2007
KARABÜK
ÜNİVERSİTESİ

DEFINING TARGET CUSTOMERS FOR VENDORS ON TWITTER WITH CONTENT-BASED FILTERING

2022
MASTER THESIS
COMPUTER ENGINEERING

Ahmed Nihad Khorsheed ALBAYATI

Thesis Advisor
Assist. Prof. Dr. Yasin ORTAKCI

DEFINING TARGET CUSTOMERS FOR VENDORS ON TWITTER WITH CONTENT-BASED FILTERING

Ahmed Nihad Khorsheed ALBAYATI

T.C.
Karabuk University
Institute of Graduate Programs
Department of Computer Engineering
Prepared as
Master Thesis

Thesis Advisor
Assist. Prof. Dr. Yasin ORTAKCI

KARABUK
July 2022

I certify that in my opinion, the thesis submitted by Ahmed Nihad Khorsheed Albayati titled “DEFINING TARGET CUSTOMERS FOR VENDORS ON TWITTER WITH CONTENT-BASED FILTERING” is fully adequate in scope and quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Yasin ORTAKCI ....
Thesis Advisor, Department of Computer Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. July 27, 2022

Examining Committee Members (Institutions)		Signature
Chairman	: Assist. Prof. Dr. Ferhat ATASOY (KBU)	....
Member	: Assist. Prof. Dr. Yasin ORTAKCI (KBU)	....
Member	: Assist. Prof. Dr. Ahmet ALBAYRAK (DU)	....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Hasan SOLMAZ ....
Director of the Institute of Graduate Programs

“I declare that all the information within this thesis has been gathered and presented by academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well.”

Ahmed Nihad Khorsheed ALBAYATI

ABSTRACT

M. Sc. Thesis

DEFINING TARGET CUSTOMERS FOR VENDORS ON TWITTER WITH CONTENT-BASED FILTERING

Ahmed Nihad Khorsheed ALBAYATI

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Yasin ORTAKCI
July 2022, 84 pages

Recommender systems (RS) have become hugely important lately. RS is used on many websites to display and sell many products. RS analyzes users' choices and also examines the properties of items. Recommendations are made based on previous preferences and interests. Using the RS in selling websites added a sophisticated tool to improve the process. The service provider needs a list of contacts to target. RS works on data containing features. One of the most important data that can be adopted in RS is Twitter data. The Twitter platform can be considered one of the most important platforms with users. This can be considered the main source of RS through data available from users and Twitter.

To reach the provision of a recommendation system that can be used for marketing purposes; in this research, we use a Twitter dataset to build Vendor Recommender System by Content-Based filtering (VRS-CB). This system can introduce people who are interested in the seller criteria. The recommendation theories and algorithms are based on content analysis. The data obtained from Twitter does not represent everything that users share, so the results are obtained only from the available data provided by Twitter. We designed and implemented this application as the Internet has become one of the most important components of our daily life. Social media via the Internet such as Facebook and Twitter have become a major role in networking and the dissemination of information. Twitter datasets are used since it is one of the best and fastest means to disseminate information at present and due to Twitter's huge number of users.

Key Words: Recommendation systems, Content-based filtering, Twitter, NLP.
Science Code : 92430

ÖZET

Yüksek Lisans Tezi

İÇERİK TABANLI FİLTRELEME İLE TWITTER'DE SATICILAR İÇİN HEDEF MÜŞTERİ TANIMLAMA

Ahmed Nihad Khorsheed ALBAYATİ

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Yasin ORTAKCI
Temmuz 2022, 84 sayfa

Öneri sistemleri (RS) son zamanlarda büyük ölçüde önem kazandı. RS birçok ürünlerin görüntülenmesi ve satışı için birçok web sitesinde kullanılmaktadır. RS, kullanıcıların seçimlerini analiz eder ve ayrıca öğelerin özelliklerini inceler. Öneriler, önceki tercihlere ve ilgi alanlarına göre yapılır. RS'nin yelkencilik veya ürün teşhir sitelerinde kullanılması, süreci iyileştirmek için karmaşık bir araç ekledi. Servis sağlayıcı, hedeflenecek bir kişi listesine ihtiyaç duyar. RS, özellikleri içeren veriler üzerinde çalışır. Tavsiye teorilerinde benimsenebilecek en önemli verilerden biri Twitter'dan elde edilen verilerdir. Twitter platformu, kullanıcıları olan en önemli platformlardan biri olarak kabul edilebilir. Bu kullanıcılar ve Twitter'dan elde edilebilecek veriler aracılığıyla RS'nin ana kaynağı sayılabilir.

Pazarlama amaçlı kullanılabilecek bir öneri sisteminin sağlanmasına ulaşmak için; bu araştırmada VRS-CB'yi oluşturmak için bir Twitter veri seti kullanıyoruz. Bu sistem satıcı kriteri ile ilgilenen kişileri tanıtabilir. İçerik analizine dayalı öneri teorileri ve algoritmalarından biridir. Twitter'dan elde edilen veriler, kullanıcıların paylaştığı her şeyi temsil etmemektedir, dolayısıyla sonuçlar yalnızca Twitter tarafından sağlanan mevcut verilerden elde edilmektedir. İnternetin günlük hayatımızın en önemli bileşenlerinden biri haline gelmesi nedeniyle bu uygulamayı tasarladık ve hayata geçirdik. Facebook ve Twitter gibi İnternet üzerinden sosyal medya, ağ oluşturma ve bilginin yayılmasında önemli bir rol haline gelmiştir. Twitter'ın çok sayıda kullanıcıya sahip olmasının yanı sıra, şu anda bilgi yaymanın en iyi ve en hızlı yollarından biri olduğu için Twitter veri seti kullanılmıştır.

Anahtar Kelimeler : Öneri sistemleri, İçerik tabanlı filtreleme, Twitter, NLP.
Bilim Kodu : 92430

ACKNOWLEDGMENT

First of all, I would like to thank my esteemed supervisor Assist. Prof Dr. Yasin ORTAKCI for his invaluable supervision, support, and tutelage during my master’s degree. My gratitude extends to the faculty of the engineering department of computer engineering, University of Karabük.

My appreciation also goes out to my wife for her encouragement and support throughout my study. Praise be to God, who has granted us the ability to complete this study and education.

CONTENTS

Page

Page

Page

LIST OF FIGURES

LIST OF TABLES

Page

Page

SYMBOLS AND ABBREVIATIONS INDEX

SYMBOLS

$a \rightarrow$	: Vector (a)
$b \rightarrow$	: Vector (b)
θ	: The angel between two vectors

ABBREVIATIONS

VRS-CB	: Vendor Recommender System by Content-Based Filtering
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise
LDA	: Latent Dirichlet Allocation
NER	: Named entity recognition
NLP	: Natural language processing
LSTM	: Long Short-Term Memory
CNN	: Convolutional Neural Network
CTR	: Collaborative Topic Regression
CBF	: Content-Based Filtering
URL	: Uniform Resource Locator
NLU	: Natural Language Understanding
DBSCAN	: Density-Based Spatial Clustering of Applications with Noise

PART 1

INTRODUCTION

The internet in general, and social media platforms in particular, have become a major part of our daily lives. These days many people spend a lot of time conducting business or watching the news or other things on these platforms. Human nature is based on people meeting and communicating with each other in different ways. One of the most popular ways today, social media, allows people to share their ideas and interests with their friends and society. Twitter is a popular social media platform and a source for data-related research in recent times. Statistics show that the number of Twitter users is more than 650 million, of which more than 150 million are daily active users. As well to this, the number of tweets is staggering, at about 500 million per day [1]. This makes Twitter a research encyclopedia in many areas, including data science. The structure of Twitter and the data generated by users gives an excellent opportunity for researchers to extract knowledge and use it in different fields. In addition, Twitter provides an API for developers to use in their scientific research.

With the development of artificial intelligence, data science, and machine learning, many systems are using user data on social networking sites to provide a tool that helps in performing some operations. This opportunity enables using the recommendation system on Twitter data for many purposes. We can define it as a subclass of information filtering systems. It has different filtering types such as collaborative filtering, content, or knowledge-based filtering. A recommendation system is built on two concepts: entities and users [2]. It filters the data utilizing the past behaviors of the users or entities they are interested in. RS can be an alternative to search engines since they help users find items they need or request. One of the areas we can utilize RS is Twitter since it is an enormous data source, which includes user information. We provide VRS-CB, which is an intelligent system based on

content filtering to make user recommendations based on Twitter data. VRS-CB is an effective tool to help the vendors obtain a target list of users according to keywords. The mechanism helps to improve the e-commerce process.
The contributions of this paper are as follows:

- We collect data from Twitter based on keywords within five special fields to provide data sets for the study
- We are building a vendor tool using content filtering to suggest a list of people to sellers within the data we have
- The system is built based on ready-made data, and then the results are compared with the collected data
- We present tables comparing the results by keywords that are used with the ready data and with our collected data

The rest of this paper is organized as follows. In Part 2, we review the relevant literature, conducting a study of the literature on RS and Twitter data. In Part 3 we do a theoretical study on the main components used. In Part 4 we present the research methodology, algorithms, and techniques in detail, and analyze the results. In Part 5 we present the experiments. After that, we discuss the results, present a summary of the working mechanisms, and future works.

PART 2

LITERATURE REVIEW

RS offers suggestions to users on what they might like without even searching. RS may be based on user behavior or data analysis. The data used with the RS may be data from sites for the sale of services and things, or social networking sites. The scope of Twitter data usage includes news recommendations, follower recommendations, tweet recommendations, and others.

The tweets have been used in different research fields, Kashfia Sailunaz and Reda Alhajj have worked to analyze and discover the sentiment of Twitter [3]. They introduced RS that recommends after analyzing the tweets and extracting the sentiments from them. They were also able to provide a study on the impact of users on others. The process focused on finding datasets about any topic to get users responses to find out how much they agree with that topic. Recommendations are made both personally and in general based on feelings towards a particular topic. In [3] they were able to provide an innovative way in which to build a personal recommendation system for social networks, specifically Twitter.

Brahim Dib et al. provide a Followee recommendation mechanism based on Semantic analysis [4]. In this article, the semantic analysis of the data contents of Twitter users and the numbers of followers and followers for that user was relied on within real data. Many experiments have been conducted and they have proven that the adoption of a semantic gap for text contents adds greater quality in the process of recommending like-minded users. They were able to build a system for suggesting to users the ideas most suitable for each other.

NadaBen Lhachemi and El Habib Nfaoui proposed Tweet Embeddings for Hashtag Recommendation [5]. The policy of creating a hashtag on Twitter makes it difficult to find appropriate hashtags for tweets, so the recommendation of the hashtag is very important for Twitter users, especially bloggers. They explained that choosing the right hashtag helps users avoid great stress in delivering information in real-time. They use word embeddings on the trained dataset and combine the extracted features with another clustering algorithm like DBSCAN. The aim of using a clustering algorithm is to filter tweets and convert them from an inconsistent group to a group of tweets that are similar in meaning. The recommendation is based on analyzing tweets whilst ignoring other Tweet components such as URLs. Link recommendation is another important field in analyzing tweets and providing RS.

Nazpar Yazdanfar and Alex Thomo proposed a link recommender to recommend URLs for Twitter users [6]. The idea was to place URLs beside tweets to make the connection with the resources easier. They focused on a neighborhood-based RS in case of link recommendation to the users. The main process was analyzing the hashtags and using them as a keyword for URLs. In this research, they proved that the collaborative filtering in the recommendation is more accurate in the case of finding similarities between the hashtag and the users, which they were able to increase the efficiency of similar systems. They concluded that the accuracy in recommending links depended on the timing of the tweets used in the system. The more recent tweets, the more accurate the recommendation.

Natural text processing involves filtering and preparing tweets to extract some features or information from them. On this basis, a large amount of literature has been created that helps assist humanity by mining information inside tweets. Louis Ngamassi et al have presented a study to analyze tweets to assist during disasters. The study is based on the tweets during the Hurricane Harvey disaster [7]. Latent Dirichlet Allocation (LDA) technology was used to extract information from tweets. The process of extracting information in times of disaster leads to a better understanding of what people need. They tried to make the best information mining by identifying a set of topics that excites Twitter users. All the topics were about the announcement of disasters and how citizens responded. The topics included canceling intercity mobility, energy threats, and climate change. After the analysis and study, a recommendation system was presented to help relief officials in the disasters.

Their recommendation included recommending that people practice writing simple tweets related to disasters, recommending the creation of guidance groups to bring life back to normal after disasters, making it easier to access information about moving between cities, and recommending the use of words and tags to facilitate the classification of tweets.

Chanchal Suman et al. presented a study based on the previous concept of text processing to identify entities with Twitter data [8]. Named entity recognition (NER) can be described as an important task in NLP. The NER was extracted from the tweets through to the presence of texts of a limited length and text that contained hashtags. Moreover, images and links are also determinants of NER. To develop methods based on deep learning, a study is presented to find entities from tweets such as images and hyperlinks by incorporating handcrafted features. A hybrid two-way model is used, the model consists of CNN and LSTM, and the results of the proposed systems are presented and compared between them.

Recommendations in RS are made based on textual data, and it may be possible to use tweets, articles, or even information based on image processing using image data, or make recommendations about an object and compare the ratings for that object or users' preference, as is done in recommending films or products. Lots of information circulates among social media users, Hao Wu et al. presented a study and provided a recommendation for users to classify items based on their social media information [9]. The study aimed to use information that has multiple sources as the comments of associated users. The model is built on the assumption that trusted friends have similar tastes and preferences. Through this study, the users' comments and the content of the elements in the social environment were combined into a single algorithm. Through the experiments that were conducted on a different set of data, the researchers reached several conclusions to improve the accuracy of the recommendations by providing a model for calculating the click rate for social network users. They found that the presence of similar people affected the user's taste and made a significant impact on decisions that affect the recommendation results.

Mainly, RS are vary with the different types of filtering used with the data. Collaborative filtering, content-based filtering, or sometimes hybrid filtering may be used. Jin Hyun Joao and others have proposed a recommendation system using collaborative filtering [10]. This study is based on analyzing the personal tendencies of the client based on the data collected when visiting clients' companies. The purpose of presenting the study was to provide a recommendation system for the local companies that were most preferred by people. GPS-measured distance data has been combined to add features for a more accurate recommendation.

To obtain practical results, mobile coupons were used. Collaborative filtering recommendation is made based on phone usage and consumption models after filtering data from redundant information. The data used during the visit to the customer service points were collected along with the time taken to know the effectiveness of the visit for a more accurate recommendation. The result of the collaborative filtering was to find purchasing information for similar users in the companies that were extracted.

Given that the RS work on the principle of prediction based on data and analysis, R.J. Kuoa et al. proposed a method by combining metaheuristic and the perturbation based on K-nearest neighbors [11]. The main aim of the proposed method was to reduce the effects of discrepancies between the data, especially the less-commonly-used data. They mentioned that if collaborative filtering is implemented to find user interests, the effects of the differences will cause incorrect recommendation results. To solve this issue, they proposed calculating and unifying the similarities that improve the algorithm's performance, which makes it possible to obtain the best recommendation result.

To improve the prediction performance for the recommendation, a hybrid system was used that works with three advanced versions of the KNN algorithm, and the results were measured with three data of different sizes. The results prove that the proposed method can recommend more accurately with the need for some time when it is used in real-time on the Internet. In addition, many systems suffered from differences in the data so they used a new method to alleviate this problem. Despite the use of RS in many fields, it remains the leader in making suggestions to users. Many sites used RS, YouTube to recommend videos, Netflix to recommend movies, and even LinkedIn to recommend jobs. Jieun Son and Seoung Bum proposed a study to improve the recommendation of items to users based on content-based filtering [12]. The proposed system can be considered a good model in this field since it does not always recommend similar elements. The elements in the dataset are compared to obtain various criteria to ensure that various elements are recommended, which increases the attractiveness of the platform being used. The network that contains the relationships and links between the elements is also analyzed, which addresses the problem of sparsity.

As was indicated, RS is used to suggest new items that were not previously shown to users. Many of these systems were built based on different mechanisms and algorithms. The main idea in each of them is to analyze and filter information about users and objects. Urszula and Michał [13] presented a Differential Evolution Algorithm supported by RS. They proposed the ranking function for directly optimizing the average precision and mentioned that users and their choices must be analyzed to create a specific preference profile. This method adjusts the generated recommendation to the user's preference. Items are represented through a feature vector generated using user-item matrix factorization. They proposed that the number of items rated by the user in the system significantly influences the results of the tests.

Table 2.1. Reviewed literature summary.

Resource	Used Methods	Summary
[3]	Naïve Bayes, SVM, and Random Forest	They reached the recommendation of most Twitter users who express feelings about a particular emotion or problem
[4]	Precision and recall for top Relevant Candidates	Introduce a system that improves 5% of the recall value in the case of recommending five followers
[5]	word2vec, and DBSCAN clustering	Introducing a robust system for recommending hashtags that are related to the entered Tweet content.
[6]	RMSE, SVD, Collaborative Filtering	Introducing a neighborhood-based system that is significantly superior to a matrix factor-based system.
[7]	LDA algorithm and KNIME Analytics Platform	Analyzing the tweets and performing four recommendations regarding the topic of relief and disasters
[8]	CNN and LSTM algorithms	The proposed architecture is showing better intricacy relational performance.
[9]	LDA and CTR	Provide a recommendation form that works with better accuracy and is more robust than modern methods
[10]	Clustering and K-means	Enable both the user and the service provider with lists of recommendations relevant to them.
[11]	K-nearest neighbors & densest imputation	Provide a model to reduce the difficulties of data sparsity and similarity in collaborative filtering
[12]	Content-based filtering and K-means	Presenting a model to address the problem of differences and the problem of specialization in the data of RS
[13]	Differential Evolution Algorithm	They improved the quality of the generated recommendations by comparing the results with the other techniques.

PART 3

THEORETICAL BACKGROUND

RS has been considered one of the most important data science tools. Many research was presented to provide a tool that helps facilitate work in one of the living areas. At some stage, the presence of the problem or obstacle helps to think about providing smart systems that help users reach better results.

3.1. PROBLEM DEFINITION

Lack of knowledge related to the target consumer audience can reduce sales. This problem generates failure to achieve the target limit in the required sales scheme. Moreover, not focusing on targeting a category in the sale leads to the accumulation of materials and their failure to sell them. Many online trading websites need techniques to develop and boost their sales strategy. RS has been widely used to recommend merchandise on electronic sales sites. It is known that in the purchasing process in all its forms, the choice is difficult with the presence of thousands of items, especially since there is no ability to review them. In the absence of RS, friends are sometimes consulted about the best elements; in that case, their recommendation is only based on their experiences. With the development of technology, RS are widely used in electronic commerce tools. RS analyzes the opinions of many users to provide a list of items that are recommended to users. Amazon sales reports have proven that the recommended items have a much higher percentage of sales than the unrecommended ones [14].

