Tribhuvan University

Institute of Science & Technology

Comparative Analysis of TF-IDF and Word2vec Algorithm for Content-based Job Recommendation System

Thesis

Submitted To:

Central Department of computer Science & information Technology
Tribhuvan University

Kirtipur, Kathmandu

Nepal

In partial Fulfilment of the requirements for the Degree of Master of science in Computer Science and Information Technology

Submitted By:
Krisha Shrestha
Roll No:112/070

July, 2020

Supervisor

Mr. Bikash Balami CDCSIT, Kirtipur, Nepal

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Student's Declaration

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

… … … … … … …….

Krisha Shrestha
Roll No:112/070
Date: July, 2020

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

Supervisor's Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Ms. Krisha Shrestha** entitled **"Comparative Analysis of TF-IDF and Word2vec Algorithm for Content-based Job Recommendation System"** be accepted as in fulfilling partial requirement for completion of Master Degree of science in Computer Science and Information Technology.

… … … … … … … … … …

Mr. Bikash Balami
(Supervisor)
Central Department of Computer Science & Information Technology
Tribhuvan University
Kirtipur, Nepal

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

LETTER OF APPROVAL

We certify that we have read this dissertation and, in our opinion,, it is appreciable for the scope and quality as a dissertation in the partial fulfilment for the requirement of Master's Degree in Computer Science and Information Technology.

Evaluation Committee

……

Asst. Prof. Nawaraj Paudel
Head of Department
Central Department of Computer
Science & Information Technology
Tribhuvan University
Kirtipur, Nepal

…….

Mr. Bikash Balami
(Supervisor)
Central Department of Computer
Science & Information Technology
Tribhuvan University
Kirtipur, Nepal

…….

(External Examiner)

…….

(Internal Examiner)

Date: July, 2020

ACKNOWLEDGEMENT

I would like to express my sincere thanks to my supervisor **Mr. Bikash Balami**, Central Department of Computer Science and Information Technology, Kirtipur, Nepal for his support, motivation, suggestions and guidance. His advice was inevitable and with his help I was able to work on my own interested field and complete my thesis on time.

I am also thankful to **Mr. Nawaraj Poudel**, Head of Department, CDCSIT who has provided all the help and facilities, which I required, for the completion of my thesis.
Moreover, I would like to express my heartfelt gratitude to all my teachers at Central Department of Computer Science and Information Technology, Tribhuvan University who have imparted knowledge in various subjects.

Last but not the least; I would like to express my thanks to all my friends and my lovely parents for direct and indirect supports for the completion of this thesis.

ABSTRACT

Recommender systems are one of the most successful and widely used application of machine learning in business as they provide users ability to gather and obtain result much quicker than ever. In the current world scenario more people are using online job portal for searching jobs which require time and effort for both employer and employee to find the right person. In order to recommend relevant jobs this research presents a content-based recommendation-based approach to provide job recommendations based on TF-IDF and Word2vec in order to evaluate the performance between the two. In order to determine whether semantic relationships of words have impact on job recommendations, WMD with Word2vec are developed and compared with TF-IDF. The Word2vec model is developed using a dataset provided by Kaggle. A series of experiments are designed starting from TF-IDF vectors and cosine similarity which is set as a baseline. Further experiments were designed after observing the results of the current experiment for Word2vec and wordvec with WMD.

Keywords:Job recommendation, TF-IDF, Word2vec, Word Embedding, Cosine Similarity, Word Mover's Distance, KNN

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CBF	Content Based Filtering
CF	Collaborative Filtering
CBoW	Continuous Bag of Words
EMD	Earth Mover's Distance
KNN	K-Nearest Neighbour
NLP	Natural Language Processing
TF	Term Frequency
TF-IDF	Term Frequency Inverse Document Frequency
WMD	Word Mover's Distance

Chapter 1 Introduction

1.1 Introduction

Recommender systems are widely and most commonly used in various aspects of online world such as in video and music services like Netflix, YouTube and Spotify, product recommenders for services such as Amazon, or content recommenders for social media platforms such as Facebook and Twitter [12]. They play a large role in our daily life where the majority of such systems are applied in the field of e-commerce for e.g. product recommendations. Most users use search engines which are inadequate to obtain promising results from vast amounts of information thus the recommender system can provide an extension to search engines when these fail to produce relevant results. Recommender systems can employ various techniques by supporting users in finding preferred items and deciding about items in a collection [16].

In business-oriented networking platforms, it is more convenient for users to have recommender systems proposed for job recommendations. Different from traditional recommendation systems which recommend items to users, job recommender systems recommend one type of users (e.g., job applicants) to another type of users (e.g., recruiters). In particular, job recommender system is designed to retrieve a list of job positions to a job applicant based on his/her preferences or to generate a list of job candidates to a recruiter based on the job requirements. Job recruitment process is generally a lengthy task and there is always a screening process where most of the candidates are disqualified. Job recommender system can provide relevant jobs recommendation that the candidate are more qualified so their chances of hire is boosted.

Recommender systems are generally classified into content-based (CB) and collaborative filtering (CF) [16]. The former analyses the similarities between users and between items by their descriptions while the latter analyses the similarity between users' actions to produce recommendations. For textual information systems used in job portals, users get a lot of job offers based on their skill, qualification, job tags, etc. Using Word2vec and WMD can rank the jobs based on the semantic meanings of the words for better recommendations. Job portal sites use multiple similar tags such as developer, cook, design, etc. However, these tags can have sub-categories such as developer can mean for java or web developer and users can miss such tags but the description might contain valuable information for the candidate. Using content based approach on jobs, the purpose of this study is to evaluate relevance of results of recommender

system using semantic relationship of words using Word2vec and compare it with traditional approach.

1.2 Statement of Problem

Recommendation systems are most useful for digital B2C business models, because that's where they can directly influence the outcome of profits and can strongly boost consumer experiences. Moreover, good recommendations are increasingly becoming the standard feature expected by consumers. Hiring the right talent is a challenge faced by all companies. This challenge is amplified by the high volume of applicants if the business is labour intensive, growing and faces high attrition rates. The job openings are advertised through multiple channels like online job portals, newspaper advertisements, etc. Nowadays, job search is a task commonly done on the Internet using job search engine sites like LinkedIn, Mero Job, and others. Commonly, a job seeker has two ways to search a job using these sites: 1) doing a query based on keywords related to the job vacancy that he/she is looking for, or 2) creating and/or updating a professional profile containing data related to his/her education, professional experience, professional skills and other, and receive personalized job recommendations based on this data. Sites providing support to the former case are more popular and have a simpler structure; however, their recommendations are less accurate than those of the sites using profile data.

