paper recommendation

October 23, 2021

1 Introduction

To better track the frontier papers of the computer science, or to obtain relevant papers more quickly and accurately that are helpful to research, it is necessary to design a paper recommendation system that is suitable for the students. However, the number of papers on the Internet is large and the content is complex. It has become a very challenging thing for students to obtain reference papers according to their own preferences and fields.

Business Value Created:

1. Intelligent and yet simple user interface to recommend papers by considering the user research field and preferences. And also recommend based upon the historical record.

2. Paper recommendation system can also utilize the information captured to understand and evaluate the trend of academic development and comparison of the paper recommendation policy in the academic session. This will lead to a better and faster development of academic session.

1.1 Purpose of the document

In daily study or completion of projects, or in the process of research, previous researchers' considerable contributions open our mind and provide us with subject inspirations and related technique thoughts. Standing on the shoulders of giants helps us to make larger achievements on academic fields. On the market of paper recommendation, there is not a well-established paper recommendation system which are suitable and targeted for students' daily study about computer science. Traditional paper search platforms usually store a great deal of papers, which are come from different subjects, different fields, different language, and have long time span. After searching in this kind of platform, it provides users a comparison of the keywords and do not take in to account the user preferences and past preferences. With the overwhelming number of papers, individuals will have more options and different perspectives of the problem, but it also means the time consuming and might not result in choosing the best paper.So, it is an imminent thing to develop a professional and accurate paper recommendation system on computer science.

Market Research:

Lot of information on the Internet are available to students. However, all these websites require users read through several pages of information, including paper summary, paper main body, personal blogs, sometimes these information are not related to our topic, hence for a subject it would be many technology field, and it would be many subfields where many people have made contributions, and these subfields are similar but not identical.

Here are certain websites which provide chances to users to have access to thousands of papers. On these platforms, users can get related papers through keywords searching.

1. **Google Scholar** is a freely accessible web search engine that indexes the fully test or metadata of scholarly literature across an array of publishing formats and disciplines. The Google Scholar index includes most peer-reviewed online academic journals and books, conference papers, theses and dissertations, preprints, abstracts, technical reports, and other scholarly literature, including court opinions and patents. Google Scholar uses a web crawler, or web robot, to identify files for inclusion in the search results.

2. **cnki** is a key national research and information publishing institution in China, led by Tsinghua University. In 1999, CNKI started to develop online databases. To data, CNKI has built a comprehensive China Integrated Knowledge Resources System, including journals, doctoral dissertations, masters' theses and so on. CNKI continues to assimilate content and develop new products in two areas: full-text academic

resources, software on digitization and knowledge management. CNKI has become the largest and most accessed academic online library in China.

3. **Paperwithcode** is a free resource for researchers and practitioners to find and follow the latest state-of-the-art ML papers and code.

Though the above websites are impartial and share detailed information. Users have different research fields and different . They might not be have access to

We have also performed knowledge elicitation by interviewing tutorials and students to obtain a deeper understanding of the papers recommendation, current pain points and operations in evaluating the right policy.

1.2 Project Scope

A system to provide intelligent recommendation on the computer science papers,

1. Considering users' interest field and evaluating his preference to choose the sequence of recommended paper.
2. Historical data of the user and other users are taken into consideration.
3. Evaluating the user's input keywords.

1.3 Scope of the document

This document is structured to provide you an understanding of how the different functionalities, tools are synchronized to operate in providing the solution to the business problem. Also there is information provided upon how the different tools are utilized in the system.

Here is the list of core modules of the system which will briefed in detail during the later sections.

Sl no	Functionality
1	User research field and preference capture
2	Keyword analysis by natural language processing
3	Determination of optimal paper by OptaPlanner
4	Data mining based on historic transactions by Orange