The presence of RS on trading websites leads to increase sales and profits and performs the knowledge about customers’ needs. On the other hand, the vendors also need to provide a group of real people to explore their sales and services. The research problem comes from the importance of suggesting and recommending a group of people based on criteria. In the operations of commerce and sales, the basic elements are the merchandise and the target group. Identifying people who will provide products and goods is the most important component of successful trading [15]. Recommender systems technology can be considered the most importantly personal service tool in the Internet marketing activities of e-commerce [16]. Although it was formerly used as a tool for developing electronic sales, it can also be used in another way and for problem-solving. In our research, we use content-based filtering to recommend a group of people as a target by analyzing their tweets. The targeted people are whom may want to purchase vendors’ services.

3.2. RECOMMENDATION SYSTEMS

3.2.1. Basic Definition

Since recommender systems are an important field in data science, they are classified as a sub-category of information filtering systems. RS usually makes predictions based on the 'rating' or 'preference' the user might give to an item [17]. RS aims to expect the user likes on a product. RS may be used to recommend a product to users. Products may be songs, videos, or items purchased from websites. It can also recommend that users be followed up, categorized, or targeted in online buying and selling. RS depend mainly on the architecture of the system, the type of used dataset, and the type of used filtering.

RS can be built using collaborative filtering or content-based filtering. Some systems may include using both types to build a hybrid recommendation system [17]. The recommendation process is done using engines. A recommendation engine is a data filtering technology that uses machine learning algorithms to propose the products that are most relevant to a user. It works by filtering the data of customer behavior, which may be obtained implicitly or explicitly.

RS has been widely used in smart systems and in a lot of literature and research. RS has recently occupied an important place in websites and most areas of life. They are designed to predict the products that may the users be interested in. RS reduce the cost of special transactions in the online shopping environment and also take advantage of time, which increases sales and profits. RS are also important to redefine users' web browsing experience, retain customers, and enhance their shopping experience [18].

3.2.2. Recommendation Approaches

There are often three main types: collaborative and content-based, and the third type is the combination of collaborative and content-based filtering. Collaborative filtering can be defined as filtering that is based on collecting and analyzing some data about the user's activity and behavior to predict what a person may prefer by calculating similarities with others [28]. Data is fetched, and matrix formulas are used to calculate similarity. Collaborative filtering has the advantage of not having to understand the content, unlike content-based filtering.

In content-based filtering, the prediction is that if a user likes a particular product, he will likely like a similar product. The similarity of the elements is calculated by comparing the type, color, shape, length of words, and their meanings in the text data. In this filter, the suggested items to the users will be similar to what they liked before [29].

In some cases, systems need both content-based and collaborative filtering. The hybrid recommendation model uses both collaborative and content-based filtering data to make the recommendation which usually beats them. Natural language processors are created, and vector equations are created to find similarities. In the following steps, items are filtered and recommended based on behaviors and preferences in the dataset [30]. Each type of recommendation system works based on the data in the system, artificial intelligence, and data science algorithms. The data determines the system's efficiency; when the data is accurate, the system will be more efficient.

3.3. RECOMMENDATION ON TWITTER

3.3.1. Twitter Data

Twitter data refers to any data or information from Twitter users. The data can contain tweets, likes, retweets, mentions, and comments shared with other users. This information can tell us about the circumstances of the Tweets when they were posted, the number of people who interacted with them, and the number of people who reposted them. On the other hand, this data may be specific to users, their pages, and even some characteristics of their accounts.

Twitter data helps build many smart systems by using it as different dataset models. The great interest of researchers in bringing Twitter data and conducting research and studies on it or exploiting it in building smart systems prompted the Twitter platform to support these researchers in 2021. Twitter has released an academic research product path that allows researchers to access the archive of old public tweets, which has allowed the researchers to pull up 10 million tweets per month [19]. A large amount of such data leads to the construction of systems with high accuracy, especially systems that do not need time-related data. Although many Twitter items can be recommended to the users, we determined the most significant three types

3.3.1.1. Tweets

Since Twitter is one of the most important global platforms for social networking, it is natural that many users exchange opinions, news, and feelings. Communication between Twitter users is done by writing posts called tweets. Tweets consist of sentences written by the user, including some links, provided that they do not exceed 280 characters [20], so they are often shortened, and some important parts of events are written without using the correct grammar in the texts. Users post tweets on Twitter that reflect their opinions.

Twitter allows using part of those tweets as text datasets. This option has led many researchers to develop artificial intelligence systems for word processing. The tweets can be considered an essential data source in many fields. For instance, many methods have been developed to automatically determine Twitter users’' users' geolocation using their tweets. A novel methodology predicting Twitter users' home locations examines the content of tweets based on Sentiment Analysis [21]. On the other hand, we can consider tweets as big data of online news that can be good sources for deep learning. In this way, we can classify the news for the user as interesting or uninteresting [22].

3.3.1.2. Hashtags

Another item of Twitter is hashtags that allow users to type their opinions on the same topics. A hashtag is a single word or some words preceded by a hash (#). The hashtag is used in many places, such as:

- A certain circumstance or event may occur that causes most users to share their opinions on that topic.
- Users need to categorize their tweets into certain categories.

Using hashtags helps spread tweets widely and reach people easily. Clicking on the hashtag will display all tweets containing that hashtag. The hashtag can be included in any part of the tweet. According to Hayley Dorney's blog [23], hashtags are extremely important to use with tweets that we want to reach the largest number of audiences. Like other platforms and websites, Twitter also has a business account. Reports have proven that targeted tweets reach the target groups more precisely with an appropriate hashtag. It was also used in marketing and advertising; It's proven that using easy-to-remember hashtags with clear meanings leads to a higher marketing funnel. The increase is 18% in messages of communication with the brand, 8% is a wide knowledge of the brand, and 3% changes the recipient’' 's intention to purchase the service [23]. Hashtags greatly benefit from gathering similar tweets for analysis [24].

At the same time, they enable researchers to conduct their studies on Twitter in a shorter time and in a more effective way. In addition, they are also used to measure the similarity and relatedness of the text of tweets. Recently, the importance of the hashtag in communicating or categorizing tweets made many literatures study the recommendation on the hashtag, such as [25][26].

In this literature, models and algorithms are built to suggest the most appropriate hashtag for tweets. Data is extracted from Twitter and applied to natural language processing algorithms tools to understand the context of the texts used to train the system. The hashtag is obtained by comparing the data in the dataset using the similarity mechanism. The great interest in hashtags shows the importance of this type of Twitter data.

3.3.1.3. User data

All Twitter user data fall into this category. User data includes the name and username, the date of joining Twitter, the number of followers and followers, and even the color and image of the account. Follower/followee are two concepts used in the Twitter platform to indicate that user accounts follow each other to see what they share. Today, it is a trend to suggest a list of Twitter accounts to follow for users who may have common interests. The mechanism of following users on Twitter can be likened to friendship in real life, where friends agree on some things and ideas or most topics. This similarity in opinions and interests added the possibility of analyzing tweets based on the evaluation of users and their followers, enabling researchers to provide many RS.

User data and ratings can be used in collaborative filtering and content-based and hybrid systems. In addition, affiliate RS has been introduced based on an interest in analyzing and filtering user data. Followee recommendation on Twitter, which is based on analyzing a set of data, is one of the application areas of RS. The followee RS is formed on semantic analysis of follower/followee topology in the Twitter user profile.

In [27], Users' data was analyzed, and a recommendation system was presented considering the modeling as the follower/followee topology was studied to recommend similar users.

3.4. CONTENT-BASED FILTERING (CBF)

CBF is a sophisticated data filtering technology that uses similar features to make decisions. Content-based filtering is one of the most important recommendation system techniques that use user data and information for the recommendation. Compared to the collaborative approach, a content-based system considers additional information about users or objects. It depends mainly on the largest percentage of similarity to make the recommendation. The idea of this system is to build a model based on the available features and feedback between the user and the objects. It is mostly used to recommend items in e-commerce stores, movies, and videos on many websites. One of the disadvantages of this system is that it is not possible to recommend new users or items because there is not enough data to achieve the required similarity ratio. Thus, evaluation-gathering techniques or features can be added in such cases to avoid system failure [31].

Content-based filtering is used for various purposes; it may be used to recommend for the classification aspect or the regression aspect. Classification predicts whether the user likes the item or not, and regression predicts the classification that the user provides for that item. When RS is used in electronic commerce operations, it may be recommended to either users or vendors. If the system is to recommend to the user, it is built to focus either on the user's features or the items [31].

The approach focuses on the item when the system is based on user features. Users' preferences may differ, so this method is less customizable. If there is a great similarity in the users' preferences, this method is more powerful. In the case of working with the features of the items, the method is user-centered. A single user deals with relatively few elements, so the recommendation is limited.

This model is less powerful than the previous model that focuses on the features of the elements [32]. However, if it is used for the vendors, it works by calculating the similarity of the system criteria to recommend goods that can be offered to a specific user or a group of people to offer a specific item to them [33].

3.5. VRS-CB CONCEPTS

Data is often used in designing and implementing intelligent systems in some fields. To simulate the evolution in data science, we analyze, filter, and prepare tweets for use within the VRS-CB. It studies the tweets obtained from Twitter on the one hand and the keywords of sellers on the other. In the first step, it filters the tweets based on those keywords, and then it finds and suggests people who are related to them. Text-processing and content-based filtering algorithms are used for tweets.

PART 4

METHODOLOGY

The project aimed to provide a target list containing several customers sharing the same requirements. The system selects the tweets from a group of people and picks up the users who have similar ideas based on their tweets. We introduce users as customers list for the online sales process since their interests are close to the vendor's standard. Firstly, natural language processing techniques are used to prepare the dataset. The preprocessing step simplifies the process of finding tweets associated with keywords, thus each tweet has an ID in each dataset. While the system will display them to the vendor. In the next step, the vendor will choose the ID of the required tweets to find similarities. The recommendation based on content analysis is applied to recommend as many users as required, and user names are suggested in the output list. The system was built using three recommendation methods and three similarity assessment metrics. In this chapter, we will explain the algorithms and tools used beside the dataset in detail during the construction of the system.

4.1. DATASET

We are using two datasets:

4.1.1 Nike Dataset

Nike had a campaign for a while called JustDoIt. For the 30th anniversary of this campaign, Nike made a partnership with Colin Kaepernick, the American famous football player. Our first data set was collected on the next day of that partnership September 7, 2018, and provided by ELIAS DABBAS on kaggle.com [41]. The dataset has 72 columns and 5000 tweets with the hashtag (JustDoIt).

Each record contained data regards to the tweet or the user. The user information fields name started with user while the fields regard the tweet started by tweet.

For using the dataset in the model we reduced the columns and extructed the only important 17 columns. The model is based on the main column of tweet_full_text that has the text of the tweet. Figure 4.1 represents the most common stop words of the dataset. as a preparation step these words are removed and the data cleaned.

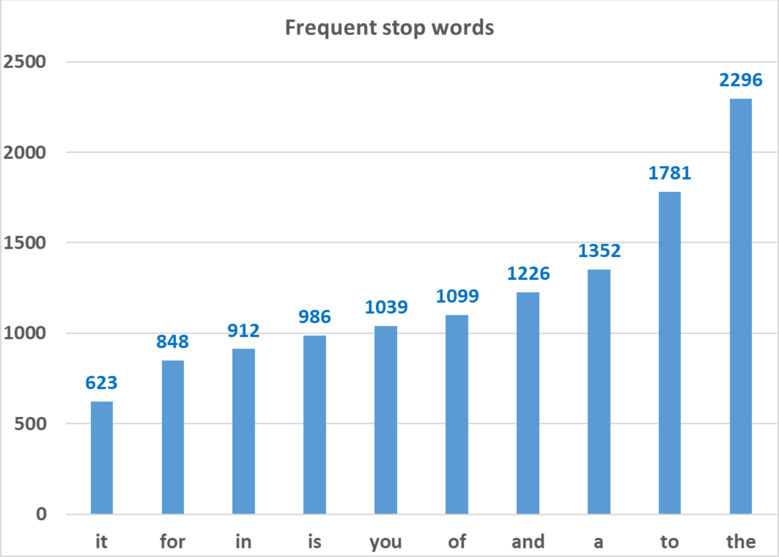


Figure 4.1. Frequent stop words of Nike dataset.

This data is collected based on keywords that are specific to a topic or hashtag, we can now review the words that are frequently mentioned in this dataset as shown in Figure 4.2.

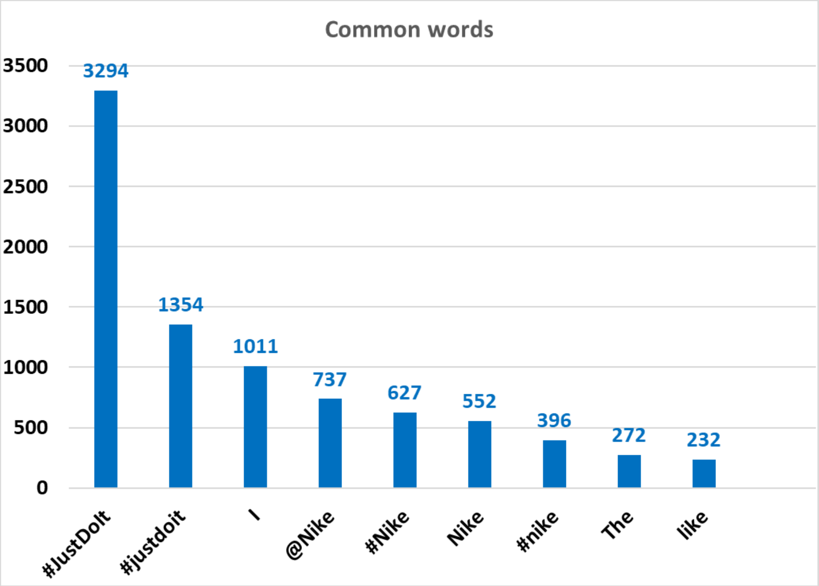


Figure 4.2. Common words in the Nike dataset.

In the stage of preparing the model, the dataset was prepared, the contents of the tweets were arranged, and the useless information was removed from them, so that the system was based on it, and then the system was applied to the data collected later.

4.1.2. Collected Dataset

The second collection of datasets is five self-collected datasets according to five categories of keywords. The datasets are collected by using Python open-source library “Twint”. This library allows getting tweets and information about the time, place, writer, and language, and also links and pictures if included in the tweet. Before using “Twint” many variables should follow and be determined.

First, the function is configured, then we have to consider the date limits to collect data during that period. The most important parameter in determining the keywords of the search. In our case and since the tool is for vendors, we have used the process to collect five types of datasets as follows:

- Cloths dataset: This dataset was collected based on keywords related to clothing and international clothing brands and accessories. The following words are used to find the appropriate tweets of this type (Clothes, T-shirts, trouser, shirts, jacket, suit, hoodies, Nike, Puma, Adidas, and kappa).

- Phones dataset: This dataset was collected based on keywords related to phones, phone brands, computers, and related staff. The following words are used to find the appropriate tweets of this type (mobile, laptop, iPhone, phone, tablet, Samsung, android, ios, hp, Lenovo, apple device, camera, iPad)
- Watches dataset: This data set was collected based on keywords related to watches, accessories, and accessories, as well as international brands of men's and women's watches. The following words were used to find suitable tweets of this type (hand watch, watch accessory, brand, hand accessories, ROLEX, TISSOT, RADO watch, OMEGA watches, CASIO watch, Fossil watch, concord watch)
- Sports dataset: This dataset was collected based on keywords related to sports and their related items. Abbreviations and names of international football leagues and some famous names were used. And also the name of real sports apps. The following words were used to find suitable tweets of this type (fpl, football, soccer, primer league, league 1, sport, la league, Bundesliga, Serie A, Messi, Cristiano Ronaldo)
- Learning dataset: This data set was collected based on words related to training, e-courses, tutorials, tests, and study programs. Tweets have been collected that can be used in some fields of science and knowledge. The following words were used to find suitable tweets of this type (course, learning, university, e-learn, skill learning, class, knowledge, study, certificate, book, science)

All this data was collected within a certain time range. Table 4.1 contains some information about the data when it was collected for the first time before processing it, and the tweets after clearing and removing the non-English tweets.

Table 4.1. Collected tweets.

Dataset	Date limit	Collected tweets no.	Word avg.	non-English tweets no.	Ready to use tweets no.
Cloths	2022/06/01 to 2022/06/10	7003	19.1023	1541	5462
Phones		7013	22.4484	3187	3826
Learning		7005	33.9370	99	6906
Sports	2022/01/01 to 2022/06/01	1777	39.8728	642	1135
Watches		1629	32.4812	16	1613

For each of the collected datasets the preparation was processed, stop wards removed, and recorded as a new dataset to be used in the model. The following Figures 4.3 – 4.8 shows the common words in each dataset before preparation.

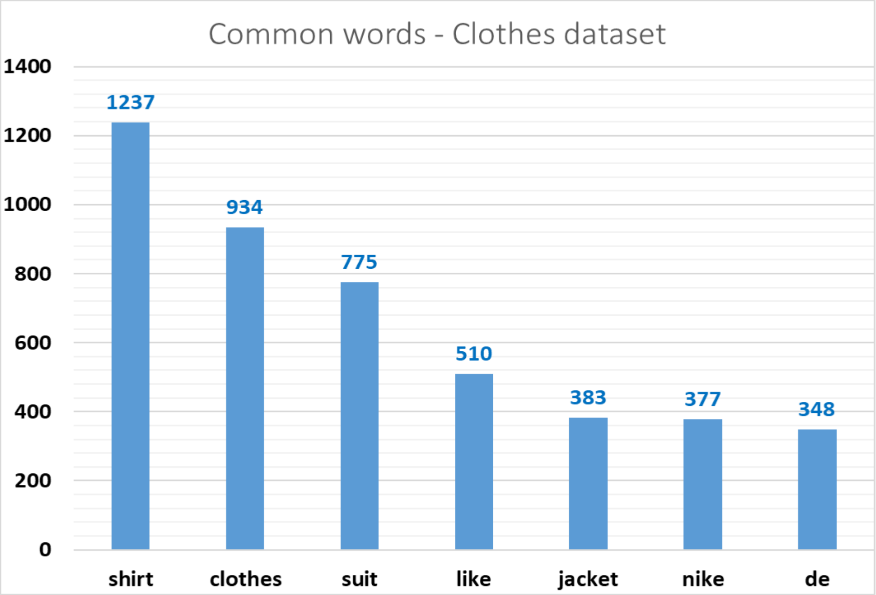


Figure 4.3. Common words in the Cloths dataset.

