which can be viewed as a weight associated with each topic).

Method

The proposed method accepts the the title, abstract, and author keywords for each paper. The data is then cleansed and prepared for input into LDA. The LDA-produced topic distribution for each paper is then used to compute distances between papers, facilitating hierarchical clustering of papers into sessions.

Preprocessing

The corpus is processed through the gensim.utils utility package [9] to transform words into lowercase and create the initial dictionary. A TF-IDF model is generated, allowing

for the removal of low weighted terms and stopwords (e.g. “and”, “the”), as well as words that may appear frequently across all documents in the corpus (making the term less useful in distinguishing the paper’s topic distribution). This common technique is shown to reduce “noise” in the data. Finally, the words are lemmatized (reducing words to their roots) and a final bag of words (BOW) description of the corpus is created.

Topic Modeling

A Java implementation of LDA developed at the University of Massachusetts called MALLET [7] is run on the corpus. LDA identifies the underlying topics within the corpus of papers and produces a D(document) by T(topic) matrix. In this unsupervised learning technique, topics are *not* seeded with values prior to the algorithm being performed. Each document row in the matrix corresponds to the distribution of topics for that paper. For example, a paper with a distribution [0.25, 0.60, 0.1, 0.05] over four topics would be 25% topic 0, 60% topic 1, etc. Prior to running LDA, the feature space for each paper is high dimensional (number of features equal the number of words in the created dictionary, typically $>$ 9,000). The topic vectors created for each paper by LDA can be thought of as a low-dimensional projection of each paper over the topic space, where T is usually less than 50. Distance and clustering operations are performed with this low-dimensional embedding.

Clustering and Assignments

Using the topic space vectors created by LDA, a D by D distance matrix is created using the cosine similarity measure. This distance matrix is used to perform hierarchical clustering [8]. A dendrogram (data on poster) is used to visualize how clusters of size 4 can be utilized to assign of papers into sessions. The quality of this assignment method is part of ongoing/future work.

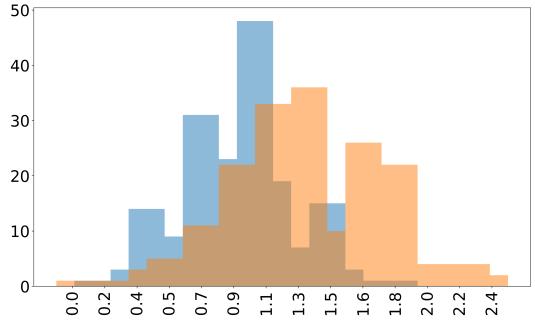

Figure 1: Distribution of historical data shown in blue, random data shown in orange and overlap shown in brown.

Preliminary Results

Our proposed method was evaluated using the data from the 2019 SIGCHI conference in order to evaluate the topic based similarity measure. For each session from the 2019 conference, we compute the sum of distances for all pairs of papers, with the expectation that the distances would in general be small. This is compared to randomly generated sessions, where the distances should be larger. The distribution for the actual sessions (light blue) and the randomly generated session (orange) are shown in Figure 1. The mass of the distribution is clearly to the left for the actual session assignments, indicating that according to the topic space similarity measure, papers in the manually-assigned session are more similar than those assigned randomly.

Conclusions and Future Work

In the preliminary results presented here, we find encouragement for our approach of autogeneration of a draft of technical conference programs, such that the “coherence” of the sessions (the relevancy of papers in a session to each other) is similar to- or better than a schedule produced manually by a volunteer domain expert.

An obvious place that future work could supplement our approach would be through validation by domain experts like those who volunteer in the scheduling role. Such a qualitative analysis could compare these volunteers' opinions of the relevancy of portions of an auto-generated schedule with their rating of historical (actual) session relevancy.

As we continue to improve our approaches for topic modeling and our measures of similarity and "coherence", we can analyze past conferences to gain insights on trends in the community's interests.

REFERENCES

[1] Paul André, Haoqi Zhang, Juho Kim, Lydia Chilton, Steven P Dow, and Robert C Miller. 2013. Community clustering: Leveraging an academic crowd to form coherent conference sessions. In *First AAAI Conference on Human Computation and Crowdsourcing*.

[2] Anant Bhardwaj, Juho Kim, Steven Dow, David Karger, Sam Madden, Rob Miller, and Haoqi Zhang. 2014. Attendee-sourcing: Exploring the design space of community-informed conference scheduling. In *Second AAAI conference on human computation and crowdsourcing*.

[3] D. M. Blei. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

[4] Lydia B Chilton, Juho Kim, Paul André, Felicia Cordeiro, James A Landay, Daniel S Weld, Steven P Dow, Robert C Miller, and Haoqi Zhang. 2014. Frenzy: Collaborative Data Organization for Creating Conference Sessions. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1255–1264. DOI: http://dx.doi.org/10.1145/2556288.2557375

[5] Juho Kim. 2016. Organic Crowdsourcing Systems. In *2016 AAAI Spring Symposium Series*.

[6] Juho Kim, Haoqi Zhang, Paul André, Lydia B Chilton, Wendy Mackay, Michel Beaudouin-Lafon, Robert C Miller, and Steven P Dow. 2013. Cobi: A community-informed conference scheduling tool. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, ACM, St. Andrews, UK, 173–182.

[7] Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. (2002). http://mallet.cs.umass.edu.

[8] Fionn Murtagh and Pedro Contreras. 2017. Algorithms for hierarchical clustering: an overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 6 (2017), e1219. DOI: http://dx.doi.org/10.1002/widm.1219

[9] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.

[10] Haoqi Zhang, Paul André, Lydia B Chilton, Juho Kim, Steven P Dow, Robert C Miller, Wendy E Mackay, and Michel Beaudouin-Lafon. 2013. Cobi: Communitysourcing Large-scale Conference Scheduling. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 3011–3014. DOI: http://dx.doi.org/10.1145/2468356.2479597