Job searching for an individual is always a difficult and time-consuming process. Another prospect of job searching is to find relevant jobs for the user without having to search for jobs similar yet with low probability of acquiring that job. The most common approach is for a user to enter search terms into a job search website. These search terms return a list of matched job advertisements which the user then self-evaluates [11]. Job seekers expect the search website to provide relevant job recommendation and the website to be able to navigate easily. While this method gives users more control over finding better jobs, this process is usually time consuming and users need more time to access whether they are more qualified for the job. A job recommender system needs to be designed where users are limited to or not required to perform manual search. Also, job recommender system should provide relevant job offers to the user based on their preference. When users send job employment on unqualified jobs, users spend more time on searches and unqualified jobs and waste valuable time going back and forth with the employer which they might not get the job. Thus, the ability of the job recommendation

system to provide relevant jobs to the user provides a contributing factor to saving time and effort for the user and getting jobs much faster.

Thus, recommender systems should employ recommendation engine that provides high degree of accuracy on its recommendations which helps job seekers and employers to focus on the qualified and right person for the job. As this process is time consuming, for job seekers expect job recommender engine to recommend jobs with high degree of accuracy and efficiency along with other features such as each of use.

1.3 Objectives

The objectives of research are as follows:

- To provide content-based recommendation for job recommendation using TF-IDF with Cosine Similarity and Word2vec with Cosine Similarity and Word2vec with WMD respectively.
- To compare results and evaluate the performance of TF-IDF and Word2vec

1.4 Thesis Organization

The flow of the thesis goes in this manner.

Chapter 1: Consists of introduction, problem statement and objectives.

Chapter 2: Describes the background study for the research and literature review of the related work by different authors.

Chapter3: Includes the overview of the methodology will detail the experiments proposed to address the research questions. It will define the methods, libraries and tools that will be used to test the hypothesis as well as the metrics required to evaluate the experimental results and ultimately answer the research questions.

Chapter 4: Contains the implementation overview of detail all the work done in order to complete the experiments. It will also present the experimental results of the various modelling approaches

Chapter 5: Conclusion.

Chapter 2 Background Study and Literature Review

2.1 Background Study

The topics that are discussed in background study are about recommendation systems that are used for recommending jobs. These recommender systems require some forms of pre-processing of data which is used in machine learning algorithms. TF-IDF and Word2vec text representation system are discussed along with distance metric used for recommending job with it. These topics are discussed in background study below:

2.1.1 Recommendation system

Recommender systems filter vital information from large data and can provide solutions to users when they are exposed to information overload by taking various parameters such as user preference, search patterns, preferred items, etc.[16] For instance, users cannot always find relevant items from vast amount of information using search engines and thus recommender systems can serve as a good complement to search engines by finding relevant recommendations based on the item they are currently viewing or by user search patterns [16]. These systems encompass a class of techniques and algorithms which are able to suggest “relevant” items to users. Ideally, the suggested items are as relevant to the user as possible, so that the user can engage with those items: YouTube videos, news articles, online products, and so on.

Recommender systems aims at providing recommendations for services that are targeted to specific users. The majority of such systems are applied in the field of e-commerce for e.g. product recommendations [16]. The concept of recommender systems is broad and spreads across several domains such as ML, information retrieval, and etc. Items are ranked according to their relevancy, and the most relevant ones are shown to the user. The relevancy is something that the recommender system must determine and is mainly based on historical data. For instance, if a user has watched YouTube videos about elephants, then YouTube is going to start showing you a lot of elephant videos with similar titles and themes.

Recommender systems are generally classified into two groups i.e. content-based (CB) and collaborative filtering (CF). The former analyses the similarities between users and between

items by their descriptions, while the latter analyses the similarity between users' actions to produce recommendations [16].

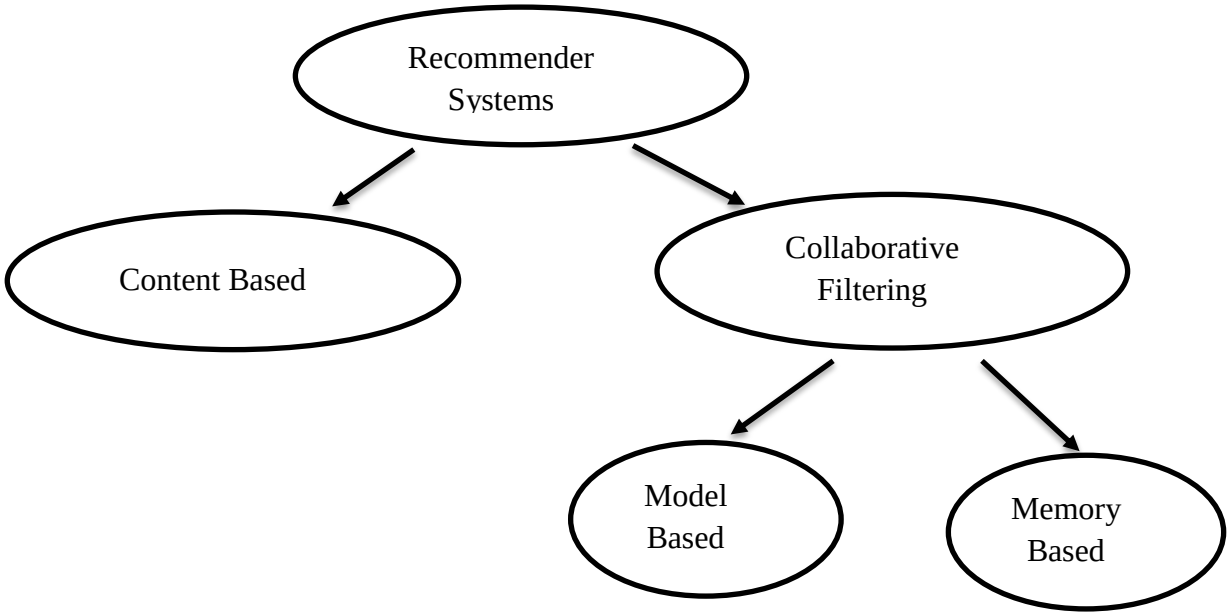

Figure 1Recommendation system classification

Content based filtering (CBF)

It is a domain dependent technique which emphasizes more on the analysis of the attributes of the items in order to make recommendations. When text documents such as publications, web articles and news are to be recommended CBF is most appropriate and successful [8]. It is one of the popular and simple recommendation systems where content or data right here refers back to the content or attributes of the products you like. So, the concept in content material-based filtering is to tag merchandise using certain key phrases, recognize what the person likes, appearance up the ones keywords within the database and propose distinctive merchandise with the same attributes.