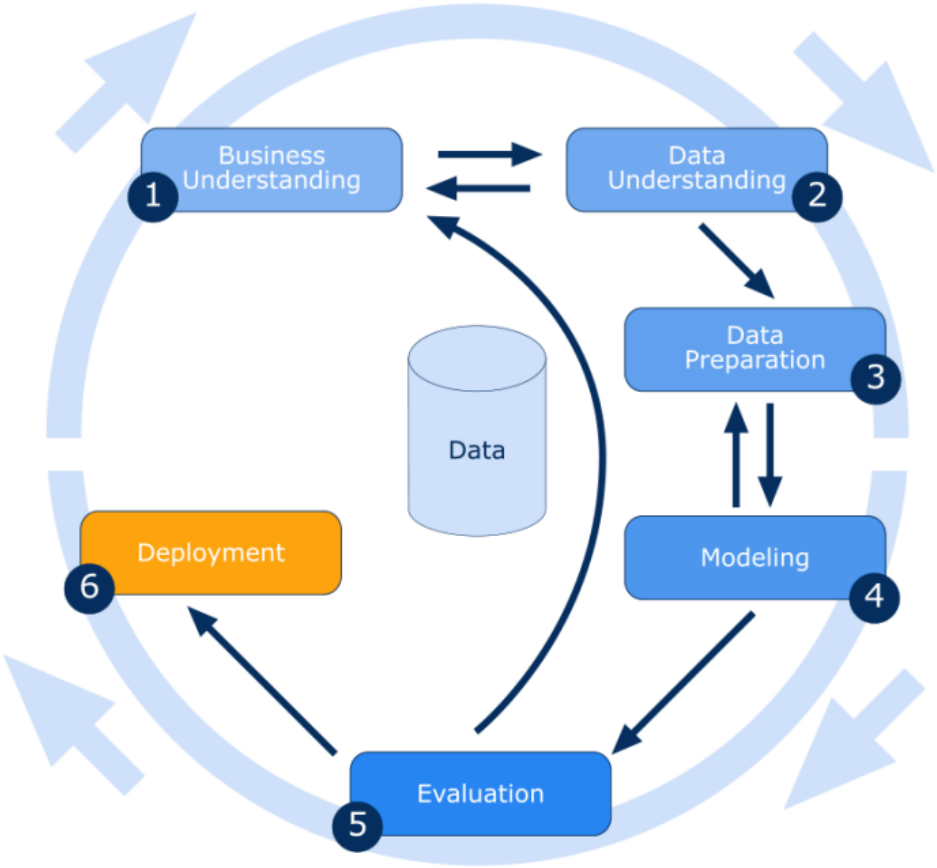
2 System/Solution Overview

This section will provide the necessary details which has supported in obtaining the necessary understanding of the different concepts required for the design of this application.

We will discuss upon the following areas: a. Data Mining b. OptaPlanner c. Content-based Recommendation

2.1 Data mining

Based on the recommendation paradigm of the supervised learning, we determine the most optimal paper based on the past publish. In the pursuit of extracting useful and relevant information from data, process of data exploration, preprocessing, modeling, evaluation and knowledge extraction is performed. Exploratory visualization using Orange tool helped to comprehend various patterns in the data set.

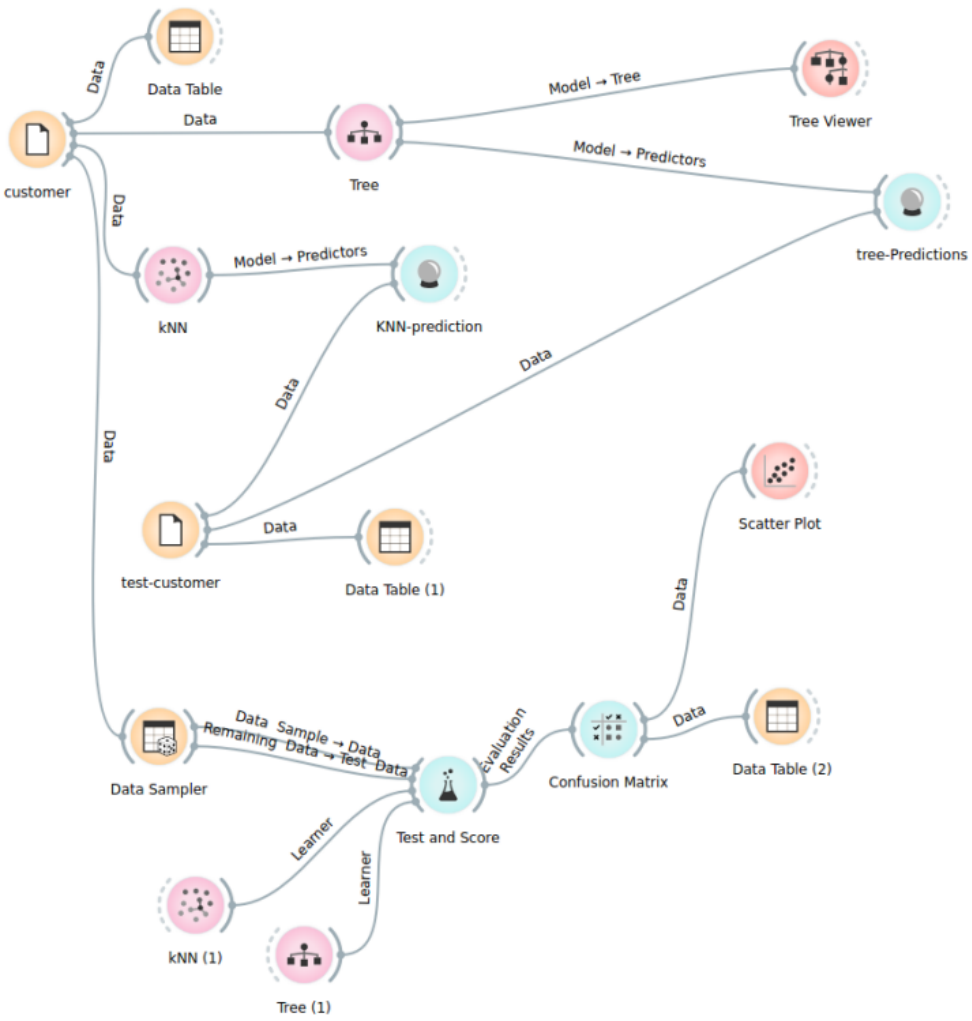


Business Understanding: The business objective is to determine the most suitable list of paper presented to users based on the users' input keywords, his pre-setting interesting field, and some historical searching and click data performed by different users previously. This result will be utilized by the students to browse or read to get the related knowledge.

