Analysis Me

**Fall**

2014

Technical Report

NASA Practicum-Trust Based Recommendation System

Enclosed in this document is the technical report of the Trust Based Recommendation System sponsored by NASA

Point of Contact: Petr Votava

Faculty Advisor: Jia Zhang

Team: Krutika Kamilla, Abhishek Mukhopadhyay,

MustafaTasdemir, Jisha Vadake Muthiyil

# Table of Contents

1 Introduction 3

2 Motivation 3

3 Related work 4

4 System design 5

5 System implementation 9

6 Experiments and analysis 13

7 Conclusions and future work 15

# Introduction

The ‘Trust-Based Recommendation System’ is a web application that facilitates scientific collaboration and is based on the ongoing research on collaborative systems done in conjunction with NASA. The primary goal of this project is to create an end-to-end solution to NASA Researchers about recommending contributors to research studies that best relate to people’s academic references and publications. The end-to-end solution will contain a web-based interface for the users that will create dynamic graphics, which reveal the relevance and trustworthiness of researchers on each other. This visualization will generate reliable inferences between people to suggest the best potential contributors into a projected study. The relation between different authors and the authors’ list of publications can be visualized. Meanwhile, the existing trust calculator is in place, it will be enhanced to include some relevant parameters that relate to researchers’ online social metadata to increase the reliability of the solution. The trust score is calculated based on the number of papers published by the author and coauthor ship. It is the novelty of our web application as on the basis of that, new researchers in the any field can get recommendations on whose papers to read and with whom they can collaborate with for researching in a particular field.

# Motivation

Studies show that, per year $1.6 trillion is being spent on Research and development worldwide out of which U.S. alone spends $465 billion. There has been a continuous contraction in budget through the past 50 years. This necessitates efficient utilization of these funds so that it will benefit humanity to the maximum possible extent.

NASA, which is the one of the most important government agencies that fund research and development in the US face the same constraints. This is coupled with scientists working on disparate domains but overlapping problem sets. A lot of redundant research working is happening which can only be avoided if there is a proper platform which helps people find what are the publications in a particular field and who are the experts in that field. There is a need to accelerate research through collaboration. Accelerating collaboration by making it easy to identify collaborators working on the same problems is important. NASA’s Earth Exchange (NEX) platform aims to provide a cloud-based platform as a service to accelerate big data analytics and scientific collaboration in Earth science. There is a possibility of extending to or integrating with subject domains too in future.

Collaboration between NASA and CMU-SV started 2 years back in order to come up with a trust based recommendation engine that would facilitate scientific collaboration. There was a need to develop a system, which recommends experts and collaborators based on domain and trust. The previous teams who worked on the trust-based system developed an algorithm to calculate trust based on knowledge reputation and other social factors. Our focus was to develop a prototype, which productized the large amount of research work that had been done in this field. Our aim was to develop a product that would help find existing research in a particular field, find the experts in that field, figure out of these experts who can act as collaborators and perform all of these tasks leveraging the user’s trust. From the very beginning of development, our prime focus was accessibility and usability of the product

# Related work

There are many products available that enable scientific collaboration. This section discusses the main features of a few products currently available.

* **ResearcherId**

ResearcherId is one of the most popular tool used for scientific collaboration. The best feature of ResearcherId is its integration with the Open Researcher and Contributor Id(ORCID). This enables data exchange between the ResearcherId system and ORCID. ResearcherId has got four main interfaces, which facilitates different types of search. There is a map view, which shows the locations conducting research in a particular topic. The biggest disadvantage of ResearcherId is that it is proprietary.

* **Researchgate**

Researchgate is a social network of researchers. It allows users to create profile, upload papers and discuss issues. Users get detailed statistics about their papers such as number of hits, citations etc. The search feature in Researchgate is not very intuitive.

