Service Recommendations Using a Hybrid Approach in Knowledge Graph with Keyword Acceptance Criteria

Jawad Ahmad 1, Abdur Rehman 1, Hafiz Tayyab Rauf 2,*, Kashif Javed 3, Maram Abdullah Alkhayyal 4 and Abeer Ali Alnuaim 5

1 Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan; jawad.ahmad@uog.edu.pk (J.A.); a.rehman@uog.edu.pk (A.R.)
2 Department of Computer Science, Faculty of Engineering & Informatics, University of Bradford, Bradford BD7 1DP, UK
3 Department of Electrical Engineering, University of Engineering and Technology, Lahore 54890, Pakistan; kashif.javed@uet.edu.pk
4 Department of Information Systems, College of Computer and Information Sciences, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia; 442204299@student.ksu.edu.sa
5 Department of Computer Science and Engineering, College of Applied Studies and Community Services, King Saud University, P.O. Box 22459, Riyadh 11495, Saudi Arabia; abalnuaim@ksu.edu.sa
* Correspondence: hafiztayyabrauf093@gmail.com

Abstract: Businesses are overgrowing worldwide; people struggle for their businesses and startups in almost every field of life, whether industrial or academic. The businesses or services have multiple income streams with which they generate revenue. Most companies use different marketing and advertisement strategies to engage their customers and spread their services worldwide. Service recommendation systems are gaining popularity to recommend the best services and products to customers. In recent years, the development of service-oriented computing has had a significant impact on the growth of businesses. Knowledge graphs are commonly used data structures to describe the relations among data entities in recommendation systems. Domain-oriented user and service interaction knowledge graph (DUSKG) is a framework for keyword extraction in recommendation systems. This paper proposes a novel method of chunking-based keyword extractions for hybrid recommendations to extract domain-specific keywords in DUSKG. We further show that the performance of the hybrid approach is better than other techniques. The proposed chunking method for keyword extraction outperforms the existing value feature entity extraction (VF2E) by extracting fewer keywords.

Keywords: knowledge graph; recommendation systems; keyword acceptance

1. Introduction

People are very fond of visiting new places and hotels worldwide in the modern era. They visit historical places, parks, or museums in their leisure time and then have lunch or dinner at a restaurant. People go to other places, cities, or even countries for business work and have to spend some days there; they also prefer to spend their time favorably. Some people are very fond of trying new and different tastes. It is a taste of the tongue that brings those people to new hotels and restaurants to eat delicious food that they may never be tasted before. Hotel owners also want more customers to use their services. The hotel management also gets credits for giving its reliable service in appreciation tips or other compliments. After visiting the place, most people share their experiences with others, which can be positive or negative. The Web performs a vital role in spreading information to others. Most people give their opinions on social media sites like Facebook, Twitter, Instagram, etc. Yelp is one of the common platforms for users to share reviews on different services.
A recommendation system is a subclass of information filtering systems that seeks to predict the rating or the preference a user might give to an item [1]. In simple words, it is an algorithm that suggests relevant items to users. Recommendation systems are used in different areas, including items, articles, news, movies, games, books, hotels, and restaurants. Regardless of innovative advancements in research and development, interest in this field is more extensive because of expanding user requests [2]. Development in recommendation systems in the past ten years has been divided into four stages, the stable period (2009–2012), the rapid growth period (2013–2015), the outbreak period (2016–2017), and the fallback period (2018); this is closely related to the development trend of artificial intelligence [3].

1.1. Yelp

Yelp brings people and local businesses together with word of mouth and gives a platform to communicate with local businesses. It is a platform where people post reviews on local businesses, such as restaurants, hotels, hostels, travel, food services, etc. People leave ratings and reviews on those businesses or services they used or visited. Yelp offers accurate recommendations to users, so it is a trustworthy platform. This makes Yelp ideal over any other similar platform. Yelp has had a significant share in online business review platforms for the past ten years. Yelp generates quarterly earnings reports—according to the fourth quarter 2019 earning report, 96 million users posted 205 million reviews on 4.9 million active local businesses. Out of 4.9 million businesses, 565,000 are paying Yelp for advertisement on its platform. In 2019, Yelp earned 1.014 billion dollars in total net revenue, an 8% increase from the previous year. Yelp’s primary income source is in advertisement, which contributes 90% of the total revenue Yelp-ir.com (accessed on 16 February 2022). In most cases, the digital advertising revenue is calculated as [4]:

\[ \text{revenue} = \text{impressions} \times \text{CPM} \]  \hspace{1cm} (1)

where CPM represents the unit price for every 1000 ad views and impressions defines the number of ad views.

1.2. Knowledge Graphs

A knowledge graph is a simple graph in which vertices and edges are considered as entities and relationships, respectively [4,5]. Objects in the real world can be considered entities, and their relevancy can be considered a relationship in the knowledge graph. This is our data structure based on which recommendations are provided by crawling on a graph through the entities’ relationships.