Data Understanding: For the data, we have the papers data, user preference data, and historical searching and click data.

Data Preparation: In this stage, as the name suggests the data requires to be prepared for further analysis. This acquired data had to be cleansed, formatted to achieve a proper organized data structure. Certain records which were missing critical data had to be omitted since this would not give much information and would become outlier.

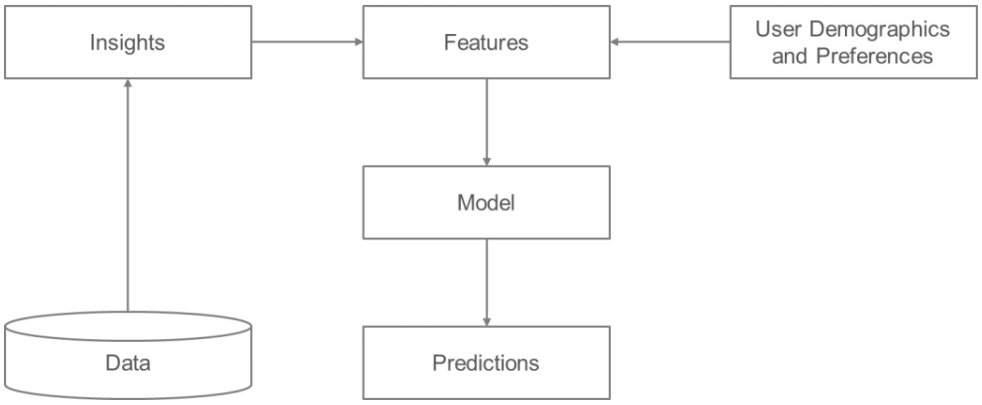
Data Analysis and Modelling: Orange tool was used for data analysis. We had narrowed upon the KNN model based on the performance and accuracy observed in Orange tool.



Orange implements functions for construction of classification models, their evaluation and scoring. Both Decision Trees and KNN model was utilized for the study. Neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. The k neighbors classification in KNeighborsClassifier is the most commonly used technique. The basic nearest neighbors regression uses uniform weights: that is, each point in the local neighborhood contributes uniformly to the classification of a query point. The default value, weights = 'uniform', assigns equal weights to all points. weights = 'distance' assigns weights proportional to the inverse of the distance from the query point. Alternatively, a user-defined function of the distance can be supplied, which will be used to compute the weights.

Evaluation: In this phase, the model results were evaluated in the context of the business objectives defined in the first phase. Based on the 2 models used, the classification accuracy and certainty was found to be better in KNN model. Hence this model was used for determining the best suitable policy.

Finally after the evaluation was successful, the code was integrated and deployed.



The above diagram provides a high level pictorial presentation on how prediction was performed.

2.2 OptaPlanner

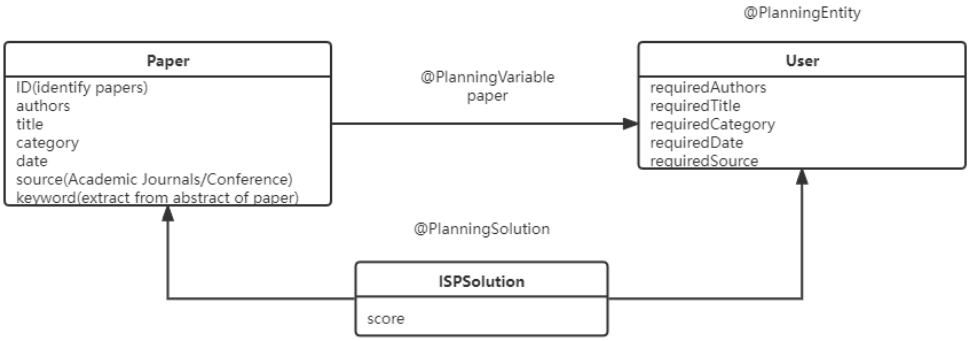
OptaPlanner is a lightweight, embedded constraint satisfaction engine satisfaction engine which optimizes planning problems. It is also known as Constraint Satisfaction Programming(which is part of the Operations Research discipline).

Usually, a planning problem has at least two levels of constrains:

1. A hard constraint should not be broken if it can be avoided.

2. A soft constraint should not be be broken if it can be avoided. For example: Teacher A does not like to teach on Friday afternoon.

These constraints define the score calculation of a planning problem. Each solution of a planning problem can be graded with a score.



