PRESS: PeRsonalized Event Scheduling recommender System

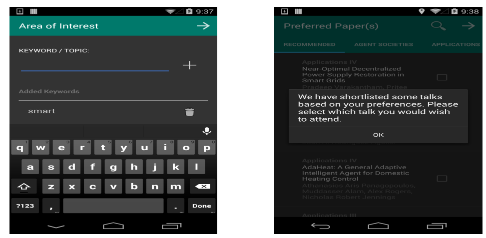
(Demonstration)

Hoong Chuin Lau, Aldy Gunawan, Pradeep Varakantham, Wenjie Wang
Singapore Management University
80 Stamford Road, Singapore 178902
{hclau, aldygunawan, pradeepv, wjwang}@smu.edu.sg

ABSTRACT

This paper presents a personalized event scheduling recommender system, PRESS, for a large conference setting with multiple parallel tracks. PRESS is a mobile application that gathers personalized information from a user and recommends talks/demos to be attend. The input from a user include a list of keyword preferences and (optionally) preferred talks. We use the MALLET topic model package to analyze the set of conference papers and classify them based on automatically identified topics. We propose an algorithm to generate a list of recommended papers based on the user keywords and the MALLET topics. An optimization model is then applied to obtain a feasible schedule. The recommended set is matched against the selected papers by the user which we obtained from a survey conducted at AAMAS-15 in Istanbul, Turkey. We show that PRESS is able to provide reasonable accuracy, precision and recall rates. PRESS will be deployed live during AAMAS-16 in Singapore.

Keywords: Recommender system; topic model; conference scheduling



(a) Area of Interest (b) Selected Papers

Figure 1: Screenshots of PRESS

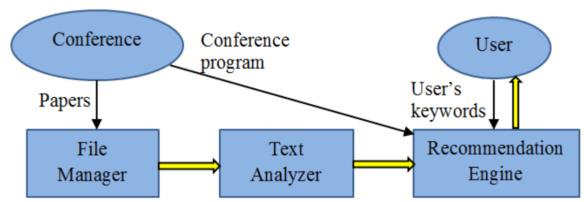


Figure 2: System Architecture

1. INTRODUCTION

In a large conference or tradeshow setting where talks are presented in parallel tracks across multiple days, it is a challenge for a conference attendee to generate a plan of talks/sessions to attend that optimize his/her preferences. And this is particularly cognitively challenging if the conference venue is large, where one may need to consider time to travel between talks. Furthermore, the system will be most helpful if a given talk/session is scheduled in more than one timeslots during the conference, and one has to decide which timeslot is best to attend that talk.

We will present a demo of a personalized recommender system where different people who use the recommender system will expect to get recommendations and plans based on their own preferences [2]. Our personalized recommender system PRESS is realized in an Android mobile application.

PRESS uses the *MA*chine *L*earning for *L*anguag*E* *T*oolkit MALLET [1] that navigates large bodies of information by finding clusters of keywords that frequently appear together, called topics. The underlying idea of topic models is that documents are made up of mixtures of topics, where a topic is a cluster of words that frequently occur together [3]. By using contextual clues, topic models connect words with similar meanings and distinguish between uses of words with multiple meanings. Each document is processed by selecting a distribution over topics, and then generating each keyword at random from a topic by using the selected distribution.

In order to generate personalized recommendations, we first build a topic model using MALLET from the list of papers to be presented at the conference. Then given the profile of a user (which he enters using the app) which consists of his preference keywords and (optionally) a list of selected papers, PRESS generates a list of recommended papers and a feasible (i.e. non-conflicting) schedule for this user that optimizes his utility score. Figure 1 shows the screenshots of PRESS. A demo of this app can be accessed via `http://bit.ly/1R5qV4p`.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. ARCHITECTURE AND SYSTEM DESIGN

PRESS consist of three components: file manager, text analyzer and recommendation engine, as shown in Figure 2. The details are given in the following subsections.

A	B	C	D	E	F
0	crowdsourcing	crowdworkers	planner	social games	requester
1	team	dishonest advisors	crowd robotics	attractions	routing
2	complex returns	committee	participation	manipulation	ranks
3	sensor motes	online planning	finite-state controllers	planning	deployment
4	theorem	logic	preference	modal logic	reason
5	customers	opinion	signals	customer	population
6	flock	cybersecurity	fluencing agents	trust	reconnaissance agent
7	advertisers	government	market	dual preference property	bid
8	metrics	antibody	synchronization	virus	maximization
9	learning	reinforcement learning	capability models	state space	policy transfer
10	mechanism design	leader	driver	landfill trash	leader equilibria
11	hedonic games	fractional hedonic games	utility	clustering	pareto efficiency