1.3. Recommendation System

A recommendation system is an algorithm that recommends the most relevant items, in our case, services, to specific users based on the user’s preferences. A sound recommendation system usually combines different machine learning models. The most common recommendation models are location-based, collaborative filtering, and content-based filtering models. The location-based model recommends services to the users based on the user’s location, the services closest to the user. Collaborative filtering gives recommendations based on the behaviors of a similar user or similar service. Content-based recommends services to users by analyzing the user preferences to find the best options. Recommendation systems (RS) gather data on user choices (e.g., books, applications, sites, hotels, restaurants, travel goals, and e-learning material). The data can be obtained explicitly (primarily by collecting users’ reviews) or implicitly (mostly by observing users’ actions, for example, applications downloaded, hotels visited, and books read). RS may use information based on demographic features of the user (like age, gender, location, nationality, etc.). Social data, like followers, followed, tweets, and posts, are typically used in Web 3.0. There is a growing tendency to use data from the Internet of things (e.g., GPS areas, RFID, real-time health signals). Items are recommended based on various factors like ratings, votes, and
reviews from users; these have become a significant measure to extract feedback in the rapid growth of technology [6]. Pattern mining is one of the best approaches used in the system for checking out user interest directions so that a large portion of the transaction can be occupied [7].

We used the reviews posted on restaurants from the Yelp dataset to construct hybrid models for recommendations. First, we conducted data preprocessing steps in which we removed stop words, cleaned up the data, and underwent lemmatization, stemming, and normalizing of the data. We also conducted descriptive analyses for insights and learning the data better in a more interactive way. We adopted two approaches; one is a hybrid recommendation in a specific way. The second is a knowledge graph that we can consider a graph-based recommendation, discussed later. We proposed criteria for keyword acceptance using the DUSKG framework. The DUSKG [8] is a domain-oriented user and service interaction knowledge graph that contains the entities and their relationships. It is a specific data structure based on which different recommendation techniques are applied. One drawback of this proposed method is that as it is a domain-dependent keyword extraction—we must first find out the best set of chunking rules for that particular domain to extract keywords and then apply recommendation techniques.

One of the essential factors to growing a business is the recommendation of the service. What if the seller already knows what product the customer likes or what service a customer wants; this can affect the business growth. One of the recent works on recommendation systems based on the knowledge graph recommends items using the existing recommendation techniques. However, there is a significant issue that needs addressing in domain-independent keyword extraction. Therefore, we proposed a chunking-based keyword extraction method.

Our significant contribution is a keyword extraction method to know user interests better and the relationship between users and restaurants. In [8], authors use rapid automated keyword extraction (RAKE), a domain-independent and language-independent method to extract keywords from review text and apply pruning rules afterward. We observe that this extracts many irrelevant keywords and rejects many relevant keywords from the recommendation process. A large number of keywords causes a large knowledge graph that requires high computational resources and space to run recommendation techniques. We consider keyword acceptance criteria rather than rejecting. We introduced a domain-dependent extraction method that extracts more refined keywords from text based on the chunking rules. The number of keywords and the execution time of keyword extraction were also reduced. We present a more scalable recommendation framework to apply various combinations of chunking rules to extract domain-specific keywords.

The remaining paper is organized as follows: Section 2 contains a literature review and previous work on recommendation systems and knowledge graphs. Section 3 represents the core methodology and DUSKG framework, whereas Section 4 contains results and discussion. The study is concluded in Section 5.

2. Related Work

Modern transportation and communication networks have transformed the world into a global village. Millions of tourists move from country to country for recreational activities. It offers an excellent opportunity for businesses worldwide to flourish to reach customers and maximize their profits. Online systems enable users to upload their views to share their experience with other people about a service. Recommendation systems use customer reviews to automatically recommend a service according to their preferences. Listed below are different types of recommendation systems.

2.1. Collaborative Filtering-Based SR

Collaborative Filtering (CF)-based service recommendations attempt to prescribe benefits regarding the comparability of users or services [9,10]. Liu et al. [11] worked for a QoS-aware web service recommendation proposing a location-aware CF approach.
2.2. Content-Based SR

The content-based service recommendation (CSR) techniques recommend services similar to other services preferred by the user, which can solve the cold-start problem. In any case, the techniques typically ask users to know the details of a service, which is hard for users. The CSR methodologies can be categorized into two types: semantic-based techniques and syntactic-based techniques. Semantic-based techniques recommend web services by using the semantic details of their processes using metaphysics information [13] extracted from service domain objectives from text data to fulfill user requirements. Xu et al. [14] present a semantic-based service discovery framework.

2.3. Graph-Based SR

A graph-based recommendation gives better performance in service recommendations with quick improvements in recommendation methods. Graduate et al. [15] built the KG for point of interest (POI) and recommended ideal POI for users by spreading initiation strategy dependent on the developed KG. A KG framework was constructed from service data and presented the keywords extraction method VF2E to recommend services based on user preferences [8]. We improved this method and proposed the method of chunking in our work [8]. Catherine and Cohen [2] report a probabilistic rationale programming approach dependent on KG for a customized recommendation. What is more, a random walk method has been prevalently connected in the graph-based recommendation.

The authors in [16] proposed a knowledge graph-based recommendation for web services. First, they analyze entities and their relationships in the data and then calculate the similarities among users and identify the nearest neighbors. Jiang [17] set forward a Bayesian personalized ranking-based machine learning method (named Hete-Learn) to get familiar with the weights of links in a heterogeneous information network (HIN). To demonstrate user preferences for a personalized recommendation, they proposed a generalized random walk with a restart model on HINs. Yao et al. [18] presented a graph-based generic recommendation structure, which develops a multi-layer context graph (MLCG) from implicit feedback data and performs a ranking algorithm in MLCG for a context-aware recommendation. The authors in [19] presented a knowledge graph-based service recommendation method that considered community effect and service content. A graph-based model for IoT web services recommendation was proposed by [20]. Similarly, authors [21] introduced a novel service recommendation technique for cyber-physical systems that uses network location as context information and contains three prediction models using random walking. The recommendation engines are also prevalent in the IoT domain, especially for social purposes. The graph-based recommendation has attracted people’s attention in academia and industry. They explored the traditional and most recent developments of filtering in recommendation systems, identified and analyzed the people-proposed methods related to knowledge graph-based recommendation systems, presented the most related contributions using application domains and future direction of research in this domain. Finally, they concluded that the knowledge graphs for recommendations are a very efficient way to connect users and items or services, providing more accurate recommendations.

