ECAI 2016
G.A. Kaminka et al. (Eds.)

doi:10.3233/978-1-61499-672-9-1797

An Intelligent System for Personalized Conference Event Recommendation and Scheduling

Aldy GUNAWAN and **Hoong Chuin LAU** and **Pradeep VARAKANTHAM** and **Wenjie WANG** [1]

Abstract.
Many conference mobile apps today lack the intelligent feature to automatically generates optimal schedules based on delegates' preferences. This entails two major challenges: (a) identifying preferences of users; and (b) given the preferences, generating a schedule that optimizes his preferences. In this paper, we specifically focus on academic conferences, where users are prompted to input their preferred keywords. Our key contribution is an integrated conference scheduling agent that automatically recognizes user preferences based on keywords, provides a list of recommended talks and optimizes user schedule based on these preferences. To demonstrate the utility of our integrated conference scheduling agent, we first demonstrated the app in the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2015) and conducted a survey to collect some data, which are used to verify the results presented in this paper. It is able to provide well calibrated results with respect to precision, accuracy and recall. We also tested the app in the 2015 WI-IAT International Conference (Singapore). The android and web-based apps have been demonstrated and deployed in AAMAS 2016 (Singapore) with positive responses from the users.

1 Introduction

In a large conference setting where talks are presented in parallel sessions across multiple days, it is challenging for a conference attendee to generate a plan of talks to attend that optimize his/her preferences. Furthermore, this adds to the cognitive challenge if the conference venue is large, where one may need to consider time to travel between talks. To reduce this cognitive load, we aim to provide an integrated conference scheduling agent that not only identifies user preferences (based on keywords) but also generates a schedule of talks to attend at different times of the conferences while considering the user preferences. We are specifically interested in academic conferences where data associated with users is easily available.

Both the individual problems (understanding user preferences and optimizing schedule accounting for preferences) have received significant interest in existing work. The first thread of related research is with respect to learning user preferences given papers has been studied extensively in the machine learning community. Statistical topic modeling has become a popular method for analyzing large sets of text collections by representing high dimensional data in a low dimensional subspace [21]. The topic model is built using MALLET, which is introduced by Andrew McCallum and his team in 2002 [10]. MALLET is able to navigate large bodies of information by finding clusters of keywords that frequently appear together, called topics.

The second thread of related work is with respect to optimizing preferences given constraints on scheduling talks. This problem is related to a single resource scheduling problem with the objective of maximizing the profitability of the resulting schedule under fixed processing times [20].

One of the best known systems in the area of academic conference event recommendation is Conference Navigator 3.0 [13]. In Conference Navigator system, users directly select preferred talks. It also collects the wisdom of the user community and makes it available through community-based recommendation interface to help users in making scheduling decisions.

Our key contribution is in providing an integrated solution for both these problems and demonstrate utility on a real conference scheduling problem. Specifically, we first employ MALLET to identify the topics of interest for a given conference, by considering papers from that conference. We then identify preferences of a given user for the topics of interest at the conference by getting the user's preferred keywords. PRESS also considers community-based recommendation in terms of the correlation among talks. These correlation values are calculated automatically based on their similarity in terms of keywords provided by the users. Based on preferred keywords, PRESS provides a list of recommended talks and optimizes user schedule based on these preferences.

For easy interaction with the users, our agent is built as an application for mobiles, namely PRESS. So, we are able to take change requests on the generated schedule and immediately provide an updated schedule. To demonstrate utility for conference attendees, we first demonstrated PRESS in the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2015) and conducted a survey to collect some data, which are used to verify the results presented in this paper. We show that the papers generated in the schedules for the users have high values of precision, accuracy and recall. We then tested PRESS in the 2015 WI-IAT International Conference (Singapore). Some feedbacks especially related to the client-facing android mobile app were collected. Finally, both android and web-based versions of PRESS have been deployed in AAMAS 2016 [7].