Figure 3: Screenshot of MALLET output

#doc name topic proportion ...											
0	Paper 1	10	0.821207	15	0.25842	23	0.061721	13	0.03214	0	0.017127
1	Paper 2	10	0.726088	15	0.249853	23	0.082568	18	0.068656	13	0.052124
2	Paper 3	10	0.870099	15	0.226938	3	0.051107	13	0.038831	19	0.010294
3	Paper 4	10	0.797018	15	0.26128	23	0.091565	13	0.047208	17	2.63E-04
4	Paper 5	23	0.737454	10	0.30123	15	0.224803	13	0.033752	17	2.48E-04
5	Paper 6	22	0.839755	15	0.200866	13	0.065629	10	0.053941	17	0.035124
6	Paper 7	24	0.741729	15	0.154617	13	0.075463	23	0.020853	16	0.004314

Figure 4: Screenshot of topic composition

2.1 File Manager

This component is responsible for converting a collection of documents (eg. pdf files) into text files and then tagging the part of speech of words in these text files.

Given a set of papers in the pdf format, the PDFMINER tool extracts the words from each file and output them into a text file. The Illinois Chunker (`https://cogcomp.cs.illinois.edu/page/software_view/Chunker`) is used to identify the semantically related words by assigning different tags. For example, in the noun words ”reinforcement learning”, the word ”reinforcement” is identified as the beginning word of a noun phrase and therefore tagged with B-NP (begins a noun phrase), however, the following word ”learning” is identified inside the same noun phrase as ”reinforcement” and therefore tagged with I-NP (inside a noun phrase).

2.2 Text Analyzer

The MALLET topic model package [1] is used to extract a set number of topics and the highest frequent words for each topic from the text documents and output the statistics of each extracted topic for each text document. MALLET allows us to filter a standard list of English stop-words from documents before processing. Unfortunately, we cannot edit the contents of this list without modifying code and recompiling. In order to rule out some trivial words, we create an extra-word file containing those trivial words. Figure 3 shows the screenshot of the MALLET output. There are 12 topics generated with 5 keywords for each topic.

The topics that compose each document including the statistics of each topic can be seen in Figure 4. For example, PAPER 1 has topic 10 as its principal topic, at about 82.1%; topic 15 at 25.8 % and so on. The topic model also suggests a connection among documents that might not at first have suspected. Papers 1, 2, 3 and 4 have topic 10 as their principal topic.

2.3 Recommendation Engine

The recommendation engine generates a list of recommended papers based on keywords entered by a user and the MALLET results. It is made up of two key components: the ranking algorithm and the optimization model.

2.3.1 *Ranking Algorithm*

Based on the MALLET outputs, we calculate the utility score of each paper with respect to the preference keywords provided by a user. We first compare how many keywords are matched with a set of keywords of each topic generated by MALLET. The utility score for each paper is calculated by finding the weighted sum of the number of keywords and utility scores for each topic.

All papers are then sorted in descending order with respect to their utility scores. The recommendation is given from the top $x\%$ of papers. This list of papers would be compared with the selected papers by a user. Take note that some papers may clash and we allow user to decide which one(s) would be attended.

2.3.2 *Optimization Model*

An optimization model is used to generate a feasible schedule. The objective is to maximize the total utility score. By considering the number of days of a conference, time slots and parallel sessions, we ensure that at each time slot, at most one paper presentation is attended. Our model also allow the user to select ”must-go” papers and ”must-skip” papers. For conferences that span a large geographical space, it is easy to extend our current model in future by considering travel time between talks.

3. RESULTS

In order to verify the goodness of PRESS, a user survey was conducted during the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-15) 2015 in Istanbul, Turkey. We managed to collect 45 respondents from the AAMAS-15 participants. Each respondent was asked to specify his/her preference keywords together with the list of talks he/she would be interested to attend. This collection of surveys serve as the ground truth. Due to a short time taken for each survey, we assume that a user will not be able to exhaustively select all preferred papers. Hence, based on a set of selected papers, we include an additional set of selected papers which have high correlation values with those papers (e.g. 0.75). All those papers are considered as the papers selected by a user.

By comparing equal number of user-selected papers and recommended papers generated by PRESS, our experimental results show that the accuracy, precision and recall values of PRESS are 92.2%, 58.6% and 58.6%, respectively, with the cutoff value of top 20% for the recommendation. Interested readers are welcome to visit our demo booth at AAMAS-16, and have their mobile devices deployed with the app.

Acknowledgements

This research is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative as well as its Corp Lab@University scheme. The authors wish to thank people who have contributed to this project: Shih-Fen Cheng, Oi Mei Wong, Chun Pong Fan, Elizabeth Lim, Eng Chye Low, Firmansyah Abd Rahman, Xiang Li, Htar Htar Hlaing & Khin Aye Mon.

REFERENCES

[1] A. K. McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[2] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3), Mar. 1997.

[3] M. Steyvers and T. Griffiths. Probabilistic Topic Models. In *Latent Semantic Analysis: A Road to Meaning*. Hillsdale, NJ: Erlbaum, 2007.