DEVELOPING A TURKISH-LANGUAGE RECOMMENDATION SYSTEM BASED ON USER CONVERSATIONS

MURAT ELİFOĞLU

BOĞAZİÇİ UNIVERSITY

2018

DEVELOPING A TURKISH-LANGUAGE RECOMMENDATION SYSTEM BASED ON USER CONVERSATIONS

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Murat Elifoğlu

Boğaziçi University

2018

DECLARATION OF ORIGINALITY

I, Murat Elifoğlu, certify that

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all sources of ideas and words, including digital resources, which have been produced or published by another person or institution;
- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature ....

Date ......05.11.2018......................

ABSTRACT

Developing a Turkish-Language Recommendation System

Based on User Conversations

Recommendation systems have recently been used in many fields. This study describes a restaurant recommendation system which is developed specifically for data that is collected from chat messages typed in Turkish. The proposed system aims to recommend best matching places to a group of users in a chat environment analyzing their conversations. In order to achieve this goal, a rule-based approach which composes of normalization, analysis and recommendation steps has been designed and implemented. Furthermore, an explanation module used for explaining why the system recommends selected places has been added. The system benefits from two data sources that are property data source and restaurant data source and a rule base. While the property source is a dataset contains features related to restaurant domain, the restaurant source has all places that can be recommended by the system. On the other hand, the rule base is a sequence of rules defined manually to extract information from chat messages in a more accurate way. The evaluation process of the system has been very difficult since no test data are available. To evaluate the system, both restaurant data source and chat messages are simulated manually.

ÖZET

Kullanıcı Yazışmalarına Dayalı Türkçe Öneri Sistemi Geliştirilmesi

Günümüzde öneri sistemleri birçok alanda kullanılmaktadır. Bu çalışma Türkçe dilinde yazılan sohbet ortamı mesajlarına yönelik özel olarak tasarlanmış bir öneri sistemi tanıtmaktadır. Önerilen sistem, sohbet ortamlarındaki grup yazışmalarını analiz ederek kullanıcılarına en iyi eşleşen önerileri sunmayı amaçlamaktadır. Bu hedefi gerçekleştirmek adına normalizasyon, analiz ve öneri aşamalarından oluşan kural tabanlı bir yaklaşım tasarlanmış ve uygulanmıştır. Ayrıca, sistemin neden seçilen yerleri önerdiğini açıklayan bir açıklama modülü eklenmiştir. Sistem, nitelik veri kaynağı ve restoran veri kaynağı adında iki veri kaynağı dışında bir de kural tabanından faydalanmaktadır. Nitelik veri kaynağı, restoranlarla ilgili özellikleri barındıran bir veri kümesi iken; restoran veri kaynağı, sistem tarafından önerilecek restoranları barındırmaktadır. Kural tabanı ise, sohbet ortamı mesajlarından daha doğru bir şekilde bilgi çıkarmak amacıyla elle tanımlanmış kurallar dizisidir. Test verisi konusunda yaşanan sıkıntı nedeniyle, sistemi değerlendirme süreci oldukça zorlu oldu. Sistemi değerlendirmek amacıyla, hem restoran veri kaynağı, hem de sohbet ortamı mesajları simule edildi.

ACKNOWLEDGEMENTS

In memory of Prof. Osman N. Darcan.

First and foremost, I would like to thank my supervisor, Prof. Tunga Güngör, for his complete guidance during my thesis study. His solid knowledge regarding the field of my study enlightened my way.

I am also very grateful to my thesis advisor, Prof. Meltem Özturan, since she accepted my thesis subject and directed me right.

Also, I would like to thank Prof. Birgül Kutlu Bayraktar, Assoc. Prof. Bilgin Metin, and Assoc. Prof. M. Fatih Amasyalı since they accepted to participate in my thesis committee and spent the time to read my study.

I would like to send special thanks to my father, mother and aunt for their endless love and unlimited support for me since I was born. Thanks to each of them, I have been a successful student as well as a man with a good character. I am also very grateful to my brother, Mert Elifoğlu, just for being my brother.

Finally, I am most grateful to my wife, Merve Elifoğlu. For everything.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Computer scientists have been studying to create computer systems that will ease people's daily stuff since artificial intelligence researches started. In earlier times, they had tried to implement systems having abilities in all areas, but they failed to achieve it. After they understood that that strategy does not work, they decided to design expert systems capable of doing specific things. Thanks to rapid development in the computer world, expert systems designed to help people for their basic needs in their daily lives have started to be adopted by their users. Nowadays, recommendation systems, a kind of expert systems, have been used by many people in many fields such as e-commerce, financial services, translation systems, etc.

1.1 Recommendation systems

A recommendation system, also known as recommender system or recommending system, generally produces a list of recommendations after analyzing some inputs coming from users in real time or being gathered from data sources. The main purpose of a recommendation system is to give the best options to its users based on their previous preferences as well as current ones. In addition to its user's preferences, the system commonly benefits from the historical data collected from other users of the system.

Three main approaches which are collaborative filtering, content-based filtering, and knowledge-based filtering are generally used to build a recommendation system. While collaborative filtering approaches create a model using users' past behaviors, content-based filtering techniques try to analyze contents with similar characteristics and select recommendations based on them. On the other hand, knowledge-based approaches generate knowledge models from users' activities and use them to make recommendations. In the literature, researchers have conducted some hybrid studies by combining all these approaches as well.

1.1.1 Collaborative filtering

Collaborative filtering, which is used by earlier recommendation systems, is one of the oldest and most common approaches. Although it is applicable in many areas, it is generally more adopted by e-commerce websites. As shown in Figure 1, user preferences and community data are two inputs of collaborative recommendation system. In the system, different types of algorithms may be applied to make recommendations. Finally, the output of the system is a recommendation list that contains calculated scores of each recommendation. In brief, a collaborative filtering can be defined as a system that tries to recommend popular items among users having similar characteristics. Since the collaborative filtering method is independent of context which means item features in terms of collaborative filtering, it is capable of recommending complex items without the need of understanding them. However, there are a few drawbacks to this approach. The most important one is the cold start problem which means the system requires a large amount of data so that it can make recommendations. The other disadvantages can be considered as scalability and data sparseness problem.

1.1.2 Content-based filtering

Content-based filtering, another common approach when building recommendation systems, is a technique based on user preferences and the features items have. In

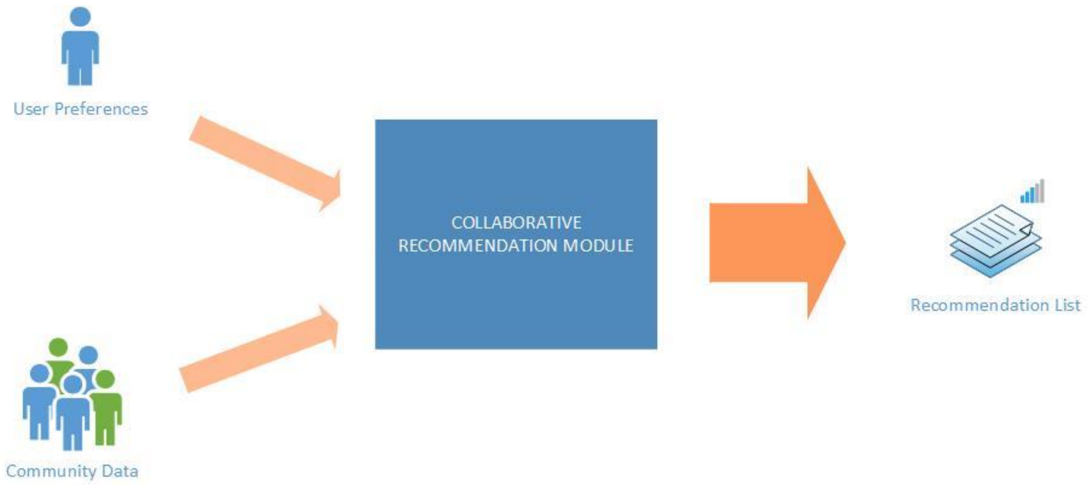

Figure 1. Collaborative recommendation system

content-based filtering, items that are similar to those that same user is interacting or interacted in the past are tried to be recommended. Figure 2 shows that how a content-based recommendation system works. User profile which composes of user's past behaviors and current preferences is an input of the system as features of items that reflect the characteristics of those are. After analyzing the inputs, the system gives the recommendation list as the output. As content-based techniques do not require a user community, they can be used even if only one user exists in the system. In the systems that content is very limited, using content-based approaches may not be comprehensive and it is not generally suggested to be used.

1.1.3 Knowledge-based approach

A knowledge-based recommendation system is a type of recommendation system that uses knowledge about and items to define a knowledge model that will be used to generate recommendations. It generally tries to make recommendations based on the question which items meet the user's requirements best. Figure 3 demonstrated

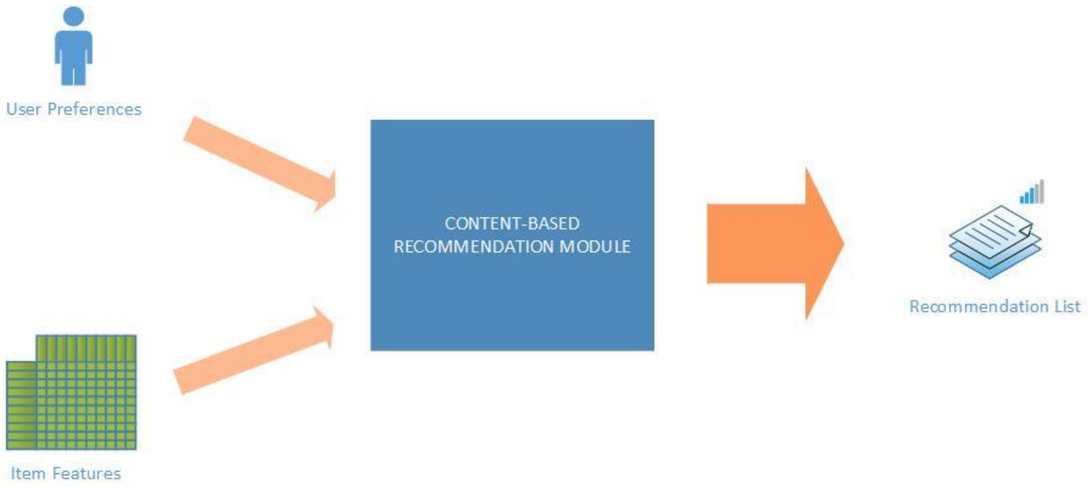

Figure 2. Content-based recommendation system

that a knowledge-based recommendation system can be considered as a content-based recommendation system which is extended to use knowledge models to give results more precisely. Since knowledge-based approaches require a bit more effort than other approaches in terms of knowledge extraction, representation, and design, it is harder to design and implement a knowledge-based system compared to collaborative or content-based ones. On the other hand, the accuracy of knowledge-based models is usually higher rather than others.