2 Related Work

Resnick and Varians [15] describe a recommender systems as follows: *In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations.*

[1] Singapore Management University, Singapore, email: {aldygunawan, hclau, pradeepv, wjwang}@smu.edu.sg

Adomavicius and Tuzhilin [1] provide a survey of the-state-of-the art and possible extensions of the recommender system. Burke et al. [4] describe two basic principles of a recommender system: a) it is personalized to optimize the experience of one user, and b) it is intended to help the user choose among discrete options. Recommender systems have been developed in various domains of applications, such as LIBRA [11] (book recommender) and INTIMATE [9] (movie recommender).

Lops et al. [8] describe two main paradigms of recommender systems. *Content-based* recommender systems generate recommended items based on items that have been liked by a user in the past, whereas *Collaborative* recommendation systems try to recommend items from other users whose preferences are similar to those of the user and recommend items they have liked. In this paper, we concentrate purely on content-based recommendation since our collected data is from a small community of users.

One method that have been used in content-based recommendation is Latent Dirichlet Allocation (LDA). LDA is a fully generative probabilistic topic model. Probabilistic topic models play an important rule in order to capture latent topical information from a large collection of data [12]. The basic underlying idea of probabilistic topic models is documents are mixtures of topics, where a topic is a cluster of words that frequently occur together [17]. By using contextual clues, topic models connect words with similar meanings and distinguish between uses of words with multiple meanings.

MALLET provides an option to use a previously generated inference file as an inference tool [10]. It uses LDA. Each document is produced by selecting a distribution over topics, and then generating each keyword at random from a topic chosen by using the selected distribution. [21] implement different methods for topic inference, such as Gibbs sampling and SparseLDA in the MALLET toolkit on streaming two different sets of documents, 13 years of full papers published in the NIPS conference and a set of journal article abstracts from Pubmed. Other applications of MALLET are in analyzing a set of personal emails [19] and a set of ratings collected on Amazon Mechanical Turk [6].

Sampson [16] introduces ”preference-based” conference scheduling (PBCS) problem. Instead of looking at the conference scheduling problem as a classical scheduling problem, the problem is treated from the customer point of view with the main objective is related to a customer-satisfaction. Other works related to the conference scheduling problem can be referred to [14, 18].

Bhardwaj et al. [3] introduce COBI as the most recent web-based, visual scheduling interface in planning a large-scale conference. COBI engages the community to play an active role in the planning process. A process that collects input from attendees and considers them as preferences and constraints in the planning process. To the best of our knowledge, no existing work incorporates the optimization mathematical model in the process of providing the recommendation papers.

3 The Proposed Approach

The overall architecture of PRESS is depicted in Figure 1. PRESS consists of four main components: Native android application (Front-End), Back-end Engine, Optimization Engine and Text Analyzer.

In the following, we provide the formal definition and formulation of the problem in the context of a large academic conference. We further explain the MALLET implementation in Text Analyzer component and two different proposed algorithms in the Optimization Engine component.

3.1 Problem Formulation

A conference consists of a set of main sessions where each main session is scheduled on one particular time period (e.g. from 09.00 - 10.00 am). In most large conferences, each main session is divided into a set of parallel sessions. We assume that each parallel session is scheduled in a particular room. Figure 2 shows an example of a conference setting on a particular day.

Let *P* be a set of papers that will be presented during a conference. Each parallel session consists of a number of talks. In order to generate a schedule that possibly contains talks across sessions, we divide each time period into multiple number of time slots (e.g. every 15 minutes). Each time slot will have one talk and only one paper $i \in P$ would be presented in that time slot for that session. We also assume that each paper will only be presented once throughout the conference. We implement MALLET to generate a set of topics T from P. Each topic $j \in T$ contains a set of keywords W_j^1 that is likely to appear together in topic j [17]. We assume that $|W_j^1| = |W^1|$ ($\forall j \in T$). See Figure 3 for an illustration.

Some methodological issues faced when using MALLET, such as how to determine the values of $|T|$ and $|W^1|$, affect the quality of the outputs. At the moment, the best way to determine the values of $|T|$ and $|W^1|$ is to run multiple analyses with different values of both and comparing the results that seem to fit ”best” [2].