* **PubMed**

PubMed is one of the richest databases of scientific papers. It was first released on 1996 and since then has been a popular search engine. PubMed for handheld devices is one of its best features. The main disadvantage with PubMed is its interface and lack of usability. To use the comprehensive search in PubMed, we need a thorough understanding of its components. Moreover, they do not have any information that could be used for collaboration and they do not track citations

Apart from the ones discussed here, most of the university research databases are good sources of information. We had the opportunity to review a handful of them and adopt the best features into our product.

# System design

As every other system requires, our practicum project also needs a system level design for a robust, reliable and maintainable use. This section discusses about the design decisions that are made throughout the whole semester that we have led after previous work done before us.

**Quality Attributes**

The key points in system design rely upon the attributes that the system must present at all times. We address these attributes according to the overall requirements of the client. Some set of requirements lead to a quality attribute that best identifies the required system attribute. We have listed the following quality attributes according to user requirements.

* + 1. **Accessibility**

Since the project has been ongoing for the last couple of years, it revealed that the outcome prototype or product was not reachable by the intended audience who are the users of the system. Therefore, the main concern for the client was to make the system accessible via modern browsers at all times. This way, there would not be any installation guideline that the users would struggle about.

* + 1. **Usability**

When we achieve the accessibility, the next biggest concern will be how usable the system is. The users will have to perform the functions that the system provides in a browser, which in some cases may create some unusable features considering that the project deals with a substantial amount of data behind. Therefore, optimizing the usability of the system taking possible use cases into consideration will be another aspect of the system design quality attributes.

* + 1. **Extensibility**

The system will offer a trust-based publication network out of which users will infer sensible information to learn about specific topics or authors in computer science field. As one of the previous teams have focused, trust-based calculation is formed with knowledge and social based factors. Each of these may contain different kinds of data sources such as co-authorship for knowledge and Twitter/LinkedIn data for social based factors in order to utilize the trustworthiness in a more reliable way. Hence, being able to add more and different types of data sources holds in our system quality attributes.

**System Architecture**

* + 1. **Dynamic View**

Since the system focuses on consuming big data and constructing meaningful representations out of it, dynamic flow of the data becomes an important aspect of the architecture. That's why the following schema has been created to show this flow as well as with the underlying physical devices that are required for deployment.

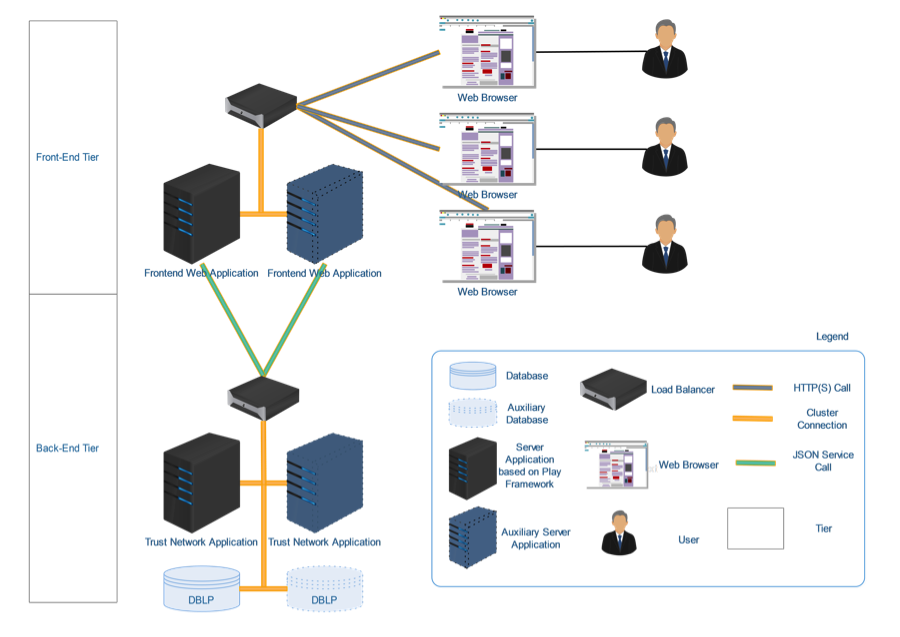


Figure 1: Dynamic View of System Architecture

As seen in Figure 1, the overall system relies on two-tiered system for better maintainability. The back-end tier consists of the processed data to be consumed and the trust modeling application based on that data. The front-end tier consists of the web application to be served to the users. As shown in the figure, the number of servers as per in the tiers can be extended for performance issues. This will require some load balancer servers to actually distribute the request load to the servers in a reasonable and equal way.