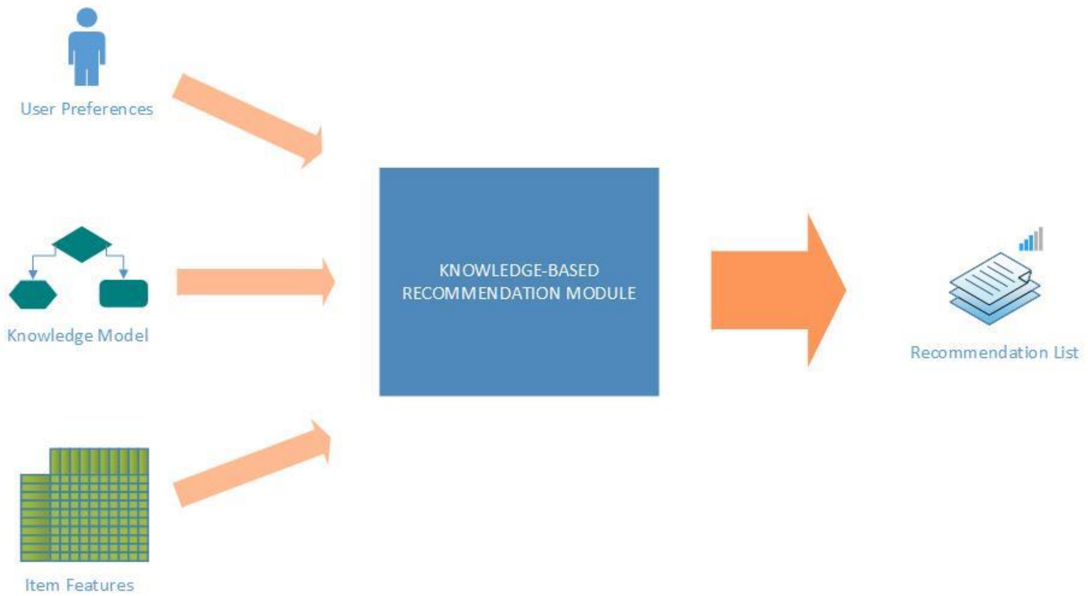

Figure 3. Knowledge-based recommendation system

1.1.4 Hybrid approach

Hybrid approaches, a composition of collaborative, content-based and knowledge-based approaches, is used to make recommendations more effectively in many cases. Hybrid recommendation systems accept multiple types of inputs and design a hybrid model by analyzing them, as shown in Figure 4. The hybrid model can be designed using different techniques such as merging all approaches that will be benefited from into a compact model, parallel use of them or pipelining them into a sequence. While hybrid recommendation systems generally provide more accurate recommendations than pure approaches, it can be very complex and time-consuming to build them.

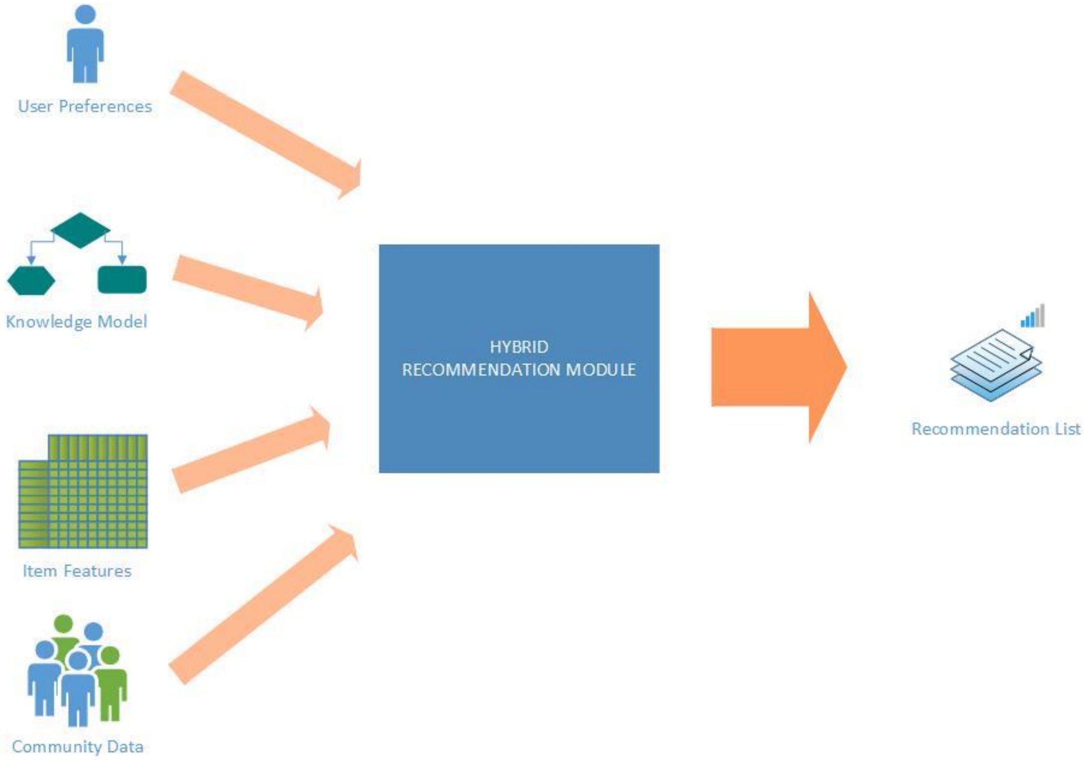

Figure 4. Hybrid recommendation system

1.2 Thesis statement

The main objective of this study is to build a restaurant recommendation system that analyzes text-based user conversations and recommends best-matching restaurants based on users' preferences. The study focuses on only text messages that are written in the Turkish language. The system is built for a group of users who may have different ideas and preferences. The study proposes a score-based method that benefits from Natural Language Processing techniques.

1.3 Organization of the thesis

The structure of the thesis is built from the following chapters. Chapter 2 conducts a literature review in recommendation systems field in terms of both English and Turkish studies. Chapter 3 proposes a methodology for a recommendation system used for chat groups to help them find restaurants. Chapter 4 presents an evaluation of the proposed system. Finally, last chapter, Chapter 5, contains a conclusion for the study and ideas for the future works.

CHAPTER 2

LITERATURE REVIEW

Researchers have been conducting many studies on recommendation systems for many years. Although an important proportion of these studies is for English based recommendation systems, in the literature, there are some researches for other languages as well. On the other hand, when recommendation system studies are classified by their domains, it can be easily seen that the number of restaurant domain related works is very low.

2.1 Approaches for recommendation systems

Having looked at the collaborative side of recommendation system researches, it is seen that many studies contributed to the development of them. Goldberg, Nichols, Oki, and Terry (1992) conducted a study that benefited from collaborative filtering in addition to content-based techniques. In that study, Goldberg et al (1992) developed a system that recorded reactions of users while they were reading e-mails and helped other users using those reactions on filtering. Proposed method in the study required that people had to know each other and it did not benefit from user ratings. Resnick, Iacovou, Suchaki, Bergstrom, and Riedl (1994) used rating functionality in their study. In collaborative filtering, probabilistic approaches are also important. Breese, Heckerman, and Kadie (1998) proposed a cluster-based approach that split users into a number of clusters using similarity vectors and statistical Bayesian models.

For content-based recommendation systems, there are also specific studies in the literature. Meteren and Someren (2000) described a content-based recommendation system that makes recommendations in a domain where the user model is very dynamic. Content-based recommendation systems are generally used in a variety of domains such as recommending news, products, restaurants, movies, etc. (Pazzani & Billsus, 2007, p. 325). According to Pazzani and Billsus (2007), many different algorithms may be used to learn user profile and should be selected depending on the representation of content (p. 339). As mentioned before, in some content-dynamic domains, collaborative methods suffer from the cold start problem. Oord, Dieleman, and Schrauwen (2013) tried to resolve this problem by proposing a latent factor model for music recommendation area. Oord et al (2013) showed that accurate recommendations can be made using the latent factor model that predicts latent factors using music audio.

In the literature, there are also many studies combining different types of approaches. Claypool et al (1999) presented a filtering approach that merges the depth of collaborative filtering and the coverage of content-based filtering. Claypool et al (1999) showed that having combined, filtering techniques can result in more accurate recommendations than having used standalone. About 2 years later, Popescul, Ungar, Pennock, and Lawrence (2001) proposed a method of unifying collaborative and content-based approaches. Using that method, Popescul et al (2001) achieved to increase the flexibility and quality of the recommendation system when data are extremely sparse. Furthermore, Yoshii, Goto, Komanati, Ogata, and Okuno (2006) presented a recommendation system that uses ratings, user preferences and item features. In that approach, Yoshii et al (2006) benefited from statistical Bayesian model estimating relations between users and contents statistically. Another study combining two approaches was conducted on social network analysis (Debnath, Ganguly and Mitra, 2008). Debnath et al (2008) proposed a hybrid approach to weight features of items computing weights thanks to a regression analysis on a collaborative social network.

