CHOOSING CONFERENCE PAPERS

Feasibility Report

Client
Kilian Weinberger
kilianweinberger@cornell.edu

Team
Mahak Garg
Syed Mutahir Hussain Kazmi
Justina Chen
James Russo
Khaleel
Yao-Chuan Chang
Sahana Tejesvi Peters

16 Feb 2017

I. Introduction

The following proposed system is intended to be used by people/organizations who organize research conferences and people who are interested to attend these conferences. Dr. Kilian Weinberger, Associate Professor in the Department of Computer Science at Cornell University, will serve as our primary client. The goal of the development team is to develop a responsive and user friendly web application which will be accessible via mobile and computer. The application will provide the organizers a way to add research conferences. They will be able to upload the papers for that conference. They can also define schedule for various sessions to manage it more effectively. The application will provide the people attending the conference with a list of sessions/papers relevant to their area of interest. This will be done using a machine learning algorithm, based on matching their papers from google scholar profile/ personal page with the papers being presented in the conference. This ML algorithm will be pluggable and can be replaced by the client in future. The overall goal of the new system is to streamline the existing process where people look at references of a research paper manually to find out which are relevant to their area of interest.

II. Preliminary Requirements Analysis

Application Overview

Objectives

The basic functionality of the system is to let research conference organizers organize conferences more effectively and provide users that are interested in the conference with a glimpse of relevant material for the upcoming sessions within a research conference to improve their conference experience.

Business Objectives

The project aims to reduce time taken for finding relevant material to papers that are about to be published in a conference. It also will provide an efficient way to those organizing the conference. This tool will serve as a one-off platform where users will be able to see schedule of upcoming conferences and gain more information about them.

Through machine learning, the system hopes to provide a good recommendation system for papers. The working prototype will be developed and implemented in time to be deployed by the end of Spring 2017.

Current Business Process and Rules

Currently, organizers put up information about the conference on a website that is primarily for academic purposes. This includes information about the conference, committee etc. Organizing and informing interested people is done manually through emails.

Users who are interested in the conference look at papers manually and use the Google Scholar profiles of presenters to gain more insight into the research the individuals are doing. They also use the “Related Work” section to find relevant papers that can be used to gain more insight on the upcoming research.

User Roles and Responsibilities

- Research Conference Owner / Organizers: These are the people who are primarily responsible for hosting and organizing the research conference. They will add the conference and presentation details, and will also optionally input conference schedule details.
- Conference Attendees: These users will upload their Google Scholar profiles or their personal academic page (with the papers they have published or of their interest). These users show interest in attending the conference or learning more information about it.

Interactions with Other Systems

The system will be built from scratch but will have an external component that is provided in the form of a machine learning algorithm by the client himself. Initially, the team will develop their own version of machine learning algorithm based on a simple similarity computation, which will provide data in a format that is acceptable to any future replacements of the same algorithm. This will be used to create the research paper recommendation system.

Production Roll Out Considerations

The central data repository design and development, as well as the design layouts are expected to be carried out in a phased manner over three months before the system is tested and put into production. The client aims to test this in an upcoming conference for Machine Learning.

Functional Requirements

Statement of Functionality

The software system is intended to create a salient and user-friendly interface for organizing research conference lectures, by combining an efficient recommendation system with a conference-schedule optimization tool.

Conference organizers will upload the master schedule of lectures with information about presentation times and locations, to the system through the website interface. Organizers have the option to update the conference schedule at any time.

Conference attendees will submit the URL of their Google Scholar profiles or personal academic websites through the website interface. The software system will scrape the PDFs of their published works from their profile page and store these files in a database. Files will be cached and co-author information included to avoid duplicate downloads.

The conference papers scraped from conference attendee's profiles will be passed through a natural-language processing algorithm that computes the distance (or similarity) between the attendees' research and the research papers presented at the conference.

Conference attendees will receive a list of recommendations for lectures to attend at the conference, based on the computed similarities. This list of recommendations will include a reason for the recommendation, and details about the lecture (time, location, keywords, paper abstract, etc.).

Security and User Capabilities

The two target users for this software product are (1) conference organizers, and (2) conference attendees. Users need to register as either an attendee or a conference scheduler in order to use the system with an email and password.

The user access level (conference or attendee) will determine the interface and options that the user sees after login.

Reporting
The recommendation process will record the number of recommendations that papers are receiving, to gauge which presentations will be most popular.