As mentioned above, the two-tier architecture let us enhance modularity of the whole system. The data is served through JSON and the front-tier applications consume them via service calls. Therefore, in case another application need emerges for trust-based publication network, the underlying application and data tiers already exist which help us focus on the third party application design and implementation separately.

The framework choices include Play Framework for all types of application servers either back-end or front-end. The reason behind this preference is that the static structures of both tiers (in terms of packages and classes) help us construct the system faster and more efficiently. Also, the models that we have used for data representations are shared within the tiers. Hence, we thought keeping the same framework for inter-cross applications would be more accurate and appropriate.

* + 1. **Static View**

Static view refers to models and objects defined for the system. Since the publication network includes different kinds of publications, we have come up with a hierarchical modeling base in order to represent the data in an efficient and easily maintainable way.  The figure below shows the schema of the data representation.

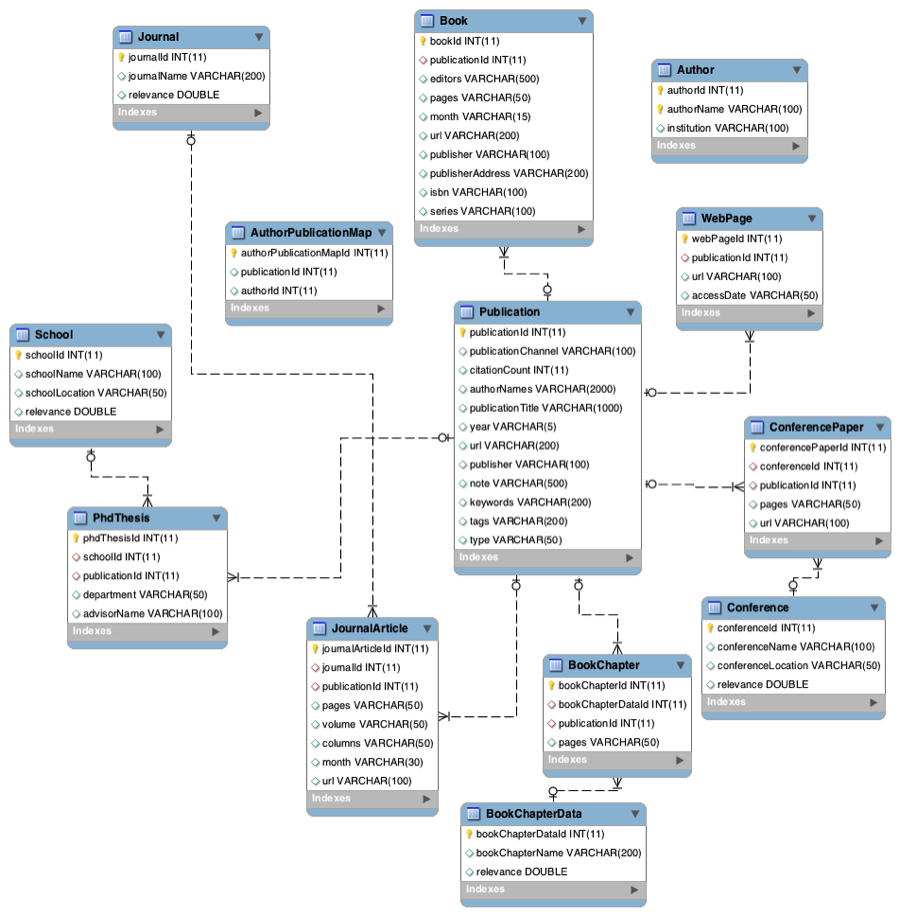


Figure 2: Database Models Schema

**How Attributes are met**

The previous work on this project had produced desktop-based offline applications that were not accessible enough for the client. Therefore, the solution of a web-based application will fulfill the accessibility requirement as suggested in previous sections. The web-based application will also require a usable interaction for a better utilization of the system and hence the required study for usability optimization has led us to web-based visualization tools that are lightweight and easy to use for possibly projected interactions. This helped us overcome the usability attribute. For extensibility, multi-tiered architecture as well as with normalized data modeling will make the system easily extensible in case of new data source must be integrated or any new business logic must be provided. Thus, the design will provide extensibility to the client for enhancement in the system.

# System implementation

* 1. **ETL (Extract & Transform & Load)**

Extract & Transform & Load (ETL) process is for preparing the data consumable by the applications developed for this project. To ensure that the data is loaded correctly and efficiently, we have defined the ETL process from scratch. The following sections reveal the details about what our approach is and how we have applied it.

* + 1. **Data Source**

Raw data source consists of XML files, which include the details about publications in computer science field and the citations associated with them. According to our estimates, there are around 4.2M publications and nearly twice as many authors within the dataset. Although the total size of the files do not even exceed 2GB, considering that they include just metadata about the publications, it becomes massive enough in data size. The format, that these raw files use are not efficient enough to use directly, especially when considering the data size. Therefore, we have defined an ETL process to prepare the data for the application.

* + 1. **Data Transformation**