2.2 Researches for the Turkish language

In the literature, the number of recommendation system studies for the Turkish language is very insufficient. Hence, papers that belong to subfields of natural language processing have been reviewed so that natural language processing approaches which are proposed for similar problems such as information extraction, question answering, and concept mining can be learned.

Sevli and Küçüksille (2016) developed an analysis and recommendation system benefiting from social networks for users who use the Turkish language. They claimed both collaborative and content-based recommendation systems miss the fact that users' behaviors may change. In order to handle that case, Sevli and Küçüksille (2016) proposed a method of extracting users' attitudes dynamically and suggesting items in real time. Gathering information from a data source like chat messages is a very hard subject in terms of natural language processing as these types of data are generally noisy which means they usually contain some words written in an ungrammatical form such as mistyped, without punctuation marks and with special characters like emotion symbols. Özyurt and Köse (2006) tried to extract gender information of users who message to each other on chat environments using a statistical and semantic approach. In order to make a meaningful analysis of data, detecting relations between the words is very important as well. Yazıcı and Amasyalı (2011) conducted a study about extracting semantic relationships between concepts automatically. They achieved to extract many types of relations such as synonyms, antonyms, sub-super relations, etc. Furthermore, Tatar (2011) presented automatic rule learning methods for the tasks which are named entity recognition and entity relation recognition. However, due to the lack of studies for Turkish in those fields, the study could not be evaluated properly.

2.3 Studies in restaurant domain

Studies which are specific to the restaurant domain have also been conducted for years. Dwivedi and Chheda (2012) proposed an architecture using a hybrid approach formed by a combination of collaborative filtering and knowledge-based filtering. The architecture they proposed used users' history, similar users' patterns and cuisine and location preferences as shown in Figure 5.

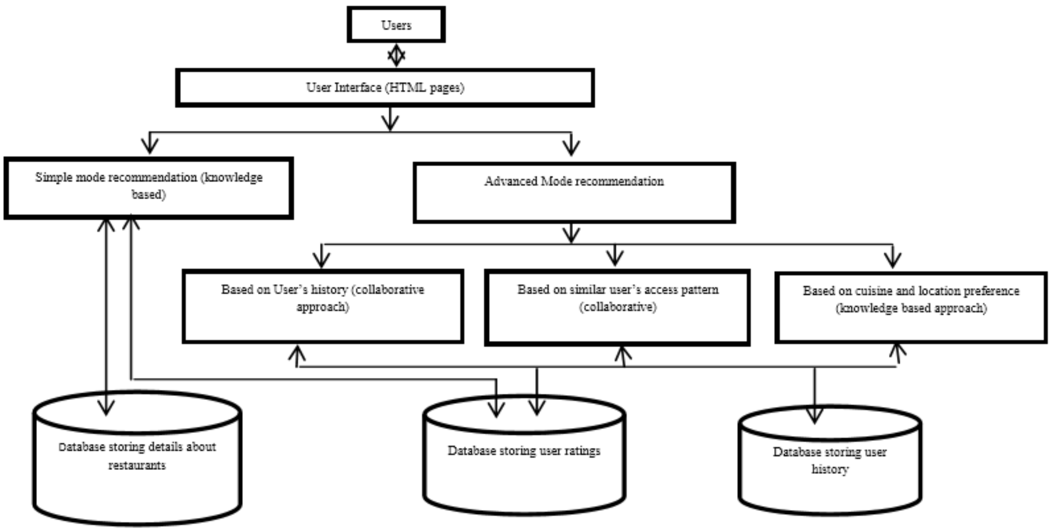

Figure 5. The architecture of the restaurant recommender system proposed by Dwivedi et al.

Thompson and Göker (2000) published their study which describes a conversational place recommendation system. In the proposed model, user preferences were gathered during the conversation in an iterative way until the number of recommendations was suitable. The text below shows a sample dialog between the recommendation system and a user who tries to find a place to have dinner (Thompson and Göker, 2000).

Inquirer: Where do you think I should eat tonight?

Advisor: Well, what type of cuisine would you like?

Inquirer: What types are there?

Advisor: Some examples are Chinese, Indian, and Mediterranean.

Inquirer: Oh, maybe Indian.

Advisor: What quality rating would you like?

Inquirer: I don’t care, as long as it’s cheap.

Advisor: How do you want to pay?

Inquirer: No, I think I’d like Chinese instead.

Advisor: Okay, we’ll switch to Chinese food. What city do you prefer?

Inquirer: How about Berkeley?

Advisor: I know three cheap Chinese restaurants in Berkeley. One is the Long Life Vegi House on 2129 University Avenue. Does that sound alright?

Inquirer: Sure, that sounds fine

Furthermore, there is a study that tries to make recommendations to a group of users instead of individuals. McCarthy (2002) proposed a new recommendation system that focuses to help a group of people who searches for restaurants based on their preferences. The proposed system uses both restaurant and user data source and it runs a group preference algorithm based on scores of each feature for each user.

CHAPTER 3

METHODOLOGY

The thesis study has been split into several parts in order to build the most convenient method for recommendation system problem in a chat environment step by step. First, sample data were collected from chat environments. After gathering the data, features that may be useful for restaurant recommendation process were extracted. Then, a content-based model was proposed based on extracted features. These two last steps recurred until the model was fit enough to solve the problem.

3.1 Data collection

Since the main subject of the thesis study aims to a very specific area in recommendation systems field, finding sample data has been very hard. The specific goal of the study is helping user groups that only use chat environments to find restaurants or places like restaurants. For conducting the study, a small set of sample data were able to be collected from friends as well as some data were simulated manually. For instance, the following group chat messages are sample inputs that are used by the system.

Kızlar bugün ne yapiyoruz?	(What are we doing today, girls?)
Hava cok soguk yaa	(The weather's really cold!)
Geçen hafta planladığımız gibi olsun işte	(Let's do what we planned last week)
Evet	(Yes)
Kapalı biyer olsun	(Let it be somewhere inside)
Hımmm	(Hımmm)

Abant tarafı çok soğuktur	(Abant side is very cold)
Yaaa ama yeşil eev	(But yeşil eev)
Bolu merkezde takılalim	(Let’s hang on Bolu center)
Hani gidecektik Yeşil Ev'e	(We would go to Yeşil Ev)
Yağmur var şimdi	(It’s raining now)
Yeşil ev çok güzel ama soğuk	(Yeşil ev is very beautiful, but it’s cold)
Pasta yiyelim	(Let’s eat cake)
Benim canım tiramisu çekti	(I want tiramisu)
Nerde vardır?	(Where can we find it?)
Offf o zaman ve kahveninki de çok güzel	(Then, ve kahve’s is very delicious)
Kübraa cok iyi fikir	(Kübraa that’s a great idea)
Ve kahvede sahlep içelim hadi hazırlanın	(Let’s drink sahlep at ve kahve, suit up)
Özledim Zaten ve kahveyi	(I’ve already missed ve kahve)
Ama ben doğal bi ortam istiyorum	(But I want a natural place)
Sigara içmelik	(Where I can smoke)
Açık alan	(Outdoor)
Sahlep mi	(Sahlep?)
Çimler falan	(Also grasses)
Şöyle otantik bir yer olsa keşke	(Authentic place though)
Tahta masalar	(Wooden tables)
Canlı müzik de olabilir ya	(Live music is also a good idea)
Tamam ozaman açık alan otantik çok soğuk olmayan biyer	(OK then, a place where is outdoor, authentic and not very cold)
Hadi bulalım	(Let’s find it)

3.2 Feature extraction

Having collected, sample data were examined to extract features that belong to the restaurant domain. In addition to analysis of sample data, websites related to restaurant domain like food ordering services were reviewed in order to detect other features which may be useful for restaurant recommendation system. As a result of this data collection and analysis process, features that will be inputs of the system are decided. The extracted features were split into three main groups which are standard properties, no-notset-yes properties, and notset-yes properties. Properties such as cuisines, locations, meals can be categorized as standard properties. Each standard property composes of sub-properties as shown in Table 1.

Table 1. Standard Features of Restaurant Domain

Standard Features	Possible Values for Standard Features
Cuisine - Mutfak	Turkish - Türk French - Fransız
Place – Mekan	Restaurant – Restoran Hotel - Hotel
Environment – Ortam	Natural – Doğal Historical - Tarihi
Meal – Öğün	Breakfast – Kahvaltı Lunch – Akşam yemeği
Foodservice – Servis	Open buffet – Açık büfe Self-service – Self servis
Price – Fiyat	Cheap – Ucuz Expensive - Pahalı
Location – Konum	Sarıyer - Sarıyer

On the other hand, no-notset-yes properties may accept three types of values based on user preferences. When users request that property, its value is set to “yes“. If they do not request it, the value of the property becomes “no”. Finally, if the property is not mentioned in the chat messages, its value takes “not-set” value. Properties such as live music and alcohol may be categorized as this type of properties. As it may be understood from its name, notset-yes property, which cannot have “no” values, is a subset of no-notset-yes property. Table 2 demonstrates all extracted features with their Turkish translations.