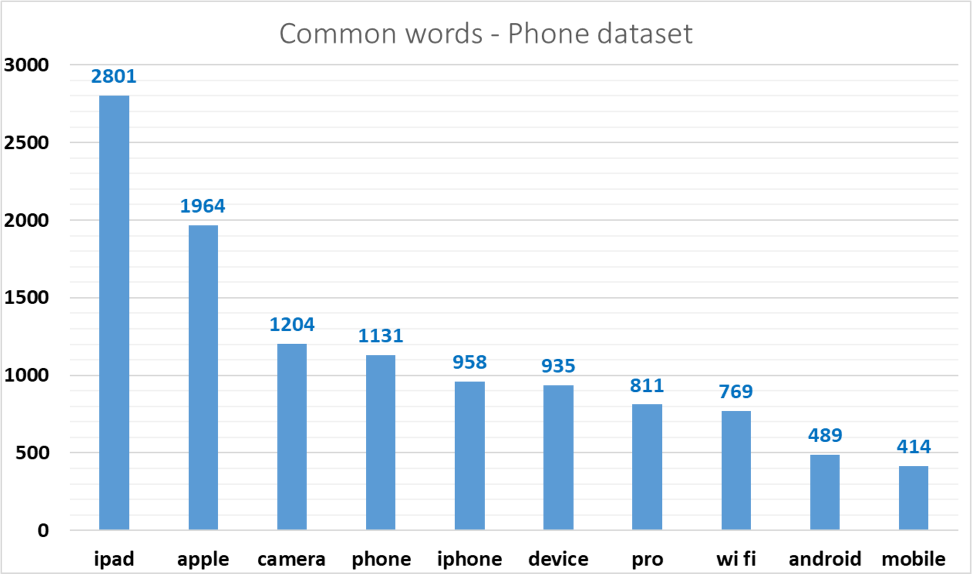


Figure 4.4. Common words in the Phone dataset.

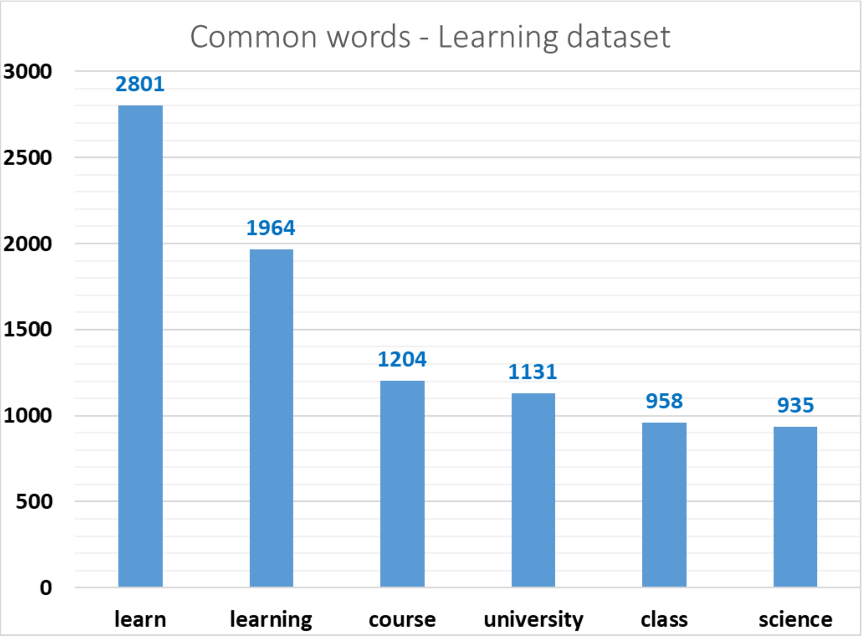


Figure 4.5. Common words in the Learning dataset.

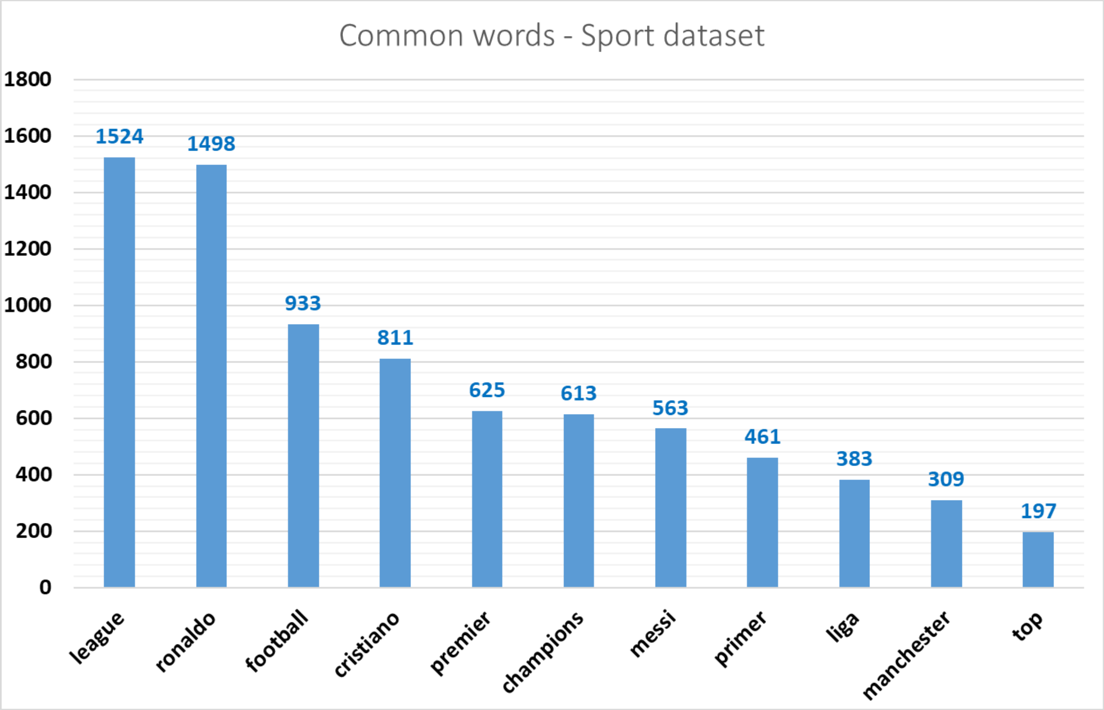


Figure 4.6. Common words in the Sport dataset.

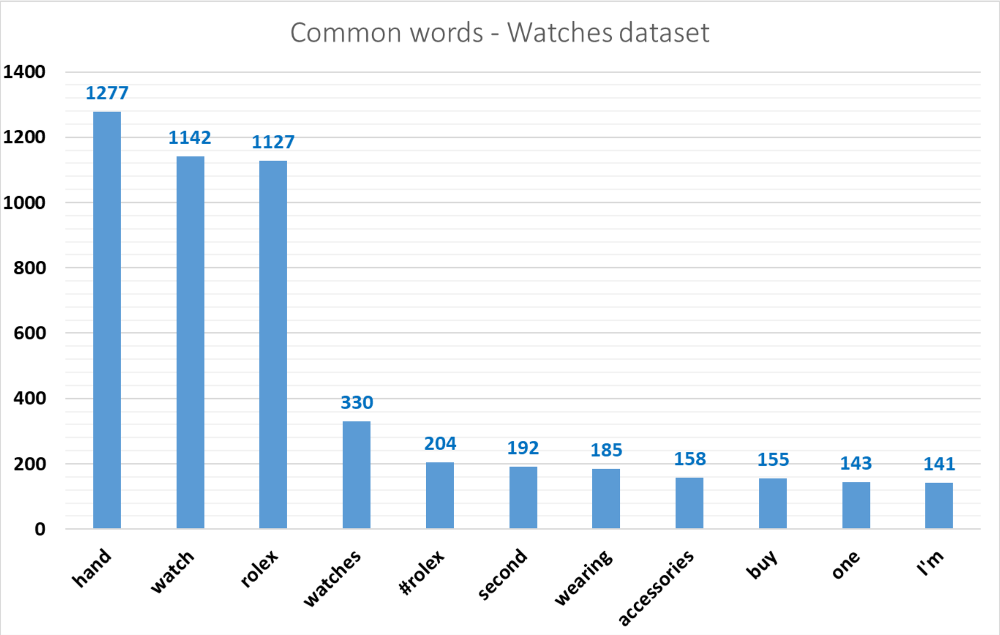


Figure 4.7. Common words in the Watches dataset.

4.2. DATA PREPARATION (NLP)

Natural language processing (NLP) is a sub-field within the branches of linguistics and artificial intelligence. The function of NLP is to provide a mechanism for the interaction between a computer and human language. Natural language processors are used for analyzing large amounts of data to understand, categorize, or make recommendations based on their contents. Natural language processing algorithms help simplify the handling of text documents.

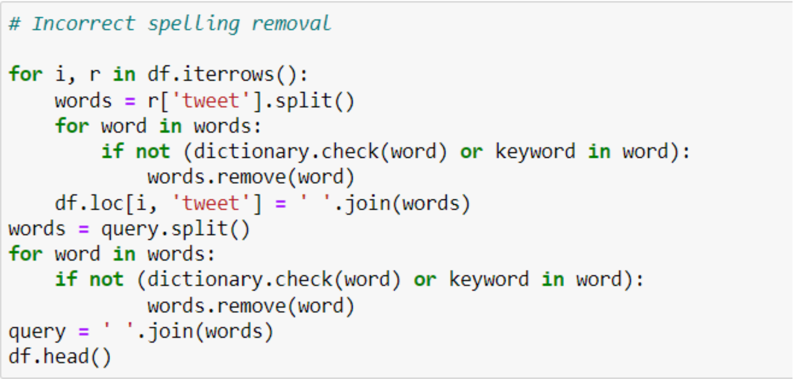
Natural language processors are used for analyzing the tweets in the dataset to understand the language and extract meanings from it. Systems that use text data have a basic need for NLP to be able to make human language readable [44]. Given its importance, it is used in machine translation, identifying the context of the text, classifying it and its characteristics, as well as in editing and simplifying texts, and it is also used to extract meanings and topics from the context of texts [45]. In each of these tasks, NLP programming follows a set of steps that include initializing and simplifying the text, and many algorithms and tools are used for this task. It is necessary to remove unnecessary words from texts, remove symbols and links, and also return words to their original form.

These steps are the preprocessing that precedes the main processing in the system. Before starting the text configuration steps in the dataset, a prior step is made, which is to remove non-English tweets. When tweets are collected, they come with a language column containing the user's profile language name. This column is used to separate English tweets so that the pre-processes are applied to them. In our system, we will follow these steps to prepare Tweets for processing.

4.2.1. Incorrect Spelling Removal

It was previously mentioned that the NLP function is to configure the texts in the dataset to be understood and processed by the system. The data used in our system is a collection of tweets shared by some users on the Twitter platform. On Twitter, writing any tweet or sharing an opinion does not require mastering the rules of the language, as every user is free to write. Sometimes words and texts with incorrect meaning or spelling may occur. Misspelling removal is used to ensure that there are no words that negatively affect the understanding of the meaning of the text [46].

Enchant is a python module which used for checking the word’s features. The checking includes spelling or if the word is in the English dictionary or not [47]. This model gives antonyms and synonyms for words, it also suggests the way to correct the incorrect words. After importing enchant model we are using the code as in Figure 4.8. In this part, tweets are fetched and divided into the words they contain. Each of these words is compared to the words in the English dictionary. If that word is not available in the dictionary, we will remove those words from the tweet.



```
# Incorrect spelling removal

for i, r in df.iterrows():
    words = r['tweet'].split()
    for word in words:
        if not (dictionary.check(word) or keyword in word):
            words.remove(word)
    df.loc[i, 'tweet'] = ' '.join(words)
words = query.split()
for word in words:
    if not (dictionary.check(word) or keyword in word):
            words.remove(word)
query = ' '.join(words)
df.head()
```

Figure 4.8. Incorrect words removal.

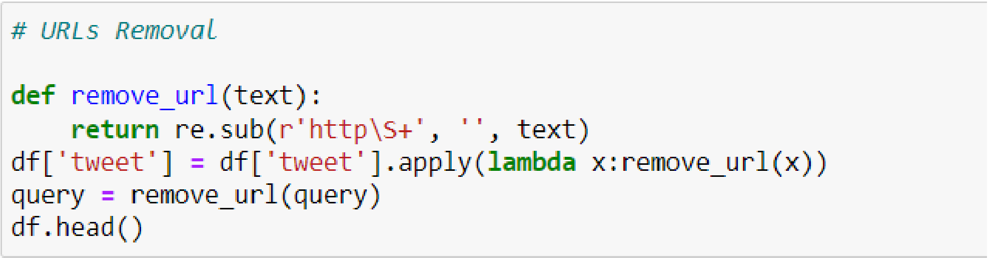
In some cases, the tweets may contain words that are not in the English dictionary and the process requires not removing them; here the function of adding those words to the dictionary can be used within the Enchant module [48]. In this case, words are compared within that user's modified dictionary.

4.2.2. URLs Removal

Tweets are often shared to express an opinion on a particular event or topic. Tweets may include referring to a page or website by including links and URLs. Preprocesses for systems that include NLP remove these links from texts and tweets [49]. A URL, sometimes called a web address, is a special line containing a web resource that specifies the location on a computer network to access that website [50].

With the spread of the Internet all over the world, it is natural that the language of Internet users is different. These URLs consist of a variety of languages and alphabets. In NLP and after removing the punctuation, the words of the URLs remain to be merged into the text or the tweet, so they are removed to not change the meanings of the tweets.

URL removal is done using the Re library. Its presence in the tweet does not provide any additional information other than its reference to a website. After importing re python model, URLs are removed using the following function, as in Figure 4.9. This function is applied to all tweets in the dataset and is considered one of the most important steps in preparing tweets for use in the system



```
# URLs Removal

def remove_url(text):
    return re.sub(r'http\S+', '', text)
df['tweet'] = df['tweet'].apply(lambda x:remove_url(x))
query = remove_url(query)
df.head()
```

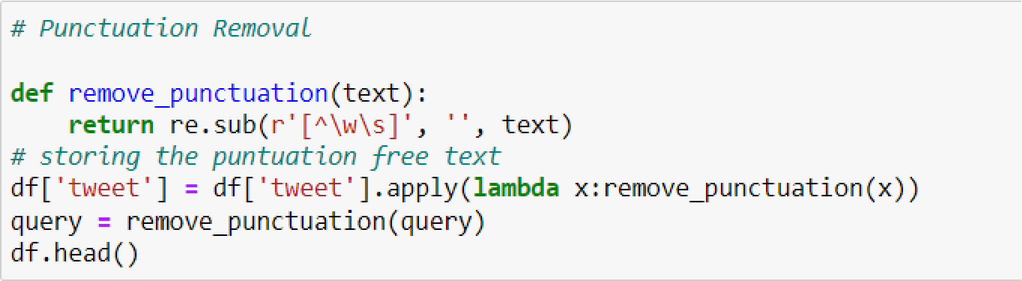
Figure 4.9. URL removal.

4.2.3. Punctuation Removal

In normal texts, the presence of punctuation marks makes sentences more accurate and clearer [51]. For NLP processing they should be removed. This removal is one of the most important words processing techniques. Removing punctuation marks helps treat texts more evenly and renders contraction words meaningless [52,53].

Sometimes it requires removing marks and leaving other marks depending on the type of system. The numbering string in Python consists of the following symbols!

"# $% & \ '() * +, -. /:;? @ [\\] ^_{|} ~`. The symbols to be removed must be specified in our function. We are removing punctuations as in Figure 4.10.



```
# Punctuation Removal

def remove_punctuation(text):
    return re.sub(r'[^\w\s]', '', text)
# storing the puntuation free text
df['tweet'] = df['tweet'].apply(lambda x:remove_punctuation(x))
query = remove_punctuation(query)
df.head()
```

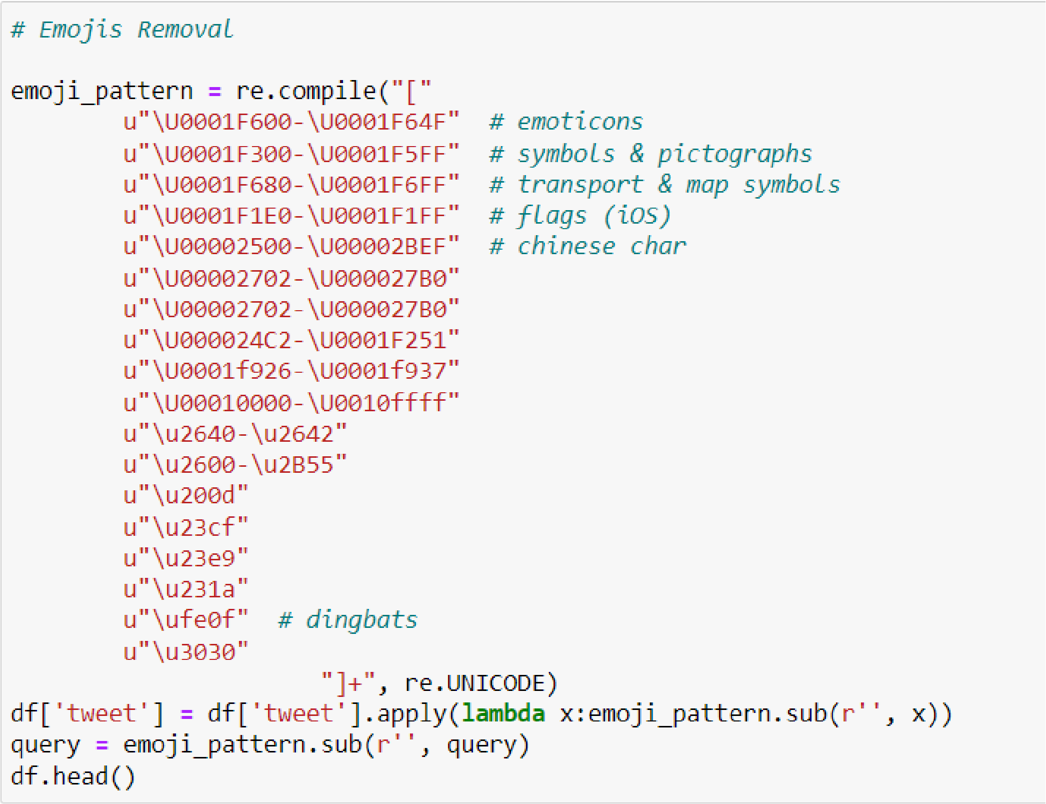
Figure 4.10. Punctuation removal.

In the English language, there are 14 punctuation marks: dot, question mark, exclamation point, comma, colon, semicolon, dash, hyphen, parentheses, parentheses, apostrophe, quotation mark, and ellipsis.