The content-based algorithm calculates the similarity between two users or items, and produces a prediction for the user by taking the weighted average of all the ratings. Similarity computation between items or users is an important part of this approach. Multiple measures, such as Pearson correlation and vector cosine-based similarity are used for this.

Now our approach for recommending the jobs would be to recommend the jobs to the user which has similar attributes or descriptive characteristics of the jobs like it would be job description, location, qualification, experience and so on.

Collaborative filtering (CF)

Collaborative filtering is a technique used to make automatic predictions about the interests of a user by collecting user preferences obtained from multiple users with similar preference. The underlying assumption of the CF approach is that if a person A has the same preference as a person B on an item, A is more likely to have B's preference on a different issue than that of a randomly chosen person. For example, a collaborative filtering recommendation system for television tastes could make predictions about which television show a user should like given a partial list of that user's tastes (likes or dislikes) [7].
Memory based techniques have found widespread success due to their effectiveness in real life applications. These can be further divided into user based and item based collaborative filtering. In user based collaborative filtering for any given user A his nearest neighbours are calculated based on their profile and ranked and top nearest neighbours are selected. The items that are rated by these users except for the ones that user A has already rated are ranked and recommended to user A [18] Item based collaborative filtering is the opposite of user based collaborative filtering where it calculates similarity between items to identify potential users.

2.1.2 Machine learning

Machine learning (ML) is the scientific study of computer algorithms used to perform and improve the performance of a specific task without using explicit instructions, relying on patterns and inference instead [16]. Most of the data in real world scenario need some forms of pre-processing before it can be used along with ML algorithms. Pre-processing generally includes multiple steps on the data such as data cleaning, feature extraction, transformation of features, handling missing data and many more which varies from data to data. ML is generally classified into two branches i.e. supervised learning and unsupervised learning. Recommendation systems can be either supervised or unsupervised or mix of both in terms of ML language.

Supervised learning

Supervised learning is the machine learning task of learning that builds a mathematical model of a set of data that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The process of model performance evaluation generally starts by keeping some sets of input-output pairs away from training set called test set. Once the learning phase is complete the model developed is used to make predictions on the test set. The predictions are then compared with the actual output and evaluation measures are calculated according to the type of task.

Unsupervised learning

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labelled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. Word2vec generally is an unsupervised learning algorithm, designed to learn vector representations of words in order to find the semantic meaning of the words.

2.1.3 Text feature representation

In order for machines to understand the text, various approaches are used for extracting features from the text. It is one of the key components of this research. There are several types of feature representations available for text data and each of them come with their own advantages and disadvantages. The traditional approach is the Bag of Words (BOW) based feature representations such as term frequency (TF) and term frequency - inverse document frequency (TF-IDF) while the state-of-art approaches used in Word2vec are word vector representations.

Term Frequency - Inverse Document Frequency (TF-IDF)

TF-IDF is a statistical measure that evaluates how important or relevant a word is for a document in a collection of documents [14]. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. It is a product of two statistics related to the text. TF-IDF helps in evaluating the

importance of a word in a document. As most documents contain a large number of redundant words such as a, an, the, this, etc. TF-IDF focuses on more rare words which might be relevant to the user. This relevance is proportional to the number of times a word appears in the document and inversely proportional to the frequency of the word in the corpus. TF-IDF is one of the most popular term-weighting schemes today. In a survey conducted in 2015, it showed that 83% of text-based recommender systems in digital libraries mostly use TF-IDF [4].

The process for TF-IDF is explained briefly below:

Term Frequency (TF)

The initial step is to calculate the term frequency of all the documents. It measures how frequently a term has occurred in a document and this alone can be

a good representation for the text documents.

TF = $\frac{Nd}{Td}$

Where,

Nd = Number of times word w occurs in a document,

Td = Total number of words in the document

For instance, if the length of Doc A is 100 words and "the" appears 15 times out of 100

then the word "the" holds a weight of 0.15.

Inverse Document Frequency (IDF)

IDF defines how important a word is based on its frequency in the corpus. While computing TF, all the words are considered equally important. However, in documents words that appear most frequently such as "the", "is", "that", "it" and so on will provide little to no meaning during text analytics. The idea is to reduce the weights of the words that appear frequently across the corpus.

IDF = $loge\left(\frac{Tc}{Dw}\right)$

Where,

TC = Number of documents in corpus

D_w = Number of documents with word "w".

For example, in a document containing 100 words, the word ‘vacancy’ appears 3 times. The term frequency (i.e., tf) for vacancy is then (3 / 100) = 0.03. Now, assume we have 10 million documents and the word ‘vacancy’ appears in one thousands of these documents. Then, IDF is calculated as log(10,000,000 / 1,000) = 4. Thus, the TF-IDF weight is the product of these TF and IDF as:

0.03 * 4 = 0.12.

Basically, a simple definition would be:

TF-IDF Weight = TF * IDF.

Word2vec

A neural network based distributed representation of words was introduced in 2013 which is a two-layer shallow neural network that is designed to process huge amounts of text data to determine semantic relationships of the words [11]. It is a two-layer neural net that processes text by “vectorizing” words where its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus. The input provided to this is a text corpus while output received is a set of feature vectors for words in that corpus which can be used to detect relationship between words. For textual based recommender systems like job recommendation, finding semantic relationships of jobs aim to provide better recommendations to the users which cannot be done on traditional methods. The authors have recommended two architectures to calculate word vectors namely, Continuous Bag of Words (CBoW) and Skip-gram. Word embedding algorithms are used to learn the numerical representations of the words that capture both the semantic and syntactic relationship in a meaningful way. In both the cases, the network uses back-propagation to learn.

CBoW

CBOW architecture is designed to predict the current word based on the context. The architecture used here is a classic feed-forward neural network. The neighbouring words with a chosen window size are provided as an input to predict the current word. Excluding the hidden layers, the linear projection layer is shared and their index corresponding to each word will be a numeric word vector for that word. According to the authors' note,[11] CBOW is faster while skip-gram is slower but does a better job for infrequent words.

Skip-gram

It is another approach to achieve word embeddings. Here the projection layer will be present at the output layer as compared to CBoW method whose projection layer was at the input. Skip-gram aims to train a feed forward neural network to predict the neighbouring words given a target word.