Table 2. Extracted Features for Restaurant Recommendation System

Standard Features	No-NotSet-Yes Features	NotSet-Yes Features
Cuisine - Mutfak Place - Mekan Environment - Ortam Meal - Öğün Foodservice - Servis Price - Fiyat Location - Konum	Cigarette - Sigara Alcohol - Alkol Seaside – Deniz kenarı Outdoor – Dış mekan	Live music – Canlı müzik Car park - Otopark Fasıl - Fasıl Fix menu – Fix menü View - Manzara Group discount – Grup indirimi Public transport – Toplu taşıma Organization - Organizasyon İftar - İftar Terrace - Teras Garden - Bahçe Indoor – İç mekan

3.3 Proposed model

The proposed model for the solution is a content-based restaurant recommendation system which takes chat messages as input and gives recommended restaurants and an auto-generated message explaining why it recommends selected restaurants. The very simple design of the proposed model can be seen in Figure 6.

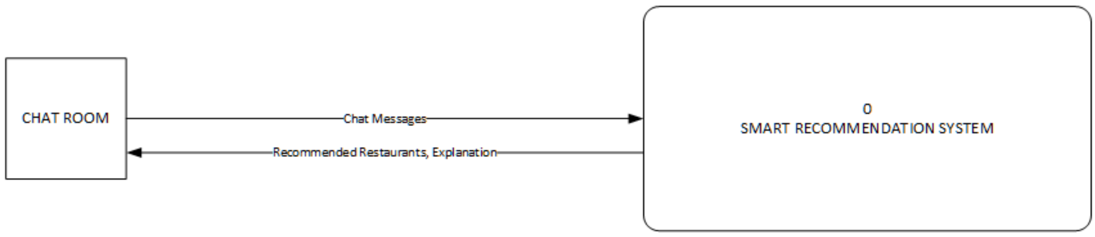

Figure 6. The simple design of the proposed model

The proposed system has been built on mainly three modules. First, the normalization module is used to normalize chat messages so that they can be analyzed easily. Analysis module which analyzes normalized messages and applies some rules to them is another important module of the system. Finally, the recommendation module is the key part of the system. It makes recommendations using restaurants data and defined rules as well as it produces an explanation why it selects recommended restaurants. These three modules are abstracted using a manager module which is a bridge between the chat room and main modules of the system. Furthermore, the system benefits from a rule base, a normalization service and two different data sources named restaurant data source and property data source. While restaurant data source stores restaurants and their available features in the system, the property data source composes of the extracted properties used by the system. Figure 7 shows the architecture of the proposed model.

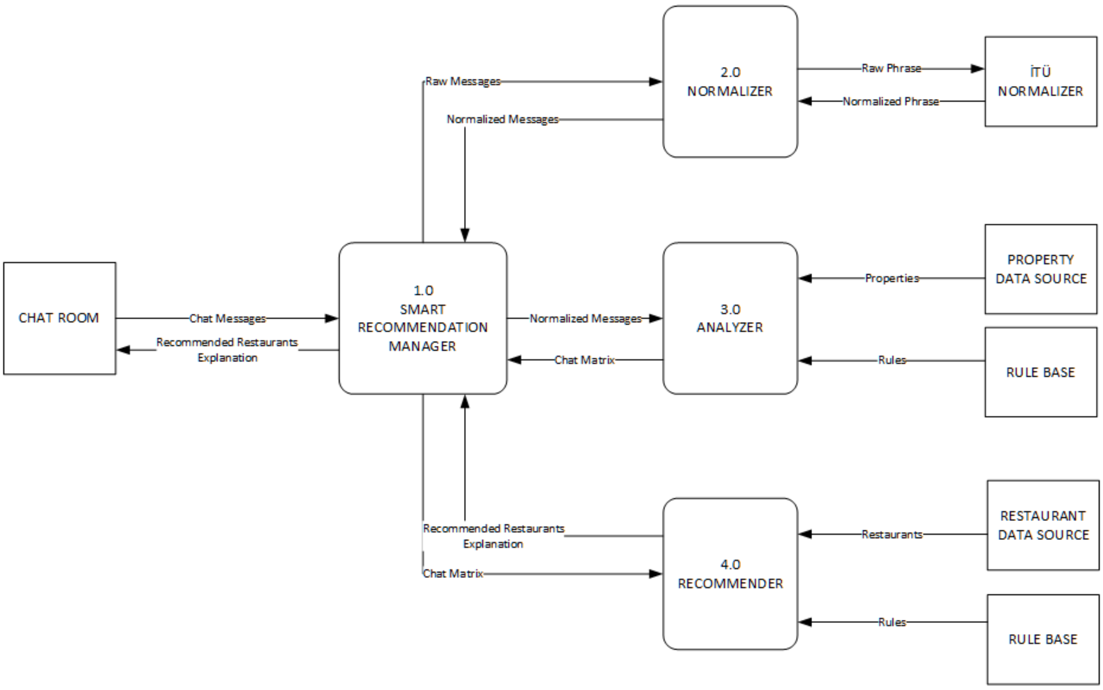

Figure 7. The architecture of the proposed model

3.3.1 Normalization module

Since data collected from chat environment are very noisy, which means written mostly in an ungrammatical form such as removed vowel letters, used numbers instead of letters, ignored some letters for ease or typing, a normalization process that will be applied on input messages is a must for the proposed model. The normalization module, designed for this purpose, uses an external normalization service for Turkish provided by Natural Language Processing Group at Istanbul Technical University (Eryiğit, 2014). ITU Turkish Language Processing Pipeline has different modules such as tokenizer, vowelizer, spelling corrector, morphologic analyzer etc.

Adalı and Eryiğit (2004) proposed a language independent hybrid model composes of a discriminative sequence classifier and a language validator for normalization of social media texts. They focused on two important problems of normalization which are diacritization and vowelization. The following sentence has two possible different meaning in terms of diactirization (Adalı and Eryiğit, 2014).

"Ruyamda evde oldugunu gordum."

Meaning 1	Meaning 2
Rüyamda evde *olduğunu* gördüm.	Rüyamda evde *öldüğünü* gördüm.
I had a dream that you *were* at home.	I had a dream that you *died* at home.

Adalı and Eryiğit (2014) also mentioned vowelization causing much more complexity rather than diacritization in that study. Vowelization means predict the complete form of given word which is written vowel reduced form. The example below shows how complex a vowelization problem might be.

“slm”

Meaning 1	Meaning 2	Meaning 3	Meaning 4	Meaning 5
selam	salam	sulama	salım	sılam
hi	salami	watering	my raft	my furlough

The normalization module of the proposed system uses the tool that is built based on the result of that study, as well. It takes raw chat messages as inputs and gives normalized chat messages as outputs using diacritization and vowelization operations.

3.3.2 Analysis module

Given normalized chat messages, analyzer produces a chat matrix that represents selected features in that chat room. It collects possible features from property data source and creates a matrix using them. The chat matrix C is a 2-column matrix where the first column shows possible features taken from property data source and the second column demonstrates values of them based on users’ preferences. The chat matrix can be represented as follows:

$$C = \begin{bmatrix} f_1 & v_1 \\ f_2 & v_2 \\ f_3 & v_3 \\ \cdots & \cdots \\ f_{m-1} & v_{m-1} \\ f_m & v_m \end{bmatrix}$$

The chat matrix is initialized with neutral values, which means value column is set to zeros as follows:

$$C = \begin{bmatrix} f_1 & 0 \\ f_2 & 0 \\ f_3 & 0 \\ \cdots & \cdots \\ f_{m-1} & 0 \\ f_m & 0 \end{bmatrix}$$

For the chat matrix, positive values mean positive attitudes to the features while negative values show that users do not request those features. On the other hand, zero-valued columns represent either not mentioned features or the features which requested and not requested equally.

After chat matrix is initialized, its values should be set based on user preferences. In order to achieve that, user preferences should be matched to features taken from property data source. This matching operation is applied with the help of two sub-modules which are root finder and negation checker as well as rules collected from rule base. Root finder aims to find the best matching word root of each word typed in chat messages. Zemberek, which is an NLP library for the Turkish language, is used to find word roots (Akın and Akın, 2007). On the other hand, the main goal of negation checker is to check whether each sentence has a positive or negative attitude. Based on this check, values in chat matrix increase or decrease for each occurrence of that property in chat messages. Different approaches are used for two different types of sentences. For name clauses, a keyword-based algorithm is used. According to this algorithm, if a sentence contains at least one of defined negation words, its attitude is negative. Or else, its attitude is set as positive. However, Zemberek is used for verb clauses as understanding that verb clauses have either the positive or negative attitude is more complex comparing to name clauses.

Here, *name clause* means a sentence with a predicate which is a name while *verb clause* means a sentence with a predicate that contains a verb. For the Turkish language, the negative form of a name clause can be made only using a specific word which is *değil*. On the other hand, verb clauses can be negated using the suffixes that are *–me*, *-ma*.

Having looked, the relations between features play a very important role for the analysis phase. Features that are stored in the property data source have relations such as synonymy and hyponymy. Synonymy, which is a symmetrical relation, means a relation between two words that have equivalent meanings. For instance, the words *brave* and *courageous* are synonyms. On the other hand, hyponymy is an unsymmetrical relation and it shows a relationship between a generic term and a specific instance of it. Having examined, the sentence *Red is a color* is considered as an example of hyponymy. Using hyponymic relationships, some standard properties are designed as tree-based features. Figure 8 shows a sample food tree that composes of food types and the relationships between them. According to the food tree, it can be easily seen that *pizza* is a *fast food*. All hyponymic relations used in the study can be shown in Figure A1, Figure A2, and Figure A3 (Appendix A) for food, dessert, and drink type hyponyms.

Features and relations between them are stored in the property source. Each feature has a *Synonyms* field, an array data structure, whose elements are similar words that can be used instead of the actual feature in the chat environment. This is a one-directional structure, so elements in the list do not have to be in the property source themselves. On the other hand, hyponymic relations are considered as a tree structure in the data source.