2.4. Hybrid SR

A hybrid service recommendation is an approach in which two or more recommendation methods are combined to achieve higher performance in the recommendation. Mainly two techniques of collaborative filtering and content-based filtering are combined to make the recommendation approach hybrid. The authors [22] proposed a hybrid approach using...
deep learning for web service recommendation. They introduce a hybrid Web service proposal recommendation by combining collaborative filtering and textual content. Mashal et al. [20] presented the hybrid service recommendation algorithm to recommend web services using the Internet of Things (IoT). They used two different methods of hybridization, weighted average and simple multiplication. The hybrid recommendation technique was also proposed by [23] for service recommendation. The authors in [24] used this technique for the Web Service API recommendation. The recommendation of services using a hybrid approach in the WIKI-WS platform was proposed by Sobecki [25]. For hotel recommendations, a hybrid multi-criteria hotel recommendation approach was proposed by [26,27]. Jain et al. [28] incorporated three components into the service recommendation process, specifically functionalities API, API usage popularities, and history of APIs. Zhang et al. [29] introduce a hybrid social network-based CF technique for recommending personalized assembling services by considering the reliable ventures and three kinds of similar enterprises with various features for computing predicted scores of competitor services. Deep learning algorithms [30,31] have been widely used for solving complex problems in big data. However, some challenges remain, such as recommendations for mobile wireless networks using big data. The authors [32] used hybrid recommendation techniques in their work to improve accuracy. Discovering service recommendations in a highly dynamic number of services brings the recommendation’s cold start and sparsity problems. To alleviate such issues, the authors [33] proposed a hybrid service recommendation for ubiquitous consumer wireless world (UCWW). A restaurant recommendation to get a better recommendation performance [34] used a hybrid approach by combining collaborative filtering and content-based filtering and showed its effectiveness visually. The authors in [35] improve the customer behavior prediction with the item2item model using a hybrid recommendation technique for restaurant recommendations.

3. Methodology

In this research, we applied pre-processing techniques to extract keywords using the chunking technique, which accepts the textual data in its real form.

3.1. Dataset Overview

The dataset sources from Yelp’s business, reviews, users, and check-in data. It was initially distributed for the Yelp Dataset Challenge, an event for students to share their data analysis. This dataset contains information on business across 11 metropolitan areas of four countries. The complete dataset includes 6.68 million reviews from 1.6 million users evaluating 192,000 businesses from 2004 to 2019. The top 5 categories within these 192,000 businesses were restaurants, shopping centers, home services, health and medical, and beauty and spas. The top 5 city rankings by the number of local businesses were Las Vegas, Toronto, Phoenix, Charlotte, and Scottsdale. The dataset is vast; analyzing all the data is difficult in terms of resources. In our research, we used all the reviews on Italian restaurants in Toronto. The original dataset consisted of 5 users, reviews, business, check-in, and tip JSON files. The join keys in the review file were user_id, review_id, and business_id. The join keys in the tip file were user_id and business_id. The join keys in the tip file were user_id and business_id. The join keys in user, business, and check-in files were user_id, business_id, and business_id. The features information in individual files are given in the snapshot, and the relationship of the three most essential entities is represented in Figure 1.

All data files are associated by a key, for example, reviews and tip data files are linked with two files, user and business, respectively.
3.2. Exploratory Data Analysis and Visualization

Data visualization is a valuable step in data science to explore the data. Some of the analyses and visualizations on an Italian restaurant in Toronto city to represent several restaurants by review count and the number of reviews by stars are as given in Figure 2.

![Figure 2. Reviews count in restaurants and stars count in reviews.](image)

Figure 3 represents the star rating distribution among restaurants and their counts.

![Figure 3. Star rating (1–5) distribution among a number of restaurants.](image)

In the Figure 3, the star rating is above average, which represents that the overall ratings given by users to these restaurants are good.

There were 177 restaurants with ratings of 4.5 in the dataset. The distribution of star ratings was normal among restaurants.
3.3. Text Preprocessing

Text processing is one of the most common tasks in natural language processing (NLP) applications that help transform the raw/unstructured text into some format that a machine can understand.

3.3.1. Removing Numbers, Punctuation

Numbers and punctuation usually do not convey value in text analysis tasks. The numbers or digits in reviews can be used as item price, the number of food items ordered, etc. For example, some numbers and punctuation marks are used in these reviews.

Review Examples:

1. “Poor service, terrible food, overpriced. Why bother?”
2. “best pizza in the city! great staff! yay strada!”
3. “One of the best restaurants in the city: great cocktail list, also: Order the Max Valiquette!”
4. “Lobster Tuesday for 25 $, good linguini and lobster.”
5. “Service was very good, friendly. 3–5 over priced ($15) for rigatoni.”

In the above examples, there are numbers and punctuation marks used, like (?!)-. The machine learning model is affected by such irrelevant data items, and leads to bad results. Therefore, in sentiment analysis like work, we first remove all the numbers or digits and punctuation from the reviews so that model may train on actual real data values.