2.1.4 Distance measures

Cosine Similarity:

Cosine similarity is often used to measure document similarity in text analysis. It is defined to be equal to the cosine of angle between two vectors and determines whether two vectors are closer to each other based on the angle. In the recommender system, cosine similarity can be used to infer which documents are closer to each other. As Cosine value ranges from -1 to 1 so, if two vectors make an angle 0, then cosine value would be 1, which in turn would mean that the sentences are closely related to each other. If the two vectors are orthogonal, i.e. cos 90 then it would mean that the sentences are almost unrelated. So, by comparing the job profile with the user profile, we can recommend job seekers with the jobs that scores are closer to 1. The similarities between two documents which are translated as vectors A and B using TF-IDF or Word2vec can be calculated by using cosine-similarity formula:

$$Cosine\ (X, Y) = \frac{X.Y}{||X||.||Y||}$$

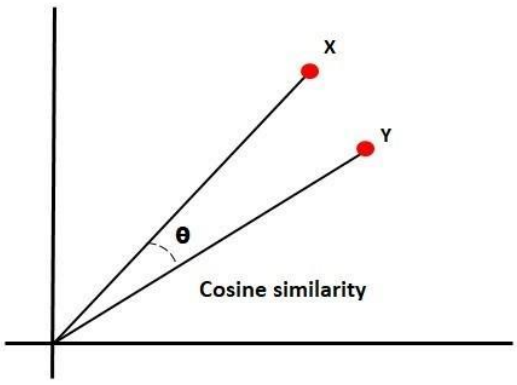

Figure 2 Cosine Similarity

Cosine can be used with BoW representations or the word vector representations to identify similarity between the text documents.

Word mover's distance:

Word Mover's Distance (WMD) is based on word embeddings that can learn semantic meanings of representations for words between two documents even when they have no words in common.[9] WMD suggests that distances and between embedded word vectors are to some degree semantically meaningful. WMD shows that this distance metric can be cast as an instance of the Earth Mover's Distance (a well-studied transportation problem for which several highly efficient solvers have been developed).

Instead of using two vectors to calculate the similarity between documents which has been done so far with TF, TF-IDF and resultant word vectors along with cosine, WMD uses word vectors for each word from both the documents. WMD shows that this distance metric is an extension to another metric called earth mover's distance (EMD) which is used to solve transportation problem [9].

Word embeddings capture a meaningful representation of each word and this is the key to calculate WMD. Words similar in meaning will be closer and transportation of such words require less work, in other words will have smaller distance. Dissimilar words will have more distance and transportation cost is high. To identify similar words between the documents
The intuition behind the method is that we find the minimum "traveling distance" between documents, in other words the most efficient way to "move" the distribution of document 1 to the distribution of document. WMD is used with advanced embedding techniques like Word2vec and Glove.

From the author's example in [9], the two sentences are in different documents: Obama speaks to the media in Illinois and: The President greets the press in Chicago. While these sentences have no words in common, they have the same semantic relationship, a fact that cannot be represented by the BOW model. In this case, the closeness of the word pairs: (Obama, President); (speaks, greets); (media, press); and (Illinois, Chicago) is not factored into the BOW-based distance.

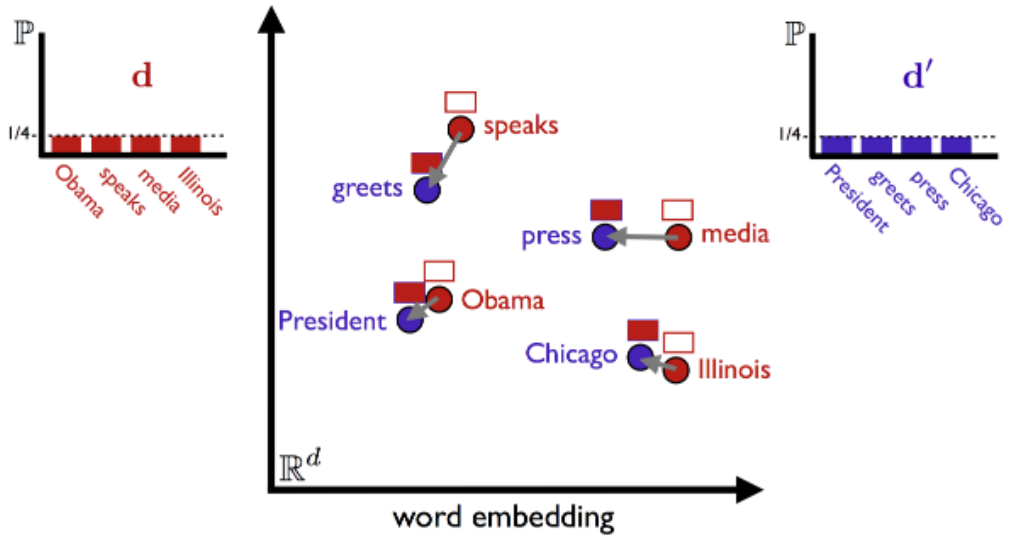

Figure 3 Word mover's distance

Here, these documents are very close to each other which shows less distance between them when using WMD and when documents with less semantic meaning have higher distances.

2.1.5 K-Nearest Neighbour (KNN)

KNN is a type of supervised instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, normalizing the training data can improve its accuracy dramatically [13]. This algorithm uses similarity measure available such as calculating distance, correlation, cosine similarity and Jaccard similarity to classify new predicted inputs from all of the provided inputs. One of the key pre-processing steps when dealing with similarity measure is to normalize the data i.e. if the data consists of different scales such as kilograms, grams, milligrams then the results will not be accurate.

2.1.5 Evaluation

In order to evaluate the performance between TF-IDF and Word2vec algorithms, evaluation metrics such as precision and recall are used. The concept of confusion matrix is used to derive metrics associated with classification [1]. The confusion matrix are used for binary, multi-class and multi-label classifications and are also used to evaluate the performance of recommender systems. Evaluation of binary classification provides a general idea of confusion matrix and the same is adapted suitably for other discrete outcome prediction

models. For recommender systems, there are generally two methods of evaluating them based on evaluation metrics like precision, recall and so on. These methods are offline evaluation and online evaluation. Offline evaluation is the most common method of evaluation of dataset and used as a baseline metric to determine whether the intended algorithm provides better accuracy than the other. While offline evaluation is performed on a dataset compiled without actual user interaction, online evaluation provides a more accurate results as it is based on the dataset of currently running system. So, the dataset differs from online to offline evaluation and confusion matrix is created to determine the evaluation metrics.

		Actual Class (Observation)	
		Y	**N**
Predicted class (expectation)	Y	TP correct result	FP unexpected result
	N	FN missing result	TN correct absence of result
TP, true positive, FP, false positive; FN, false negative; TN, true negative.			

Table 1 Confusion matrix

The structure of a binary classification confusion matrix is shown in the table 1. In the figure, we have 2x2 matrix that consists of following elements

True positive (TP): Instances of items that were predicted to be true and were true in actual.

True negative (TN): Instances of items that were predicted to be false and were false in actual.

False positive (FP): Instances of items that were predicted to be true but were false in actual.

False negative (FN): Instances of items that were predicted to be false but were true in actual.