In summary, MALLET generates two different outputs (Figures 4 and 5) that would be kept in the database and used as inputs for the optimization engine:

- $\mathbf{M}_{|T| \times |W^1|} = [w_{jk}^1]$, where w_{jk} represents keyword k of topic j ($\forall j \in T, k \in W_j^1$).
- $\mathbf{U}_{|P| \times |T|} = [u_{ij}]$ where u_{ij} represents the utility score of paper i related to topic j ($\forall i \in P, j \in T$).

Let W_i^2 be the set of keywords stated on paper $i \in P$. As mentioned in Section 1, we consider both keywords generated by MALLET and from papers directly and both would be kept in the database.

3.2 MALLET Implementation

The Text Analyzer component consists of two sub-components: the PDFMINER tool and the MALLET topic model package. Take note that both sub-components: PDFMiner and MALLET, are run offline and generated results would be kept in the database. PDFMiner (`https://pypi.python.org/pypi/pdfminer/`) is a tool for extracting information from PDF documents. This sub-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.

In most cases, information has no structure, some pre-processing steps are required to convert unstructured information and extract structured relevant information. 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). Likewise, other types of phrases such as a verb phrase will be tagged with B-VP (begins a verb phrase) and I-VP (inside a noun phrase), respectively.

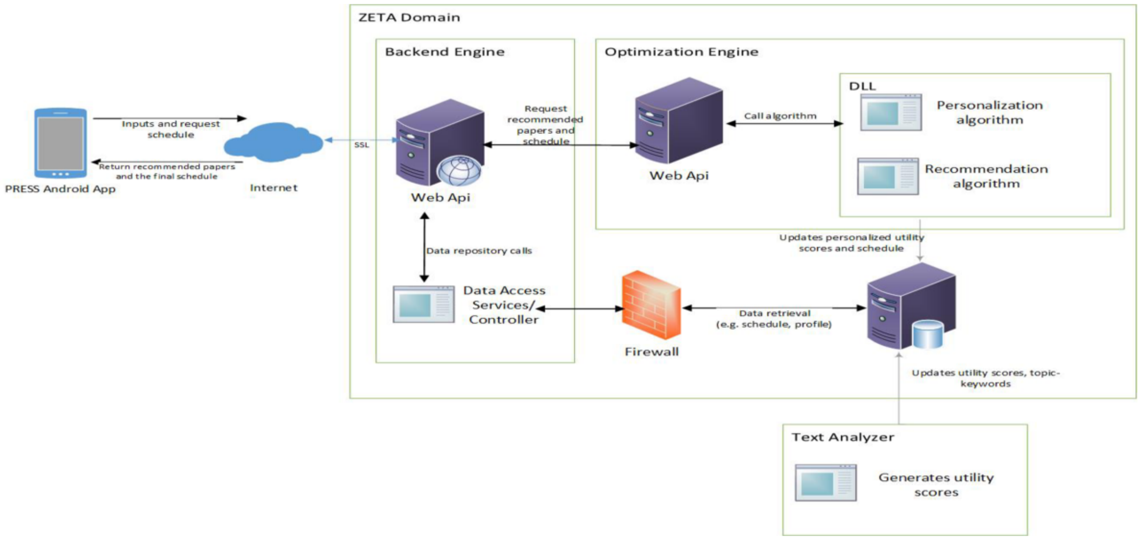


Figure 1: System Architecture of PRESS

		Room			
Time Period	Main Session	Room 1	Room 2	Room 3	Room 4
08.00 - 09.00	Session#1	PS1	PS2	PS3	PS4
09.00 - 10.00	Session#2	PS1	PS2	PS3	PS4
10.00 - 10.30	Coffee break				
10.30 - 11.30	Session#3	PS1	PS2	PS3	PS4
11.30 - 12.30	Session#4	PS1	PS2	PS3	PS4
12.30 - 14.30	Lunch time				