As mentioned above, the data format is not consumable directly. Therefore, we need to transform the data for our modeling. In order to do this, we first came up with the models.  Since data source had a hierarchical data representation for publications (such as Book, Web page, Conference etc.), we have adopted the same approach. Because, we thought it would be more to the purpose for our extensibility in case we will expand the data set with different types of other data sources especially from social media.

The way that we have transmitted the data from files to our destination database management system consists of two parts. The first part is actually the transformation part in which we are changing the data structure of the raw data source into the same representational form as our data modeling. This way, we can directly load the data and apply performance optimization depending on our need. In detail, we have written a custom XML streaming parser to consume the data source. We have experimented several approaches such as loading all sources content into memory and streaming from the source. We found out that streaming is way more efficient and reliable than memory processing. On the other hand, the implementation became more sophisticated. After streaming the content, the parser exports each individual records that it detects to a CSV format file for corresponding models. So, the data becomes ready for loading phase.

However, one problem we have faced was the fact that the data source was not consistent and for some records not all the attributes were existent. Therefore, we needed to find out these edge cases and exclude them so as to refine the data set.

* + 1. **Data Load**

After all above processes, data is ready to bulk load into database management system and apply optimization on it for better performance. We have automated all the scripts and put them under a separate main folder in the project. The transformed data can be loaded by running these script files directly.

* 1. **Data Fetch**

MySQL database is being used in the backend. To fetch the data from the database, prepared statements have been used. Using prepared statements makes it easy to insert parameters into the SQL. Prepared statements pre-compile SQL queries and hence it is more efficient than calling SQL statements. It skips the plan generation step on subsequent calls thereby increasing the efficiency and reusability. Each of the SQL queries have been put into different methods to ensure maximum modularity.

* 1. **Trust calculation**

Trust score includes two components: a knowledge factor and a social factor. The knowledge factor depends on publications and knowledge reputation. The reputation is decided by the publication channel and citation power. Each component is given different weightage depending on the time of the publication. The weights also vary according to the type of the publication. Each of these weights was modeled using the dblp dataset.

The social factor of the trust takes into account coauthorship. Coauthorship is also given a time scaled weightage such that the papers co-authored in the recent past gets higher weightage compared to the ones published in distant past. The recent papers include the ones published in the past six years with respect to the date of query and the intermediate papers include the ones published in the last 6-12 years.