There will be a short delay in between when conference attendees submit their academic profiles, either as personal websites or Google Scholar profiles, and when the algorithm has finished computing distances and generating recommendations. When this process is complete, attendees will receive an email notification with a link back to a static webpage containing their schedule.

Non-functional requirements
The system will require server space and domain access to function. These requirements will be chosen at the discretion of the client, although the team will make recommendations if necessary. Initially team will use free services and then at later stages or the release stage think of deploying in paid services.

Optional Features
Ideally, the system will include a conference-scheduling tool along with the recommendation system. This tool might include an auto-generator feature, which auto-generates a personalized and optimized schedule based on the system recommendations, and the master conference schedule. Conference attendees will have the option to revise their auto-generated schedules as per their preferences. This scheduling interface could be a dynamic webpage (like Cornell's scheduler tool), and include an option to email a printable version.

The number of attendees per presentation will be tallied and updated as conference attendees are generating their schedules. The conference organizer can use this tool to gauge the number of attendees per lecture.

The system also may include a notification system where the users are informed about any changes happening to various schedules, reminders, tasks being completed etc.

Usability
This project will require rigorous user, program and acceptance testing. Usability concerns, such as interface organization and layout, feedback and notifications, algorithmic efficiency, automation, and resourceful data collection with be continuously considered.

Rigorous testing procedures will ensure that usability considerations meet the client's standards and users' needs.

Scope

The scope of our software system focuses on requirements for the two user groups—conference organizers and attendees. The system will support initial entry and updates of conference information and research scholar profiles, for each group respectively.

The software system and website are not intended to replace the conference website entirely. Conference organizers will still have to create a main site for their conference with information about fees, lodging, registration, and other logistics. This scheduling tool will try to pick up as much information as it can from the website and the rest will be done manually.

This software system is also not intended to be an organizing tool for conference organizers on the optimal layout and structure of their conference. The room sizes and research presentation times need to be calculated beforehand by the conference organizer, then inputted into the software system.

The system will also not support direct PDF upload; researchers need to have a personal academic website or a Google Scholar profile where information about publications can be collected.

III. Process to be Followed

For this project, we have decided to follow an iterative refinement software development model which involves initially creating a prototype and eventually adding the functionality until the client requirements are met. This model allows the client/user to evaluate the system at each iteration and provide feedback. Such things are very useful when developing a system where UI design is involved. Iterative Design provides an incremental method for more effectively involving the client in the complexities that often surround the design process. The team can quickly create a user interface for the client to evaluate. If necessary and time permitting, changes can be made to the interface in the next iteration based on client feedback. While the user interface is revised at each iterative step, the back-end team can work on adding functionalities like pdf scrapping, pdf to text, word vector creation.

Below is the proposed outline of the iteration stages and milestones including what the team expects to have completed at each stage:

1st iteration (March 12, 2017)

Requirements Document
The team will create a formal document stipulating all the requirements from the client. The client will decide on which requirements are “must have”, “good to have” and “optional”. After the client approves the document, the front-end team will go ahead and create an initial design for the UI, while the back team will start working on one or more functionalities.

Mockups: Simple User Interface Design
After the approval of the requirements document, the front end will go ahead and design the UI. The design will contain all the required data entry fields, but not all of them may be functional. In the first milestone, the UI will be simple and might not contain all the proposed fields. Depending on the implementation at the back end, if the pdf scraping engine is complete, we may add a pseudo web page to demonstrate the results to the client for the feedback.

2nd iteration (April 9, 2017)

Design Document and Presentation
A formal draft of the system architecture and design will be done as a part of 2nd iteration. In the design report the software and hardware requirements will be specified. It will give the client an idea of the overall architecture of the system as well as the finer details of each component. This can help the team as well as the client. For the team, it will outline all the parts of the software and how those parts will work together, thus coordinating them under a single vision. For the client, it will serve as a reference for future maintenance and extension activities.

Revised UI
Based on the feedback from the client from the first iteration, the team will modify the user interface design as needed. More features and abilities will be added to the system. By this time, the backend would be at a stage that the product is a MVP (minimum viable product). The client should be able to see the core functionalities working such as actually getting the match of the conference papers based on user profile.

3rd iteration (May 7, 2017)

Final Testing Period
The team will reserve the last two weeks to thoroughly test the functionalities of the system. This will involve user testing, program testing and acceptance testing. The client will use the second week to test the system on real users and data. By this time all the functional requirements should be complete and only a few minor changes should be required.