4.2.4. Emojis Removal

Emojis are shapes, graphics, and symbols that are included in texts and tweets. These symbols have clues to add more meaning to sentences and tweets. These emojis can indicate feelings to fill the missing emotion in the tweets. The term emoji refers to images that can be represented using encrypted characters [54]. After 2010, many phones added emojis to their operating systems, which helped spread its use widely throughout the world [55]. For this reason, the possibility of emojis in tweets is very large because there are many forms and connotations for all situations, and feelings, and also using them instead of some words.

Emojis are removed as a preprocessing step. Many emojis are meaningless and do not provide useful information in the Tweet. Programmatically, the Emojis are removed by a piece of code designed for this issue see figure 4.11.



```
# Emojis Removal

emoji_pattern = re.compile("["
        u"\U0001F600-\U0001F64F"  # emoticons
        u"\U0001F300-\U0001F5FF"  # symbols & pictographs
        u"\U0001F680-\U0001F6FF"  # transport & map symbols
        u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
        u"\U00002500-\U00002BEF"  # chinese char
        u"\U00002702-\U000027B0"
        u"\U00002702-\U000027B0"
        u"\U000024C2-\U0001F251"
        u"\U0001f926-\U0001f937"
        u"\U00010000-\U0010ffff"
        u"\u2640-\u2642"
        u"\u2600-\u2B55"
        u"\u200d"
        u"\u23cf"
        u"\u23e9"
        u"\u231a"
        u"\ufe0f"  # dingbats
        u"\u3030"
                      "]+", re.UNICODE)
df['tweet'] = df['tweet'].apply(lambda x:emoji_pattern.sub(r'', x))
query = emoji_pattern.sub(r'', query)
df.head()
```

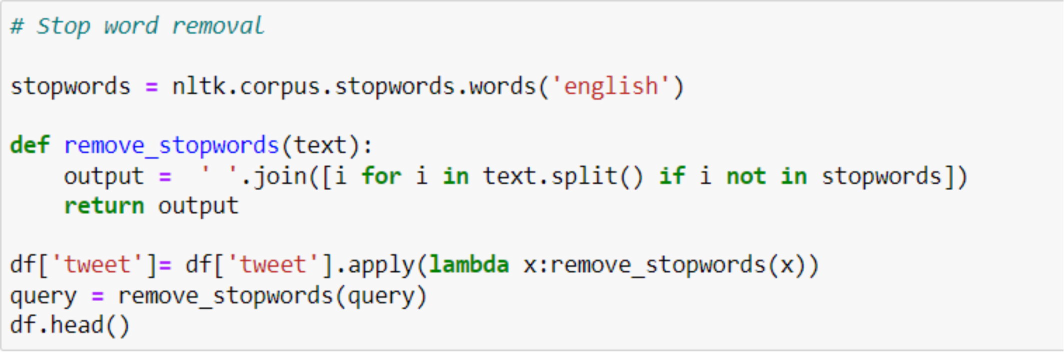
Figure 4.11. Emojis removal.

It is worth to mention, the user respond by typing Emojis more than words, so the mechanism for converting those emojis into words should be defined to preserve the sentence structure and meaning [56] and this format is comprehensive for evaluating some services using Emojis. Normally, it is deleted from the tweets since it has no effect.

4.2.5. Stop Word Removal

The term Stop-words means all words frequently repeated in the texts or sentences of any language, not just English; and do not affect the concept of the sentence. Stop words have an impact on many systems that contain word processing. Removing commonly used words in tweets makes the system focus on important words in the context of the tweet.

The process of removing stop words is removing the low-level information from the tweets to get focus on the important information [57]. After importing nltk, we use the function as in Figure 4.12 on all tweets to remove frequently repeated words.



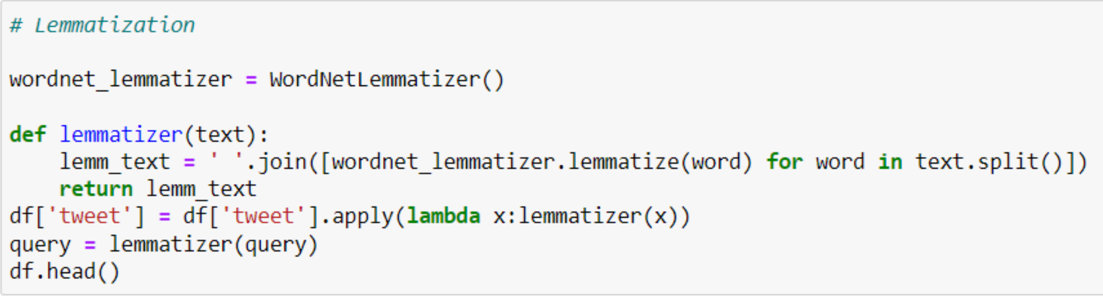
```
# Stop word removal

stopwords = nltk.corpus.stopwords.words('english')

def remove_stopwords(text):
    output =  ' '.join([i for i in text.split() if i not in stopwords])
    return output

df['tweet']= df['tweet'].apply(lambda x:remove_stopwords(x))
query = remove_stopwords(query)
df.head()
```

Figure 4.12. Stop words removal.

In general, most stop words are functional words, often words that do not have a meaning on their own but help in the formation of the sentence. Many of the words contained in the tweet, such as adjectives, nouns, and verbs are not counted as stop words.

4.2.6. Lemmatization

Lemmatization is a text normalization method used in the NLP process. Essentially, lemmatization is a method that converts any word to the base root mode by returning multiple forms of words to the basic form with the same meaning. It is often used for indexing, information retrieval, and labeling. The functions of Lemmatization focus on the morphological analysis of the used vocabulary, removing the inflectional endings and returning the word dictionary to the correct form [58]. The purpose of using lemmatization in NLP applications is to treat words with different inflections and analyze them as a single element. The function used in our system is shown in Figure 4.13.



```
# Lemmatization

wordnet_lemmatizer = WordNetLemmatizer()

def lemmatizer(text):
    lemm_text = ' '.join([wordnet_lemmatizer.lemmatize(word) for word in text.split()])
    return lemm_text
df['tweet'] = df['tweet'].apply(lambda x:lemmatizer(x))
query = lemmatizer(query)
df.head()
```

Figure 4.13. Lemmatization.

Lemmatization is considered the most important part of natural language understanding (NLU) and natural language processing (NLP). The processes in artificial intelligence are compared to Stemming, the stemming cuts words without knowing their context in tweets to achieve derivation, while Lemmatization finds out the origin of the word in the context of the tweet before making the derivation [59]. Lemmatization is distinguished by accuracy because it gets the roots of words from the dictionary and this helps the systems recognize the tweets after completing preprocessing steps.

4.3. TF-IDF

The term TF-IDF refers to an algorithmic measurement method that can be applied to texts. This algorithm measures the relevance of words to the text.

In our system, it is applied to all tweets as the first step of completing the recommendation. It is usually used in systems that contain NLP since the basis of its work is comparing the amount of increasing the number of times a word appears in a certain text to another text [37]. The tweets that contain some words such as (is, I, it) which are repeated in most other tweets come in a low rank, because those words are not influential and are not special. In case of such a word appeared in a tweet and not appeared in the others, this means the word is related to the tweet and it has an impact on understanding the meaning of the speech.

TF-IDF has two values, either the number of repetitions of the word in the text or a single tweet is the TF value, while the IDF value got by calculating the logarithm of the number of tweets in the dataset divided by the tweets containing the keyword [38]. To build RS based on text data, TF-IDF can be used with one of the similar calculating parameters to provide the elements of the recommendation [39]. According to [60] the equations of TF-IDF could be as (4.1) and (4.2) shown below:

$$TF(t) = \frac{Number\ of\ times\ word\ t\ appear\ in\ a\ document}{Total\ number\ of\ words\ in\ the\ document} \tag{4.1}$$

$$IDF(t) = log_e \frac{Total\ number\ of\ documents}{Number\ of\ documents\ contains\ word\ t} \tag{4.2}$$

According to these equations, the value will be near zero if the searched word is widely appearing in several tweets otherwise it will be close to 1. The word is related to the tweet if the combination of the two values is higher. In the model and after importing the feature_extraction model of the sklearn library, we perform the TF-IDF algorithm on the tweets. The tweets are compared with the vendor’s choice to find related tweets and recommend the users.

4.4. BAG OF WORDS

The bag-of-words (BoW) model is streamlining the representation of text which is used in information retrieval and NLP.

The BoW represents the text as a bag containing all the words of the text, the bag ignores the grammar and the word order in the text but it keeps the repetitions number of repeated words [61]. This model is considered a vector space model. A vector model can be defined as an algebraic method that is used with text and other data to represent a vector. Converting word combinations in texts and tweets helps to process them in the system. The vector model is used in data science to filter and retrieve information and systems that deal with text.

BoW’s approach is flexible, simple, and easy. It can be used in many ways for extracting text features. It is the description of a word’s appearance in the text. It is involving the vocabulary and the measure of known words. When we are using BoW for the similarity measurement, the approach assumes that the texts are similar when they have similar content. Further, just from the content of the text, we can learn something about its meaning [62].

The vectors generated by BoW contain a large number of empty data with large dimensions that create the generation of discontinuous vectors. To apply BoW on tweets, removing stop words from them should be implemented first with preprocessing steps as mentioned earlier. Stop words are insignificant so they must be removed and ensure they are not in the word bag, which takes more time to process. Tokenization to all sentences is also applied to each tweet. Tokenization makes it easy to separate and manipulate words in tweets. In our method, the text feature_extraction was imported from the sklearn library for a BoW application using CountVectorizer.

4.5. WORD2VEC

Word2vec is a natural language processing method that uses a neural network to learn the association of words from large texts. It is used to find synonyms of words, suggest words for partial sentences, or discover features of words in sentences. This model converts words into a set of numbers called vectors. Words numbers were chosen with great care to represent the semantic similarity of the word they represent [63]. Word2vec consists a set of nested functions that produce word vectors.

These overlapping functions can be represented as a two-layer neural network which trained to prepare word vectors and build their linguistic context. The input of these functions in Word2vec is a set of texts or tweets and results in vector spaces. Although hundreds of vectors are generated Word2vec allocates one for each word within the text document. If there are common words in the context of the speech in the tweet then the vectors of those words are close to each other in the vector space [64].

The presence of different model variables may greatly affect the quality of the word2vec model, so the number of vector dimensions was increased. The result of improving the word2vec model is followed by an increase in the complexity of the mathematical calculations and an increase in the processing time [65, 66].

Word2Vec is an unsupervised method. Word2Vec internally uses a supervised classification model to obtain these embeddings from the corpus as it can provide a corpus without any label information and the model can produce packed word embeddings [67]. We used the (gensim) library for implementing the word2vec method on the tweets.

4.6. COSINE SIMILARITY

The cosine similarity is an algorithm for calculating the similarity between vectors. The algorithm detects the similarity of the generated tweet text vectors the similarity measured in the internal product space. The similarity is measured by determining whether the vectors point in the same direction or not [40]. If the angle between two vectors was 90 degrees the cosine similarity will be zero (i.e. the vectors are perpendicular to each other).

This algorithm is to measure the cosine angle between two vectors. The measure of the cosine is a judgment of orientation, not magnitude, concerning the origin [69]. The cosine value for an angle with zero degrees equals 1, which means similarity, while the cosine value of an angle with 90 degrees equals zero, which means a difference.

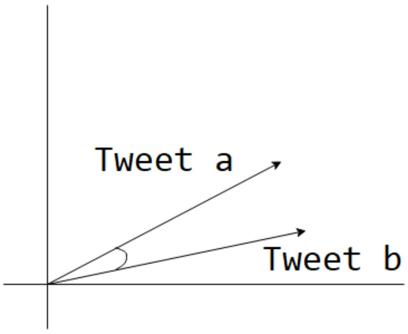


Figure 4.14. Tweets as vector.

If the tweets are converted to vectors as shown in Figure 4.14 the similarity calculated by the fallowing equation:

$$Similarity(a, b) = Cos\,\theta = \frac{a\rightarrow . b\rightarrow}{||a\rightarrow||.||b\rightarrow||} \tag{4.3}$$

Where:

a→ is the vector (a)
b→ is the vector (b)
θ is the angle between them.

4.7. EUCLIDEAN SIMILARITY

The Euclidean similarity is an algorithm for calculating the similarity and difference between vectors. This method can be considered the basis for many similarity calculation algorithms. It can be widely used to identify trends in optimization applications or smart text and number systems. If there are two vectors A and B then the distance between them can be calculated by taking the square root of the sum of the square of the difference between the vectors [70].

4.8. CORRELATION SIMILARITY

Correlation similarity is finding the identical between two different vectors or entities. The way this algorithm works is similar to calculating the cosine similarity between the two vectors, but it also calculates the correlation between the random variables involved in the distribution. In other words, the value of the cosine similarity between the x and y centered versions is again constrained between -1 and 1 to compute the similarity by the correlation similarity method [71].

4.9. VRS-CB DEVELOPMENT

Today, in any electronic sale that takes place on any website a lot of artificial intelligence tools are used, whether for the customer or the vendor. The most famous sites have recently implemented many mechanisms to attract those who want to buy by suggesting items and recommending services. To provide a tool to help vendors we worked on a problem of finding people to target in the sales process. The development of VRS-CB passed through several stages. A software development model was used to construct the project implementation steps. Initially, we designed the idea of the project and collected datasets (texts and tweets) from Twitter. Some of the vendor's keywords are set to finding similar tweets and then recommend people to target using the content-based recommendation mechanism. The first model was built as shown in the diagram below Figure 4.15.

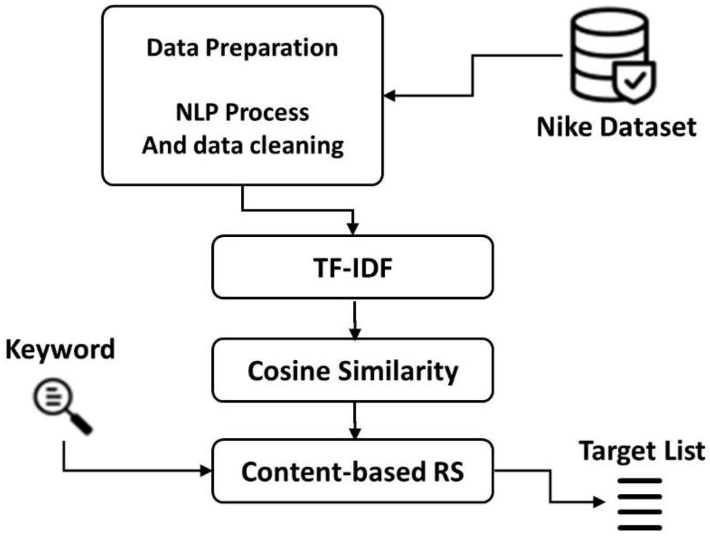


Figure 4.15. VRS-CB first designed model.

The first model was built using Nike's JustDoIt tweets [41]. After fetching the dataset then organizing and extracting processes implement only for the required columns which were used as the system dataset. In the next step, we put the tweets into NLP preprocessing to remove incorrect words, stop words, some external links, and emojis. In the step of the recommendation system based on content filtering, TF-IDF and cosine similarity methods were used. Preliminary results were obtained from the system and the system worked well in the first step of its construction. The system's performance is evaluated with the accuracy of the recommendations [68]. The accuracy of the recommendation was 86.24%. The accuracy of the system was good as a recommender based on several tests that included topics related to dataset vocabulary.

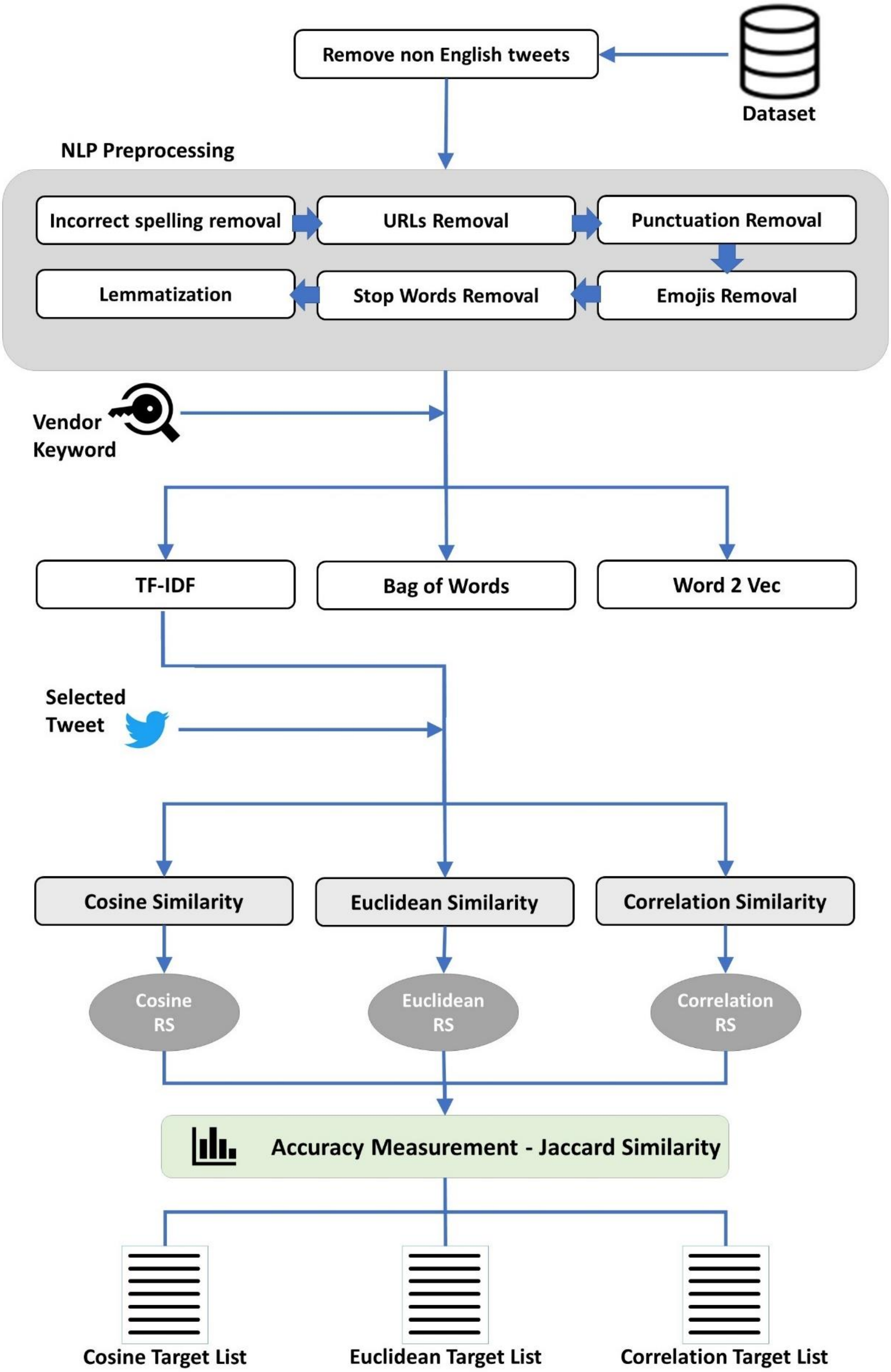
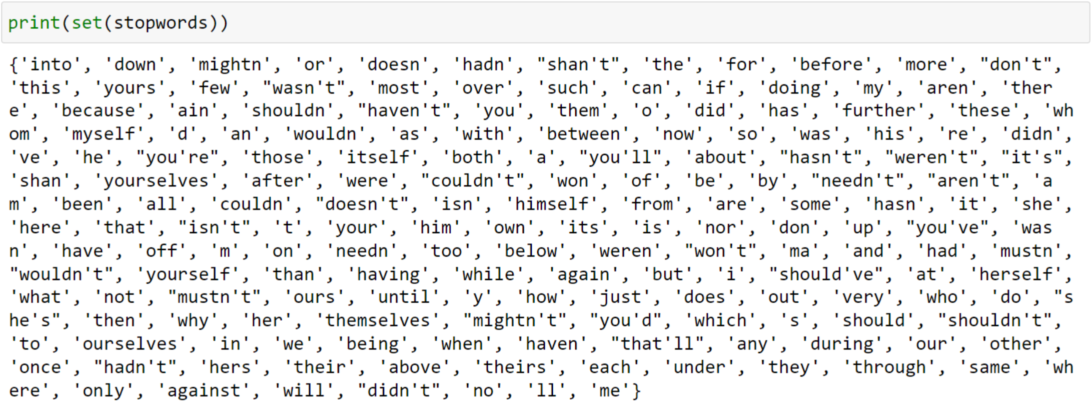


Figure 4.16. VRS-CB Improved architecture.