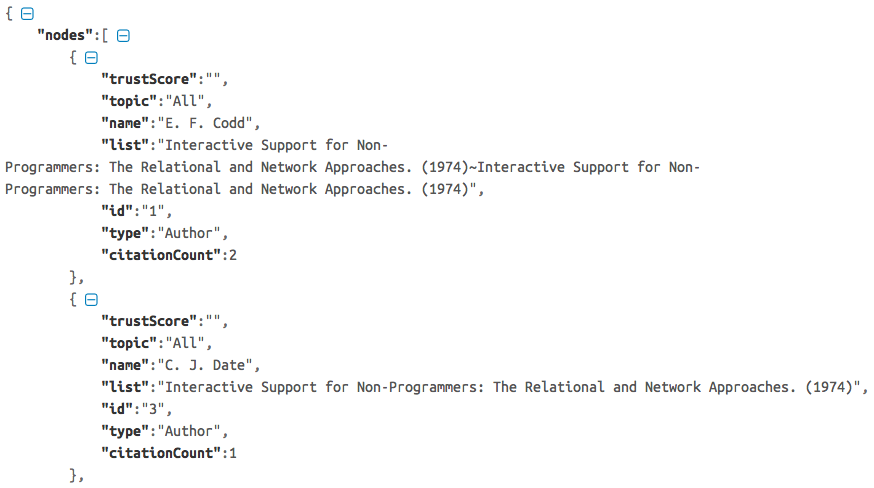
The final trust score is a combination of the knowledge factor and social factor. Final trust score = (CoauthorshipCount\*TimeScaledCoauthorship)+KnowledgeFactor

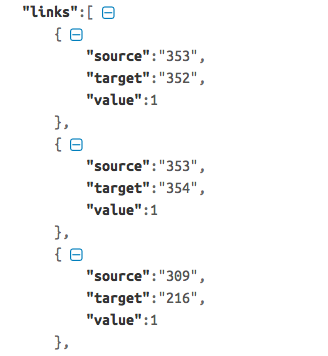
* 1. **Data Provision layer Implementation**

Controller class function is called to render the graph directly to the view JavaScript. This also calls the transform encoding helper as the topic parameter uses JavaScript’s encode Uri component which outputs a different format when compared to Java's URL Encoding/Decoding, which the frontend - backend applications use to transfer parameters. API call of util is used to make the API call to the backend and return the JSON data to the frontend to render it.