The picture above depicts the design in OptaPlanner, Planning entity: the class(or classes)that doesn't change during planning[User]. In the process of solution, every paper entity and User entity will calculate and get a result. Finally, take the highest score as the final recommendation. Planning variable: the property(or properties) of a planning entity class that changes during planning. In this example, that's the paper. Solution: the class that represents a data set and contains all planning entities.[ISPSolution]

		Author Constraint	Category Constraint	Date Constraint	Title Constraint	TotalScore
Weight1(1,1,1,1)	paper A	80*1	20*1	40*1	30*1	170
	paper B	80*1	20*1	30*1	50*1	180
Weight2(1,1,3,1)	paper A	80*1	20*1	40*3	30*1	250
	paper B	80*1	20*1	30*3	50*1	240

This table show an example of calculating Paper and User. Under different weights, we get different result. 1. AuthorConstraint: The rule is implemented by calculating the similarity between Authors and requiredAuthors, and the similarity is used as the score 2. CategoryConstraint: The rule is implemented by

calculating the similarity between Category and requiredCategory, and the similarity is used as the score 3. DateConstraint: The rule is implemented by calculating the similarity between Date and requiredDate, and the similarity is used as the score 4. TitleConstraint: The rule is implemented by calculating the similarity between Title and requiredTitle, and the similarity is used as the score 5. TotalScore: Calculate the similarities by weights and sum up.

2.3 TextRank

In general, some keywords in article can well represent the main point of the article. Through searching the keywords, we can find relevant papers which can provide some reference for our study. In the processing of searching, extracting keywords is the first thing that the paper recommendation system do. The second is matching the input with the paper keywords. TextRank and Cosine Similarity are the pivotal technique.

Graph-based ranking algorithms have been successfully used in citation analysis, social networks, and the analysis of the link-structure of the World Wide Web. Arguably, these algorithms can be singled out as key elements of the paradigm-shift triggered in the field of Web Search technology, by providing a Web page ranking mechanism that relies on the collective knowledge of Web pages. In short, a graph-based ranking algorithm is a way of deciding on the importance of a vertex within a graph, by taking into account global information recursively computed from the entire graph, rather than relying only on local vertex-specific information.

Applying a similar line of thinking to lexical or semantic graphs extracted from natural language documents, results in a graph-based ranking model that can be applied to a variety of natural language processing applications, where knowledge drawn from an entire text is used in making local ranking/selection decisions. Such text-oriented ranking methods can be applied to tasks ranging from automated extraction of keyphrases, to extractive summarization and word sense disambiguatio.

Graph-based ranking algorithm are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph. The basic idea implemented by a graph-based ranking model is that of "voting" or "recommendation". When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex. Moreover, the importance of the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

2.3.1 Keywords Extraction

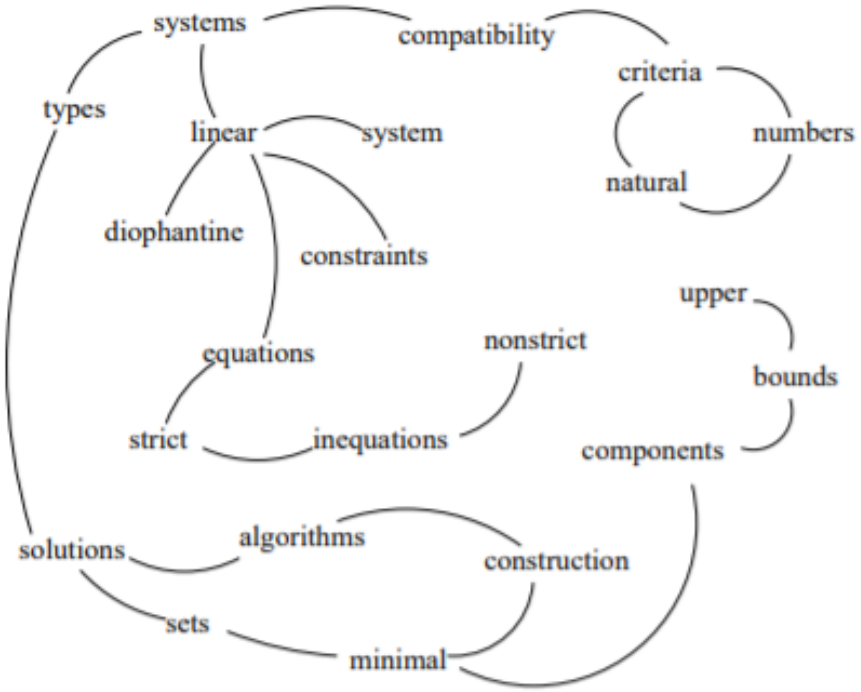
The task of a keyword extraction application is to automatically identify in a text a set of terms that best describe the document. Such keywords may constitute useful entries for building an automatic index for a document collection, can be used to classify a text, or may serve as a concise summary for a given document. Moreover, a system for automatic identification of important terms in a text can be used for the problem of terminology extraction, and construction of domain-specific dictionaries.

The expected end result for this application is a set of words or phrases that are representative for a given natural language text. The units to be ranked are therefore sequences of one or more lexical units extracted from text, and these represent the vertices that are added to the text graph. Any relation that can be defined between two lexical units is a potentially useful connection(edge) that can be added between two such vertices.

example

Here is a piece of text. The picture shows sample graph build for keyphrase extraction from an Inspec abstract. Keywords assigned by TextRank are similar to keywords assigned by human annotators.