Contractions are the short form of words (or a group of words) written in certain letters and pronounced differently than the complete word(s). In most contractions, an apostrophe (‘) represents the missing letters. Most of the time, reviewers use contractions in their reviews. It also represents a particular person’s writing style as the review itself is not something that must be very clear In the sense of following grammatical rules and word representation. The most common contractions contain verbs, auxiliaries, or models attached to other words. Some examples of reviews containing contractions are as follows:

- Most bland veal sandwich I’ve ever had.
- Shouldn’t have been surprised but the food quality wasn’t great and the service took forever.
- Would’ve expected more for $19 but the quality is there.
- Won’t be going here again.
- Fantastic service. I can’t say a negative thing.

In the above review examples, some contractions are used, they are: I’ve, Shouldn’t, Would’ve, Won’t, can’t.

3.3.2. Text Tokenization, Normalization, and Stop-Words

Text tokenization is the very first step in text preprocessing before applying any operation at the word level. It is an algorithm that breaks down the text strings into individual words and other punctuation symbols by white-spaced characters. We had to conduct this process first before feeding the data into the part-of-speech (POS) tagger. Afterward, we normalized the raw text into canonical form. A single English word like “connect” may have multiple forms “connects”, “connected”, “connection”, “connecting”, “connectivity”, etc. Text normalization converts all such forms into their original word or root word. We do this to allow NLP to recognize words with similar meanings. Text is normalized using two approaches, stemming or Lemmatization.

3.3.3. Stemming and Lemmatization

Stemming and Lemmatization are text normalization techniques in NLP (natural language processing). Usually, a word has multiple meanings based on its context in the text; in the same way, different words convey different meanings. We use different word forms in our sentences based on the grammar rules to convey the complete and correct message. However, in the field of NLP, ML models do not work on such different forms of words; the models treat such multiple forms of words as different individual entities that cause more extensive storage and high computation with no use. The ML model is
also affected by such variations of words, so we converted all different forms of words into their root word in data preprocessing activities.

- I liked dining here.
- The Asian guy working was very respectful.
- The phone seems to be disconnected and hasn’t opened up in weeks.

In the above short reviews, users use ‘liked’, ‘working’, ‘respectful’, ‘disconnected’. These are the extended forms of words. These can also be liking, likeness, worked, respected, respectfulness, respected, disconnection, etc. Therefore, in NLP, we convert such words into their root forms: ‘like’, ‘work’, ‘ respect’, ‘disconnect’, using both stemming and lemmatization techniques by SnowballStemmer and WordNetLemmatizer, respectively.

Although both techniques are used to obtain root words, the difference is that stemming works by cutting the end or beginning of the word and extracting the common word form among all its variants as the final root word. Most of the time, it works successfully, but not always. For example, the words ‘study’, ‘studying’, ‘studies’, ‘studied’ stemming would extract the word ‘stud’ as the root word, because changes are after the letter ‘d’, but the word ‘stud’ is not the correct word and is wrong in this case. In contrast, lemmatization extracts the root word based on its dictionary meaning. For the sample example above, lemmatization will extract the word ‘study’ as the root word. However, the drawback of lemmatization is that it is significantly slower than stemming as it has to look up the word in the dictionary for the correct root.

The existing techniques used for recommendation are collaborative filtering and content-based filtering. However, these approaches have data sparsity, cold start problems, and scale-ability issues. Our work revolves around graph-based service and hybrid service recommendations. We get the hybrid service recommendation by combining these two approaches using a knowledge graph.

To better understand the process of this recommendation using knowledge graph framework, a flow diagram is given in Figure 4, which demonstrates the end-to-end flow of the recommendation framework.

![Figure 4](image_url)

Figure 4. A flow diagram representing the end-to-end process of recommendation, including data processing, knowledge graph constructions, and recommendation.

### 3.4. DUSKG Framework

The DUSKG framework consists of three entities with five relations among them. The three entities are user, service, and value features (VFs). These value features are basically the keywords extracted from the reviews. The user is an entity/object that reviews services and for whom the recommendations would be generated. A review is an entity based on which we extracted the value preferences to get the interest and taste of users towards
services. It tells us what recommendations are made and to whom. To extract VFs from reviews, the Rake tool is used for keyword extraction, but VFs extracted from it are huge, and some of them are not desired keywords. Some rules are made according to the need that is further applied to the extracted keywords to filter them as required. For this purpose, a new algorithm is developed in which new objectives are achieved in terms of POS tagging of more than just nouns and verbs. Value preferences to VFs of the service are calculated using the sentiment analysis technique Textblob, which gives the polarity of the keyword. Service is an entity on which the reviews are written, and that is what is to be recommended to the target user.

The five relations among entities were identified. The relations were FOCUSON, BELONGTO, USIMILAR, FSIMILAR, SSIMILAR. FOCUSON relations exist between the user and VF entities; it tells us the aspects based on which the user has concern toward any service in the review. For example, if a user writes a review “Food is tasty but the wait time is long”, it means that the user wants to say something about tasty food and complains about the wait time being long. Thus, the relation between a user entity and two VF entities (food and wait time) is FOCUSON because the user’s focus in his review is on the food and wait time. The BELONGTO relation exists between VF and service entities; it tells us that a particular VF belongs to that service. It means this observation is identified in that specific service. In the example given, the relation of those VF entities with its service for which the review is written is the BELONGTO relation. SIMILAR is a relation between two user entities to check similarity between them. A SIMILAR relation exists between two VF entities. A tool word2vec is used to check the similarity between two VFs. A SIMILAR relation exists between two service entities. Weight vectors are associated with each relation implying the specific similarity values.