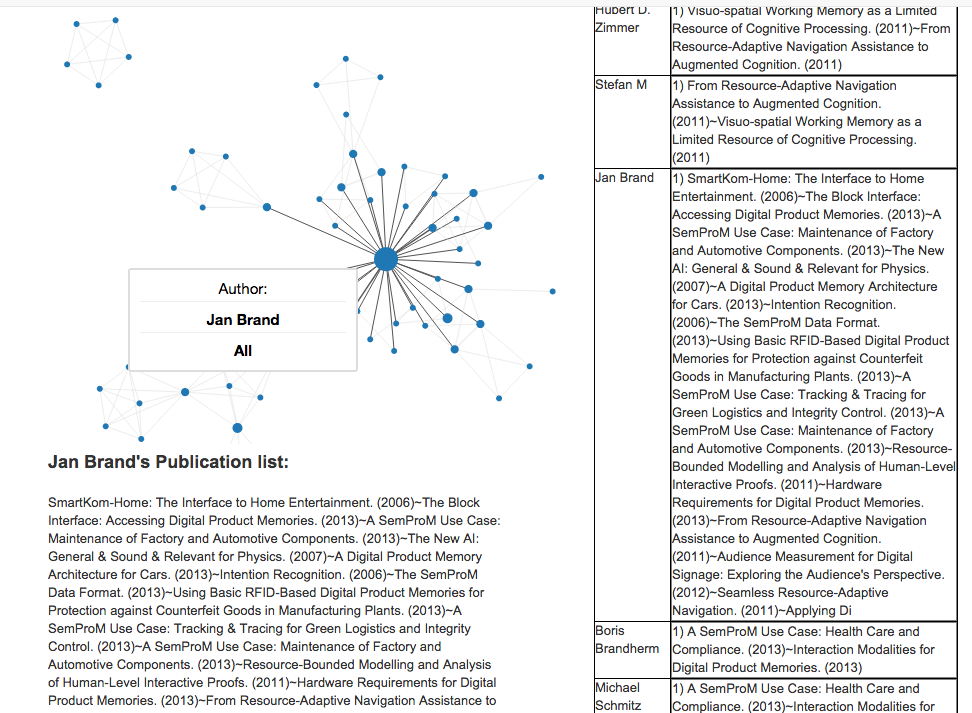
* 1. **UI Implementation**

Json data which comes from the backend through ajax call is fed as input to the vis.js and the nodes and links objects create the directed graph. Figure below shows a sample node and link json object.



****

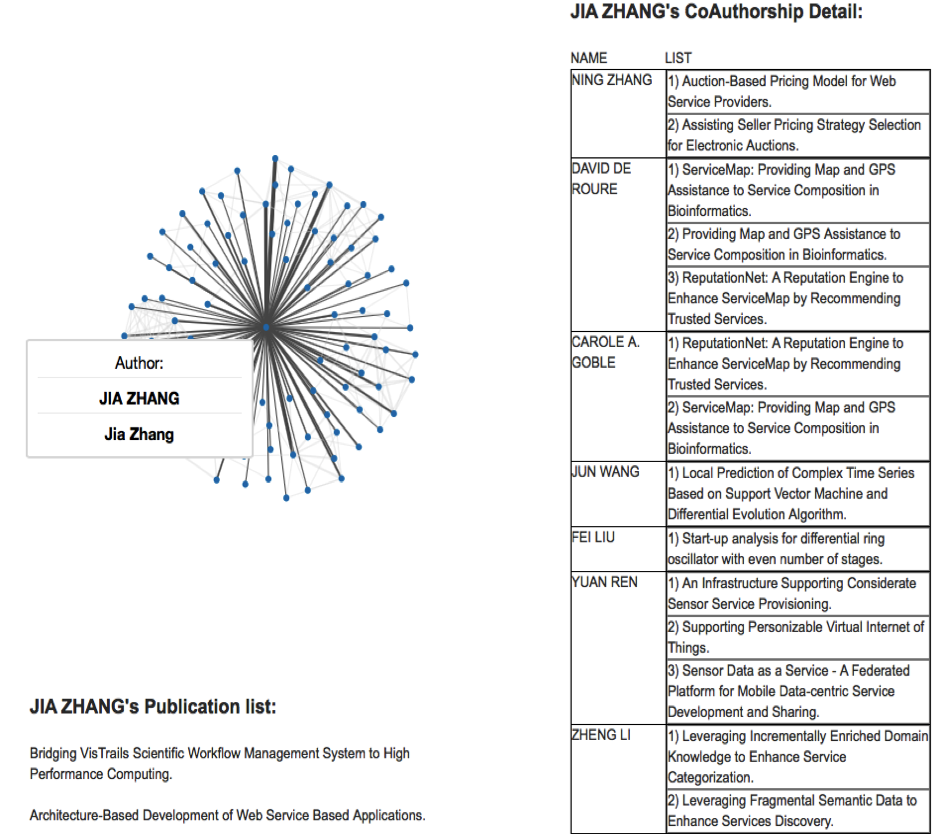
There is significance behind the size of each node and the thickness of the links connecting the nodes. Radius of the node is directly proportional to the citation count of the author’s publications. The more the citation count, the bigger the radius of the circle of the author node. Thickness of the link between two nodes is proportional to the number of publications that the two authors have coauthored.