"Compatibility of systems of linear constraints over the set of natural numbers.Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types."



2.3.2 Sentence Extraction

In a way, the problem of sentence extraction can be regarded as similar to keyword extraction, since both applications aim at identifying sequences that are more "representative" for the given text. In keyword extraction, the candidate text units consist of words or phrases, whereas in sentence extraction,we deal with entire sentences. TextRank turns out to be well suited for this type of applications, since it allows for a ranking over text units that is recursively computed based on information drawn from the entire text.

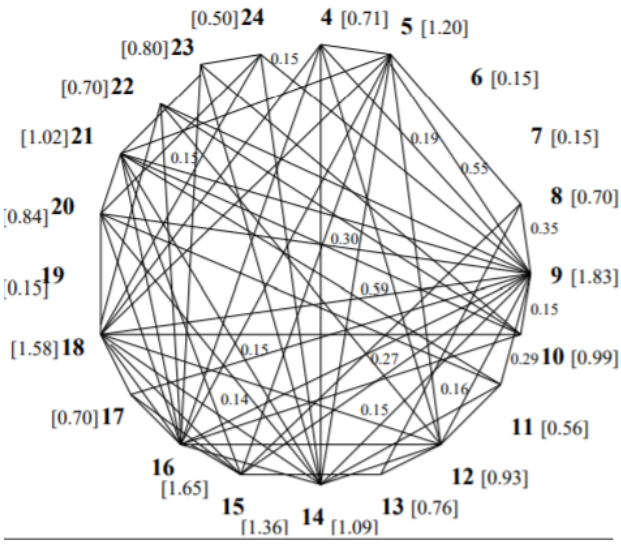
To apply TextRank, we first need to build a graph association with the text, where the graph vertices are representative for the units to be ranked. For the task of sentence extraction, the goal is to rank entire sentences, and therefore a vertex is added to the graph for each sentence in the text.

example

"TextRank extractive summary: Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil De fense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo. The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with Gilbert brought coastal flooding, strong southeast winds and up to 12 feet to Puerto Rico's south coast."

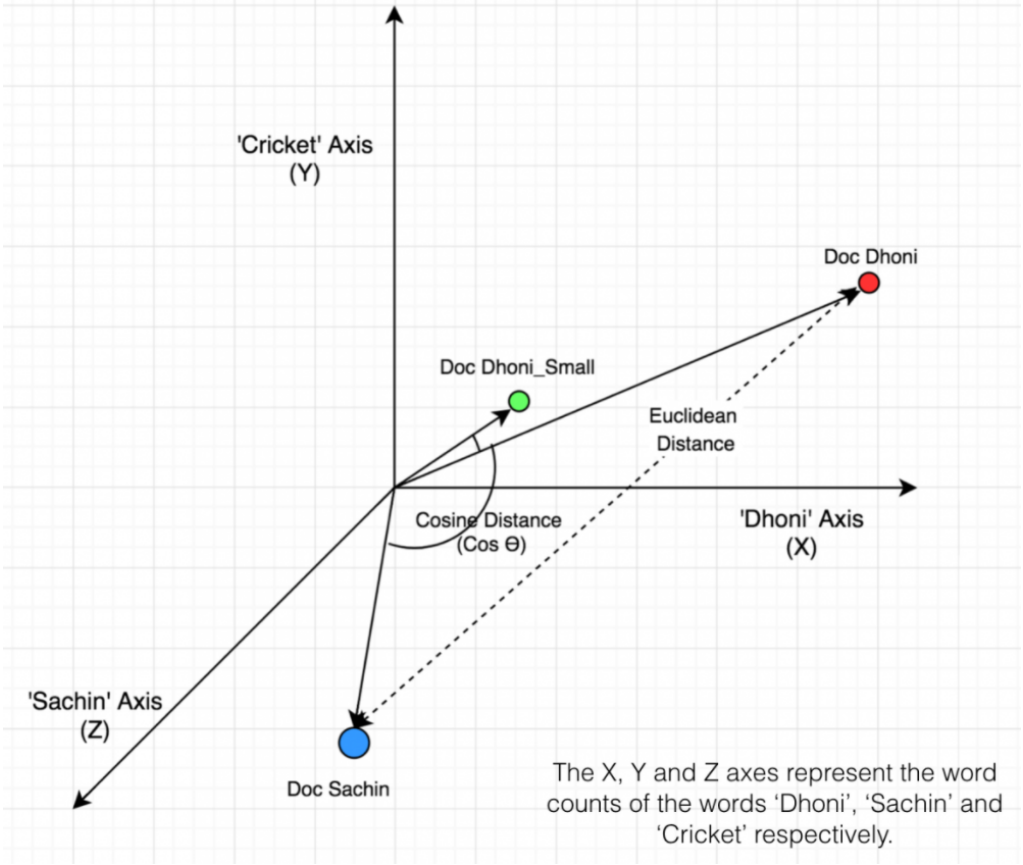
Sample graph build for sentence extraction from a newspaper article. Manually assigned summaries and TextRank extractive summary are also shown.