4.9.1. VRS-CB Structure

The first stage is illustrated in Figure 4.15 which is the basic of RS. To achieve a more comprehensive principle of the system we used the (Twint) tool to collect our data and tweets from Twitter. Five datasets were collected to evaluate the system in five different areas where sale or service provision could occur. These areas were clothing, watches and accessories, phones and computers, education, and sports. The system architecture has been improved; the diagram in Figure 4.16 represents the improved architecture. At first, the step of arranging the dataset was added and the non-English words were removed from the datasets. The preprocessing steps start with deleting the incorrectly written words and phrases from the tweets in the incorrect spelling removal step. After that, all tweets will be free of incomprehensible or misspelled words. Then, the links are removed from tweets. As it was previously indicated that most of the tweets contain a link to access Internet addresses, these are removed in this step. In the third step, the punctuation marks are removed from the tweets and added to all those symbols that affect the results of the comparison and recommendation. Then, in the fourth step, the symbols, emojis, and shapes that are included in the tweets are removed. Tweets should be removed from those emojis by writing the code for emoji groups in the removal function. The last two steps of NLP preprocessing are stop-words removal and lemmatization. Tweets usually contain many words that are repeated frequently in context. These words do not affect the meaning of the text in the tweet and are called stop words. The step of removing stop words from tweets is one of the most important preprocessing steps for text initialization in natural text processing systems. We review the removed words in Figure 4.17.



```
print(set(stopwords))
```

```
{'into', 'down', 'mightn', 'or', 'doesn', 'hadn', "shan't", 'the', 'for', 'before', 'more', "don't",
'this', 'yours', 'few', "wasn't", 'most', 'over', 'such', 'can', 'if', 'doing', 'my', 'aren', 'ther
e', 'because', 'ain', 'shouldn', "haven't", 'you', 'them', 'o', 'did', 'has', 'further', 'these', 'wh
om', 'myself', 'd', 'an', 'wouldn', 'as', 'with', 'between', 'now', 'so', 'was', 'his', 're', 'didn',
've', 'he', "you're", 'those', 'itself', 'both', 'a', "you'll", 'about', "hasn't", "weren't", "it's",
'shan', 'yourselves', 'after', 'were', "couldn't", 'won', 'of', 'be', 'by', "needn't", "aren't", 'a
m', 'been', 'all', 'couldn', "doesn't", 'isn', 'himself', 'from', 'are', 'some', 'hasn', 'it', 'she',
'here', 'that', "isn't", 't', 'your', 'him', 'own', 'its', 'is', 'nor', 'don', 'up', "you've", 'was
n', 'have', 'off', 'm', 'on', 'needn', 'too', 'below', 'weren', "won't", 'ma', 'and', 'had', 'mustn',
"wouldn't", 'yourself', 'than', 'having', 'while', 'again', 'but', 'i', "should've", 'at', 'herself',
'what', 'not', "mustn't", 'ours', 'until', 'y', 'how', 'just', 'does', 'out', 'very', 'who', 'do', "s
he's", 'then', 'why', 'her', 'themselves', "mightn't", "you'd", 'which', 's', 'should', "shouldn't",
'to', 'ourselves', 'in', 'we', 'being', 'when', 'haven', "that'll", 'any', 'during', 'our', 'other',
'once', "hadn't", 'hers', 'their', 'above', 'theirs', 'each', 'under', 'they', 'through', 'same', 'wh
ere', 'only', 'against', 'will', "didn't", 'no', 'll', 'me'}
```

Figure 4.17. Removed stop words.

lemmatization is the last step of NLP in our model. In this step, a final review of the tweets' words is made to prepare them for processing. Lemmatization refers to getting tasks done correctly. In NLP, its function is to handle the morphological analysis of the vocabulary of a tweet. In this step, the inflection and the increase in words are removed. After that, each word returns to the base form as in the dictionary. This step is important, especially before the procedures of finding similarities and recommendations. As in Figure 4.16, the tweets here are prepared for a recommendation.

4.9.2. Recommendation Models

To build an advanced model and to include most of the models that are used with the content-based recommendation the TF-IDF, a bag of words, and word2vec vectorizers are proposed. Here, the generated tweets are entered into three forms. The VRS-CB is built to make recommendations based on three vectorizers. In this stage, the keyword is requested from the user. The associated tweets with the keywords and the user has to choose the appropriate Tweet ID. The system recommends displaying a list of people to target in the sales process by tweets content-based filtering. Each TF-IDF, bag of words, and word2vec models need a similarity assumer algorithm for completing the recommendation process.

They also need to prepare recommended target list according to the similarity accuracy management metric. In the similarity algorithms aspect, we used three algorithms the Cosine similarity, Euclidean similarity, and Correlation similarity algorithms with each of the three models. For evaluating system issues, we used the Jaccard index algorithm. Figure 4.16 shows the TF-IDF vectorizer method based on three similarity measurements.

For the recommendation model each one of cosine, Euclidean, and correlation measurements were used for checking the similarity among the tweets according to the selected one. The best similar tweets with the username of each are listed for the vendor as a recommendation target list.

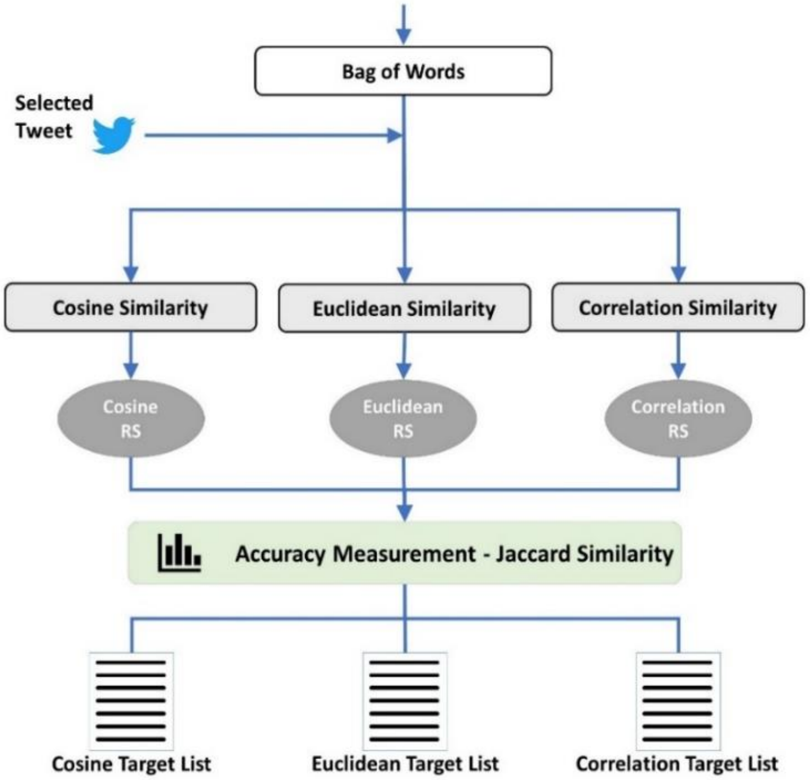


Figure 4.18. Bag of words recommender model.

The accuracy measurement by the Jaccard index algorithm will measure the recommendation accuracy for that case. Besides each target user in the recommendation list, a percentage of the similarity which came from the algorithm is also listed. The top 10 users will be recommended. If the similarity percentage of the top target users was less than 50% the user will dismiss even if the number of users in the list is less than 10. In the recommendation model, each algorithm will prepare a target list of users according to the mathematical steps and calculations of the method to find the similarity among the tweets. The method for the bag of words and word2vec models is the same to the TF-IDF model in terms of structure.

Figures 4.18 and 4.19 illustrated the structure of the bag of words and word2vec recommendation models. They are working on the same procedure with different methods. In each case, the vendor will be sure that he is getting the best recommendation for the list of targetable users.

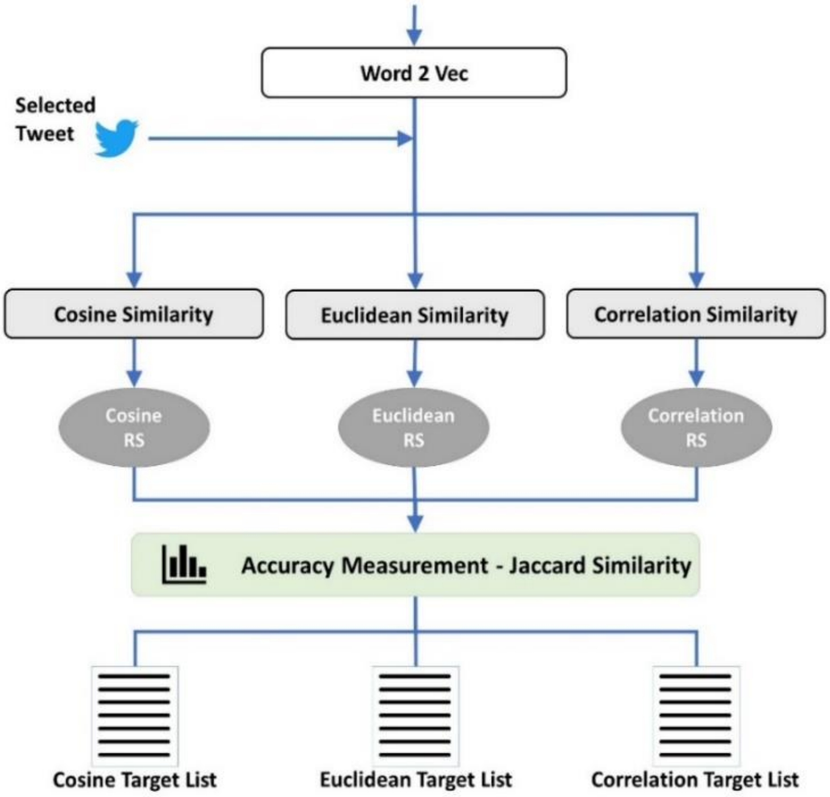


Figure 4.19. Word2vec recommender model.

4.9.3. System Input

After completing the system architecture, it is easy to use it with any data set containing tweets with usernames. The system input defines as the keywords of the vendor. In the experimental stages of the system, the keywords that were used to collect the dataset were inserted into the systems as inputs, but there are no limits to the use of all types of keywords. The use of keywords related to the type of data or the time in which it was collected.

The common use topics in the tweets lead to more results because users have a common idea. System and comparison are applied to find the most suitable people to recommend. The accuracy of the system depends on its use, each time the accuracy is calculated against the base values.

4.9.4. Proposed Output

In general, the output is always a list of people that sellers can target. The list is based on the similarity of tweets. The similarity is checked according to the keyword. The list changes with different keyword cases. The system consists of three RS each of them depending on more than one algorithm. The similarity of users to recommend them is measured and calculated in more than one way. The method of calculating similarity and recommendation differs from one algorithm to another. Next to each list of recommended people is printed the percentage of similarity. The system is designed to suit the uses of all types of service providers. Multiple results can be obtained by changing the keyword or the primary tweet.

4.9.5. Evaluation Metrics

To evaluate the system and determine its accuracy we used the Jaccard index for checking the similarity and kappa statistic for ensuring the true vectorization.

4.9.5.1 Jaccard Similarity

Jaccard similarity is defined as a statistical parameter or algorithm for measuring similarity and diversity between different types of data. It can be used with numeric, text, and relative data. To measure similarity, a comparison is made between tweets, and the similarity is measured within the range from zero to 100 [72]. A higher percentage means more similarities in tweets. The equation used to calculate the similarity of the two text variables A and B is as follows

$$J(\mathrm{A},\mathrm{B}) = \frac{|A \cap B|}{|A \cup B|} \qquad (4.4)$$

It deals with the values of text variables and compares the words of the tweets to find the percentage of similarity. All feature data for these two variables are combined into a single numeric value. The similarity in the features of the variables means the similarity between them [73].

In each process of recommending and showing results within the system, the similarity is tested and the recommendation system is evaluated using this technique.

4.9.5.2 Kappa Statistic

It is one of the most important statistical measures that are used to indicate the validity and reliability of values. Using of Cohen's kappa is very important since it represents the likely range of values to be true. Its presence, along with tools that deal with numbers and vectors, allows the RS to ensure that the used values in the recommendation are accurate and therefore the results are accurate. We are using it to indicate the validity and accuracy of the vectors of tweets before checking the similarity.

4.9.6. System Features

VRS-CB provides a helpful tool for the vendors. We analyzed the data in the first stage and use content-based filtering to suggest a list of people. The most important feature of VRS-CB is based on data analysis. The process of analyzing data before using it increases the possibility of obtaining good results due to the removal of non-influential words. It works to find the targets according to the keyword.

The vendors can use the required keywords and can get more than one result by changing the words with similar keywords and with different meanings. Documented results can be obtained by considering the use of evaluation techniques along with the people-suggestion process and the avoidance of irrelevant people recommendations. The use of the application is not limited to a specific time, but can be used repeatedly.

PART 5

RESULTS

In this section, we view the experiment results of the VRS-CB application based on the three vectorizers (TF-IDF, bag of words, and words to vector), each one is used with three similarity measurements; Cosine Similarity, Euclidean Similarity, and Correlation Similarity (i.e. one different measurement in each time). We are testing the recommendation on five data sets. In each dataset, we are using two keywords as shown in Table 5.1. The system was tested on different cases while the accuracies were recorded for each case using the Jaccard index metric. To achieve more realistic results, we used more than one random ID in each case.

We are using symbols and letters instead of the long terms in Table 5.1, the similarity measurements are named (a, b, and c) for (Cosine Similarity, Euclidean Similarity, and Correlation Similarity) respectively. The term (rand) means random, (Rec) means recommendation, and (Rec. users) means recommended users

Table 5.1. Experiments results.

Case	Dataset	Keyword	Rand. tweet ID for Rec.	No. of Rec. users with TF-IDF	No. of Rec. users with BoW	No. of Rec. users with Word2vec
1	Clothes	Clothes	1414	a=4 b=4 c=4	a=5 b=10 c=5	a=10 b=10 c=10
2		Nike	3882	a=2 b=2 c=2	a=2 b=10 c=2	a=5 b=10 c=5
3	Sport	football	49	a=1 b=1 c=1	a=6 b=0 c=3	a=7 b=1 c=3
4		football	1655	a=3 b=4 c=4	a=6 b=0 c=5	a=7 b=0 c=2
5	Watches	rolex	721	a=2 b=2 c=2	a=10 b=0 c=10	a=10 b=8 c=10
6		accessories	1227	a=5 b=2 c=4	a=7 b=7 c=4	a=10 b=10 c=10
7	learn	Class	229	a=2 b=2 c=2	a=2 b=1 c=2	a=10 b=1 c=1
8		university	6918	a=5 b=5 c=5	a=10 b=6 c=9	a=10 b=5 c=6
9	phone	iphone/ipad	581	a=1 b=1 c=1	a=4 b=9 c=4	a=3 b=0 c=3
10		iphone	3444	a=2 b=2 c=2	a=6 b=5 c=4	a=2 b=1 c=2

5.1. EXPERIMENTS OF CLOTHES DATASET

The clothes dataset was tested using two experiment cases, case 1 and case 2. The keywords of the two cases were "clothes" and "nike". Although the keywords (clothes, nike) belong to the same category (Clothes) there are differences in their results based on their frequency in the dataset.

5.1.1. TF-IDF Results on Clothes Dataset

While testing case 1, the similarity results for recommendation were in the range 52.45~90.61 for this keyword. All three similarity measurements obtained the same number of recommended users with very close accuracies as in Table 5.2, while the Jaccard accuracy achieved 80.4%.