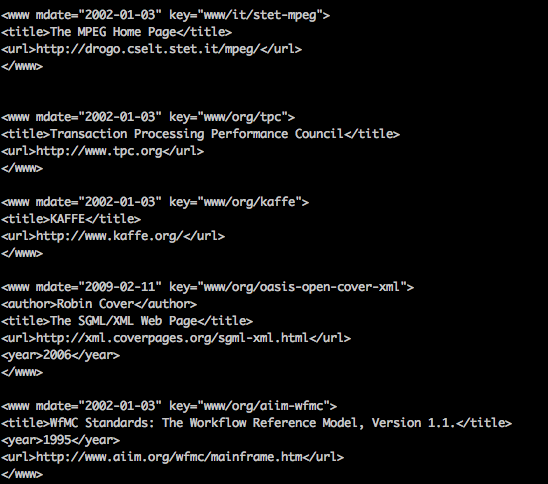
# Experiments and analysis

This section describes how the results of the final web application were tested.

* 1. DBLP dataset is huge, so there is a need to verify the accuracy of data that the application produces on querying. This verification was done using CMU professor names as input to the tool and checking if the publications and coauthors listed are reliable and trustworthy. For example, when we queried for Professor Jia Zhang, it gave her publications and coauthor network. This is shown in the figure below. This was a very important step after the development work was over.



* 1. Dataset is not complete- After some analysis, it was found that DBLP dataset that is used as the source of information for our tool is not complete, as some attributes in it are missing for some webpage elements (author attribute is missing) as shown in the figure below.



* 1. Since the whole ETL framework is in place, we can import any useful scientific dataset in future and leverage our framework without spending any additional effort for building the data load process.

# Conclusions and future work

The trust based recommendation system is a fully functional prototype that has many features useful to the research community. We could learn and experiment with many new and exciting technologies. The system supports topic based and author based searches and calculates a reliable trust score.

The project has the potential to be developed into a full fledged website and we suggest the following steps to be taken next:

* 1. **Improve the back end:**

The data is currently stored in a MySQL database. With over four million publications and eight million authors, a MySQL database is not the perfect choice for the system. As the application scales, the back-end has to be improved. We suggest a high performance database such as HANA, a NoSQL database such as mongo dB or an inverted index. While inserting new records into a database would be easier than an index re-creation, we believe that creating an inverted index would be the ideal choice for the backend.

* 1. **Provision to track the citations:**

Citation count plays an import role in deciding the relevance of the paper and in trust calculation. A provision to track citations and update them accordingly would be essential to show accurate information

* 1. **Creating user profile**

The system currently does not have a provision to create a user profile. This brings in additional validations, as we need to ensure that the information entered by the user is correct.

* 1. **Improve trust score calculation**

The trust score takes into consideration the number of publications, citations and coauthorship information. Different weights have been modeled according to the different times when the papers were published. While considering co-Authorship, we could consider different factors as the possibility of a mentorship and assign weights accordingly. For e.g., a PhD student working under a reputed professor many have published 10 papers with him/her. But that should not make him a more trusted co-author compared to another professor who might have co-authored just 6-7 papers with him. Another interesting factor to be considered would be to integrate LinkedIn profile information. Since LinkedIn is one of the largest professional networks, it could give a fair amount of information about a person’s social acceptance.

* 1. **Pagination**

Currently we have options to limit the results by a number, which the user can select from a drop down. It would be nice to have a feature to show the next n relevant publications/authors

**Appendix:**

The following documents have been submitted along with this technical report

1. Readme file: that briefly describes the purpose of the project
2. Code
3. Access information: URL, user name/password
4. Download and installation documents with step-wise descriptions
5. Executive summary
6. Background and motivation
7. Assumptions and considerations
8. Design documents (architectural design documents and various diagrams e.g., UML files)
9. Discussions
10. Presentations (ppt file)
11. Tutorial: step-by-step usage file with screen shots included
12. Future work: to-do list and descriptions