3: BC–HurricaineGilbert, 09–11 339
4: BC–Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.



2.4 Cosine Similarity

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The two vectors are arrays containing the word counts of two documents. The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance because of the size of the document(like, the word 'cricket' appeared 50 times in one document and 10 times in another) they could still have a smaller angle between them. The smaller the angle, higher the cosine similarity.



3 Functional Specifications

This section will explain the overall functionality of the system.

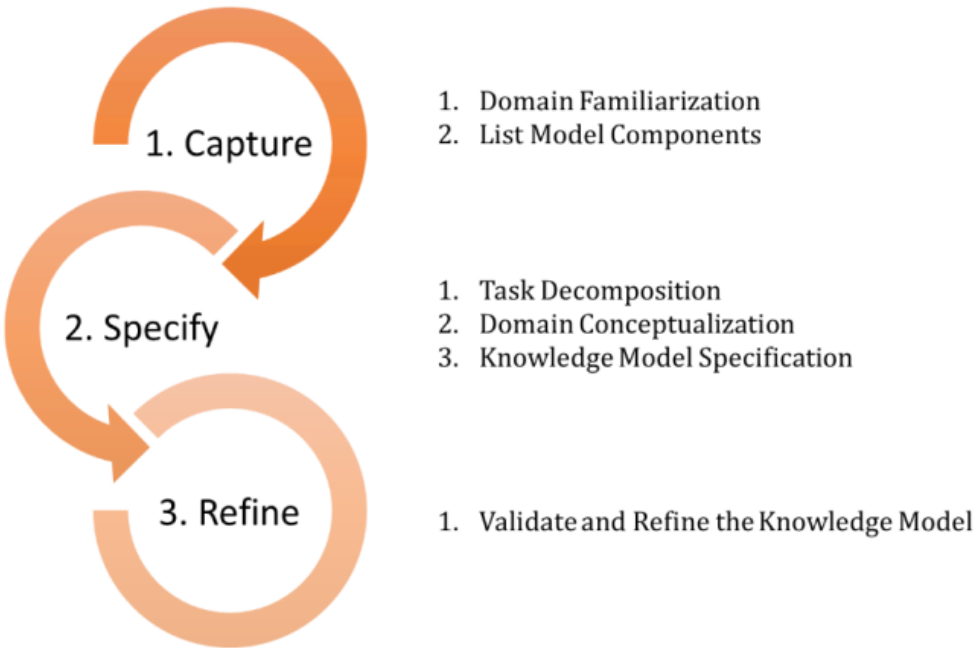
We begin the section by providing an overview of the knowledge modelling process and followed by use cases, flow chart, acticity diagram, field list, rules and followed by detailed functional explanation.

3.1 Knowledge Modelling

The diagram mentioned here, capture the differences in the Data, Information and Knowledge. How the data is synthesized to represent useful knowledge is the core objective of the knowledge modelling process.

	characteristic	*example*
Data	uninterpreted raw	...---...
Information	meaning attached to data	S O S
Knowledge	* attach purpose and competence to information * potential to generate action	emergency alert ->start rescue operation

For an synthetic tasks the system does not yet exist: the purpose of the task is to construct a system description. Knowledge modellung helps to construct an abstract description of a system of how the different knowledge components play in problem-solving, in a way that is understandable for humans. Knowledge modelling is 3 stage process as pictorially presented here.



1. Knowledge Capture: In this initial phase we began with the exercise of encompassing the data. And this data can be categorized primarily into the following categories:

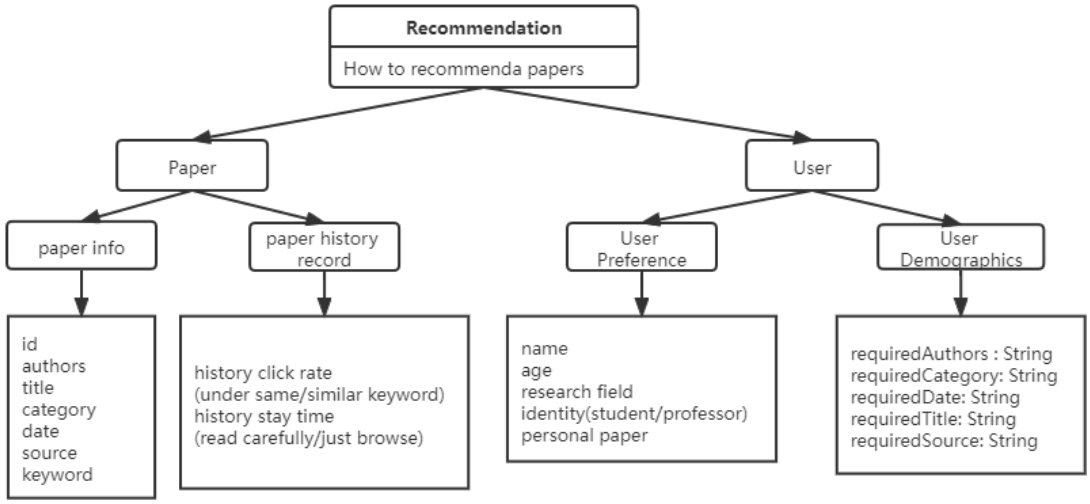
Sl No	Source	Description	Approach to obtain
1	Paper datasets	Provide information on the paper subject, field, abstract, authors, and so on	Performing web search and python crawler
2	User	Provides information on their research fields, preference and search history	Elicitation of this information by user's profile and system recording