PS: parallel session

Figure 2: Example of conference setting

	Parallel Session			
Time Slot	Room 1	Room 2	Room 3	Room 4
08.00 - 08.15	Paper#1	Paper#2	Paper#3	Paper#4
08.15 - 08.30	Paper#5	Paper#6	Paper#7	Paper#8
08.30 - 08.45	Paper#9	Paper#10	Paper#11	Paper#12
08.45 - 09.00	Paper#13	Paper#14	Paper#15	Paper#16

Figure 3: Example of talks in a particular time period

A	B	C	D	E	F
0	leader	trust	landfill trash	society	merchant
1	complex returns	bisimulation	autonomous mobile agents	normative systems	collective behaviour
2	cybersecurity	ranking	envy-free division	ranks	manipulation
3	theorem	preference	modal logic	logic	reason
4	tactics	crowd robotics	robot	control	team
5	election	vote	manipulation	voter	maximin
6	planning	endogenous discounting	learning	complexity	judgment aggregation
7	autonomy	learning	platform	planning	attack surface
8	reasoning	analysts	knowledge base	defects	complexity
9	technology	violation	obligations	guard function	shared information
10	mechanism design	market	advertisers	government	partial verification

Figure 4: 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 5: Screenshot of topic composition

The second sub-component, the MALLET topic model package [10], is used to extract a set 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 4 shows the screenshot of the MALLET output. There are 11 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 5. 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.

3.3 Proposed Algorithms

Given a set of keywords K that the user is interested in and the results of MALLET tools, we calculate the personalized utility score for each talk and generate a list of recommended talks.

Personalization Algorithm

We present the personalization algorithm for providing a list of recommended talks, as shown in Algorithm 1. The objective is to calculate $\tilde{u}_{ij}$,the modified utility score of paper $i \in P$ related to topic $j \in T$, with respect to the set of keywords K given by the user. We compare the number of keywords $|K|$ which are matched with a set of keywords W_j^1 of topic j, represented as Tot_j $(\forall j \in T)$. For each paper i, the utility score u_{ij} is multiplied by Tot_j in order to get the value of $\tilde{u}_{ij}$. Finally, we calculate the total personalized utility score of paper i, $TotU_i = \sum_{j \in T} \tilde{u}_{ij}$ $(\forall i \in P)$ (LINES 1 - 18).

The next step is to compare K with the keywords from paper i, W_i^2 $(\forall i \in P)$. If a match exists, the value of $TotU_i$ will added by one for each matched keyword (LINES 19 - 27). For each user, all papers would be sorted in descending order with respect to the values of $TotU_i$ (LINES 28 - 29). The recommendation is given from the top $x\%$ of papers. This is a naive way in order to provide a list of recommended talks without considering possible conflicts.

The user will then select or remove some talks from the list. Those selected talks would be in the "must-go" and "must-skipped" lists, respectively. PRESS continues to call the recommendation algorithm in order to provide the final schedule that maximizes the total personalized utility score and ensures there is no conflicts among talks.

Algorithm 1 Personalization Algorithm

for $h = 1\ to\ |K|$ **do**
  for $j = 1\ to\ |T|$ **do**
    $Tot_j = 0$
    for $k = 1\ to\ |W^1|$ **do**
      if (h^{th} keyword from the user is matched with k^{th} keyword of topic j) **then**
        $Tot_j + = 1$
      end if
    end for
  end for
end for
for $i = 1\ to\ |P|$ **do**
  for $j = 1\ to\ |T|$ **do**
    $\tilde{u}_{ij} = Tot_j \times u_{ij}$
  end for
end for
for $i = 1\ to\ |P|$ **do**
  $TotU_i = \sum_{j \in T} \tilde{u}_{ij}$
end for
for $h = 1\ to\ |K|$ **do**
  for $i = 1\ to\ |P|$ **do**
    for $k = 1\ to\ |W^2|$ **do**
      if (h^{th} keyword from the user is matched with k^{th} keyword of paper i) **then**
        $TotU_i + = 1$
      end if
    end for
  end for
end for
Rank all papers based on $TotU$ values in the descending order
return the top $x\%$ of papers

Recommendation Algorithm

In the recommendation algorithm, we introduce a mathematical model to formulate the scheduling problem. The time slots of talks are taken into consideration in this model. The mathematical programming model is solved by the commercial solver CPLEX Optimization Studio 12.6.1.