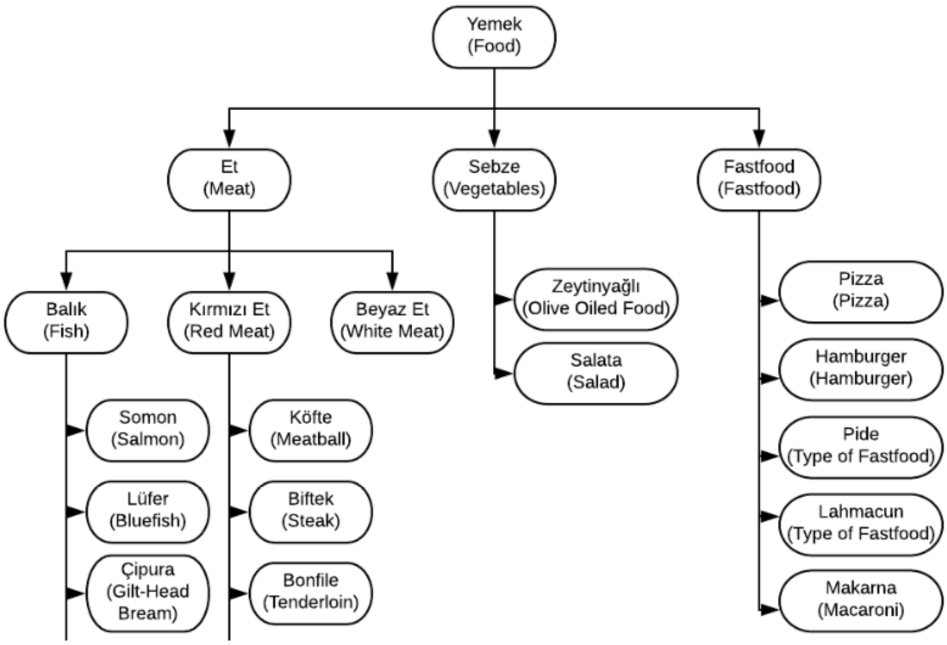

Figure 8. Sample hyponymic relations between food types

Child features have a *ParentKey* property that links the feature to its parent. However, the value in the parent key field has to be another instance of the data source in contrast to the elements of synonyms list. Finally, a single entity in the property source also contains a *Word* property which stores usage of a feature in the Turkish language, and a *Category* property that holds the type of that feature. A very little snapshot of the property data source can be shown in the following example.

```
[
  {
    "Word": "türk",
    "Category": { "Type" : 1, "Name" : "Cuisine" },
    "ParentKey": null,
    "Synonyms": [ "osmanlı" ]
  },
  {
    "Word": "fransız",
    "Category": { "Type" : 1, "Name" : "Cuisine" },
    "ParentKey": null,
    "Synonyms": [ "fransa" ]
  },
```

```
{
   "Word": "kırmızı et",
   "Category": { "Type" : 2, "Name" : "Food" },
   "ParentKey": "et",
   "Synonyms": [ ]
 },
{
   "Word": "kebap",
   "Category": { "Type" : 2, "Name" : "Food" },
   "ParentKey": "kırmızı et",
   "Synonyms": [ ]
 }
]
```

In addition to synonym and hyponym relations between features, a rule base is also used for the analysis module. The rule base contains a few semantic rules which are used to extract a feature from a sentence. For instance, the sample rule below is a rule that is applied to a sentence to extract *car park* feature.

A sentence contains the word “car (araba)” refers to the feature “car park (otopark)”.

I will come by car. --> I need a place having a car park.

Arabayla geleceğim. --> Otoparkı olan bir mekan olsun.

Figure 9 shows all mentioned sub-modules and data sources as a workflow diagram. Thanks to all these sub-modules, values of chat matrix can be decided iteratively based on weights that are calculated based on property relations.

3.3.3 Recommendation module

Recommendation module, the final module in the pipeline, uses two sub-modules. The first one is used for calculating similarity scores between chat matrix and restaurant matrix, and the other one is for generating explanations for why the system recommends selected restaurants.

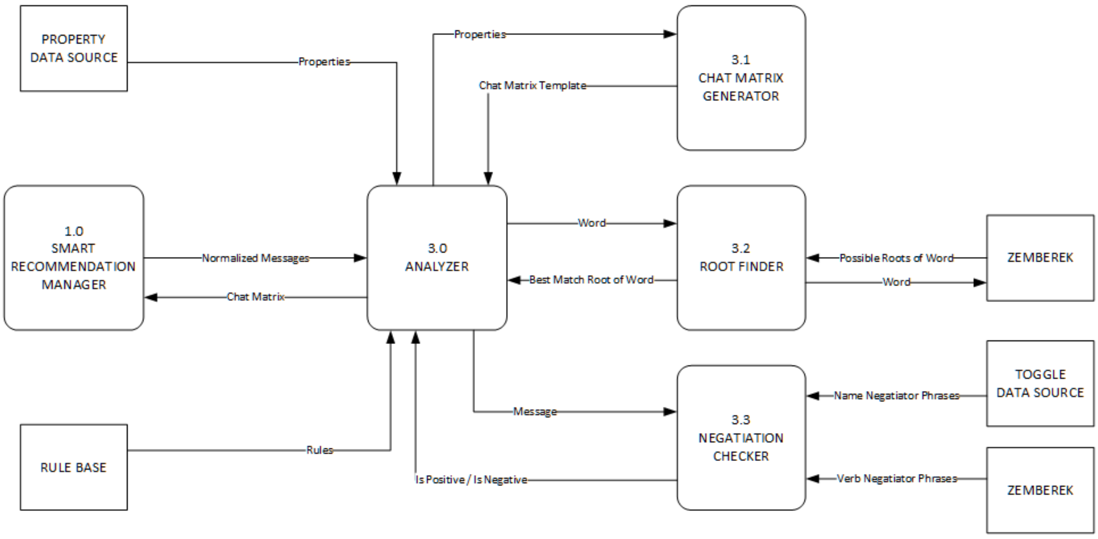

Figure 9. Data flow diagram of the analysis module

The restaurant matrix is very similar to the chat matrix in terms of its structure. It contains property values for each restaurant in the system. Features of each restaurant are stored in restaurant data source with its name and description. The object below can be shown as an example of a single restaurant entity.


```
{
    "Id": 1,
    "Name": "Name comes here",
    "Description": "Description comes here",
    "Cusines": [ "türk", "osmanlı" ],
    "Places": [ "hotel" ],
    "Environments": [ "doğal", "tarihi" ],
    "Meals": [ "kahvaltı", "öğle yemeği", "akşam yemeği" ],
    "FoodServices": [ "alakart", "açık büfe" ],
    "Prices": [ "pahalı" ],
    "Locations": [ "sultanahmet", "fatih" ],
    "Others": [ "sigara", "deniz kenarı", "otopark", "manzara", "toplu taşıma",
                "organizasyon", "iftar", "teras",   "kapalı alan" ],
    "Foods": [ "köfte", "bonfile", "beyaz et", "sebze" ],
    "Desserts": [ "trileçe", "sütlaç", "baklava", "kadayıf", "dondurma" ],
    "Drinks": [ "şarap", "rakı", "soğuk içecek" ]
}
```

For each restaurant, a matrix, which is very similar to the chat matrix, is generated mapping features on a single restaurant entity to a feature vector. As a result, after the feature vector is built for each restaurant, they are combined to create a restaurant matrix as shown below.

$$R = \begin{bmatrix} f_1 & v_{11} & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & v_{1(n-1)} & v_{1n} \\ f_2 & v_{21} & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & v_{2(n-1)} & v_{2n} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ f_{m-1} & v_{(m-1)1} & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & v_{(m-1)(n-1)} & v_{(m-2)n} \\ f_m & v_{m1} & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & v_{m(n-1)} & v_{mn} \end{bmatrix}$$

Here, the value of each cell is calculated using the same algorithm which is used for each iteration of the chat matrix generation process. Having created, the restaurant matrix is used for similarity calculation as well as the chat matrix. The two matrices are compared to each other over their first column which holds feature information. To make this comparison, a dot product operation is applied to chat matrix and restaurant matrix. The value column of chat matrix is produced by the value column of each restaurant in restaurant matrix in terms of dot production. Each dot production is accepted as the similarity score of that restaurant. For each restaurant R_i, the similarity score S_i is calculated with the formula below.

$$S_i = \vec{C}_1 . \vec{R}_i$$

Using this formula, the similarity score for each restaurant R_i, *i* is between *0* and *(n-1)* where *n* is the number of restaurants. Having looked, the formula can be shown implicitly as follows. In this formula *m* is the number of features which is supplied from property source.

$$S_i = \sum_{j=0}^{m-1} c_j\, r_{ji} = c_0\, r_{0i} + c_1 r_{1i} + c_2 r_{2i} + \cdots + c_{(m-2)}\, r_{(m-2)i} + c_m\, r_{mi}$$

After similarity score for each restaurant is calculated, restaurants are sorted by their scores in a descending order. In order to have more comparable scores in terms of end-user perspective, similarity scores are normalized between *0* and *100* using normalization formula below.

First, scores are normalized to [0,1] range with classical normalization formula:

$$y_i = \frac{x_i - min(x)}{max(x) - min(x)}$$

Here, x_i represents values in S_i similarity vector. While *min(x)* shows the minimum score, *max(x)* shows the maximum score. Finally, y_i is the *[0, 1]* normalized form of x_i. Converting this normalized value into *[0, 100]* range requires multiplying the value by *100* as shown in the formula below. After applying this formula, we get the result n_i which is *[0, 100]* normalized value.

$$n_i = y_i * 100$$

Explaining why the system chooses given recommendations is a very important topic in recommendation systems since users wonder about it. A basic explanation module was designed to cover this issue. The explanation module uses the chat matrix and the restaurant matrix as the similarity score calculation module does. Turkish sentence templates are merged with meaningful properties for that chat to create explanation sentences. Sentences are chosen from templates randomly based on property types. For instance, the sentence below is an explanation template that can be used on cuisine property.