DUSKG can be demonstrated as DUSKG = (E, R, TE, TR, fE, fR, A, fEA, fRA).

Here, E, R, TE, and TR represent the set of entities, services, entity type, and relation type, respectively, in KG. The terms used in the DUSKG framework are explained below.

E = {User, VF, Service}

R = {FOCUSON, BELONGTO, USIMILAR, FSIMILAR, SSIMILAR}

R ⊆ {u, v} | (u ∈ E) ∧ (v ∈ E) relation can only exist between two entities u and v, and both entities must belong to entity set E

fE ⊆ E × TE function assigns a specific type to entity from entity set E

fR ⊆ R × TR function assigns a specific type to relation from relation set R

A represents the attributes of entities or relations with their corresponding values

fEA ⊆ E × A function assigns attributes with values to the entity

fRA ⊆ R × A function assigns attributes with values to a relation

The relation among entities in DUSKG are defined according to a rule given below:

fR(R) = FOCUSON if ((fE(E) = User) ∧ (fE(E+1) = VF))

fR(R) = BELONGTO if ((fE(E) = VF) ∧ (fE(E+1) = Service))

Rules : fR(R) = USIMILSAR if ((fE(E) = User) ∧ (fE(E+1) = User))

fR(R) = FSIMILAR if ((fE(E) = VF) ∧ (fE(E+1) = VF))

fR(R) = SSIMILAR if ((fE(E) = Service) ∧ (fE(E+1) = Service))

The above rule represents E as a first entity and E+1 as a second entity, the relation FOCUSON can only exist between the user entity and VF entity. The relation BELONGTO can only exist between the VF entity and the service entity. The relation USIMILAR, FSIMILAR, and SSIMILAR can only exist between two user, VF, and service entities, respectively.

The Yelp data of the user, reviews, tips, restaurants, and check-ins are used as input in the whole recommendation process. In Figure 5, yellow-colored boxes represent entities, and blue-colored text represents the relations among them. The green-colored boxes show
our contribution to the recommendation process. The first step is to extract restaurant category data from the business dataset, and then based on this data, the data from other files are extracted. In the second step, the required preprocessing is performed on the data. In preprocessing, some tasks occur, such as stop words removal, tokenization, stemming, lemmatization, etc.; for this purpose, we used the Natural Language Toolkit (NLTK) library in python, which is for text processing, tokenization, parsing, classification, stemming, tagging, and semantic reasoning in textual data. A RAKE tool is used in mining features from user reviews, as discussed before, and then sentiment analysis is performed using the TextBlob toolkit discussed above. In the third step, relationships among all the entities are confirmed. In other words, triples are created in this step, then to construct the knowledge graph, we used Knowledge Graph, which interprets a relation. After constructing the knowledge graph, a recommendation algorithm is applied in the fourth step, specifically a hybrid algorithm. In the end, the model will give recommendations to the user based on his interest and profile.

We did not perform data preprocessing on the data because the chunking technique works on the grammar rules and the sentence structure, including all the POS and their proper forms. They include preprocessing techniques, including removing stop-words, lemmatization or stemming, etc., no need to be performed on the data.

There can be eight POS in a sentence or text: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection. All POS words play a specific role in the sentence structure. The most important POSs are noun, adjective, and interjection in sentiment analysis-like tasks. We applied chunking with the following rules on the text.

NP: {
  <NN.*><VB.*><JJ>*<RB>*<UH>*
  {<VB.*><NN><JJ>*<RB>*<UH>*
  <NN.*><JJ>*
  {<JJ.*><NN.*
  {<NN.*><UH>
  {<UH><NN.*
  {<FAC|ORG|LOC|PRODUCT|EVENT><JJ.*>
  {<JJ.*><FAC|ORG|LOC|PRODUCT|EVENT>

Figure 5. User, restaurant, and keyword extracted from reviews using chunking-based methods are entities and their relations FOCUSON in user-keyword, BELONGTO in keyword-restaurant, and their similarity relations among entities themselves form the knowledge graph in matrices [8].
We observed from the previous studies that people use noun phrases with adjectives in chunking rules to find the polarities from reviews or text. The other parts of speech, like pronouns, adverbs, and prepositions, do not play an important role in getting polarities or sentiment analyses. The noun (NN) words are used to represent a person, place, thing, or idea in a sentence or review. A noun can be a subject or object, about which, the user writes something useful, interesting, or informative in their review. This is what the user provides the data for. For example, nouns can be a restaurant name, food item, location name, etc. The adjective (JJ) words describe a noun in a sentence. They represent the state of a noun, for example, ‘beautiful’, ‘fantastic’, ‘good’, ‘better’, ‘best’, ‘worst’ etc. The verb (VB) expresses the action of anything being completed in the sentence. The verbs in reviews can be ‘eat’, ‘drink’, ‘clean’, etc. The interjection words express emotions, for example, ‘Oh!’, ‘Wow!’, ‘Oops!’, etc. The adverb (RB) modifies or describes a verb, an adjective, or another adverb. For example, the adverbs can be ‘gently’, ‘extremely’, ‘carefully’, etc. Finally, the star (*) is used to acquire different forms of the POS words, e.g., nouns can be singular or plural, so using an asterisk, both types of nouns are considered. Moreover, sometimes there may be a verb or adverb between a noun and adjective, so we also used an asterisk to extract both cases.