The scheduling problem is defined as follows. We define $MUST$ and $SKIP$ as "must-go" and "must-skip" lists, respectively. Let assume the conference is held within a set of days D. Each day $d \in D$ is divided into a set of time slots S_d. Each time slot $s \in S_d$ on day $d \in D$ consists of a set of parallel sessions N_{ds}. A talk would be held in one parallel session at each time slot.

The decision variable X_{dsn} is a binary variable. Its value equals to 1 if a talk in parallel session n on day d at time slot s is selected.

$$Maximize \sum_{d \in D} \sum_{s \in S_d} \sum_{n \in N_{ds}} \hat{u}_{dsn} \times X_{dsn} \tag{1}$$

The objective function (1) is to maximize the total personalized utility score of selected talks. Let $\hat{u}_{dsn}$ is the utility score of the talk in parallel session $n \in N_{ds}$ on day $d \in D$ at time slot $s \in S_d$. The utility scores are collected from $TotU_p$ ($p \in P$) values with respect to the time slot. For example, if paper p_1 is presented on Day 1, time slot 1 and parallel session 1, the value the talk $\hat{u}_{111} = TotU_{p_1}$.

$$\sum_{k \in N_{ds}} X_{dsn} \leq 1 \;\; \forall d \in D, s \in S_d \tag{2}$$

Equation (2) ensures that at each time slot, only one talk is attended.

$$X_{dsn} = 1 \;\; \forall (d, s, n) \in MUST \tag{3}$$

Equation (3) ensures that talks in the "must-go" list, $MUST$, are attended.

$$X_{dsn} = 0 \;\; \forall (d, s, n) \in SKIP \tag{4}$$

Equation (4) enforces that talks are in the "must-skip" list, $SKIP$, would not be attended since they are out of the user interest.

$$X_{dsn} \leq M \times \hat{u}_{dsn} \;\; \forall d \in D, s \in S_d, n \in N_{ds} \tag{5}$$

Equation (5) guarantees that only talks with non-zero personalized utility scores would be selected. Let M be a very large number.

4 Architecture and System Design

Figure 1 illustrates the various individual components and their interactions. All communications among main components are implemented by using RESTful web service published on one of Singapore University Management servers, called ZETA server.

Android Application (Front-end Engine)

This is a client-facing android mobile app that allows a user to enter preferred keywords, view recommended talks, select preferred talks (indicated as "must-go"), remove non-preferred talks (indicated as "must-skip") and view the final schedule. This component serves as an interface for the user to construct the user profile. All information provided by the user will be sent to the back-end engine.

Back-end Engine

This component is responsible for coordinating and delegating tasks between the front-end and the optimization engines. The back-end engine is also responsible for storing and retrieving all information related to the conference in the database, including keywords from papers and text analyzer outputs.

First, it collects the user-profile from the front-end engine and pass it to the optimization engine. The optimization engine will call the personalization algorithm in order to generate a list of recommended talks. This list would be passed back to the front-end engine so the user can indicate and select his preferred talks ("must-go") and remove some non-preferred talks (must-skip").

The back-end then consolidates "must-go" and "must-skip" lists together with other information from database, such as conference schedule, and passed to the optimization engine. The recommendation algorithm will be called in order to generate the final schedule. At the end, the back-end engine pass back the final schedule to the front-end engine an display it to the user.

Optimization Engine

The optimization engine consists of two algorithms: personalization and recommendation algorithms. As described in Section 3, this component interacts with the back-end engine in order to generate the list of recommended papers and the final schedule.