Final Documentation and Presentation
A final document will be made for the product and presented to the client so that the product can be maintained and extended. This will include information on all the functionalities that have been implemented. The team will also like to add which features could be added in the future. A demonstration of the system and training will be provided to the client so that the client can understand how the user interface is intended to be used.

Final System
The final system will have all the functionalities that were deemed “must have” by the client at the start. Depending on the bandwidth of the resources some of the “good to have” and “optional” functionalities may also be included in the system.

IV. Suggested Deliverables

The goal in this project is to create a system to automate and streamline the recommendation process for research presentations at conferences. To succeed in this goal, the following set of products will be delivered to the client.

Prototype Demonstrations
Over the course of the software system production, the team has scheduled three official prototype demonstrations following production milestones. The client will receive a thorough update and demonstration of the progress that has been made between each milestone. Because the team is following an iterative model for software development, each prototype should be a fully functional software product, with subsequent prototypes containing more features, updates, and improvements. The presentation may not cover all areas of the system production uniformly—

at one presentation, the user interface may be addressed more heavily, while the next presentation may focus on the database structure, and yet another may focus on integration and rigorous testing.

Regular Status Updates

To ensure that software system aligns with the client's interests throughout the development process, the team will meet and correspond with the client on a regular basis. These status reports will focus on updating the client on progress, raising any questions about expectations, and receiving critical feedback. The client meetings will most likely occur on a weekly basis at the beginning of each development cycle, but may become less frequent as the software system matures.

Onboarding Documentation

To ensure that the final software system is utilized to the fullest extent, the team will conduct rigorous user testing on the user interface throughout the software development cycle. However, in addition to making sure that the system is user-friendly with a salient interface for features, the team will also be delivering documentation for user onboarding (for both attendee and organizer user types) that explicitly describes the functions and features of the software product.

In addition, the team will provide high-level written overview of the system structure and backend processes. The client will be updated in detail on the system structure, requirements, and operations throughout the software process. This high-level overview will include a summary of these details, and is intended for the client to have a written record—it is not intended or necessary for the users to have access to this overview.

Business Considerations Documentation

At the beginning of the software development process, the team will deliver a document with explicit information about ownership of the final product, and expectations and deliverables throughout this process. This document will be based on a conversation with the client regarding expectations of deliverables and features and ownership of the final product. Both the client and development team must approve changes in expectations or ownership made throughout the development process, and the agreement document updated accordingly.

System Interface

A. ***Conference Attendee Interface***: A login and main page for conference attendees to register and upload their academic profiles, in exchange for a personalized recommendation of lectures to attend. This interface will also allow conference attendees to update or resubmit their profiles, if their scholar profile is updated with new publications, for example. The conference attendee-side of the software product will also include the email notification system, and a static webpage with recommendations listed.

B. ***Conference Organizer Interface***: A login and main page for conference organizers to upload their conference's master schedule and presentation details. Conference organizers will be able to update the initially submitted information, to reflect changes that arise in the conference scheduling.

C. ***Database***: The software system will include a database system that will store hashes calculated from PDFs collected and file path of the PDFs which will likely be stored on the filesystem directly. The database(s) will also include details about the publications, including authors and co-authors (to avoid duplicate entries and caching), computed

similarities to other entries, and abstracts (to include for easy user-access in the recommendation reports).

D. ***Integrated Algorithm***: The initial similarity score algorithm that calculates the distance (or similarity) between the conference attendees' publications and the publications being presented at the conference will be developed by the team. The algorithm will be integrated in a way that can be easily swapped out for a different Machine-Learning based algorithm later on, developed by the client and at the client's discretion. Initially, the recommendation system will only take the computed similarities from this algorithm into account. However, a second version of the recommendation system will also include an option for user feedback (for users to thumbs up or down their recommendations). The second version of our feedback system will take this user feedback into account when generating a recommendation list.

E. ***(Optional) Attendee Scheduler***: If it's feasible in context of time constraints and the progress of other required components, the software system will include a scheduler tool for conference attendees. This scheduler tool would involve a dynamic web page that generates an optimal schedule for each attendee based on their personalized recommendations, but that can be changed at the attendee's preferences. The scheduler will also keep track of the attendee schedules, helping conference organizers gauge demand for each presentation.