Table 5.2. Case 1 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	2147	Nycthemic	90.58
	2562	Nysportsmike	75.23
	2720	Mollieewalkerr	61.95
	2667	Stapeathletic	52.47
Euclidean similarity	2147	Nycthemic	90.57
	2562	Nysportsmike	75.21
	2720	Mollieewalkerr	61.93
	2667	Stapeathletic	52.45
Correlation similarity	2147	Nycthemic	90.61
	2562	Nysportsmike	75.34
	2720	Mollieewalkerr	62.1
	2667	Stapeathletic	52.63

In case 2 the recommendation results achieved 57.4 as the lower bound and upper bound 82.27 as in Table 5.3. The accuracy of this case according to Jaccard is 82.05%

Table 5.3. Case 2 experiment results using TF-IDF.

algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	4086	zonkedin_mia	81.89
	6450	bru_raw	57.4
Euclidean similarity	4086	zonkedin_mia	81.87
	6450	bru_raw	57.38
Correlation similarity	4086	zonkedin_mia	82.27
	6450	bru_raw	58.45

5.1.2. Bag of Words Results on Clothes Dataset

In case 1 here, the best number of recommended users was achieved by Euclidean Similarity with mean accuracy of 70.76%, while the Jaccard accuracy achieve 80.4%. The results of both cosine similarity and correlation similarity are similar unlike Euclidean similarity as in Table 5.4. This is the result of measuring the distances between vector elements by Euclidean

Table 5.4. Case 1 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	2147	Nycthemic	86.6
	2562	Nysportsmike	81.65
	2720	Mollieewalkerr	66.67
	2667	Stapeathletic	57.74
	2518	Aaronjsuch	51.64
Euclidean similarity	2562	Nysportsmike	80.38
	2147	Nycthemic	80.38
	2720	Mollieewalkerr	73.12
	5633	Geovanky	67.68
	640	imb3llaaa	67.68
	4758	nerdchristina2	67.68
	591	ed_stevens125	67.68
	5261	m3sskutz	67.68
	3470	Leafkazoo	67.68
	3106	Thenewapplejuic	67.68
Correlation similarity	2147	Nycthemic	86.66
	2562	Nysportsmike	81.74
	2720	Mollieewalkerr	66.81
	2667	Stapeathletic	57.88
	2518	Aaronjsuch	51.81

In case 2 as shown in Table 5.5, the recommendation of Euclidean similarity was achieved for 10 tweets with 64.0% accuracy on average, which is very close to the

TF-IDF vectorizer's average accuracy result. According to the Jaccard index, the recommendation accuracy was 66.67%. The difference between TF-IDF and BoW is based on the way tweets are represented by each vectorizer. While TF-IDF generates the vectors of tweets, the bag of words represents the tweet as word frequency basis. The similarity is calculated once by measuring the angle between the vectors and once by the frequency of words. This difference in results is a feature of VRS-CB

Table 5.5. Case 2 experoment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	4086	zonkedin_mia	81.65
	6450	bru_raw	57.74
Euclidean similarity	4086	zonkedin_mia	76.85
	6450	bru_raw	68.27
	6420	Suavekp	61.85
	2374	masonx1	61.85
	2322	Snkrfrkrmag	61.85
	1824	cesar_loaks	61.85
	4135	torribaby30	61.85
	1506	_mrsolodolo_	61.85
	5335	marqel_	61.85
	5370	michael84628780	61.85
Correlation similarity	4086	zonkedin_mia	82.04
	6450	bru_raw	58.82

5.1.3. Word2vec Results on Clothes Dataset

The results of word2vec in case 1 are obtained the optimal number of recommendations with very high accuracy for each tweet. The best similarity was achieved by the Correlation similarity algorithm with mean accuracy of 98.05% as shown in Table 5.6, while the Jaccard index accuracy was 91.32%

Table 5.6. Case 1 experimant results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	2147	Nycthemic	91.21
	2562	nysportsmike	89.39
	2720	mollieewalkerr	88.62
	1881	phonybigcharles	88.14
	6753	tony_be	86.44
	3301	_jdmodel_	85.05
	241	x_alizeee	85.05
	2065	Iamrashae	85.05
	3185	freshinsight3	85.05
	5230	1dlarrielove	85.05
Euclidean similarity	2147	Nycthemic	87.82
	2562	nysportsmike	86.12
	2720	mollieewalkerr	85.38
	1881	phonybigcharles	84.9
	6753	tony_be	83.1
	241	x_alizeee	82.44
	3301	_jdmodel_	82.44
	5230	1dlarrielove	82.44
	3185	freshinsight3	82.44
	2065	Iamrashae	82.44
Correlation similarity	2147	Nycthemic	99.44
	2562	nysportsmike	99.02
	2720	mollieewalkerr	98.79
	1881	phonybigcharles	98.64
	6753	tony_be	97.99
	3233	starfallenjaxx	97.34
	5230	1dlarrielove	97.34
	241	x_alizeee	97.34
	3185	freshinsight3	97.34
	3301	_jdmodel_	97.34

In case 2, the accuracy according to Jaccard was 79.6%. As shown in Table 5.7 by comparing the experimental results with the previous results we find the Word2vec with the Euclidean similarity extremely appropriate to clothes dataset

Table 5.7. Case 2 experimant results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	6450	bru_raw	69.21
	4135	torribaby30	67.85
	4086	zonkedin_mia	67.69
	1506	_mrsolodolo_	64.58
	2494	leeuh_mucis	51.12
Euclidean similarity	6450	bru_raw	91.5
	1506	_mrsolodolo_	91.05
	4135	torribaby30	90.5
	4086	zonkedin_mia	89.83
	2322	Snkrfrkrmag	89.48
	5370	michael84628780	89.12
	6837	freeman13711792	88.77
	1988	miok9661	87.71
	5720	Thedailyretina	85.37
	1633	_temi_tope	84.72
Correlation similarity	6450	bru_raw	82.13
	4135	torribaby30	80.19
	4086	zonkedin_mia	79.86
	1506	_mrsolodolo_	75.25
	2494	leeuh_mucis	51.56

5.2. EXPERIMENTS OF SPORT DATASET

The sport dataset is tested using the same keyword "football" twice, each time the test is made with a different tweet ID. The experiments with the sport dataset are included in cases 3 and 4. The results are evident that the clothes category achieved higher results than the sport dataset. This depends on the selected tweet and the number of its words, as well as depends on the number of tweets that the dataset contains regarding that topic.

5.2.1. TF-IDF Results on Sport Dataset

The best result of case 3 as shown in Table 5.8 is with the cosine similarity which obtained 80.61%. The accuracy by using the Jaccard index was 73.68%.

Table 5.8. Case3 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	13	Mohanourali	80.61
Euclidean similarity	13	Mohanourali	75.97
Correlation similarity	13	Mohanourali	76.53

The results of case 4 are shown in Table 5.9. The results of similarities ranged between 50.66~82.68 while the accuracy was 73.68%.

Table 5.9. Case4 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	148	mikekin73778218	82.68
	540	pessi_hgh_abusa	82.68
	1277	haikalharis96	82.68
	120	Bananasportsfc	63.89
Euclidean similarity	148	mikekin73778218	79.23
	540	pessi_hgh_abusa	79.23
	1277	haikalharis96	79.23
	120	Bananasportsfc	56.73
Correlation similarity	148	mikekin73778218	79.32
	540	pessi_hgh_abusa	79.32
	1277	haikalharis96	79.32
	120	Bananasportsfc	50.66

5.2.2. Bag of Words Results on Sport Dataset

The results of both cases 3 and 4 are shown in Table 5.10 and Table 5.11 goes with the cosine similarity algorithm for the two experiments which obtained 73.0% for case 3 and 79.5% for case 4. The accuracy according to the Jaccard index recorded 73.68% for Case 3 and 77.78% for Case 4 followed by the Correlation similarity algorithm while the Euclidean similarity algorithm did not obtain any results so it has been neglected.

Table 5.10. Case 3 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	13	Mohanourali	88.39
	148	mikekin73778218	75
	540	pessi_hgh_abusa	75
	1655	ilab1612	70.71
	1277	haikalharis96	67.08
	120	Bananasportsfc	61.24
Correlation similarity	13	Mohanourali	80.24
	148	mikekin73778218	59.09
	540	pessi_hgh_abusa	59.09

Table 5.11. Case 4 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	148	mikekin73778218	88.39
	540	pessi_hgh_abusa	88.39
	1277	haikalharis96	88.39
	120	Bananasportsfc	79.06
	49	mukam73844499	70.71
	13	Mohanourali	62.5
Correlation similarity	148	mikekin73778218	82.29
	540	pessi_hgh_abusa	82.29
	1277	haikalharis96	82.29
	120	Bananasportsfc	62.84
	49	mukam73844499	54.86

5.2.3. Word2vec Results on Sport Dataset

In word2vec vectorizer experiments, the results of case 3 and case 4 are shown in Table 5.12 and Table 5.13. The best similarity goes with cosine similarity for the two experiments which obtained (7) recommendations with mean accuracy of 66.2% for Case 3 and 72.9% for Case 4. The accuracy by using the Jaccard index was 70.0% for both Cases. The Euclidean similarity did not obtain any results in the second experiment so it has been also neglected.

Table 5.12. Case 3 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	13	Mohanourali	81.87
	1277	haikalharis96	68.81
	1655	ilab1612	67.61
	148	mikekin73778218	64.24
	540	pessi_hgh_abusa	63.95
	120	bananasportsfc	60.83
	510	17b___g	56.09
Euclidean similarity	13	Mohanourali	59.45
Correlation similarity	13	Mohanourali	88.66
	1277	haikalharis96	53.14
	1655	ilab1612	52.77

Table 5.13. Case 4 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	1277	haikalharis96	79.6
	13	Mohanourali	78.91
	148	mikekin73778218	73.36
	120	bananasportsfc	72.84
	540	pessi_hgh_abusa	70.78
	49	mukam73844499	68.08
	510	17b___g	66.9
Correlation similarity	1277	haikalharis96	70.36
	13	Mohanourali	67.5

5.3. EXPERIMENTS OF WATCHES DATASET

The watches dataset is tested in case 5 with the keyword "rolex", and case 6 with the keyword "accessories".

5.3.1. TF-IDF Results on Watches Dataset

In case 5 the accuracy was 73.68%, in the same time the recommendation result shows very low prediction values for a “rolex” keyword as in Table 5.14. VRS-CB capabilities allow vendors to use it multiple times with different tweets. The small percentage of similarity is based on the content of the tweet. Although this keyword is repeated a lot in the dataset, this does not prevent the presence of a few tweets similar to the ones that were selected.

Table 5.14. Case 5 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	639	Mondaniweb	66.92
	97	goyal_sanchit	61.23
Euclidean similarity	639	Mondaniweb	63.54
	97	goyal_sanchit	57.28
Correlation similarity	639	Mondaniweb	63.4
	97	goyal_sanchit	59.84

Unlike case 5, the case 6 similarities accuracy was very high like 99.5% for the recommendation results as Table 5.15. The keyword "accessories" is not a high common keyword in the dataset thus it was selected to demonstrate similarity calculation differs.

Table 5.15. Case 6 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	562	vinseo608	79.41
	259	Vamprvnge	52.36
	738	Taseiyu	51.07
	578	iitsskayy_	51.07
	761	samgee_gamwise	50.53
Euclidean similarity	562	vinseo608	78.64
	259	Vamprvnge	50.6
Correlation similarity	562	vinseo608	79.79
	259	Vamprvnge	54.13
	578	iitsskayy_	54.03
	738	Taseiyu	54.03

5.3.2. Bag of Words Results on Watches Dataset

Case 5 with BoW vectorizer recorded Jaccard accuracy equal to 52.63% which is less than TF-IDF. The recommendation result as shown in Table 5.16 shows acceptable prediction values for Cosine similarity and Correlation similarity while the Euclidean similarity algorithm has been neglected.

Table 5.16. Case 5 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	796	criswilson02	82.5
	961	xh487	81.41
	639	Mondaniweb	77.61
	1024	Jewellerssarum	76.7
	1580	nkedaudiologist	75.93
	1578	Davidjhodges	75.93
	909	365romandays	72.76
	512	mealesy82	68.8
	422	oracle_time	67.27
	452	theconnor_welch	67.27
Correlation similarity	796	criswilson02	76.86
	961	xh487	74.79
	639	Mondaniweb	69.8
	1024	Jewellerssarum	68.17
	1580	nkedaudiologist	67.47
	1578	Davidjhodges	67.47
	909	365romandays	63.66
	512	mealesy82	56.85
	422	oracle_time	56.47
	452	theconnor_welch	54.67

Case 6 results accuracy according to Jaccard was 99.5% which is the same as TF-IDF accuracy with this case. It is worth mentioning that the (accessories) is a distinguished keyword because in the tweet script it is always followed with the types of the accessories such as (swatch, necklace, etc.) therefore it achieved high

Jaccard accuracy, while the recommendation results for each algorithm were low range values as in Table 5.17.

Table 5.17. Case 6 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	562	vinseo608	67.08
	761	samgee_gamwise	60.3
	738	Taseiyu	57.74
	578	iitsskayy_	57.74
	119	_miintee	50
	259	Vamprvnge	50
	723	iamnot_mpho	50
Euclidean similarity	578	iitsskayy_	57.78
	562	vinseo608	57.78
	738	Taseiyu	57.78
	119	_miintee	51.88
	259	Vamprvnge	51.88
	723	iamnot_mpho	51.88
	505	blindp01	51.88
Correlation similarity	562	vinseo608	61.34
	738	Taseiyu	52.9
	578	iitsskayy_	52.9
	761	samgee_gamwise	51.6

5.3.3. Word2vec Results on Watches Dataset

The case 5 recommendation result shows good prediction values for Cosine similarity and Correlation similarity while the Euclidean similarity got the lowest results shown in Table 5.18. Despite the good results, The Jaccard accuracy recorded 40.0%. This ratio means that the algorithms were able to find similarities for 40% of the tweets that were similar according to the Jaccard index.

Table 5.18. Case 5 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	495	betyouwantmenow	79.05
	1412	nancyoakley1	76.03
	1367	sashatsigan	75.18
	1580	nkedaudiologist	74.79
	97	goyal_sanchit	74.64
	796	criswilson02	73.68
	369	ledoo_duma	73.5
	786	hattonjewels	72.24
	62	Paulcerro	70.8
	909	365romandays	68.52
Euclidean similarity	495	betyouwantmenow	64.23
	1412	nancyoakley1	60.89
	1580	nkedaudiologist	58.98
	796	criswilson02	57.22
	369	ledoo_duma	56.91
	786	hattonjewels	55.64
	1367	Sashatsigan	54.48
	62	Paulcerro	51.89
Correlation similarity	495	Betyouwantmenow	92.76
	1412	nancyoakley1	89.76
	1367	Sashatsigan	88.76
	1580	Nkedaudiologist	88.22
	97	goyal_sanchit	88.21
	796	criswilson02	86.9
	369	ledoo_duma	86.62
	786	Hattonjewels	85.15
	62	Paulcerro	82.95
	909	365romandays	79.56

In case 6 the Jaccard accuracy got 77.43% which is less than the previous method, while the recommendation results for Cosine and Correlation algorithms achieved high accuracies and the optimal number of tweets shown in Table 5.19

Table 5.19. Case 6 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	578	iitsskayy_	94.35
	761	samgee_gamwise	94.23
	738	Taseiyu	88.05
	830	sarrasb0	84.6
	354	Tsumenkin	84.35
	119	_miintee	82.4
	723	iamnot_mpho	82.36
	259	Vamprvnge	81.56
	110	Kuchiistarah	81.09
	505	blindp01	80.93
Euclidean similarity	578	iitsskayy_	87.03
	761	samgee_gamwise	86.18
	354	Tsumenkin	72.56
	738	Taseiyu	69.51
	110	Kuchiistarah	65.24
	505	blindp01	64.85
	259	Vamprvnge	64.15
	675	bini_ph	60.91
	830	sarrasb0	57.13
	305	legendofsebong	50.56
Correlation similarity	578	iitsskayy_	99.28
	761	samgee_gamwise	99.24
	738	Taseiyu	93.2
	830	sarrasb0	85.86
	354	Tsumenkin	85.52
	119	_miintee	79.82
	723	iamnot_mpho	79.69
	259	Vamprvnge	76.65
	110	Kuchiistarah	74.93
	505	blindp01	74.5

5.4. EXPERIMENTS OF LEARN DATASET

The testing of this dataset is made by the keywords "class" and "university". these keywords were selected as the common words of the dataset to obtain case 7 and case 8.

5.4.1. TF-IDF Results on Learn Dataset

In case 7 the results in all similarities were close as in Table 5.20. The accuracy according to Jaccard was 90.45%.

Table 5.20. Case7 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	5999	Emmswsd	66.92
	4108	Falmouthprimary	52.87
Euclidean similarity	5999	Emmswsd	66.9
	4108	Falmouthprimary	52.85
Correlation similarity	5999	Emmswsd	70.1
	4108	Falmouthprimary	55.91

The Case 8 results extract very high recommendations for each similarity measurement. The upper bound was 100% and the lower bound was 69.25, the results shown in Table 5.21. In this case, the accuracy according to the Jaccard index was 99.5%.

Table 5.21. Case8 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100
	5874	kaala_naaag	100
	3119	corridor_24	94.87
	627	antiproton_com	75.37
Euclidean similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100
	5874	kaala_naaag	100
	3119	corridor_24	93.59
	627	antiproton_com	69.25
Correlation similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100
	5874	kaala_naaag	100
	3119	corridor_24	95.09
	627	antiproton_com	74.85

5.4.2. Bag of Words Results on Learn Dataset

Case 7 results as shown in Table 5.22 shows low similarity, the recommendations obtained close results unless Euclidean similarity which achieved less accuracy, on the other hand, the accuracy according to the Jaccard was high at 89.55%.

Table 5.22. Case 7 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	5999	Emmswsd	70.71
	4108	falmouthprimary	50
Euclidean similarity	5999	Emmswsd	59.98
Correlation similarity	5999	Emmswsd	73.58
	4108	falmouthprimary	53.23

The case 8 experimentation results are listed in Table 5.23. We observed significantly improved results for all similarities compared with the previous experiment. In this case, the accuracy according to Jaccard was 95.24%

Table 5.23. Case 8 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100
	5874	kaala_naaag	100
	3119	corridor_24	94.87
	627	antiproton_com	94.28
	4017	creed_l	81.65
	5984	Margoasnipe	81.65
	3135	coach17w	75
	1317	Sacredheartuniv	70.71
	6843	Brinleyhineman	57.74
Euclidean similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100
	5874	kaala_naaag	100
	3119	corridor_24	61.14
	4017	creed_l	61.14
	5984	Margoasnipe	61.14
Correlation similarity	876	Pumpkinpaichi	100
	5393	deepeacemaker9	100

	5874	kaala_naaag	100
	3119	corridor_24	92.79
	627	antiproton_com	92.02
	4017	creed_l	72.73
	5984	margoasnipe	72.73
	3135	coach17w	62.93
	1317	sacredheartuniv	55.05

5.4.3. Word2vec Results on Learn Dataset

With Word2vec vectorizer, case 7 results are shown in Table 5.24. The recommendation results show low quality and quantity unless the cosine algorithm obtained 10 recommendations, also with poor accuracies. The accuracy according to Jaccard was 55.56%.

Table 5.24. Case 7 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	5999	Emmswsd	63.42
	778	Bizhighlight	54.6
	4335	undergroundyxe	54.58
	6612	Jsmmcmanes	54.18
	6063	digitalequityct	54.08
	4989	Bethersdensch	53.08
	194	Antiobroni	52.12
	4908	tx_troublemaker	52.07
	4907	envir490	52.06
	2910	Gentrywmd	51.35
Euclidean similarity	5999	Emmswsd	53.76
Correlation similarity	5999	Emmswsd	66.81

Case 8 with word2vec obtained an accuracy according to the Jaccard index equal to 91.74%. This cases results are shown in Table 5.25.