These four measures are used to create some of the commonly used metrics to explain the performance of the model. Thus, after calculation of confusion matrix, evaluation of recommender systems can be calculated using metrics like precision, recall and F1-score.

Precision: It is defined as the proportion of the outcomes predicted that were relevant. It is the measure of the correctly identified positive cases from all the predicted positive cases. Thus, it is useful when the costs of FP are high.

$$Precision = \frac{TP}{TP+FP}$$

Recall: It is defined as the proportion of the relevant cases that were found among all the relevant cases. It is the measure of the correctly identified positive cases from all the actual positive cases. It is important when the cost of FN is high.

$$Recall = \frac{TP}{TP+FN}$$

Accuracy and F1-score are also used as an evaluation metric where accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are crucial. A brief description on accuracy and F1-score is given below:

Accuracy: It is the measure of all the correctly identified cases. It is mostly used when all the classes are equally important.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

F1-score: It is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. It gives a better measure of the incorrectly classified cases than accuracy metric.

$$F1\text{-}score = 2*\frac{Precision*Recall}{Precision+Recall}$$

2.1.6 Model Evaluation

There are two types of experiments that can be performed to evaluate the performance of the different recommender systems, offline experiments and online experiments. An offline experiment is the baseline study and relatively inexpensive way of evaluating recommender systems, typically evaluating systems based on historical data of recorded user interactions with the system [19]. The drawback of offline experiments is that they cannot directly measure the perceived quality of evaluations from a user perspective. It state that offline experiments typically provide answer to filter out inappropriate approaches leaving behind smaller set of algorithms to be tested for online experiments[19]. Online experiments are conducted by recruiting test subjects and letting them interact with the recommender system while recording behaviour or asking qualitative questions regarding the system's performance [19]. User studies are expensive to conduct, but in return, assumptions about user preferences is not necessary, as it is in offline experiments [19].

2.2 Literature Review

It states that words are represented by real valued vectors, typically in the range of few tens to hundreds of dimensions, as opposed to dimensions equal to the size of the vocabulary in traditional CBoW-based representations such as TF/TF-IDF. The distributed vector representation of the words is learned depending on the usage of the words as found in the corpus [10].

How a job recommender system could be designed to match a candidate profile by using the candidate's previous job transitions during its career. This transition model approach was compared with a more common way of producing job recommendations to candidates, using the cosine similarity method to measure the suitability of a user profile towards job advertisements. The analysis was conducted by examining 2400 LinkedIn users, and the results yielded that the transition model outperformed the cosine similarity-based approach in terms of successful recommendations [2].

Also performed analysis of content-based recommender on Irish public legal judgments to solve the problem of labelling Irish legal documents using KNN recommender using TF-IDF and Word2vec with WMD which shows promising results when using WMD even though only a sample of data was involved for it [17].

Implication of word embeddings as an application of neural networks that takes words from the vocabulary of corpus as input and translates them to lower dimensional space and applies back-propagation techniques to fine-tune the weights. Word embeddings are the weights of the first layer of the network, which is usually referred to as embedding layer or projection layer [6].

Bayesian model based on cosine similarity weighted modified classifier in disease classification prediction was studied in this paper. By constructing an improved Bias model based on cosine similarity weighting and performing it on the Spark platform, the accuracy of the improved model was compared with that of the traditional Bias model which showed that under the cluster Spark platform, the improved Bayes model has the highest efficiency in disease prediction classification [15].

Chapter 3 Methodology

3. Methodology

The methodology chapter includes implementing TF-IDF/Word2vec algorithm on a dataset to create vector representation which can be used to determine distance between the jobs for recommendations. It involves performing experiments initially from a simpler approach to slowing growing towards incorporating the state-of-the-art components in a search to identify the best recommender for job descriptions for a given job. The study involves using traditional TF-IDF approach which will be compared to Word2vec word embeddings. Each of the modules is analysed using cosine distance to recommend jobs to the user. Also, WMD is used on Word2vec embedding to identify whether it provides better recommendation to the user. Thus, the baseline starts at using TF-IDF vectors and cosine similarity and steps are taken to explore and grow towards the embeddings approach to find a similarity between the documents to recommend job titles. This research utilizes the similarity-based approach on text representations of job descriptions to recommend jobs. The recommender technique used in this research is Content Based Filtering. The impact of feature representation type and the similarity techniques are explored in this research in an attempt to find a suitable approach to generate relevant recommendations.

3.1 Dataset

The dataset from Kaggle consists of job listings of US based companies of about 10,000 entries which will be termed as dataset 1 and is our main dataset. This is a pre-crawled dataset, taken as a subset of a bigger dataset (more than 4.6 million job listings) that was created by extracting data from Dice.com, a prominent US-based technology job board. This dataset will be used for recommending jobs to the user. Since Word2vec requires large amounts of data in order to provide better semantic relationship of words [11] so to create our own domain specific model another dataset of job listing from Kaggle is about 80000 job posting which is used in addition to 10,000 job entries to create domain specific Word2vec model. These job postings were used to get word embedding for Word2vec vector representations.

3.1.1 Data Acquisition

In the data acquisition module, data from different job companies are collected and stored in a dataset. These datasets for this research are obtained in csv file format and following column of database are used and presented as below:

Crawl_time	Job_title	Category
Company	City	State
Country	Inferred_city	Inferred_state
Post_date	Job_description	Job_type
Salary_offered	Job_board	Geo
Cursor	Contact_email	Contact_phone_number
Uniq_id		

Table 2 Column of Database used

For simplicity, we select job_title, category, city, company and job_description from the dataset as text that will be used for TF-IDF feature extraction and Word2vec model creation. Before using them for text feature representation, the data is pre-processed to obtain text relevant for recommendation.

3.1.2 Data Cleaning

The dataset needs to be cleaned from null values, html tags and other special characters before passing it for feature representation.
The steps involved in data cleaning process is given below:

- Remove all entries of null values in columns in dataset table in 'job_title','category','company' and 'job description'
- Parsing the HTML text to extract text
- Remove all escape characters, punctuation, single character words, extra white spaces.
- Convert text to lower case letters.
- Removal of stop words and rare words. (For only traditional feature representations)
- Tokenize sentence to words

After data cleaning process, it was observed that data entries were reduced from 10,000 entries to about 7508 job entries. The processed dataset is saved to another file so it can be used for generating word vectors in TF-IDF and Word2vec model generation. Each row in the table from the dataset used in this research can also be referred to as a report or more technically a document. In order to make recommendations, they are merged together to form a single document.

3.2 Research Design overview

In this section it provides a workflow on how the TF-IDF and Word2vec with distance metrics which is used as a basis for providing job recommendations. Multiple experiments on TF-IDF and Word2vec algorithms using traditional and state-of-the-art techniques and these models will be evaluated. The research aims at the recommender performance on each algorithm with each incremental change made with either the feature representation or the similarity measure. The efforts are only made to study the impact of the changes made in the approach rather than the embedding model fine-tuning. Default settings are used with all the models developed and they are very similar to Google's pre-trained embedding.