5 Experimental Results

5.1 User Study Details

PRESS was first demonstrated during the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-15) which was held from 4 - 8 May 2015 in Istanbul, Turkey. The conference consists of 6 main sessions. Each main sessions is labeled by an alphabet which represents a particular time period, e.g. main session B is held on Wednesday (6 May 2015) from 11.00 -

12.30. Each main session is further divided into 5 different parallel sessions, numbered from 1 - 5. Each talk is given a predetermined time slot (e.g. 15 minutes). In total, there are 166 talks. Each parallel session is related to one of particular research area/topic, such as Game Theory, Applications and others. The detailed schedule, including the information about the papers, can be found in `http://www.aamas2015.com/en/program.asp`.

In order to verify the effectiveness of PRESS, a user survey was conducted at AAMAS-15. We collected 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 and would be used for analysis purpose.

5.2 System Components

We also tested the app in the 2015 WI-IAT International Conference. Some feedbacks especially related to the client-facing android mobile app (e.g. the design of a sign-up page, the layout and so on) have been collected. We include some final screenshots for the Android app. The opening screen requests the user either to sign in or to register (Figure 6(a)). The registration is required for the first times (Figure 6(b)). The user also needs to agree with the terms and conditions of the app (Figure 6(c)). Figure 6(d) summarizes the profile of the registered user.

Figure 7(a) shows the screen for the user to input the preferred keywords. Once the arrow button on the right top corner is clicked, the list of recommended talks which are generated by the personalization algorithm (Algorithm 1) would be displayed. The user then select and remove some talks. Those would be treated as ”must-go”and ”must-skip” lists, as shown in Figure 7(b). The details of one particular talk can also be displayed (Figure 7(c)). All those information would be sent back to the back-end engine and the recommendation algorithm would be called. Finally, the final schedule for each day would be displayed, as seen in Figure 7(d).

5.3 Insights

After demonstrating PRESS and conducting a survey at the AAMAS-15, we analyze the goodness of PRESS in recommending the list of talks. Out of 14 research areas, the top three most selected areas are Application, Game Theory and Learning which cover up to 42%. Due to a short time taken for each survey, we assume that a user will not be able to exhaustively select all preferred talks. Hence, based on a set of selected talks, we include an additional set of selected talks which have high correlation values with those talks. All those talks are considered as the talks selected by a user as well. The higher the correlation value is, the more similar two papers are in terms of topics including keywords generated. The correlation between two talks is calculated using the Cosine Coefficient formula:

$$\cos(i, i') = \frac{\sum_{j \in T} u_{ij} u_{i'j}}{\sqrt{\sum_{j \in T} u_{ij}^2}\sqrt{\sum_{j \in T} u_{i'j}^2}} \quad \forall (i, i') \in P \tag{6}$$

We evaluate the performance of PRESS by comparing three statistical measures: accuracy, precision and recall rates. The accuracy is the proportion of true results (true positives and true negatives) among the total number of cases examined. Precision (positive predictive value) is the fraction of retrieved cases that are relevant, while recall (sensitivity) is the fraction of relevant cases that are retrieved. Precision can be seen as a measure of quality, whereas recall is a measure of quantity.

By setting the numbers of user-selected papers from the ground truth and recommended papers generated by PRESS to a cut-off of top 10% $\times$ 166 talks which equals to 16 talks with the highest total personalized utility scores and a cut-off correlation value (e.g. 0.75), our experimental results show that the accuracy, precision and recall rates of PRESS are 92.02%, 58.61% and 58.61%, respectively. Other results with different cut-off correlation values can also be seen in Table 1.

We conclude that the higher the cut-off correlation value, the lower the values of measures are. It is intuitive correct since the selected talks by the user during the survey would be fewer. If we do not include talks with high correlation values, the three measures are much lower since the users are not aware with similar talks.