These POS words are essential in the sentence to get polarity, semantics, or user opinion about that particular restaurant. The verbs are also helpful in the context of keywords similarity. For example, if one user shares their experience at restaurants and writes in the review, “I enjoyed my stay at the restaurant and especially the swimming in the pool”. Furthermore, another user writes a review like “We eat healthy food and do swimming with our friends, great time, making fun”. Therefore, the verb “swimming” is being used in both reviews posted by different users. The verb swimming represents this restaurant, so these two reviews can be similarly based on the word swimming for that particular restaurant. We can recommend such restaurants to those users who like swimming or who wrote about swimming in their reviews.

In the above rules, it is observed that extracted noun phrase chunks from the text, in which nouns mainly come with an adjective before or after them. In the second pattern, the verb first comes with nouns and adjectives, and after that, the adverb or interjection is the same as before. In the same way, the other such rules acquire noun first and adjective second, an adjective first and noun second, noun first and interjection second, interjection first and noun second. For now, we used three POSs noun, verb, adjective in patterns considering them as essential to get the user opinion from their reviews. We can also use different POSs or with different sequences in patterns.

We performed experiments to understand the working of our keyword extraction method and evaluate its performance. We randomly selected 1000 reviews to compare their approach and our proposed method on keyword extraction. The results we achieved are the following.

3.5. Keywords Extraction Execution Time

The results depend on how we deployed this model in the application or recommendation system. However, things that take time in their execution matter in recommending items to users, specifically real-time recommendations. However, at some level, whenever we create a model with a lot of reviews, we realize how execution time affects our recommendation system. In the whole recommendations process, keyword extraction is the only activity that takes time. We compared the time taken by the keyword extraction technique VF2E and with our proposed method of chunking as follows:

1. 1000 it [03:28, 4.79 it/s] (VF2E);
2. 1000 it [01:12, 13.88 it/s] (Chunking)

In the above stats, 1000 it means the iterations over 1000 reviews, and in the square brackets, the time taken and the number of iterations per second are given. We also observed VF2E [8] which took 3 min and 28 s, whereas our proposed chunking method took 1 min and 12 s to extract keywords from 1000 reviews.
3.6. Number of Keywords Extracted

As many keywords are extracted, the size of the knowledge graph and the computations increase, and we also have to utilize high-end resources. Thus, one of our concerns is to reduce the size of the knowledge graph by extracting the more minor but more specific, meaningful, and relevant keywords. In terms of keywords selection, the approach VF2E follows a rule that considers what keywords need to be rejected, whereas, in the proposed method of chunking, we follow the criteria of keywords based on some grammatical rules. In this approach, the main concern is what keywords need to be rejected based on some simple rules mentioned above. We extracted keywords from 1000 reviews and noticed the number of keywords:

1. 20,962 keywords extracted by VF2E;
2. 2397 keywords extracted by chunking

There is a huge difference between the number of keywords extracted by their technique and, in our chunking method, the number of keywords was much lower.

3.7. VF2E vs. Keyword Extraction with Chunking

Furthermore, some other comparisons between VF2E and the proposed approach are in finding polarity of just keywords; we performed chunking in the reviews based on some grammar rules defined in our method. In this way, we calculated the polarity, but for that particular chunk from the review, either polarity of all the sentences where that chunk appears in a text or only of that chunk. To better clarify this point, we took some examples to get the polarity of the words. If the word ‘very delicious’ was extracted as a keyword, the polarity of this would be positive. However, in reality, this word was used in a sentence like “The food was not very delicious”. Thus, the polarity of the whole sentence would be negative. Moreover, when we passed this sentence to the chunking method, it gave us some essential chunks from the text like ‘food not delicious’.

Both methods aim to find out the most relevant services/restaurants and recommend them to users. Generally, this process does not follow the traditional machine learning techniques like splitting the dataset into train and test parts, and getting the model trained on the train set and then tested on some unseen test instance to evaluate the performance/learning using classifiers/algorithms. The basic concepts of both recommendation approaches are the same—we have not only training datasets like customer reviews. This is precisely why we prepare our data and construct a particular structure in which we can recommend items to users based on their reviews posted on a service. In our recommendation system, we attempt to recommend services to users without knowing what services they have visited before any step. The system is unaware of the services visited by the users; therefore, the proposed method blindly recommends services to customers in terms of their previously visited services. Although we have some models, structures, and environments in which we do some processing on the data provided by the customer and then come to a decision about what services should be recommended to which user. The foremost important step is keyword extraction from the reviews, which then become entities based on which relationships are created in the knowledge graph. When we have to recommend services to a user, the algorithm crawls on a graph to extract the best recommendations. At this step of extracting keywords, we compared the proposed method with the existing one. They use pruning rules based on which they reject the keywords; their main concern on keyword extraction is basically which keywords should be rejected from many keywords. In the extraction approach, RAKE is used, even though, in VF2E [8] the authors presented their approach as an improved RAKE method with a set of pruning rules named value feature entity extraction (VF2E) even then the number of keywords extracted is considerable comparatively by our proposed method. As they have already mentioned, the keywords extracted by RAKE are large in number. However, after extracting the keywords by RAKE the authors in VF2E apply pruning rules, and the three rules about which we have concern are as follows:
Rule 1: Exclude VFs, which includes one word and is an adjective, adverb, or interjection.

Rule 2: Exclude VFs, which have the uppercase first letters and whose named entity is identified as GPE (geopolitical entity) or PERSON.

Rule 3: Exclude VFs without a noun.