The dataset block contains datasets on jobs provided by Kaggle and the dataset is pre-processed accordingly and there are subtle differences in their steps and are mentioned in the previous section. Pre-processed job documents are used to develop and test traditional TF-IDF with cosine similarity. These job documents after pre-processing are provided to Word2vec algorithms to develop word embeddings. Word embeddings are trained using combination of two datasets where both cosine and word mover's distance can be applied to identify the similarity between the documents. The baseline starts at using TF-IDF vectors and cosine similarity and steps are taken to explore and grow towards the embeddings approach to find a similarity between the documents to recommend jobs. There are three experiments for providing job recommendations using cosine similarity for TF-IDF and Word2vec as well as Word2vec with WMD distance in this research.

3.3 Recommender types

3.3.1 TF-IDF with Cosine Distance Metric

For the first experiment, TF-IDF will be used to obtain cosine similarity of jobs in order to get job recommendations. The pre-processing steps are already performed at this point and clean data is used for the combined dataset.

The below are the steps taken to develop this recommender:

1. Extract TF-IDF vectors from the data for each document.
2. Compute cosine similarity between each document using TF-IDF vectors
3. For each document identify similar documents and rank them in top-N recommendations.
4. Compute KNN with k=5 and obtain accuracy and weighted average score for precision, recall and F1-score

3.3.2 Word2vec with Cosine Distance Metric

Word2vec is used to develop word embedding from a combined dataset. Word2vec provides word vectors which are the numerical representation for the words in the document. In this approach, the Word2vec model is developed and by using the word embedding, jobs are compared using cosine distance to find which jobs are closer to one another.

The steps taken to develop Word2vec recommender with cosine similarity are:

1. Develop word embedding using job documents.
2. Extract word vectors for each word in job descriptions and calculate resultant vectors to represent the document.
3. Compute similarity between each document using vectors representing context of the documents.
4. For each document identify similar documents and rank them in top-N recommendations.
5. Compute KNN with k=5 and obtain accuracy and weighted average score for precision, recall and F1-score

3.3.3 Word vectors and WMD metric

In this method, WMD is used instead of cosine similarity. In this approach, the steps taken for providing job recommendations are:

1. Develop word embedding using job documents.
2. Extract word vector for each word in combined jobs dataset.
3. Use word vectors of each document to calculate similarity matrix using WMD.
4. For each document identify similar documents and rank them in top-N recommendations in ascending order. WMD is a distance metric and numerical value higher means higher difference between the documents.
5. Compute KNN with k=5 and obtain accuracy and weighted average score for precision, recall and F1-score

Chapter 4 Implementation and Analysis

4.1 Implementation

The TF-IDF, Word2vec, Word2vec with WMD algorithm are implemented in python programming language of community free edition PyCharm 2019 and python version of 3.7in windows platform of 64 bit OS with core i5 processor. As for providing rating for the job recommendation from the user for job recommendations, website for displaying jobs and its recommendation is implemented using Django web framework. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.

4.1.1 Python Programming Language

The programming language used in the implementation is python programming language. It's an interpreted, high-level, general-purpose programming language. It also provides great support for libraries required for machine learning such as scikit-learn, pandas, spacy and genism which is used for job recommendation.

- **Scikit-learn**
  Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It provides support for creating matrices of TF-IDF features and perform cosine similarity on the documents.
- **Pandas**
  It is a great library for reading data on csv formats where most of the datasets and results are saved for machine learning.
- **Spacy**
  spaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython. It is used for loading our domain specific Word2vec model which is used to perform cosine distance and provide top-N recommendations of the selected job.

- **Genism**

  It is a great package for processing texts, working with word vector models (such as Word2vec, Fast Text etc) and for building topic models. By using the job dataset, Word2vec model is created using this library. It also provides WMD distance on the Word2vec model which is used to provide top-N recommendations of the selected job.

4.1.2 PyCharm

PyCharm 2019.3 is a python integrated development environment for developing computer software. It is developed by JetBrains, and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition. Both can be used for commercial development.

IntelliJ IDEA Community Edition is the open source version of IntelliJ IDEA, a premier IDE (Integrated Development Environment) for Java, Groovy and other programming languages such as Python.

4.1.3 Django Web Framework

Django is a free and open source high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development without needing to reinvent the wheel [5]. For web development, the web interface provided a list of searchable jobs. Each of these job links shows job recommendations from TF-IDF, Word2vec and Word2vec with WMD. Users can rate these recommendations for specific jobs from 1 to 5 which is later used for evaluation. In order to show job recommendations to the users for obtaining user rating, Django is used to create a website for displaying jobs, its recommendations using TF-IDF and Word2vec where users can rate them from 1 to 5.

4.1.4 Test Environment

All of the algorithms are tested on the Python 3.7 environment and PyCharm as IDE for code execution. The implementation of the algorithm is done in Acer Nitro 5 with Intel (R) core ™ i58300U CPU @ 2.40GHz core processor with Installed RAM of 12 and system type of 64-bit operating system and x64 based processor.

4.2 Evaluation

Experiments for different types of recommenders have been outlined in the previous chapter and they will be used in this chapter in the experiment sections. Implementation section under experiments shows how each recommender type has been implemented and compared. Experiment summary section under experiments provides a brief explanation about the results obtained and compared. For our experiments, online experiments was chosen with a small number of people to evaluate the recommendations with user ratings since the dataset did not have user ratings of job recommendations and only job details.

Each experiment is performed with a question to answer while progressing towards solving the research problem. Experiment 1 provides research on TF-IDF and based on the results observed their further experiments will be designed accordingly. For further experiments, word embeddings of Word2vec are used which are trained using dataset provided from Kaggle so it is termed as domain specific word embeddings. For each job recommendations, user ratings were accumulated through Django based website and KNN algorithm was used to predict the user ratings. Top-10 recommendations for the job will be provided to user and user ratings from the user are obtained which will be used in KNN. These user ratings were accumulated and then used to compare with predicted ratings obtained from KNN. Different values of k were experimented where k=5 was found to provide more accuracy in precision and recall.

The parameters for evaluation of the different experiments used are provided below:

Dataset used: Pre-processed version of the dataset provided by Kaggle.

Recommender Type: Using cosine or WMD as similarity and TF-IDF/Word2vec vectors used to represent job descriptions.

Evaluation and comparison: Using KNN to calculate evaluation metrics i.e. precision and recall.

This experiment is performed using a python *programming* environment and the configuration of the computing device and time taken to develop are given below.