Tastes from {PropertyName} cuisine are served here as you requested [Adverb].

[Adverb] {PropertyName} mutfağından lezzetler bu mekanda yer almaktadır.

Here, *[Adverb]* part is replaced with an adverb which is chosen based on the difference between chat and restaurant matrix values. An empty string replacement is also available here, which means ignoring *[Adverb]* part of the sentence. On the other hand, *{PropertyName}* part in this example can only be replaced with cuisine types such as Turkish (*Türk*), Chinese (*Çin*), etc. Appendix B lists all explanation templates that are used in the explanation module of the system.

Figure 10 shows the entire structure of the recommendation module as a data flow diagram.

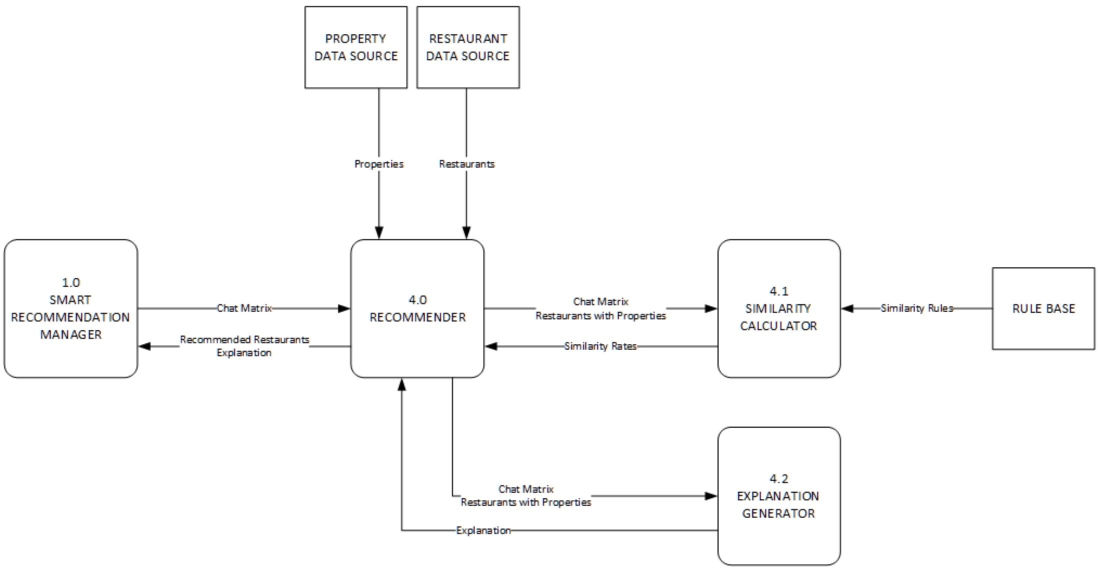

Figure 10. Data flow diagram of the recommendation module

CHAPTER 4

EVALUATION

Evaluating the developed system properly requires real chat data that should be obtained from group chat messages as well as restaurant data. Although collecting restaurant data seems easier than accessing group chat messages, required restaurant data could not be obtained. A few e-commerce services serving in restaurant-domain were asked to get their data, but unfortunately, the result was negative. On the other hand, even if the required data is collected, creating a training set using the collected data is a very challenging process and it requires manual operations. In order to create only one training sample, anyone should analyze given chat messages and select only a few restaurants from restaurant data source containing maybe more than 100 restaurants based on their properties. Thus, training data creation process is also very hard.

Since collecting data and creating training data for them was very challenging, we had to create our test data by simulating them. Simulation data were produced with a distribution containing most of the cases that the system may encounter. For chat messages, 10 different group chat message set was produced that contains both short and long messages in terms of message length, both grammatical and ungrammatical messages in terms of grammar and messages having both poor and rich meaning in terms of properties. The following messages demonstrate sample chat data. A few messages of sample chat data were selected as corrupted or grammatically incorrect in order to compare them with their normalized versions.

Evet arkadaşlar geleneksel iftarımız için beyin fırtınasına hepiniz hoşgeldiniz:)	(OK guys, welcome to the brainstorming session for our traditional iftar :))
:) Konu malum zaten, fikirleri bekliyoruz:)	(:) The topic is already known, waiting for the ideas:))
Bir kere fix menü şart:))	(First, fix menu should be included:)))
Aynn fix menü garanti:)	(I agre, fix menu is important:))
Geçen sefer otele gitmiştik, bu sefer daa otantik bişeyler olsun:)	(Last time we were in a hotel, this time let's try somthng mor traditional:))
Evet böle osmanlı mutfağı faln:)	(Yes, for exmple ottoman cuisine:))
Neden olmasın:)	(Why not!:))
Hava çok sıcak yalnız, mutlaka balkonu terası falan olan bir yer olsun:)	(It's very hot, the place should have balcony or terrace:))
Ya da bahcesi:)	(Or garden:))
Bahçe güzel konsept bak:)	(I think, garden is nice concept, too:))
Evet, çok da pahalı olmasn:)	(Yeah, it shuldn't be too expensive, either:))
Yer olarak neresi diyelim:)	(What about the location?:))
Sultanahmet ya da eyup olabilir:)	(Sultanahmet or eyup:))
Olmadı faith, eminönü:)	(If not, faith, eminönü:))
Tarihi yarımadadan uzaklaşmayalım katılıyorm:)	(I agre, we should not be far away from historic half-island:))

Furthermore, 30 different restaurant samples were produced with a similar logic. Properties of these restaurants were distributed in terms of their types as follows: 99 standard, 4 no-notset-yes, and 12 notset-yes. The JSON document below demonstrates a sample restaurant entity with properties. Here the no-notset-yes and notset-yes properties can be seen 'others' field of the JSON document.

```
{
  "Id": 14,
  "Name": "Restaurant M",
  "Description": " Restaurant M description",
  "Cuisines": [ "türk", "osmanlı" ],
  "Places": [ "yalı" ],
  "Environments": [ "doğal", "tarihi" ],
  "Meals": [ "öğle yemeği", "akşam yemeği" ],
  "FoodServices": [ "alakart" ],
  "Prices": [ "pahalı" ],
  "Locations": [ "sarıyer" ],
  "Others": [ "manzara", "toplu taşıma", "iftar","kapalı alan" ],
  "Foods": [ "köfte", "bonfile", "beyaz et", "sebze" ],
  "Desserts": [ "kazandibi", "sütlaç", "kadayıf"],
  "Drinks": [ "soğuk içecek" ]
}
```


4.1 Sample use-case

Given a sample input containing chat messages that are listed above, the system first produces normalized chat messages as listed below.

Evet arkadaşlar geleneksel iftarımız için beyin fırtınasına hepiniz hoşgeldiniz @smiley[:)]	(OK guys, welcome to the brainstorming session for our traditional iftar @smiley[:)])
@smiley[:)] konu malum zaten , fikirleri bekliyoruz @smiley[:)]	(@smiley[:)] The topic is already known, waiting for the ideas @smiley[:)])
Bir kere fix menü fix @smiley[:))]	(First, fix menu should be included @smiley[:))])
Aynen fix menü garanti @smiley[:)]	(I agree, fix menu is important @smiley[:)])
Geçen sefer otele gitmiştik , bu sefer da otantik birşeyler olsun @smiley[:)]	(Last time we were in a hotel, this time let’s try something more traditional @smiley[:)])
Evet böle Osmanlı mutfağı falan @smiley[:)]	(Yes, for example ottoman cuisine @smiley[;)])
Neden olmasın @smiley[:)]	(Why not! @smiley[:)])
Hava çok sıcak yalnız , mutlaka balkonu terası falan olan bir yer olsun @smiley[:)]	(It’s very hot, the place should have balcony or terrace @smiley[:)])
Ya da bahçesi @smiley[:)]	(Or garden @smiley[:)])

Bahçe güzel konsept bak @smiley[:)]	(I think, garden is nice concept, too @smiley[:)])
Evet , çok da pahalı olmasın @smiley[:)]	(Yeah, it shouldn’t be too expensive, either @smiley[:)])
Yer olarak neresi diyelim @smiley[:)]	(What about the location? @smiley[:)])
Sultanahmet ya da Eyüp olabilir @smiley[:)]	(Sultanahmet or Eyüp @smiley[:)])
Olmadı Fatih, Eminönü @smiley[:)]	(If not, Fatih, Eminönü @smiley[:)])
Tarihi yarımadadan uzaklaşmayalım katılıyorum @smiley[:)]	(I agree, we should not be far away from historic half-island @smiley[:)])

These normalized messages, which are generated using İTÜ NLP pipeline, are more meaningful for the language analysis. After the normalized messages are processed by the analysis module, a chat vector is created in the system. In terms of ease of display, the following chat vector only shows meaningful features for the above conversation. Values of all properties not included in the chat vector can be considered as 0.