Table 5.25. Case 8 experiment results using Word2vec.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	5393	deepeacemaker9	89.59
	4017	creed_l	85.67
	876	Pumpkinpaichi	83.81
	5984	Margoasnipe	77.81
	5874	kaala_naaag	77.8
	3135	coach17w	76.55
	627	antiproton_com	74.87
	3119	corridor_24	72.49
	6843	Brinleyhineman	67.69
	1317	Sacredheartuniv	67.58
Euclidean similarity	5393	deepeacemaker9	78.99
	4017	creed_l	71.77
	876	Pumpkinpaichi	67.7
	5984	Margoasnipe	50.88
	5874	kaala_naaag	50.84
Correlation similarity	5393	deepeacemaker9	95.6
	4017	creed_l	88.68
	876	Pumpkinpaichi	84.51
	5984	Margoasnipe	64.05
	5874	kaala_naaag	63.83
	3135	coach17w	57.45

5.5. EXPERIMENTS OF PHONE DATASET

The testing process included different keyword types. Case 9 experimented with the combined keyword "iphone/ipad" while case 10 tested with the keyword "iphone" separately.

5.5.1. TF-IDF Results on Phone Dataset

In Case 9, the recommendation results were in the range 73.1~75.05 as in Table 5.26. while the accuracy according to the Jaccard index records 95.24%.

Table 5.26. Case 9 experiment results using TF-IDF.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	3584	Shoppingbooms	73.12
Euclidean similarity	3584	Shoppingbooms	73.1
Correlation similarity	3584	Shoppingbooms	75.05

In case 10, we got two recommendations for each similarity. The recommendation results were low as shown in Table 5.27 on the other hand the accuracy achieved 90.45%.

Table 5.27. Case 10 testing results using TF-IDF.

algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	4278	Eadraffan	65.81
	1362	Alexsavkovic	56.41
Euclidean similarity	4278	Eadraffan	61.46
	1362	Alexsavkovic	50.87
Correlation similarity	4278	Eadraffan	68.12
	1362	Alexsavkovic	60.42

5.5.2. Bag of Words Results on Learn Dataset

The recommendation results accuracies were closed for all similarities in case 9 as Table 5.28. The accuracy according to Jaccard records 95.24%.

Table 5.28. Case 9 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	3584	shoppingbooms	69.63
	401	Samagamec	57.74
	2570	Ozsavas	57.74
	3957	mohanav91	57.74
Euclidean similarity	401	samagamec	64.06
	2570	Ozsavas	64.06
	3957	mohanav91	64.06
	4715	ryanlounsbury	56.79
	2780	Asegar	56.79

	3268	videomasterapp	56.79
	1884	mooregare	56.79
	685	Mxriarose	56.79
	900	simonapperley	50.75
Correlation similarity	3584	shoppingbooms	71.71
	401	samagamec	62.26
	3957	mohanav91	62.26
	2570	Ozsavas	62.26

The results of case 10 according to Table 5.29 shows the number of recommendations between 4~6 and the accuracies was acceptable for all similarities. The accuracy of Jaccard index achieved 94.74% for this experiment

Table 5.29. Case 10 experiment results using Bag of word.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	1362	Alexsavkovic	81.65
	1031	navimarketplace	70.71
	192	Maorisara	66.67
	4278	Eadraffan	66.67
	97	ag_theblade	57.74
	3999	_liamcass	57.74
Euclidean similarity	1362	Alexsavkovic	69.87
	192	Maorisara	58.71
	4278	Eadraffan	58.71
	97	ag_theblade	50.35
	1031	navimarketplace	50.35
Correlation similarity	1362	Alexsavkovic	79.06
	1031	navimarketplace	65.28
	192	Maorisara	60
	4278	Eadraffan	60

5.5.3. Word2vec Results on Learn Dataset

Case 9 recommendation results accuracies were closed for all similarities in word2vec as the Table 5.30. The accuracy according to the Jaccard index was 80.0%. The contrast between the recommendation results values and the accuracy value is noticeable. This means that the algorithms were able to find similarities for a good proportion of the tweets that were similar according to Jaccard’s similarity

Table 5.30. Case 9 experiment results using Word2vec model.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	4477	Kansasacc	55.07
	4478	Missouriacc	55.07
	4476	acc_oklahoma	55.07
Correlation similarity	4476	acc_oklahoma	62.94
	4478	Missouriacc	62.94
	4477	Kansasacc	62.94

In case 10 results we got two recommendations for cosine and correlation similarities and one recommendation for Euclidean as in Table 5.31. The accuracy of the Jaccard index achieved 90.0%.

Table 5.31. Case 10 experiment results using Word2vec model.

Algorithm	Tweet id	Username	Similarity by Algorithm%
Cosine similarity	1362	Alexsavkovic	55.34
	4278	Eadraffan	53.7
Euclidean similarity	1362	Alexsavkovic	65.85
Correlation similarity	1362	Alexsavkovic	57.31
	4278	Eadraffan	54.62

5.6. ALL EXPERIMENTATION SUMMARY

After conducting the tests on the datasets, we can summarize the results obtained to be studied and evaluated individually. The results are classified according to the number of recommendations in each testing case and again according to the best similarity ratio for each case. As the number of recommendations for each case is shown in Table 5.1, we present using separated tables that includes the best similarity ratio for each case. In Table 5.32, the best similarities for all cases with TF-IDF are listed. The highest similarity is highlighted for configuring the best similarity in that case. Table 5.33 is the summary of BoW results, where Table 5.34 is the word2vec cases' best similarities.

Table 5.32. The best recommendation similarities with TF-IDF.

The case	Cosine Similarity	Euclidean Similarity	Correlation Similarity
1	90.58	90.57	**90.61**
2	81.89	81.87	**82.27**
3	**80.61**	75.97	76.53
4	**82.68**	79.23	79.32
5	**66.92**	63.54	63.4
6	79.41	78.64	**79.79**
7	66.92	66.9	**70.1**
8	100	100	100
9	73.12	73.1	**75.05**
10	65.81	61.46	**68.12**

Table 5.33. The best recommendation similarities with BoW.

The case	Cosine Similarity	Euclidean Similarity	Correlation Similarity
1	86.6	80.38	**86.66**
2	81.65	76.85	**82.04**
3	**88.39**		80.24
4	**88.39**		82.29
5	**82.5**		76.86
6	**67.08**	57.78	61.34
7	70.71	59.98	**73.58**
8	100	100	100
9	69.63	64.06	**71.71**
10	**81.65**	69.87	79.06

Table 5.34. The best recommendation similarities with Word2vec.

The case	Cosine Similarity	Euclidean Similarity	Correlation Similarity
1	91.21	87.82	**99.44**
2	69.21	**91.5**	82.13
3	81.87	59.45	**88.66**
4	**79.6**		70.36
5	79.05	64.23	**92.76**
6	94.35	87.03	**99.28**
7	63.42	53.76	**66.81**
8	89.59	78.99	**95.6**
9	55.07		**62.94**
10	55.34	**65.85**	57.31

Relying on all the experiments and results, we can say that for every vectorizer there is a similarity measurement that works best with it. This can be summarized in Table 5.35 according to the number of recommendations and the best similarity ratio.

Table 5.35. The best similarity measurement with each vectorizer.

According to a number of recommendations		According to best similarity accuracy	
With TF-IDF	All are close	With TF-IDF	Correlation Similarity
With BoW	Cosine Similarity	With BoW	Cosine Similarity
With Word2vec	Cosine Similarity	With Word2vec	Correlation Similarity

PART 6

DISCUSSIONS CONCLUSIONS AND FUTURE WORK

6.1. DISCUSSIONS

To discuss the tested results and indicate the importance and strengths of the system, we have to say that the system depends mainly on the dataset. The system contains these tools and algorithms that are similar in their work and differ in the way they perform. In our tests, keywords and tweets were chosen to match all possibilities or choices. We have a single or double keyword, and some keywords have also been used for more than one case. Also, we have selected keywords that are not very common in that dataset. These differences help to show the difference also in the results so that we can determine the best mechanism in each case.

Referring to the method of recommendation in TF-IDF, Bag of words, and Word2vec. The TF-IDF makes a recommendation based on the relevant importance of the keyword in those Tweets. Bag of words recommends by frequency of keywords in tweets so that the tweet vectors have the frequency of each word within that context. The main difference between a bag of words and TF-IDF is that the Bag of words does not generate Inverse Document Frequency (IDF) and recommends by frequency number (TF). Word2Vec is an algorithm that uses a non-deep neural network to produce word vectors. This algorithm differs from TF-IDF in that TF-IDF is a statistical scale that forms a vector of tweets, while word2vec produces a vector of tweet terms one by one and with some arithmetic, converts that set of vectors to a single vector. In addition, word2vec, unlike TF-IDF, takes into consideration the context of the word in the tweet.

With each recommendation system, the cosine similarity, euclidean similarity, and correlation similarity algorithms are used. The main function of each is to find similar tweets and thus complete the recommendation process. There are some differences in their features and the way they work, so each of them can be a strength of the system. In each search for a list of people to target, each algorithm prepares a list of people. The lists are sometimes the same, a little different, or a lot different at others.

Cosine similarity finds the similarity by the measure of the cosine of an angle between each vector. Euclidean Similarity measures the distance between vector elements to determine the similarity of two tweets. Between these two algorithms, the result of cosine similarity is better when the two tweets are separated by an euclidean distance. The small angle between two tweet vectors generates greater similarity using the cosine similarity. But in the case of a part of the contents of the tweet completely identical to the part of the other tweet, here euclidean similarity measures the optimal similarity.

Correlation similarity explains the strength and direction of an association between any variables. While the cosine similarity computes the similarity between two variables, whereas the correlation similarity computes the correlation between two jointly distributed random variables

Depending on the results, we can summarize the compatibility and quality of performance for each model and algorithm. TF-IDF represents tweets as vectors before sending them to similarity determination algorithms. According to the TF-IDF recommender summary figure, the performance of the cosine similarity algorithm is consistent in most cases for this model. The correlation similarity algorithm is somewhat similar to the cosine similarity in this model. The difference between them is only in two cases. We can say that tweets containing more words have a greater correlation in comparison. Therefore, correlation similarities results were more than cosine similarity unlike the case in tweets with a few words. Euclidian’s performance is similar in this model to other algorithms.

A bag of words converts tweets into a set of words, each with a repetition weight. This model reduces the knowledge of the association between words. According to the bag of words recommender summary Figure, we can find that the results of cosine similarity are superior in most cases compared to correlation similarity. Whereas in the cases of tweets with many words the recommendation to use correlation similarity may be overcome. In this model, the difference is usually found in the euclidean similarity. In this case, the measure of similarity between tweets is by measuring the frequency of words and the distance between their dimensions.

The presence of words with connotations in two tweets leads to their similarity according to the cosine similarity algorithm, despite the difference in significance. While the significance is the reason for the similarity according to euclidean similarity. Therefore, we find the difference between them according to the tweet that we are dealing with.

The performance of the word2vec model depends on converting tweets into vectors, but by representing each word in it with a vector and then merging them. In the case of tweets with a few words, the cosine similarity is superior to other algorithms. Only if there is affinity in word numbers does euclidean similarity take precedence. Correlation is somewhat similar to the cosine algorithm in the case of short and medium tweets. In the case of long tweets, the difference between the two algorithms can be seen due to the possibility of correlation between the word vectors. Word2vec models summary shown in Word2vec recommender summary figure. In addition, we also can perform the best performance for each similarity measurement as follows:

- Best performance for Cosine similarity is with BoW then with TF-IDF
- Best performance for Euclidean similarity is with BoW then with Word2vec
- Best performance for Correlation similarity is with Word2vec then with TF-IDF

6.2. CONCLUSIONS

In today's world, technology companies meticulously audit every step that users take. Every company has a massive set of data about their millions of users and they collect it to target ads and build things such as suggestion systems. This information can add to the enjoyment of online purchases or watching movies and videos on some sites but if consumers feel that the company knows too much about them, they are likely to be reluctant to continue using the service. Personalized content or services can be addictive or threatening depending on your point of view. It can help us find things we like, but it can also create unintended biases amongst users. Personalization helps us have a better user experience for customers, but it is a trade-off and we need to find a good point between privacy and personalization.

The main objective of building our system is to provide a distinctive tool based on several algorithms of RS. The system is built with this architecture to ensure that the results are valid based on several possibilities in each recommendation process. The use of the system depends mainly on the data used, keywords, and comparison tweets to find people for their recommendation. We applied the system to five data sets. In each of them, more than one keyword was used with all possibilities to get the best recommendation. By relying on the Jaccard Similarity algorithm to evaluate the process each time, the user can adopt the percentage of similarity to the accuracy of the system.

The accuracy ratio of the system can be compared with the similarity ratio obtained from the algorithm itself. After running ten test cases, the results showed that the system works on recommendations in an accurate, flexible, and malleable manner. The feature required in the systems is used by service providers, according to the different services they provide. The main requirement of any system is to be applicable and not specific, as is the case with our system.

Depending on the obtained results, the system is working on allocating a list of users and recommending them in several different cases. Each model or algorithm has its advantages and using them together gives the system the ability to make the recommendation despite tweets differing in length. Compared with the performance of the algorithms in the cases we have discussed, it turns out that in each case, we are going to have a high percentage of similarity in the results of the recommendation.

6.3. FUTURE WORK

In the future, a built-in system can be made using collaborative filtering recommendation techniques to consider some features of the recommended users. In this case, the RS will be converted from content-based filtering systems to hybrid systems. Within our system, it will also be possible in the future to integrate the outputs and make a special algorithm based on the features of the used algorithms. The ultimate goal is to use the system directly with the Twitter platform and feed the input with real-time tweets to obtain the latest user data.

REFERENCES

1. Antonakaki, Despoina, Paraskevi Fragopoulou, and Sotiris Ioannidis. “A Survey of Twitter Research: Data Model, Graph Structure, Sentiment Analysis and Attacks.” ***Expert Systems with Applications*** (2021).

2. Hosein J, H. Sim R. Saadat. “Recommendation Model for Large Databases.” ***International Journal of Information and Education Technology*** (2012).

3. Sailunaz, Kashfia, and Reda Alhajj. "Emotion and sentiment analysis from Twitter text." ***Journal of Computational Science 36*** (2019).

4. Dib, Brahim, Fahd Kalloubi, and Abdelhak BOULAALAM. "Semantic-based followee recommendations on Twitter network." ***Procedia Computer Science 127*** (2018).

5. Ben-Lhachemi, Nada. "Using tweets embeddings for hashtag recommendation in Twitter." ***Procedia Computer Science 127*** (2018)

6. Yazdanfar, Nazpar, and Alex Thomo. "Link recommender: Collaborative-filtering for recommending urls to twitter users." ***Procedia Computer Science 19*** (2013)

7. Louis Ngamassi, Hesam Shahriari, Thiagarajan Ramakrishnan, Shahedur Rahman "Text mining hurricane harvey tweet data: Lessons learned and policy recommendations." ***International Journal of Disaster Risk Reduction*** (2022).

8. Suman, Chanchal, Saichethan Miriyala Reddy, Sriparna Saha, and Pushpak Bhattacharyya. "Why pay more? A simple and efficient named entity recognition system for tweets." ***Expert Systems with Applications*** (2021).

9. [9] Hao Wu, Kun Yue, Yijian Pei, Bo Li, Yiji Zhao, Fan Dong “Collaborative Topic Regression with social trust ensemble for recommendation in social media systems” (2016).

10. Jooa, JinHyun, SangWon Bangb, and GeunDuk Parka. "Implementation of a recommendation system using association rules and collaborative filtering." ***Procedia Computer Science*** (2016).

11. R.J. Kuo, Ch. Chen, Sh. Keng, "Application of hybrid metaheuristic with perturbation-based K-nearest neighbors algorithm and densest imputation to collaborative filtering in recommender systems" (2021).

12. Son, Jieun, and Seoung Bum Kim. "Content-based filtering for recommendation systems using multiattribute networks." ***Expert Systems with Applications*** (2017).

13. U. Boryczka, M. Bałchanowski, "Using Differential Evolution in order to create a personalized list of recommended items," ***Procedia Computer Science*** (2020).

14. Hussein, Mohsin Hasan, and Rand Abdulwahid Albeer. "The Importance of Recommendation Systems in Electronic commerce." (2022).

15. Susan MaGee "Identifying Your Market, determine why a customer would want to buy your product/service." ***Edward Lowe Foundation - https://edwardlowe.org/how-to-identify-a-target-market-and-prepare-a-customer-profile/*** (2022).

16. Kangning Wei; Jinghua Huang; Shaohong Fu "A Survey of E-Commerce Recommender Systems" (2007).

17. Francesco Ricci and Lior Rokach and Bracha Shapira, "Introduction to Recommender Systems Handbook, Recommender Systems Handbook" (2011).

18. Nujoud Hashem Al Waleedi "Recommendation systems can predict the preferences of users and the products they are looking for on the Internet" ***tech-ye.com/recommender-systems*** (2021).

19. Wasim Ahmed "Using Twitter as a data source an overview of social media research tools" (2021).

20. Arnout B. Boot, Erik Tjong Kim Sang, Katinka Dijkstra, Rolf A. Zwaan " How character limit affects language usage in tweets"(2019)

21. Aml Mostafa, Walaa Gad, Tamer Abdelkader, Nagwa Badr "Pre-HLSA: Predicting Home Location for Twitter Users Based on Sentimental Analysis." (2022).

22. Beakcheol Jang, Inhwan Kim, Jong Wook Kim "Word2vec Convolutional Neural Networks for Classification of News Articles and Tweets." (2019)

23. Hayley Dorney "The dos and don'ts of hashtags" ***https://business.twitter.com/en/blog/the-dos-and-donts-of-hashtags.html*** (2021).

24. Brahim Diba, Fahd Kalloubia, El Habib Nfaouia "Semantic-Based Followee Recommendations on Twitter Network ." (2018).

25. Ben-Lhachemi, Nada. "Using tweets embeddings for hashtag recommendation in Twitter." ***Procedia Computer Science*** (2018).

26. Kumar, Nagendra, Eshwanth Baskaran, Anand Konjengbam, and Manish Singh. "Hashtag recommendation for short social media texts using word-embeddings and external knowledge." ***Knowledge and Information Systems*** (2021).

27. Dib, Brahim, Fahd Kalloubi, and Abdelhak BOULAALAM. "Leveraging topic feature for followee recommendation on Twitter network." ***In 2020 International Conference on Intelligent Systems and Computer Vision (ISCV) IEEE*** (2020).

28. Awan, Mazhar Javed, Rafia Asad Khan, Haitham Nobanee, Awais Yasin, Syed Muhammad Anwar, Usman Naseem, and Vishwa Pratap Singh. "A recommendation engine for predicting movie ratings using a big data approach." (2021).