Computation device configuration:

- Processor: Intel core- i5 - 8330U
- RAM: 12GB

Index	Text representation	Distance metric	Embedding type
1	TF-IDF	Cosine	-
2	Word2vec	Cosine	Domain specific(Word2vec)
3	Word2vec	WMD	Domain specific(Word2vec)

Table 3 Recommenders Developed

4.2.1 Experiment 1

For first experiment on TF-IDF, job recommendations are calculated for TF-IDF vector using cosine similarity and user ratings for the job for top-10 recommendations were provided. By obtaining all the ratings of users for the jobs, KNN is used with k as 5 to obtain evaluation metrics using confusion matrix. The results obtained which shows precision, recall and f1-score for 10 different labels of ratings that user provided are shown below:

Class	Precision	Recall	F1-score
0.5	0.3125	0.084034	0.13245
1	0.5	0.045064	0.082677
1.5	0	0	0
2	0.09151	0.089827	0.090661
2.5	0.16875	0.036585	0.060134
3	0.299841	0.195517	0.236694
3.5	0.210089	0.164912	0.18478
4	0.284259	0.488055	0.359268
4.5	0.267575	0.19851	0.227925
5	0.355174	0.390551	0.372023
Accuracy	0.278795	0.278795	0.278795
Weighted Avg	0.272713	0.278795	0.256481

Table 4 Evaluation Result for TF-IDF

Observing the recommendation results of user ratings at different rating from 1 to 5, it shows precision and recall accuracy of 0.278795 along with average accuracy of 0.278795 for F1-score. Also, weighted average precision is shown as 0.272713 and recall as 0.278795 along with F1-score of 0.256481. These results will be used as a baseline to compare experiments 2 and 3 that uses Word2vec to provide job recommendations.

4.2.2 Experiment 2

From this experiment onward, word embedding techniques is used to evaluate TF-IDF and Word2vec. The aim of this experiment is to use domain specific embedding of Word2vec algorithm for job recommendations using cosine similarity and user ratings for the job for top-10 recommendations were provided. By obtaining all the ratings of users for the jobs, KNN approach to obtain evaluation metrics using confusion matrix. For k=5, the results obtained which shows precision, recall and f1-score for 10 different labels of ratings that user provided are shown below:

Class	Precision	Recall	F1-score
0.5	0.26	0.136555	0.179063
1	1	0.032189	0.06237
1.5	0	0	0
2	0.144695	0.048701	0.072874
2.5	0.142857	0.055556	0.08
3	0.294077	0.296207	0.295138
3.5	0.171894	0.280201	0.213074
4	0.354296	0.53477	0.426216
4.5	0.268589	0.188398	0.221458
5	0.406	0.239764	0.301485
Accuracy	0.298523	0.298523	0.298523
Weighted Avg	0.308417	0.298523	0.277711

Table 5 Evaluation Result for Word2vec with Cosine Similarity

Observing the recommendation results of user ratings at different rating from 1 to 5, it shows precision and recall accuracy of 0.298523 which is an increase of 7% than of TF-IDF whereas F1-score shows an accuracy of 0.298 which is increase of 7.6% more than of TF-IDF. Also, weighted average precision shows an increase of about 13% for precision and 7% increase in recall and F1-score increase of about 8.2%.

4.2.3 Experiment 3

For the final experiment, WMD algorithm is used to provide recommendations of jobs for the users. The aim of this experiment is to use domain specific embedding of Word2vec algorithm for job recommendations using distance metric of WMD and provide user ratings for the job for top-10 recommendations. By obtaining all the ratings of users for the jobs, KNN approach to obtain evaluation metrics using confusion matrix. For k=5, the results obtained which shows precision, recall and f1-score for 10 different labels of ratings that user provided are shown below:

Class	Precision	Recall	F1-score
0.5	0.262931	0.128151	0.172316
1	0.162162	0.051502	0.078176
1.5	0	0	0
2	0.1175	0.101732	0.109049
2.5	0.154545	0.023035	0.040094
3	0.274327	0.172069	0.211485
3.5	0.173313	0.169925	0.171602
4	0.324504	0.635161	0.429551
4.5	0.24422	0.089941	0.131466
5	0.356356	0.280315	0.313795
Accuracy	0.288972	0.288972	0.288972
Weighted Avg	0.262841	0.288972	0.251747

Table 6 Evaluation result for Word2vec with WMD

Observing the recommendation results of user ratings at different rating from 1 to 5, it shows precision and recall accuracy of 0.288972 which is an increase of 4% more than of TF-IDF whereas F1-score shows an accuracy of 0.29 which is increase of 3.6% more than of TF-IDF. Also, weighted average precision shows a decrease of about 3.6% for precision and 3.65% increase in recall and F1-score decrease of about 1.2%.

4.4 Experiment results

For content-based recommendation of jobs a total of three experiments based on TF-IDF, Word2vec and Word2vec with WMD were performed and evaluated using KNN. Experiment 1 was conducted using traditional feature representations TF-IDF. The results obtained in this experiment showed TF-IDF accuracy with score at 0.27 by both precision and F1 across the value of K=5 with top-10 job recommendations. This experiment provided the research with a benchmark to compare with Word2vec based recommendation. Experiment 2 and 3 showed increase in precision and F1-score where domain-specific embedding used. In experiment 2, it showed precision and recall accuracy of 0.298523 which is an increase of 7% than of TF-IDF. Also, weighted average precision shows an increase of about 13% for precision and 7% increase in recall with F1-score increase of about 8.2%. Experiment 3 was a slight turn around for domain-specific embedding when WMD was used instead of cosine similarity where precision and recall accuracy of 0.288972 which is an increase of 4% more than of TF-IDF whereas F1-score shows an accuracy of 0.29 which is increase of 3.6% more than of TF-IDF. However, weighted average precision shows a decrease of about 3.6% for precision and 3.65% increase in recall and F1-score decrease of about 1.2%. Also, weighted average precision shows a decrease of about 3.6% for precision and 3.65% increase in recall and F1-score decrease of about 1.2%. The results appeared promising with 0.29% precision and recall which showed an increase of 4% more than of TF-IDF. Here, the results show that increase in accuracy is not vastly different and that is because our Word2vec model was created from combining two datasets of about 80,000 documents consisting of word count of about 100,000 to a million words whereas Word2vec model generally consists of millions of words. When the dataset is small and the context is domain specific, TF-IDF may work better than Word2vec.

As Word2vec gives a unique vector for each word based on the words appearing around the particular word whereas TF-IDF is obtained from straightforward linear algebra. This is also one of the reason why Word2vec has more accuracy than TF-IDF.