F. ***(Optional) Notification System:*** If it is feasible, the system will include a notification system where the users are informed about any changes happening to various schedules, reminders, tasks being completed etc. This will be in the form of both emails and web notifications.

V. Technical Feasibility

To determine the feasibility of our technical requirements we tried to explain at least one option for implementation. If we are able to think of an implementation, then the requirement is considered feasible even if the implementation is replaced by a better one in production.

Requirements

A. ***Create a web application:*** We will be creating our web application to be supported for all different screen sizes including web browsers.

B. ***Hosting:*** Hosting of our web application will either occur on Heroku, AWS, or a local Cornell server depending on our client's preference. The quality of the product will determine this. Initial testing will involve using Heroku though.

C. ***Download all PDF's on a given webpage****:* We must be able to find and download all PDF's off a given URL such as a google scholar link or personal website.

D. ***Store all PDFs in a database****:* The PDF's we download must be saved in some way to be reused and have minimal overlap. Some papers have many authors and if we have already downloaded a paper we don't want to download it again. We will store an md5 hash of all the PDF's to make sure we don't store any duplicate PDF's. We will most likely use a relational database such as PostgreSQL to store the database in an efficient way.

E. ***Compute Similarity Measure between PDF's****:* An important aspect is to use a similarity based algorithm to create a recommendation system. The system will use PDFs gathered and convert them to Text before applying similarity algorithm. The output will be generic enough to give the system with the capability with replacing the algorithm with any other algorithm that may or may not be provided with by the client. Once we have compared two PDF's we will then save the similarity of them in our database so it does not need to be computed again.

F. ***Suggest conference papers relevant to the user interests****:* Once we have computed the similarity of the papers of a given user and that of a conference we will alert them of similar papers via an email. They must give us their email when we get the link to their papers. This functionality is built in to rails but there are also many gems that allow for sending of email. We will use the information stored in our database to let the users know which papers were most similar to which so they can better understand the results as well. And we will also send the user a static webpage with this information in the email.

G. ***Conference organizer interface****:* Conference organizers will be able to add conferences, which can then be looked at by the audience. We will allow organizers to also give us a url that contains PDF's of all the papers accepted to the conference. Attendees can show interest in the conference and get updates about the conference along with any other relevant information about related conference papers. The interface will allow conference organizers to add meta-data such as room locations and times for different paper talks. All of this will be handled by built in rails functionality and saved into the database.

H. ***User Login****:* Both attendees and conference organizers will have user logins. This login will consist of an email and password for each user. We will be able to do this using built in rails functionality and storing the credentials in our database using the above-mentioned database and gem.

VI. Visibility

The team aims to maximize the interaction between the client and the development team at various stages of the development process. This will eventually ensure that there are no deviations from agreed specifications and any corrections can be made through client feedback. The following are the methods that the team looks to use for communication purposes:

Communication

Client meeting: In person meetings and emails will serve as the primary form of communication during the whole stage of software development. In person meeting with our client will be done on a biweekly basis and primarily at 9am on Friday. We will update the client about our weekly progress and gather direct feedback from the client.

Team meeting*:* The team will also hold weekly general meetings every Wednesday to ensure all the members are effectively taking part in the project and understand their roles during the software life cycle. We are using the software" slack" as the team messaging tool to post any discussion and get feedback from one another.

Intermediate Deliverables and Presentations

Live demonstrations: The client will be provided with prototype of the website through biweekly presentations. By demonstrating functionality and UI design of the website, we can have direct feedback from our client in each iteration process.

Presentations: Presentations will include design layouts of screens and demos of working functions. The team will provide this to the client in keeping with the goal of updating the current progress of the team consistently.

Reports: The client will be presented with documentation at various phases of the software lifecycle. These documentations will enable the client to have a record about the details of the project.

VII. Risk Analysis

This section of the report lays out a broad overview of the possible risks associated with the project that impede and impact the development process. They broadly fall under the following categories:

Functional Requirements
The risk with this is that the result of the project does not meet the client's expectations. An example could be the client may have a different idea than the team about the user interface design or the way a conference paper is suggested to a scholar. To mitigate this risk, the team will have regular meetings and a visibility plan with the client to ensure a clear channel of communication and the setting of expectations.

Given the limited period of a semester, additional functionalities may be added later as part of improved versions in the future.

Time Risk
The risk under this is that the project does not go live with all the functionalities that the client had indicated given the time constraint of one semester. Incorrect estimates of time vis-à-vis implementation of coding tasks may also cause delays in the release of system.