Table 1: Statistical measures

Correlation value	Measure		
	Accuracy	Precision	Recall
0.75	92.02%	58.61%	58.61%
0.80	91.73%	57.08%	57.08%
0.85	91.62%	56.53%	56.53%
0.90	91.57%	56.25%	56.25%
0.95	90.87%	52.64%	52.64%

6 Conclusion

We introduce a personalized event scheduling recommender system, PRESS. PRESS is an android mobile app that gathers personalized information from a user and recommends talks. Although there is a bunch of recommender systems in different domains, so far as we are concerned that PRESS is the first android app that incorporates an optimization model for generating a feasible schedule.

We demonstrated PRESS at AAMAS-15 in Istanbul, Turkey. The generated predictions by PRESS is compared against the ground truth. We observe that PRESS achieves reasonable accuracy, precision and recall rates. Some feedbacks have also been collected during the 2015 WI-IAT conference. We have also deployed the android and web-based versions of PRESS during AAMAS-16 (Singapore). Positive responses have been given by around 140 users.

The current version of PRESS uses the direct keyword matching among keywords generated by MALLET and provided by the user. We will consider more advanced techniques which allow going beyond the direct keyword matching. We also consider other possible scenarios. Some talks may be scheduled in more than one timeslot so the attendee has to decide which timeslot should be attended. This is related to the capacity constraint of rooms which is currently negligible. Other combinations of precision/recall (e.g. product of criteria or using a varying linear coefficient) will also be included for our future work.

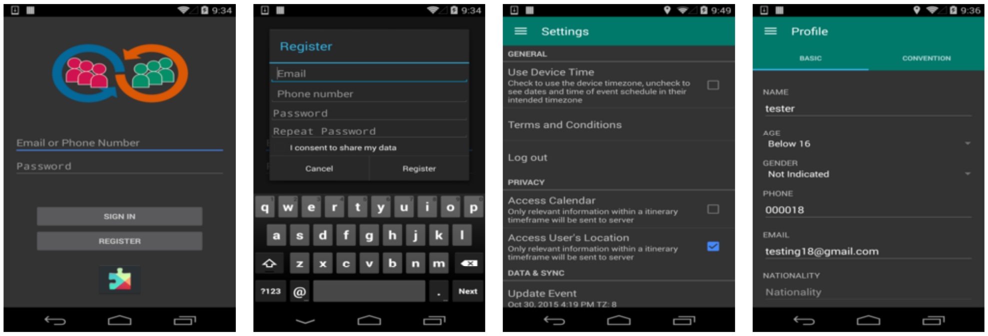
ACKNOWLEDGEMENTS

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, Khin Aye Mon & Htar Htar Hlaing.

REFERENCES

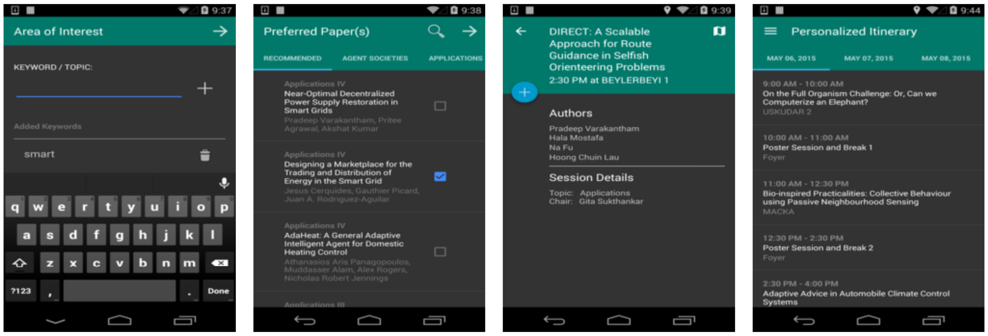
[1] G. Adomavicius and A. Tuzhilin, ‘Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions’, *IEEE Transactions on Knowledge and Data Engineering*, **17**(6), 734–749, (2005).