For this research study, the results are mainly limited to the sample size of data, in the offline experiments, and test subjects. Due to the limited availability of interactions with the system it could be evaluated for limited number of users. In the case of offline experiments, the user study relies on an assumption, the assumption that the users' ratings of the candidates indicate the true quality of the recommendations. This assumes that users, with great certainty, know which candidates are relevant to them and which are not. To strengthen the assessment of the systems, further studies evaluating the systems in an online environment should be conducted.

Chapter 5 Conclusion

5.1 Conclusion

A recommender system is a viable option for dealing with information overload in several contexts, including identifying appropriate candidates for employers on a recruitment platform. The ability of the job recommendation system to provide relevant jobs to the user provides a contributing factor to saving time and effort for the user and getting jobs much faster. Thus, job recommender system should employ new and effective techniques to provide better recommendations to the users.

In this research study, the dataset acquired from Kaggle along with pre-processing steps were performed to evaluate TF-IDF and Word2vec using evaluation metrics. TF-IDF was used as a baseline for content-based recommendation to compare it with state-of-the-art techniques like Word2vec and WMD to determine its effectiveness on job recommendation. TF-IDF vectors do not account for semantic similarities in language. Word embeddings try to capture these relationships by relying on an idea that words that appear in the same context have similar meanings. Thus, Word2vec can provide better recommendations to the job seekers based on the semantic meanings of words in the different job profiles. TF-IDF and Word2vec evaluation metrics were shown have difference of accuracy by 7% as the dataset was taken from the subset of larger dataset. While experiment 3 provided results with precision and recall accuracy of 0.288972 which is an increase of 4% more than of TF-IDF whereas F1-score shows an accuracy of 0.29 which is increase of 3.6% more than of TF-IDF but weighted average precision shows a decrease of about 3.6% for precision and 3.65% increase in recall and F1-score decrease of about 1.2%. with respect to TF-IDF. TF-IDF can provide better results Word2vec model only consists of our dataset 1 instead of combining two datasets since there will be less word embeddings to work with. Through implementation of three prototype recommender systems, this study has shown that using Word2vec recommendation provides better results in comparison to TF-IDF. For each job recommendations, user ratings were accumulated through Django based website and KNN algorithm was used to predict the user ratings. By using KNN algorithm, the results showed TF-IDF accuracy with score at 0.27 by both precision and F1-score across the value of K=5 with top-10 job recommendations whereas further experiments on Word2vec it showed accuracy of 0.298523 which is an increase of 7% than of TF-IDF. Also, weighted average precision shows an increase of about

13% for precision and 7% increase in recall. WMD with Word2vec also showed promising results with 0.29- precision and recall which showed an increase of 4% more than of TF-IDF.

The domain specific embedding developed from combination of two dataset of jobs from Kaggle is not enough as pre-trained model from google which has billions of words but the experiments showed promising results for Word2vec and the accuracy of the evaluation metric can be increased with the increase in dataset from jobs dataset. These experiments provide a baseline for further research on online experiments on real world scenario with thousands of concurrent users for further evaluation of these models on content-based recommendation.

References

[1]Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing Management, 39(1), 45 - 65. Retrieved from http://www.sciencedirect.com/science/article/pii/S0306457302000213 doi: https://doi.org/10.1016/S0306-4573(02)00021-3

[2] Bradford Heap , et.al (2013). Combining Career Progression and Profile Matching in a Job Recommender System

[3] Bradley, K., Smyth, B.: Personalized Information Ordering: A Case Study in Online Recruitment. Knowledge-Based Systems 16, 269–275 (2003)

[4] Breitinger, Corinna; Gipp, Bela; Langer, Stefan (2015-07-26). "Research-paper recommender systems: a literature survey". International Journal on Digital Libraries. 17 (4): 305–338. doi:10.1007/s00799-015-0156-0. ISSN 1432-5012.

[5] Django. [Online]. Available: https://www.djangoproject.com/. [Accessed: 11-Feb-2020]

[6] Fathi, E., & Shoja, B. M. (2018). Chapter 9 - Deep Neural Networks for Natural Language Processing. In V. N. Gudivada & C. R. Rao (Eds.), Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications (Vol. 38, pp. 229 – 316). Elsevier. Retrieved from http://www.sciencedirect.com/ science/article/pii/S016971611830021X doi: 10.1016/bs.host.2018.07.006

[7] Gronvall, P., Huber-Fliflet, N., Zhang, D. J., Keeling, R., Neary, R., & Zhao, D. H. (2018, December). An Empirical Study of the Application of Machine Learning and Keyword Terms Methodologies to Privilege-Document Review Projects in Legal Matters.

[8] Isinkaye, Folasade & Folajimi, Yetunde & Ojokoh, Bolanle. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal. 16. 10.1016/j.eij.2015.06.005.

[9] Kusner, Matt & Sun, Y. & Kolkin, N.I. & Weinberger, Kilian. (2015). From word embeddings to document distances. Proceedings of the 32nd International Conference on Machine Learning (ICML 2015). 957-966.

[10] "Machine Learning textbook". www.cs.cmu.edu. Retrieved 2020-05-28.

[11] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings. Retrieved from http://arxiv.org/abs/1301.3781

[12] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh WTF:The who-to-follow system at Twitter, Proceedings of the 22nd international conference on World Wide Web

[13] Piryonesi S. Madeh; El-Diraby Tamer E. (2020-06-01). "Role of Data Analytics in Infrastructure Asset Management: Overcoming Data Size and Quality Problems". Journal of Transportation Engineering, Part B: Pavements. 146 (2): 04020022. doi:10.1061/JPEODX.0000175.

[14] Ramos, Juan. (2003). Using TF-IDF to determine word relevance in document queries.

[15] Ran, Z.. (2017). Application of improved Bayesian model based on cosine similarity weighted in prediction of disease classification. Acta Technica CSAV (Ceskoslovensk Akademie Ved). 62. 143-152.

[16] Ricci, Francesco & Rokach, Lior & Shapira, Bracha. (2010). Recommender Systems Handbook. 10.1007/978-0-387-85820-3_1.

[17] Sandesh Gangadhar(2020). Content-based Filtering Recommendation Approach to Label Irish Legal Judgements Legal Judgements.

[18] Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). The Adaptive Web. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), (pp. 291–324). Berlin, Heidelberg: Springer-Verlag. Retrieved 2019-12-24, from http://dl.acm.org/ citation.cfm?id=1768197.1768208

[19] Shani, Guy & Gunawardana, Asela. (2011). "Evaluating Recommendation Systems" .10.1007/978-0-387-85820-3_8.

Appendix

Dataset 1:

https://www.kaggle.com/PromptCloudHQ/usbased-jobs-from-dicecom

Dataset 2:

https://www.kaggle.com/kandij/notebook