A solution to this is for the team to carefully define each stage the project is at with enough time for fallback and testing periods.

Technical Resources
Resources that come under this are servers and systems essential to the project. The team and client must decide on a hosting plan and a domain name that is acceptable and ensure that the website can serve a set of users. In case of moving to SSL, the certificates may have to be purchased. Additionally, since the system is web based, the team realizes that there can be a difference of user interface on different internet browsers and thus will look to make a good responsive website.

The team will make these things clear to the client to ensure that technical difficulties are well known and can be catered before the team is during the stages where it intends to release the software.

Non-Functional Requirements
The risk here is that the project must consider functionalities that are implicit in nature. The number of users the server could handle, performance considerations and its security has to be laid out.

The team will keep frequent communication with the client and look for effective feedback to cater for the non-functional requirements that the team may have missed or may have overlooked. This will be done in a timely manner to avoid any unwanted delays towards the end of the project.

VIII. Business Considerations

Several business considerations surround any practical software product. As part of the Feasibility Study and Plan, the following points briefly demonstrate our business plan and consideration.

The project team which consists of the following individuals, James Russo, Justina Chen, Khaleel, Mahak Garg, Sahana Tejesvi Peters, Syed Mutahir Hussain Kazmi and Yao-Chuan Chang is hereafter referred to as “the team”.

Ownership of code
As students of Cornell University, the team owns the copyright for any code or tools written / created for the project. The team will not transfer ownership of the copyright. However, a license which will outline the policy of use will be drafted. The team will ensure that any code written, is their own work. Any work taken from other sources, will be credited accordingly as per the legal guidelines. As far as possible, the team will strive to maintain the code base as their own work written by themselves as opposed to source from a third party regardless if it is open source or not. The client or any member from the team cannot reuse the code for commercial purposes without explicit permission of the team.

Based on the academic nature of the project, the team believes that a license is the most appropriate option. Given that this project is likely to be extended into a second phase of development, it is vital that we define the outline of fair usage and future enhancements, if any.

The team will provide the client with a limited license accordingly which will incorporate the usage policy for the written code and future developments. The limited license permits the client to use and modify the code for an unlimited period of time. The future modification of code, if any, may or may not be done by the team. The team will not be responsible for any modification done to the code by any third party outside the team once it is delivered to the client by the team. However, the team will assist the client by addressing any further concerns/questions. Furthermore, the team reserves the right to reuse, demonstrate and replicate any part of the code or the whole application for educational purposes or for prospective employers.

Trademarks and Patents
The team does not plan to trademark any name at the moment. However, it reserves the right to do so in the future.

No part of the system is eligible for any patent application in the near future. Nonetheless if on a later date, if any part of the system is found to be patentable, the team reserves the rights to the uncontested patent and any derivative works based therein while the client will automatically gain

non-exclusive rights to use the system, and will have full rights to the use and modification of the system regardless of any patent rights held by the team.

Trade-secrets and NDA
Since the software may contain sensitive and/or personal information about its users and conference owners, including but not limited to their personal information or intellectual property in their conference paper, a Non-Disclosure Agreement (NDA) is required for this aspect.

The NDA will tentatively be signed towards the completion of the release stage when we hand over the deliverables to avoid any conflicts of interest or prevent any intentional/unintentional sharing of the aforementioned information by the team or the client.

Keeping the above guidelines into consideration, the system will be designed to incorporate a password based protective measure to avoid unauthorized users from getting this information.

Future commercial use and financial returns
There will be no duty on the part of the team, individually or as a group, to account for any return on subsequent commercial use or development based on the code or material developed for this project.

The company intends to initiate the first release without any returns. Should the client or the client's team realize any monetary value, royalties or financial returns from commercial use or licensing any of the code or material developed for this project, the authors (team) are be taken into consideration and will have a right to the returns as well. The ratio of said returns will be decided whenever the decision is being made.

IX. Conclusion

From the feasibility study, the team is able to conclude that the a tool that helps organizers manage research conferences while recommending relevant materials to attendees is viable in terms of technicality, skill of team members and time. With the time constraint of exactly one semester (approximately 3-4 months) the team believes the scope of the project is manageable and that the client's requirements can be satisfactorily fulfilled upon system completion. The conclusion of the feasibility report is to proceed with software development of the project.

X. Appendix

Gantt Chart Schedule