In the critical analysis considering all the rules mentioned above, we found that they accept only those keywords which contain at least one noun. Thus, it is evident that nouns are the most crucial entity in the graph for which user’s post reviews or talk about a restaurant for that specific entity in their reviews. However, in this rule, it is also evident that we have all the noun words as keywords in the results, and the minimum length of the keyword is two letters; so in this case, the simple rule causes a tremendous amount of noun words as keywords. Rule 3 is a general rule, impacting all keywords, that any keyword without a noun would be discarded. Thus, they accept only those keywords that must include at least one noun. Therefore, if any keyword is not a noun itself, it can only be accepted if it is extracted with any other noun word as one keyword. On the other side, Rule 1 is if any keyword is one word and is an adjective or adverb.

Alternatively, interjection would also be discarded. Finally, it comes to the result that a noun with any other word except these three POS words would be accepted. It leads us to irrelevant and unexpected keywords. For example, in a review, the extracted keywords were:

Review: “The food quality is good. The service is great. Clean environment and the ambiance is incredible.”

Keywords extracted by VF2E ['ambience', 'clean environment', 'service', 'food quality'].

Keywords extracted by the Chunking method ['quality is good', 'service is great', 'clean environment', 'ambiance is incredible'].

The keywords ‘ambiance’ and ‘service’ represent little. Just because it has been considered single nouns, the VF2E accepts them according to their pruning rules. Moreover, the proposed method did not accept the single noun as keywords. In our rules, nouns must come up with any other word, most probably the adjectives before or after, to convey proper positive or negative information about that noun, which may affect our recommendations process. The word ‘clean environment’ is acceptable because the noun ‘environment’ in this keyword comes up with ‘clean’. This is the correct keyword that our chunking method also extracted. The keyword ‘food quality’ also does not portray anything worthy, because we do not know what it is about the food quality—is this bad, worst, good, or best. We get such information from this keyword; so, in our method, the extracted keywords properly represent the state of the noun, e.g., ‘service is great’, ‘clean ambiance’, ‘ambiance is incredible’, and the ‘quality is good’. These keywords are much more important and relevant and have a great impact on our recommendations process. Although there may be issues in the keyword, like, ‘quality is good’—the quality referenced is unclear, food quality, staff quality, etc. Therefore, there may be drawbacks of such hidden information in the process at the bottom level. Nevertheless, the difference between the keywords extracted by VF2E and our chunking method is prominent. Some other points to be focused on in terms of our proposed method are that they are applying the POS tagging on words level in the competitor work. Because some words can be treated in multiple forms in the review, for example, the word ‘clean’ is an adjective itself, but if we use this in a sentence like “The environment is spotless”, it is being used as an adjective. However, in the sentence “The environment should have a clean”, the word clean is used as a verb. When we apply any POS tagging technique on a single word, it does not capture the context of that particular word in the sentence. Nevertheless, we applied the POS tagging at the sentence level to obtain the contextual POS tag of the word.
4. Results and Discussion

We used the Yelp dataset in our research, and because of the vast amount of data, we filtered it only for Italian restaurants in Toronto city. However, the existing approaches were applied on a different datasets in the sense of data filtering. Nevertheless, the domain and category we extracted the data for are the same. Therefore, after finalizing the proposed methods and other existing techniques to give recommendations, we applied existing and proposed methods on the same dataset in order to evaluate and compare them. We obtained better results over the competitor’s methods. We verified the performance of hybrid recommendation techniques over the other recommendation techniques, giving the recommendations for five randomly selected different users. The performance comparison of hybrid recommendation techniques against other techniques for five different users is illustrated in Figure 6.

![Figure 6. Performance comparison of hybrid recommendation technique over other techniques for five different users. The blue, red, yellow, and green lines show performance score of the recommendation techniques UCF, ICF, CBF, and hybrid, respectively.](image)

In the above, Figure 6, we compared the hybrid recommendation technique (hybrid) with other techniques, which are user-based collaborative filtering (UCF), item-based collaborative filtering (ICF), and content-based filtering (CBF). The graphs show the hybrid technique’s performance on several recommendations. Here, we observed that the hybrid technique outperforms all other recommendation techniques, and also, as the number of recommendations increases, the performance of recommendation techniques also increases. That green line represents the hybrid recommendation technique performance and is at the top in all the graphs.
In Table 1, we calculated the average scores of the groups of three users each. The baseline recommendation methods of VF2E show the score values on the different number of recommended sets of services. Our proposed methods of chunking show their score values in bold. Our proposed methods perform better on each recommendation count than the other respective baseline method. We also noticed that the score of recommendation methods increases as the number of recommendations increases. Furthermore, the performance of hybrid recommendations remained consistently higher than the other techniques.

Table 1. Performance comparison of baseline methods and proposed method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Number of Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>UCF_VF2E</td>
<td>5.86</td>
</tr>
<tr>
<td>UCF_Chunking</td>
<td>6.02</td>
</tr>
<tr>
<td>ICF_VF2E</td>
<td>3.57</td>
</tr>
<tr>
<td>ICF_Chunking</td>
<td>5.72</td>
</tr>
<tr>
<td>CBF_VF2E</td>
<td>3.73</td>
</tr>
<tr>
<td>CBF_Chunking</td>
<td>8.75</td>
</tr>
<tr>
<td>Hybrid_VF2E</td>
<td>4.41</td>
</tr>
<tr>
<td>Hybrid_Chunking</td>
<td>10.68</td>
</tr>
</tbody>
</table>

The other techniques are not consistent in their performance because of their recommendation process. However, the hybrid technology works in that it combines the recommendations item/service sets in one list and the order matters based on the weightage of the recommended items assigned to them by the recommendations techniques. The above graph’s hybrid recommendation technique performance line consistently increases order and mainly performs better than the other techniques for all five users.