[2] L. AlSumait, D. Barbará, J. Gentle, and C. Domeniconi, ‘Topic signif-



(a) Front screen (b) Registration (c) Setting (d) Basic profile

Figure 6: Screenshots of PRESS showing: a) the opening screen, b) the registration screen, c) settings of the app and d) the user profile



(a) Keywords selection (b) Recommended papers (c) Paper details (d) Schedule

Figure 7: Screenshots of PRESS showing: a) the list of selected keywords, b) the list of recommended talks, c) the details of one particular talk and d) the final schedule

icance ranking of LDA generative models', in *Machine Learning and Knowledge Discovery in Databases*, eds., W. Buntine, M. grobelnik, D. Mladenić, and J. Shawe-Taylor, volume 5781 of *Lecture Notes in Computer Science*, 67–82, Springer, (2009).

[3] A. P. Bhardwaj, J. Kim, S. Dow, D. R. Karger, S. Madden, R. Miller, and H. Zhang, 'Conference scheduling: a personalized approach', in *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2014)*, pp. 2–10, (2014).

[4] R. Burke, A. Felfernig, and M. H. Goker, 'Recommender systems: an overview', *AI Magazine*, **32**(3), 13–18, (2011).

[5] O. Celma and X. Serra, 'FOAFing the music: bridging the semantic gap in music recommendation', *Web Semantics*, **6**(4), 250–256, (2008).

[6] J. Chuang, S. Gupta, C. Manning, and J. Heer, 'Topic model diagnostics: Assessing domain relevance via topical alignment', in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, eds., S. Dasgupta and D. Mcallester, volume 28, pp. 612–620, (2013).

[7] H. C. Lau, A. Gunawan, P. Varankantham, and W. Wang, 'PRESS: PeRsonalized Event Scheduling recommender System (Demo track)', in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, (2016).

[8] P. Lops, M. de Gemmis, and G. Semeraro, 'Content-based recommender systems: state of the art and trends', in *Recommender Systems Handbook*, eds., F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 73–105, Springer, (2011).

[9] H. Mak, I. Koprinska, and J. Poon, 'INTIMATE: a web-based movie recommender using text categorization', in *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pp. 602–605. IEEE Computer Society, (2003).

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

[11] R. J. Mooney and L. Roy, 'Content-based book recommending using learning for text categorization', in *Proceedings of the 5th ACM Conference on Digital Libraries*, pp. 195–204. ACS Press, (2000).

[12] M. Ovsjanikov and Y. Chen, 'Topic modeling for personalized recommendation of volatile items', in *Machine Learning and Knowledge Discovery in Databases*, eds., J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, volume 6322 of *Lecture Notes in Computer Science*, 483–498, Springer, (2010).

[13] D. Parra, W. Jeng, P. Brusilovsky, C. Lopez, and S. Sahebi, 'Conference Navigator 3.0: an online social conference support system', in *Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP 2012)*, (2012).

[14] J. Quesnelle and D. Steffy, 'Scheduling a conference to minimize attendee preference conflicts', in *Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2015)*, pp. 379–392, (2015).

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

[16] S. E. Sampson, 'Practical implications of preference-based conference scheduling', *Production and Operations Management*, **13**(3), 205–215, (2004).

[17] M. Steyvers and T. Griffiths, 'Probabilistic Topic Models', in *Latent Semantic Analysis: A Road to Meaning*, Hillsdale, NJ: Erlbaum, (2007).

[18] B. Vangerven, W. Passchyn, A. Ficker, D. Goossens, and F. Spieksma, 'Conference scheduling: a personalized approach', in *Proceedings of the 30th Annual Conference of the Belgian Operations Research Society*, pp. 15–16, (2016).

[19] X. Wang and A. McCallum, 'Topics over time: A non-markov continuous-time model of topical trends', in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 424–433. ACM, (2006).

[20] B. Yang and J. Geunes, 'A single resource scheduling problem with job-selection flexibility, tardiness costs and controllable processing times', *Computers & Industrial Engineering*, **53**(3), 420–432, (2009).

[21] L. Yao, D. Mimno, and A. McCallum, 'Efficient methods for topic model inference on streaming document collections', in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 937–946, New York, NY, USA, (2009). ACM.