2. Knowledge Specification: goal of this stage is to get a complete specification of the knowledge model. The following activities was carried out to build such a specification:
a. choose a task template;
b. construct an initial domain schema;
c. specify the three knowledge categories.

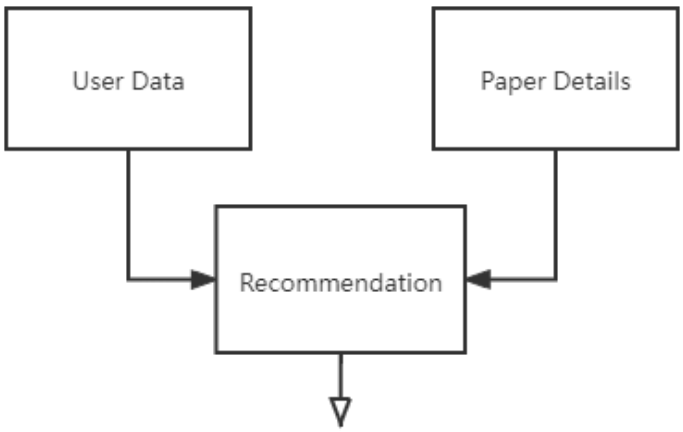
Task template can be captured as below:

Task Features	Explanation
Nature of Output	How to recommend papers
Nature of Input	Paper features, User Preferences
System Nature	Analyzing the data and constructing a recommender system
Constraints	Matching user preferences to find the suitable policy

Based on the above task template, inference can be analyzed as consisting of the problem-solving factors arranged in a hierarchical structure, it provides a abstracting mechanism over the details of the reasoning process.



Based on the inference, we have constructed the initial domain schema.



3. Knowledge Refinement Is a process of validating the knowledge model to verify whether the model is right? This was performed by testing the system built based on the knowledge model. After testing the various domain information with the respective knowledge model action for the expected behavior. Necessary fine tuning and fix was performed on the model built to ensure the expectation is achieved.

3.2 Use cases

Description of how a person who actually uses that process or system will accomplish a goal. Here we provide the details upon the different possible use cases:

1. User only provides the demographics and preference to determine the optimal paper.
2. User provides the keywords and the preferences to determine the suitable papers.
3. User would like to check upon the previous history click record to browse papers which are in hot field, or have high citation, or are latest.
4. User can choose to restart the recommendation by re-choosing the preferences again.

UC-1	Current Optimal Paper recommendation
Primary Actor(s)	User A
Initiation	User will provide the below details 1. keywords 2. paper preference information
Conditions	User chooses to provide the keywords of paper and other preference infomation such as author, source and date.
Post-conditions	The system will consume the user-provided preference details to perform the intelligent evaluation of papers to Screen related papers

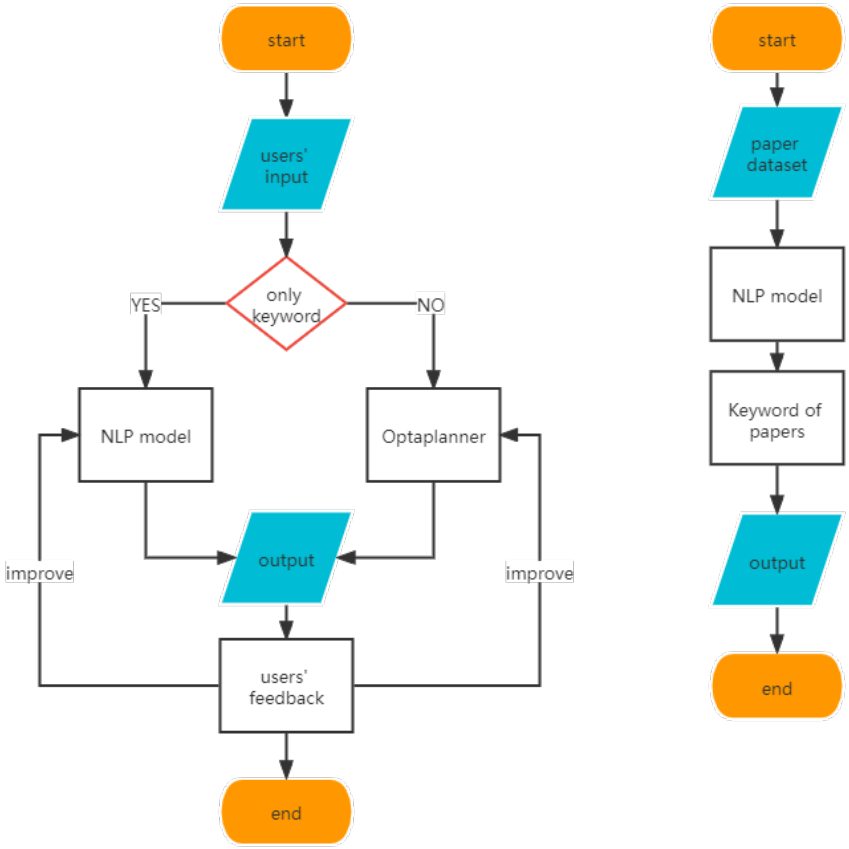
UC-2	History Click Record Determination
Primary Actor(s)	User A
Initiation	User will provide the below details 1. keywords
Conditions	User choose to browse the paper list based upon his basic information details
Post-conditions	System will consume the basic user details to perform the intelligent evaluation of all the historical data to provide a list of papers