Osmanlı (Ottoman)	3
hotel (hotel)	1
pahalı (expensive)	-1
otantik (authentic)	1
tarihi (historical)	-1
iftar (iftar)	1
teras (terrace)	2
fix menü (fix menu)	3
bahçe (garden)	2
Sultanahmet (Sultanahmet)	1
Eyüp (Eyüp)	1
Fatih (Fatih)	-1
Eminönü (Eminönü)	-1

In the recommendation module, a nearly same process is also applied to all restaurants in the system in order to create restaurant matrix. The vector below demonstrates the values of some properties for a single restaurant vector as an example.

doğal (natural)	1
romantik (romantic)	-1
kahvaltı (breakfast)	1
akşam yemeği (lunch)	-1
alakart (a la carte)	1
açık büfe (open buffet)	-1
sigara (smoking)	-1
alkol (alcohol)	1
deniz kenarı (seaside)	-1
açık alan (outdoor)	-1
canlı müzik (live music)	1
otopark (car park)	-1
toplu taşıma (public transport)	1
organizasyon (organization)	1
bahçe (garden)	1
yemek (food)	-0.58
et (meat)	-0.17
balık (fish)	1
köfte (meatballs)	-1
fast food (fast food)	-0.33
tatlı (dessert)	0.17
içecek (drink)	-0.21
kahve (coffee)	0.2
türk kahvesi (Turkish coffee)	1
latte (latte)	1
mocha (mocha)	-1

Table 3. Scores of Restaurants

Restaurant Name	Similarity Score	Normalized Score [0-100]
Restaurant I	0.02	100.00
Restaurant E	-0.02	80.12
Restaurant H	-0.02	80.07
Restaurant A	-0.02	80.04
Restaurant J	-0.05	60.00
Restaurant B	-0.05	59.85
Restaurant F	-0.08	40.40
Restaurant D	-0.08	39.94
Restaurant G	-0.11	19.98
Restaurant C	-0.14	0.00

Table 4. Explanations of the System

Restaurant	Recommending	Not Recommending
Restaurant I	Mekanda çok talepte bulunduğunuz fix menü bulunmaktadır. Özellikle istediğiniz teras burada mevcut. Burası bir hotel. Bu mekan otantik bir ortama sahiptir. Tarihi bir mekan değil. Bu mekan fatih civarında yer almıyor. Mekanda iftar bulunmaktadır.	Fazlasıyla arzuladığınız osmanlı mutfağı özelinde hizmet vermemektedir. Mekanda fazlasıyla arzuladığınız bahçe bulunmamaktadır. Burası fiyat olarak pahalı bir mekan. Bu mekan sultanahmet civarında yer almıyor.
	There is a fix menu that you request a lot in the place. The terrace you want is available here. This is a hotel. This place has an authentic environment. It's not a historical place. This place is not located around fatih. Iftar is not available in this place.	It does not serve ottoman cuisine. There is not any garden as you wish. Here is expensive in terms of price. This place is not located around sultanahmet.
Restaurant E	Çok talepte bulunduğunuz fix menü burada mevcut. Burası bir hotel. Tarihi ortam talebiniz bu mekan tarafından karşılanmamaktadır. Burası fiyat olarak pahalı bir mekan değil. Bu mekan sultanahmet civarında bulunmaktadır. Bu mekan eyüp civarında bulunmaktadır.	Yoğun bir şekilde talep ettiğiniz osmanlı mutfağından lezzetler bu mekanda mevcut değildir. Mekanda çok talepte bulunduğunuz teras mevcut değildir. Mekanda çok istediğiniz bahçe bulunmamaktadır. Mekanda iftar bulunmamaktadır.
	The fix menu you have requested is available here. This is a hotel. Your historical environment request is not covered by this place. Here is an expensive place. This place is located around sultanahmet. This place is located around eyüp.	The dishes of the ottoman cuisine you are demanding are not available in this place. There is not a terrace where you request a lot. There is not any garden you want in the place. Iftar is not available here.

When both chat vector and restaurant matrix are generated, similarity scores are calculated with the help of the recommendation module. Table 3 shows similarity scores with their normalized values.

As shown in Table 3, Restaurant I is the winner. Following restaurants E, H and A have nearly the same scores, which means they are nearly identical in terms of their properties that are requested by users of the system. Table 4 demonstrates why the top 2 restaurants are selected by the recommendation system.

CHAPTER 5

CONCLUSION

A recommendation system has been developed for the restaurant area within the scope of this thesis study. First, a literature review had been conducted in recommendation systems. In the scope of literature review, both studies for English language and Turkish language had been reviewed. Then, a model was proposed and implemented as a solution for recommendation problem in restaurant domain. Due to the lack of real data, a test environment was simulated and the proposed system was evaluated using that simulation environment.

The solution is aimed at a smart assistant who can support groups of users who are planning to eat or drink outdoors in a chat environment. It is foreseen that the presence of such an assistant will be the solution to this problem since it is difficult to identify common favorites as the number of people in such user groups increases and therefore make a common decision.

In this study, an application for only the Turkish language has been implemented. However, similar studies can be performed for other languages using a different normalization module and data sources with different contents and rule bases. Since the system uses a keyword-based model that is developed for the Turkish language, it is not easy to reach the correct meanings while analyzing some complex sentence structures using this model. In order to understand such cases clearly, a sub-module that can analyze sentences semantically can be included in the proposed system. Within the scope of the study, the identity and profile information of the users were ignored. Improving the proposed system is also possible benefiting from this information. In addition, users' past behaviors can be saved, then a suggestion module that works using the features in the history data may be developed.