29. Lops, Pasquale, Dietmar Jannach, Cataldo Musto, Toine Bogers, and Marijn Koolen. "Trends in content-based recommendation." ***User Modeling and User-Adapted Interaction*** (2019).

30. Huang, Zhenhua, Chang Yu, Juan Ni, Hai Liu, Chun Zeng, and Yong Tang. "An efficient hybrid recommendation model with deep neural networks." (2019).

31. Baptiste Rocca "Introduction to recommender systems, Overview of some major recommendation algorithms" ***towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada*** (2019).

32. Thorat, Poonam B., Rajeshwari M. Goudar, and Sunita Barve. "Survey on collaborative filtering, content-based filtering and hybrid recommendation system." ***International Journal of Computer Applications*** (2015).

33. Yang, Wan-Shiou, Hung-Chi Cheng, and Jia-Ben Dia. "A location-aware recommender system for mobile shopping environments." ***Expert Systems with Applications*** (2008).

34. Webster, Jonathan J., and Chunyu Kit. "Tokenization as the initial phase in NLP." ***COLING The 14th International Conference on Computational Linguistics*** (1992).

35. Eger, Steffen, Paul Youssef, and Iryna Gurevych. "Is it time to swish? Comparing deep learning activation functions across NLP tasks." (2019).

36. Singh, Ravinder, et al. "A framework for early detection of antisocial behavior on Twitter using natural language processing." ***Conference on Complex, Intelligent, and Software Intensive Systems*** (2019).

37. Chen, Chien-Hsing. "Improved TFIDF in big news retrieval: An empirical study." ***Pattern Recognition Letters*** (2017).

38. Huang, Cheng-Hui, Jian Yin, and Fang Hou. "A text similarity measurement combining word semantic information with TF-IDF method." ***Jisuanji Xuebao Chinese Journal of Computers*** (2011).

39. Tajbakhsh, Mir Saman, and Jamshid Bagherzadeh. "Microblogging hash tag recommendation system based on semantic TF-IDF: Twitter use case." ***In 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops*** (2016).

40. Fócil-Arias, Carolina, et al. "A tweets classifier based on cosine similarity." ***Working notes of CLEF 2017, Conference and Labs of the Evaluation*** Forum (2017).

41. **https://www.kaggle.com/datasets/eliasdabbas/5000-justdoit-tweets-dataset?select=justdoit_tweets_2018_09_07_2.csv**

42. Gunawardana, Asela, and Guy Shani. "A survey of accuracy evaluation metrics of recommendation tasks." ***Journal of Machine Learning Research*** (2009).

43. Himanshu Sharma "Complete Tutorial On Twint: Twitter Scraping Without Twitter's API" (2020).

44. Chowdhary "Natural language processing." ***Fundamentals of artificial intelligence*** (2020).

45. Kowsari, Kamran, et al. "Text classification algorithms: A survey." (2019).

46. Etaiwi, Wael, and Ghazi Naymat. "The impact of applying different preprocessing steps on review spam detection." ***Procedia computer science*** (2017).

47. Perkins, Jacob. Python "Text Processing with NLTK 2.0 Cookbook: LITE. Packt Publishing Ltd" (2011).

48. **https://www.geeksforgeeks.org/enchant-request_pwl_dict-in-python/#:~:text=Enchant%20is%20a%20module%20in,exists%20in%20dictionary%20or%20not.**

49. Srujan, K. S., S. S. Nikhil, H. Raghav Rao, K. Karthik, B. S. Harish, and H. M. Keerthi Kumar. "Classification of amazon book reviews based on sentiment analysis." ***In Information Systems Design and Intelligent Applications, pp. 401-411. Springer, Singapore*** (2018).

50. Saxe, Joshua, and Konstantin Berlin. "eXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys." (2017).

51. Parkes, Malcolm Beckwith. "Pause and effect: An introduction to the history of punctuation in the West. Routledge" (2016).

52. Kashish Rastogi " Text Cleaning Methods in NLP" (2022).

53. Wael Etaiwi and Ghazi Naymat "The Impact of applying Different Preprocessing Steps on Review Spam Detection" (2017).

54. Fisher, Jonathan "Here's how people in different countries use emoji". Business Insider Australia (2021).

55. Blagdon, Jeff "How emoji conquered the world". ***The Verge. Vox Media*** (2013).

56. Shiha, Mohammed, and Serkan Ayvaz. "The effects of emoji in sentiment analysis." ***Int. J. Comput. Electr*** (2017)

57. Kaur, Jashanjot, and P. Kaur Buttar. "A systematic review on stopword removal algorithms." ***International Journal on Future Revolution in Computer Science & Communication Engineering*** (2018)

58. Bergmanis, Toms, and Sharon Goldwater. "Context sensitive neural lemmatization with lematus." ***In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*** (2018)

59. Balakrishnan, Vimala, and Ethel Lloyd-Yemoh. "Stemming and lemmatization: A comparison of retrieval performances." (2014)

60. tfidf.com | Tf-IDF stands for term frequency-inverse document frequency.

61. Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. "Understanding bag-of-words model: a statistical framework. "***International journal of machine learning and cybernetics."*** (2010)

62. Jason Brownlee "A Gentle Introduction to the Bag-of-Words Model, Deep Learning for Natural Language Processing" (2017).

63. Mikolov, Tomas; Chen, Kai & Corrado, Gregory S."Computing numeric representations of words in a high-dimensional space" (2015).

64. Muhammad, Putra Fissabil, Retno Kusumaningrum, and Adi Wibowo. "Sentiment analysis using Word2Vec and long short-term memory (LSTM) for Indonesian hotel reviews." ***Procedia Computer Science*** (2021).

65. Rong, Xin. "word2vec parameter learning explained." (2014).

66. Mikolov, Tomas "Efficient Estimation of Word Representations in Vector Space" (2013).

67. Mimi Dutta "Word2Vec for Word Embeddings -A Beginner's Guide" (2021).

68. Albayati, Ahmed Nihad Khorsheed, and Yasin Ortakci "Recommendation Systems on Twitter Data for Marketing Purposes using Content-Based Filtering." ***International Congress on Human-Computer Interaction, Optimization and Robotic Application*** (2022).

69. Dehak, Najim, Reda Dehak, James R. Glass, Douglas A. Reynolds, and Patrick Kenny. "Cosine similarity scoring without score normalization techniques." (2010).

70. Miyamoto, Sadaaki, Ryosuke Abe, Yasunori Endo, and Jun-Ichi Takeshita. "Ward method of hierarchical clustering for non-Euclidean similarity measures." ***In 2015 7th International Conference of Soft Computing and Pattern Recognition*** (2015).

71. Ma, Yong, Shihong Lao, Erina Takikawa, and Masato Kawade. "Discriminant analysis in correlation similarity measure space." ***In Proceedings of the 24th International Conference on Machine learning*** (2007).

72. Bag, Sujoy, Sri Krishna Kumar, and Manoj Kumar Tiwari. "An efficient recommendation generation using relevant Jaccard similarity." ***Information Sciences*** (2007).

APPENDIX A.

SUMMARY OF TEST CASES

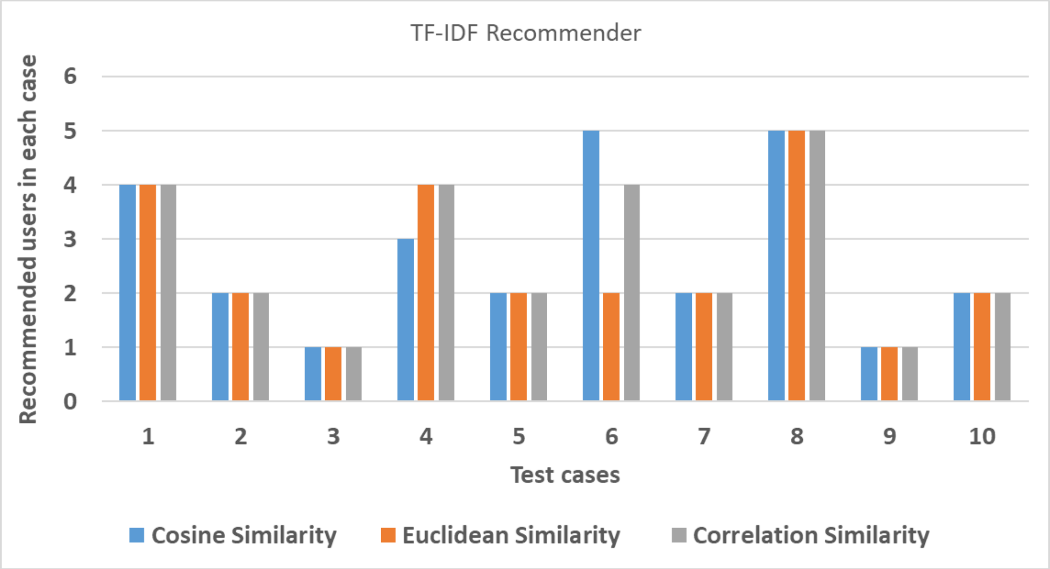


Figure Appendix A.1. TF-IDF Recommender summary.

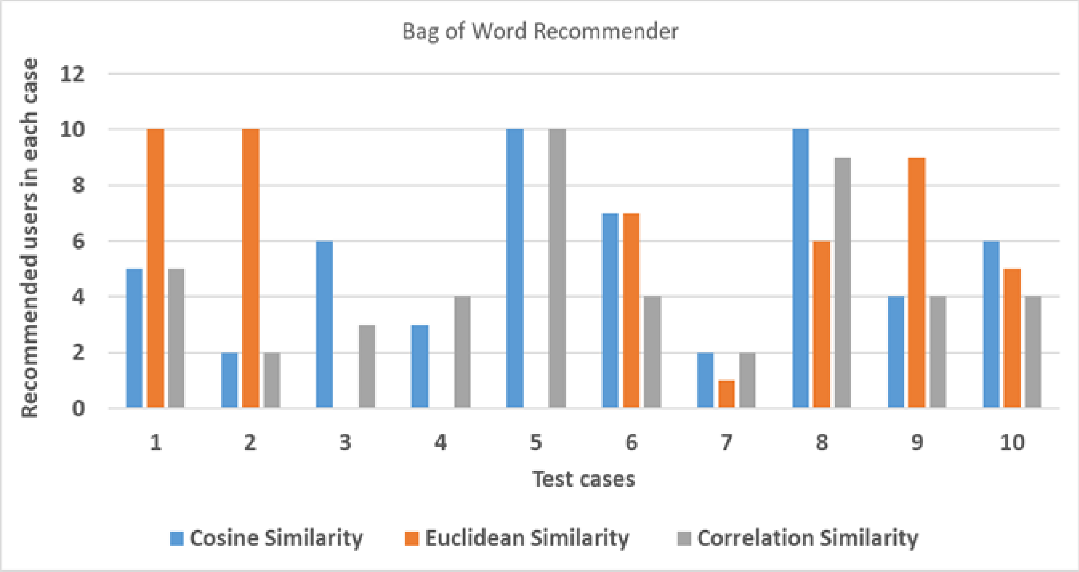


Figure Appendix A.2. Bag of word Recommender summary.

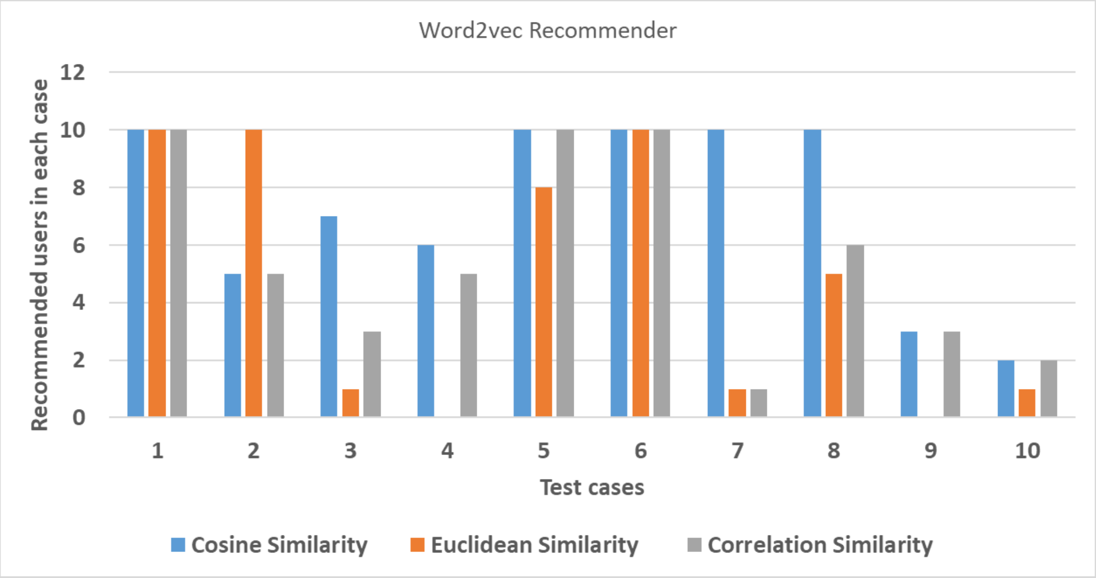


Figure Appendix A.3. Word2vec Recommender summary.

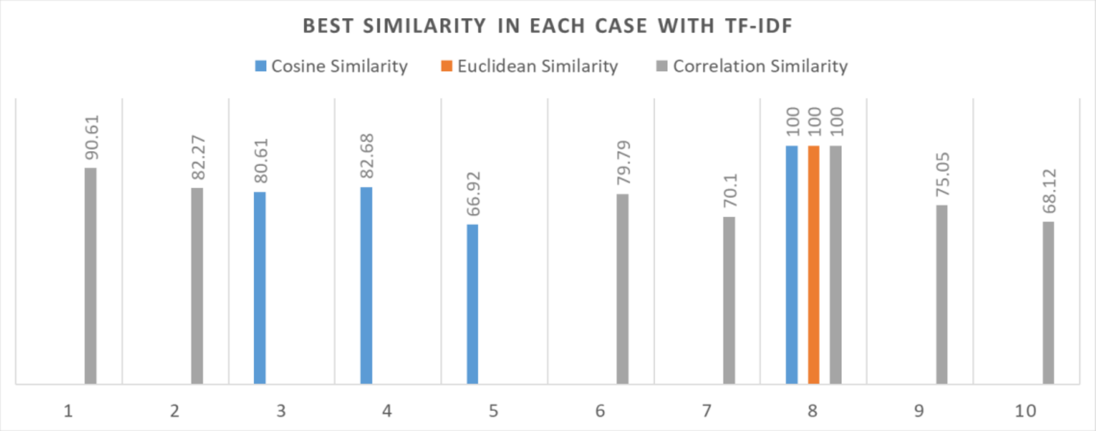


Figure Appendix A.4. TF-IDF best similarity summary.

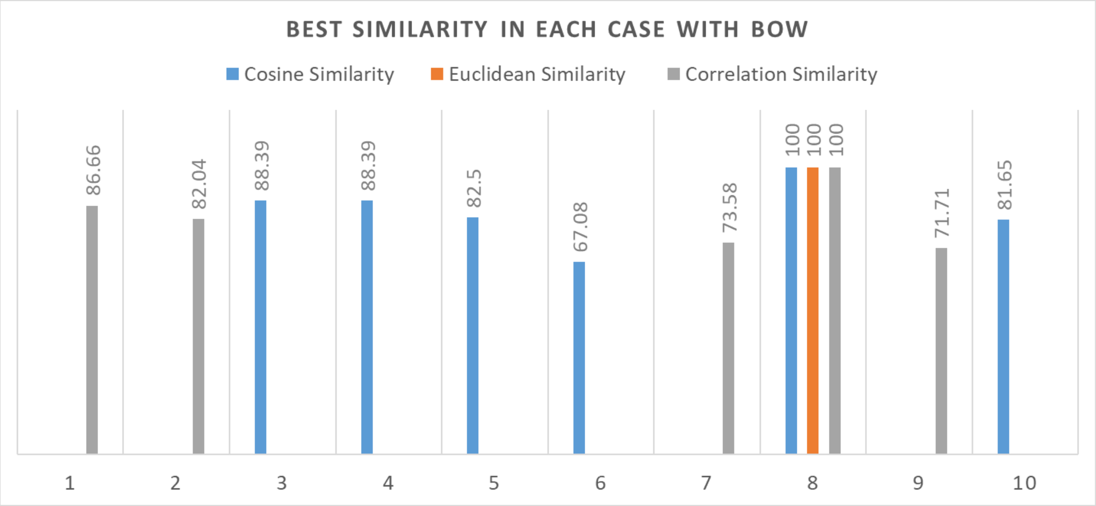


Figure Appendix A.5. BoW best similarity summary.

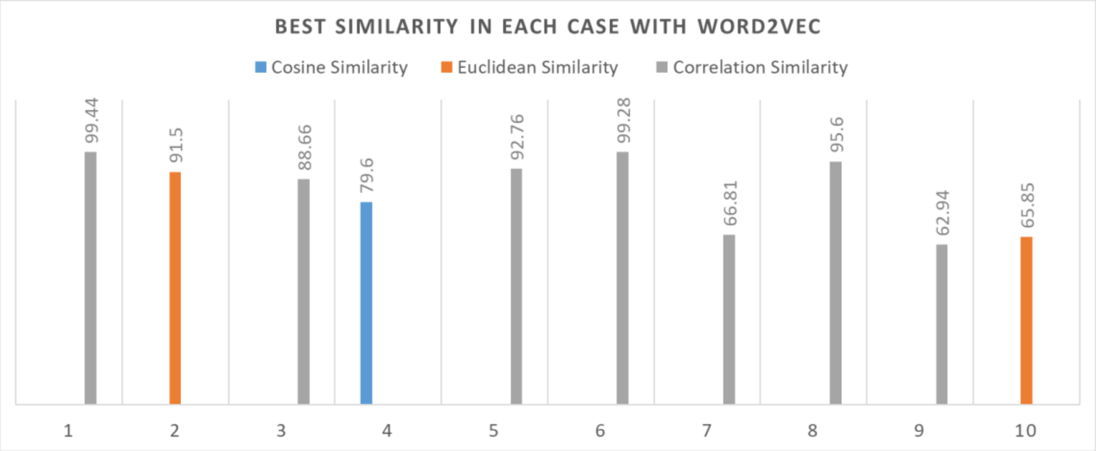


Figure Appendix A.6. Word2vec best similarity summary.

RESUME

Ahmed Nihad Khorsheed ALBAYATI was born in Iraq, graduated from software engineering department in Kerkuk, tech collage / engineering departments. Started Master of computer engineering in Karabük since 2020 and work in Jafer Alsadiq University in Kerkuk.