3.3 Form Business Rules and Dependencies

Drools Rules:

Rule Name	Business Rules
similarity	calculate the similarity between users' preference and paper info (ex: requiredauthor and author)

3.4 FlowChart



The flowchart diagram helps to understand the workflow designed in the system.

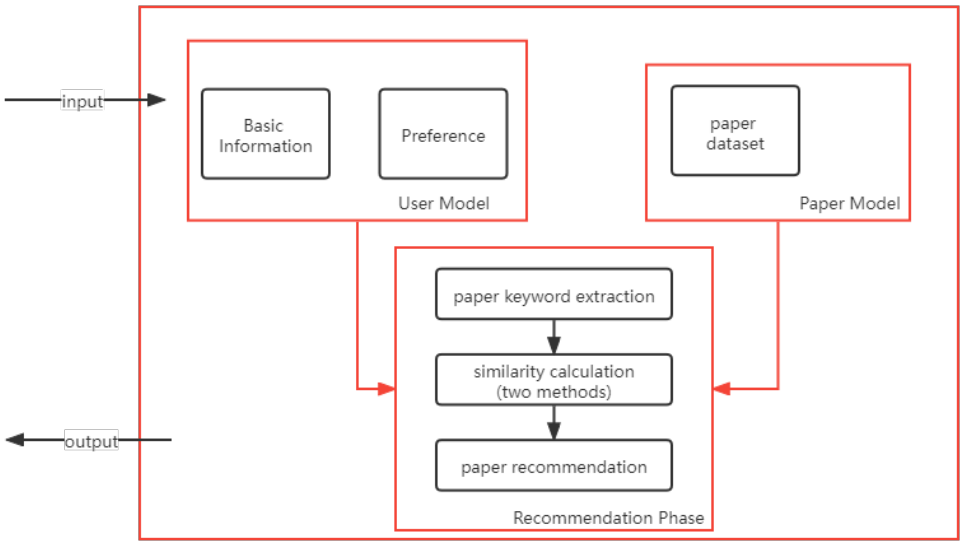
1. Process the paper dataset, extract the keywords through NLP model.
2. To capture the user's input keyword and preference.
3. When users only provide keyword, the system will enter NLP model branch. The system use cosine similarity to return recommendation result.
4. When users provide keyword and other preference information, the system will enter OptaPlanner branch. The system return recommendation result through numeric field weighted similarity.
5. Users can choose to provide the feedback and restart the process again or close the application.
6. Users can review the last recommendation result.

4 System Model

SDLC(Software Development Life Cycle)-Waterfall model is followed to build this recommender system. With the well-structured flow of phases, it has helped to build a high quality software ready to use.

Initially we begin with the clear definition of the problem statement for paper recommendation. Followed by the knowledge modelling study which has helped to guide us upon the constructive problem solving methodology. During the design phase, we discussed on the various possible approaches which can be utilized for the model, also how to provide the best experience to the user. Logical data flow diagram and a draft requirement blue print of design specification is documented based on the knowledge model.

The picture below presents logical data flow diagram, this describes how data flows and how it is utilized for the recommendation.



System Architecture

The system architecture diagram shown below is used to show relationship between the different components of the system. It is built upon the main principles of scalability, user friendliness, simplicity, high intelligence, robustness and modularity.

In addition, we have several external systems as mentioned here,

The key components of the application are: 1. Artifacts generated from KIE component for: OptaPlanner and Drools for constraint solving, workflow management system which executes business processes, converting

business logic into assets such as cases, processes, decision table and to execute rules.

2. Model: Java package containing Data Objects for persistence storage. Built with Java Persistence API(JPA).

3. Service: Backend application to expose REST based API for Frontend application usage.

4. Maven for dependency management and Django framework for application runtime.

5. MySQL Database to store all the information using the relational model that is structured in tables. Database normalization is performed to reduce data redundancy and improve data integrity.

During the Development phase, actual development started. Development is performed using the Django model. Django module is used to create a standalone, production-based application. It follows a layered architecture in which each layer communicates with the layer directly below or above. The picture below represents the Django flow architecture.