APPENDIX A

HYPONYMIC RELATIONS

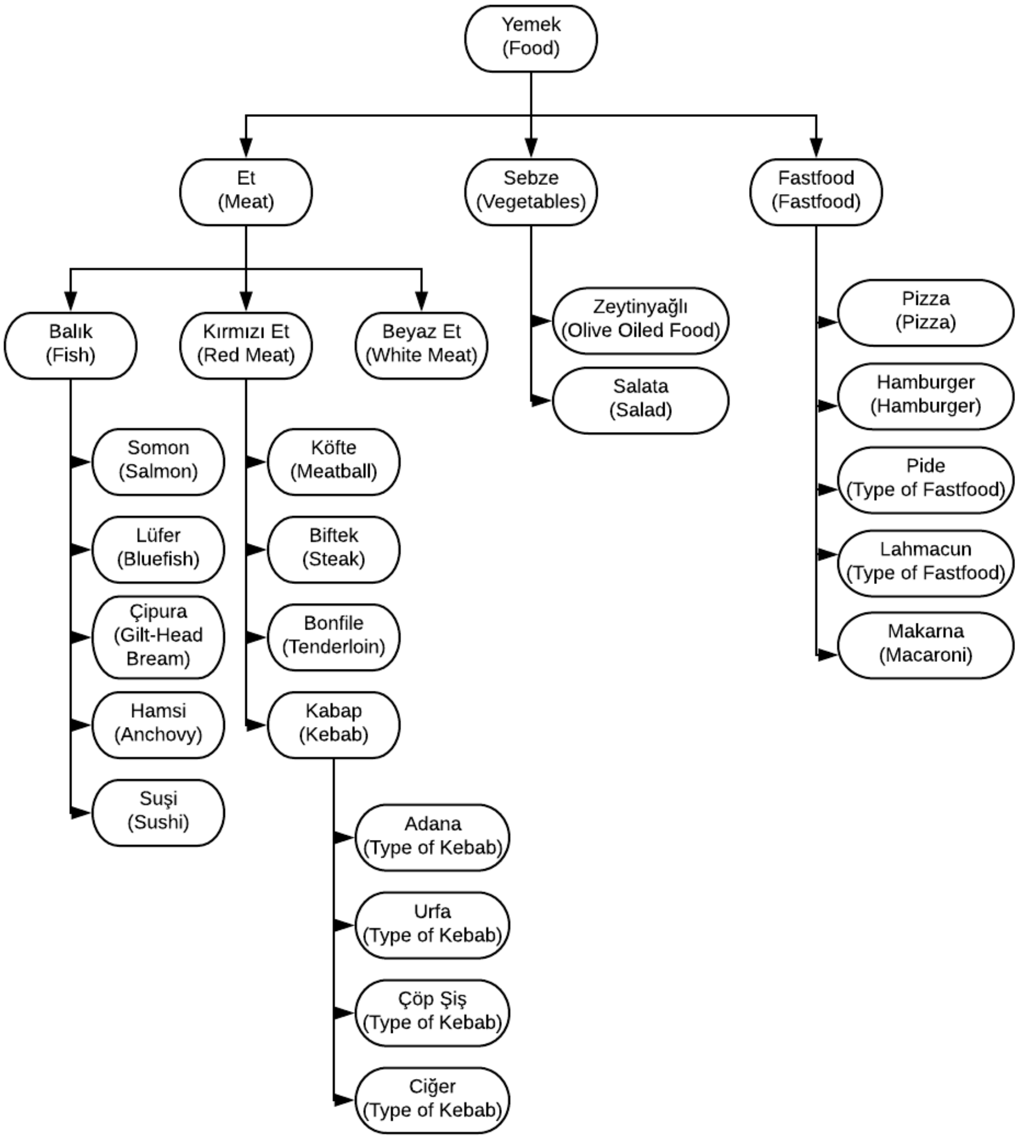

Figure A1. Hyponymic relations between food types

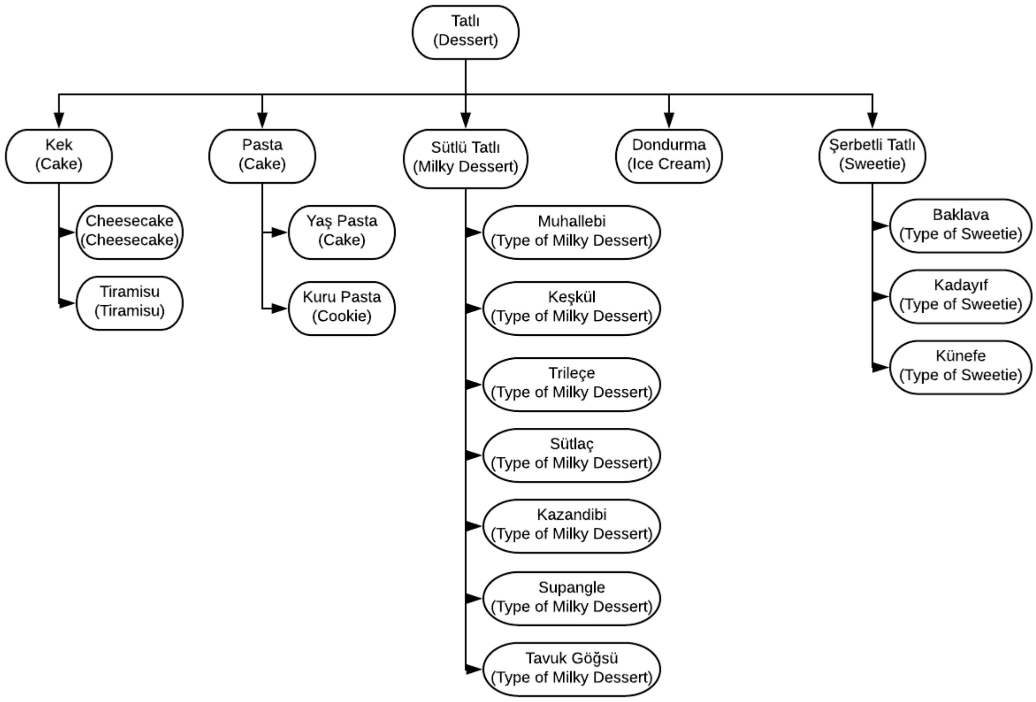

Figure A2. Hyponymic relations between dessert types

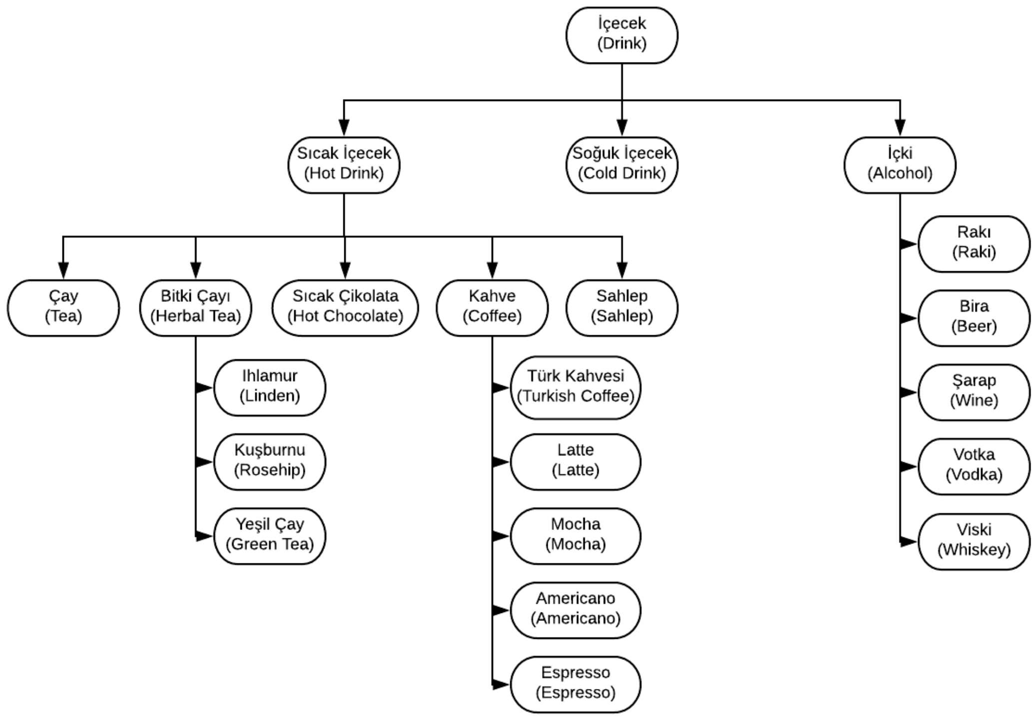

Figure A3. Hyponymic relations between drink types

APPENDIX B

EXPLANATION TEMPLATES

Templates for General Usage

{PropertyName} [Adverb] istenmiş, bu mekanda {PropertyName} mevcuttur.
({PropertyName} is demanded [Adverb], it is available here.)

Mekanda {PropertyName} bulunmaktadır.
({PropertyName} is available in this place.)

Burada {PropertyName} vardır.
(There is {PropertyName} here.)

[Adverb] {PropertyName} özelliği bu mekanda mevcuttur.
({PropertyName} feature is available in this place as you requested [Adverb].)

[Adverb] talep ettiğiniz {PropertyName}, bu mekan tarafından sağlanmaktadır.
([Adverb] demanded {PropertyName} is supplied by this place.)

[Adverb] istediğiniz özellik olan {PropertyName} mevcuttur.
({PropertyName} is available here as you requested [Adverb].)

[Adverb] tercih ettiğiniz {PropertyName} mevcuttur.
({PropertyName} is available here as you preferred [Adverb].)

Templates for Cuisine Property Type

[Adverb] {PropertyName} mutfağından lezzetler bu mekanda yer almaktadır.
(Tastes from {PropertyName} cuisine are served here as you requested [Adverb].)

Bu mekan {PropertyName} yemekleri üzerine çalışmaktadır.
(This place cooks {PropertyName} foods.)

Templates for Place Property Type

Burası {PropertyName} olarak hizmet vermektedir.
(Here serves as {PropertyName}.)

{PropertyName} talebinizi karşılayan bir mekandır.
(This place meets your {PropertyName} demand.)

Templates for Environment Property Type

{PropertyName} ortam talebinizi bu mekan karşılamaktadır.
(This place meets your {PropertyName} environment demand.)

Burası istediğiniz gibi oldukça {PropertyName} bir mekan.
(Here is a pretty {PropertyName} place.)

Templates for Meal Property Type

Bu mekan {PropertyName} için tercih edilebilir.
(This place can be preferred for {PropertyName}.)

{PropertyName} için gönül rahatlığıyla tercih edebilirsiniz.
(You may prefer here for {PropertyName} with a peace of mind.)

Templates for Food Service Property Type

Bu mekanda {PropertyName} hizmet verilmektedir.
({PropertyName} service is available in this place.)

[Adverb] {PropertyName} servis şekli burada mevcuttur.
({PropertyName} service is available here as you requested [Adverb].)

Templates for Price Property Type

Burası fiyat olarak {PropertyName} bir mekan.
(Here is {PropertyName} place in price.)

Burası [Adverb] {PropertyName} olarak değerlendirilebilir.
(This place can be considered as {PropertyName} as you requested [Adverb].)

Templates for Location Property Type

Bu mekan {PropertyName} civarında bulunmaktadır.
(This place is located around {PropertyName}.)

{PropertyName} bölgesi için burasını tercih edebilirsiniz.
(In {PropertyName} you can choose here.)

Templates for Food Property Type

Mekanın menüsünde [Adverb] {PropertyName} bulunmaktadır.
({PropertyName} is available in the menu as you requested [Adverb].)

[Adverb] {PropertyName} bu mekanda servis edilmektedir.
({PropertyName} is served here as you requested [Adverb].)

Templates for Dessert Property Type

Tatlı seçeneklerinden [Adverb] {PropertyName} mevcuttur.
({PropertyName} as a dessert option is available as you requested [Adverb].)

[Adverb] {PropertyName} bu mekan tarafından sunulmaktadır.
({PropertyName} is served by this place as you requested [Adverb].)

Templates for Drink Property Type

İçecek olarak [Adverb] {PropertyName} vardır.
({PropertyName} is available as drink here.)

Bu mekanda {PropertyName} içebilirsiniz.
(You can drink {PropertyName} in this place.)

REFERENCES

Breese, J. S. Heckerman, D. & Kadie, C. (1998). *Empirical analysis of predictive algorithms for collaborative filtering* (Research Report No. MSR-TR-98-12). Redmond, WA: Microsoft Corporation.

Claypool, M. Gokhale, A. Miranda, T. Murnikov, P. Netes, D., & Sartin, M. (1999). Combing content-based and collaborative filters in an online newspaper. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.145.9794 &rep=rep1&type=pdf

Debnath, S. Ganguly, N., & Mitra, P. (2008). Feature weighting in content based recommendation system using social network analysis. *Proceedings of the 17th International Conference on World Wide Web*. Beijing, China.

Dwivedi, P., & Chheda, N. (2012). A hybrid restaurant recommender. *International Journal of Computer Applications*, *55(*16*).*

Goldberg, D. Nichols, D. Oki, B. M. & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM - Special issue on information filtering, 35,* 61-70. New York City, NY: ACM New York.

Göker, M. H., & Thompson, C. A. (2000). Personalized conversational case-based recommendation. *Advances in Case-Based Reasoning. EWCBR 2000. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), 1898,* 99-111. Berlin, Germany: Springer.

McCarthy, J. F. (2002). *Pocket restaurant finder: A situated recommender system for groups*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download ?doi=10.1.1.496.4366&rep=rep1&type=pdf

Meteren, R., & Someren, M. (2000). *Using content-based filtering for recommendation*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc /download?doi=10.1.1.25.5743&rep=rep1&type=pdf

Oord, A. Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. *NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, 2*, 2643-2651. Lake Tahoe, NV: Curran Associates.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. *The adaptive web* (pp. 325-341). Berlin, Germany: Springer.

Popescul, A. Ungar, L. H. Pennock, D. M., & Lawrence, S. (2001). *Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments*. Retrieved from https://arxiv.org/ftp/arxiv/papers /1301/1301.2303.pdf

Resnick, P. Iacovou, N. Suchak, M. Bergstrom, P. & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of Netnews. *Proceedings of ACM, Conference on Computer Supported Cooperative Work* (pp. 175-186). Chapel Hill, NC: ACM New York.

Yoshii, K. Goto, M. Komatani, K. Ogata, T., & Okuno, H. G. (2006). Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. *ISMIR 2006: 7th International Conference on Music Information Retrieval*. British Columbia, Canada: University of Victoria.