In the literature review, we studied the recommendations technique used to give recommendations to any particular group of users, or the items belonging to a specific group of items being recommended to users. Sometimes recommendations are made for individual users, and sometimes items are recommended to a group of users based on some criteria. Qian et al. [36] focused on recommending visualizations personalized for each individual based on their past interactions.

In Figure 7, we first selected all the users who posted at least ten reviews; in other words, those who have visited at least ten services. After this, we created five user groups and randomly selected three groups for each recommendation technique. Furthermore, in the above graphs, all three groups for their respective recommendation techniques were represented in each graph. We also divided the combined groups into smaller graphs in the figures for better understanding. The comparison was performed between the recommendation techniques with our proposed keywords extraction method and the baseline recommendation techniques with the keywords extraction method of VF2E. These graphs show that our proposed chunking method represents the baseline technique UCF_VF2E and the red line represents our proposed technique with chunking. In each of the three graphs, UCF_Chunking outperformed the UCF_VF2E. Similarly, our proposed methods of recommendations with chunking performed better than the other respective baseline recommendation techniques of VF2E in each graph.
Figure 7. Comparison among our proposed techniques (Chunking) with baseline methods (VF2E) for a group of three users in each graph (combined). The red, green, blue, and light green lines show the performance score of our proposed methods UCF_Chunking, ICF_Chunking, CBF_Chunking, and Hybrid_Chunking methods over the various baseline methods of VF2E.

In Figures 8–11, our proposed method of keyword extraction with chunking outperformed all other respective baseline methods of VF2E. We evaluated the performance of our method on three different groups of five users for each recommendation technique and gave the comparison of baseline methods in the same way. These graphs were just the separation of the above-combined graphs given in Figure 7 for better understanding and to represent the recommendation techniques’ performance. This shows the performance comparison of the average results of three groups. It is shown in all graphs mentioned that performance of the hybrid recommendation technique was better than all other recommendation methods, whether those are baseline methods or the chunking itself. Moreover, we calculated the average of three groups of five users to combine all our results into one graph. Comparison among our proposed techniques (Chunking) with baseline methods (VF2E) on average results of three groups is given in Figure 12.

The overall performance of the recommendation technique with the proposed method of chunking in keyword extraction performed better than all other techniques. The lines red, green, light blue, and light green in the graph represent the recommendation techniques with chunking-based keywords extraction method, which are UCF_Chunking, ICF_Chunking, CBF_Chunking, and Hybrid_Chunking, respectively. The base article could achieve the maximum score of 31% on 25 recommendations with a hybrid recommendation. Therefore, we concluded that this technique performed better than all other recommendation techniques. We also showed in our experiments that the hybrid recommendation results were always better than the other techniques, achieving scores of 22.22%, 34.48%, 32%, 36.84%, 41.18% with 50 recommendations for five individual users, respectively. Using the proposed chunking-based keyword extraction method, the average score was 33.34%. In comparison, they achieved a 30% score for 50 recommendations. The score of recommendations is increased as the number of recommendations increases. In each figure shown above, the x label is the number of recommendations on average, representing the average scores of each set of three graphs.
Figure 8. Performance comparison of proposed UCF_Chunking technique with baseline methods (VF2E) for a group of three users in each graph and the average results graph of these groups.

Figure 9. Performance comparison of proposed ICF_Chunking technique with baseline methods (VF2E) for a group of three users in each graph and the average results graph of these groups.
Figure 10. Performance comparison of proposed CBF_Chunking technique with baseline methods (VF2E) for a group of three users in each graph and the average results graph of these groups.

Figure 11. Performance comparison of proposed Hybrid_Chunking technique with baseline methods (VF2E) for a group of three users in each graph and the average results graph of these groups.
Figure 12. Comparison among our proposed techniques (Chunking) with baseline methods (VF2E) on average results of three groups (combined). The blue, red, yellow, and green lines show performance score of the recommendation techniques UCF, ICF, CBF, and Hybrid, respectively.

5. Conclusions

We compared our proposed method’s performance, and its recommendation set results with the base paper findings in our experiments. We used the same data structure of knowledge graph in our approach and similarity measures, like user-to-user, service-to-service, and keyword-to-keyword similarities. The average calculated scores of the user groups on 80 recommendations for the baseline methods UCF_VF2E, ICF_VF2E, CBF_VF2E were 26.56%, 28.87%, 16.96%, 27.44%, respectively, and the results of our proposed methods UCF_Chunking, ICF_Chunking, CBF_Chunking, and Hybrid_Chunking outperformed the others, achieving the scores of 33.59%, 30.78%, 34.68%, 36.34%, respectively. Therefore, we conclude that the results of our proposed methods are better in all formats for recommendations, whether it is for individual users or groups of users, and all four proposed chunking methods remained at the top in the scores graph. In the future, we may introduce new pruning rules for chunking to improve the relations among entities in a knowledge graph. For now, we used a specific pattern of words in the review to be extracted as a chunk from the text or review. We extract the noun phrases as keywords from the reviews using the mentioned rules in Section 3. We can apply different combinations of such POS words to compare the results in the future. We can also apply different similarity measures based on the data of the features and can compare our proposed method on other domains like ‘food’, ‘hotels and travel’, etc. We may determine what pattern of POS chunks significantly impact the recommendations depending on categories to give better and more personalized recommendations.


Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Researchers Supporting Program (Project number RSP-2021/314), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.
References


