TU WIEN Informatics

Content-Based Restaurant Recommendation Systems Using Textual and Visual Data

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Wirtschaftsinformatik

eingereicht von

Dante Godolja
Matrikelnummer 11929150

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Wien, 29. April 2023

Dante Godolja | Julia Neidhardt

TU WIEN Informatics

Content-Based Restaurant Recommendation Systems Using Textual and Visual Data

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Business Informatics

by

Dante Godolja
Registration Number 11929150

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass.in Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Vienna, 29th April, 2023

Dante Godolja

Julia Neidhardt

Erklärung zur Verfassung der Arbeit

Dante Godolja

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 29. April 2023

Dante Godolja

Kurzfassung

Inhaltsbasierte Restaurantempfehlungssysteme verwenden Eigenschaften wie Art der Küche, Preis und Standort, um den Benutzerinnen und Benutzern Restaurants vorzuschlagen. Durch die Analyse des Inhalts von Restaurants können diese Systeme Empfehlungen generieren. In der Forschung wird aktuell untersucht, wie ihre Effektivität und Anpassungsfähigkeit verbessert werden kann. In dieser Arbeit werden verschiedene Ideen für den Aufbau eines robusten Empfehlungssystems untersucht. Diese Ideen umfassen Text- und Bildverarbeitung. Für Restaurantvorschläge mittels Bildverarbeitung untersuchen wir Ansätze, die entweder auf der Farbähnlichkeit von Bildern basieren oder Eigenschaften mithilfe trainierter Bildmodelle extrahieren. Für die Textverarbeitung verwenden wir als Baseline-Modell TF-IDF sowie das State-of-the-Art-Modell SBERT. Anhand eines Prototypen werden die vorgeschlagenen Modelle präsentiert. Mit der richtigen Vorverarbeitung kann TF-IDF ähnliche Ergebnisse wie SBERT erreichen und diese je nach Szenario auch übertreffen. Jedoch bietet SBERT mehr neuartige Empfehlungen als das Baseline-Modell. Je nach Szenario können beide Modelle verwendet werden, um sinnvolle Restaurantempfehlungen zu generieren.

Abstract

Content-based restaurant recommender systems use features such as cuisine type, price range, and location to suggest dining options to users. By analyzing the content of restaurants, these systems can generate recommendations. Current research explores ways to improve their effectiveness. In this thesis we explore different ideas on how to build a robust recommender system. Such ideas include text and image processing. For image processing we explore suggesting restaurants based on image color similarity or feature extraction using pre-trained image models. For text processing we explore TF-IDF as a baseline and the state-of-the-art model SBERT. These ideas are then used in a practical case. Results show that with proper preprocessing, TF-IDF can achieve similar scores to SBERT and depending on the scenario even outperform it. However SBERT still provides more novel recommendations than TF-IDF. Depending on the scenario, both models can also be used to make meaningful restaurant recommendations.

Contents

CHAPTER 1

Introduction

The food and dining industry has been revolutionized by the internet, providing customers with a wealth of information and tools to enhance their dining experiences[11]. The rise of social media, food blogs, and review websites has empowered customers to share their opinions and experiences with a vast online community, while also allowing restaurants to showcase their menus and services to a broader audience.

In recent years, the use of recommendation systems has become increasingly popular in many different fields, including the restaurant industry[5, 8, 7, 2]. These systems use algorithms to analyze data about customers' preferences, behaviors, and previous interactions with a particular restaurant to generate personalized recommendations. The goal of these systems is to help customers make informed choices about where to eat, while also providing restaurants with valuable insights into their customers' needs and preferences.

1.1 Content-Based Recommendation Systems

Content-based recommendation systems are a class of recommendation algorithms that suggest items to users based on their preferences and past interactions with similar items [12]. These systems are based on the idea that if a user likes a particular item, then they are likely to enjoy other items that have similar features or characteristics.

The features used in content-based systems can vary widely depending on the domain and the type of items being recommended. For example, in a music recommendation system, the features may include genre, artist, tempo, and instrumentation, while in a movie recommendation system, the features may include genre, actors, director, and plot keywords. For the case of restaurants, restaurant metadata such as location, price-range, opening-hours can be used, while also utilizing restaurant description and/or images.

One advantage of content-based systems is that they do not require information about other users' preferences or behavior to make recommendations. This makes them useful in situations where user data is limited or difficult to obtain. Additionally, content-based systems are able to recommend niche or unique items that may not have a large user base, as they are not dependent on popularity or ratings.

1.2 An Overview of the Current Literature on Recommendation Systems

All relevant state-of-the-art research relies heavily on user data to build their Recommender Systems (RSs):

Gupta et al. [8] uses a user's geolocation and assumes that a user will like similar restaurants as the ones that they already visited. A user profile is created based on the restaurants visited and similar restaurants nearby are recommended. Another approach is utilizing user reviews by performing sentiment analysis and deriving a user's favorite food [2]. This information is then used to compare with restaurant menus and making a reasonable recommendation to the user.

User reviews could be also used for a venue's amenities instead of the food. The user should enter their favorite amenities into the system. After that, restaurants that offer these amenities are extracted and from these, the reviews of the amenities are analyzed by sentiment analysis. The restaurants that have the best ratings are then suggested [7].

An interesting idea is also the use of restaurant description together with photos from the restaurant. This content data is then combined with user data in order to build a hybrid RS [5].

1.3 GDPR and Its Impact on Constructing RSs With User Data

In a data-aware European society it is however difficult to have unlimited access to user data. The General Data Protection Regulation (GDPR)[1] has had a seismic impact on the collection and use of user data, drastically altering the landscape of recommendation systems. This sweeping legislation has imposed strict limits on the collection of user data, requiring companies to obtain explicit consent from users before collecting, processing, or storing their personal data.

Falter Verlagsgesellschaft m.b.H.[2] (from now on addressed to as Falter), an Austrian media portal, has a comprehensive list of restaurants that it displays to their readers and wants to develop a better way to do this by utilizing recommendation systems. It wants to achieve this goal without the need for extensive user data thus adding a hard limitation

[1] `http://data.europa.eu/eli/reg/2016/679/oj` (24 April 2023)
[2] `https://www.falter.at` / `https://www.falterverlag.at` (24 April 2023)

on the construction of such system. The goal of this thesis is to build a content-based RS that relies solely upon restaurant data to provide users with meaningful recommendations. This is done by assuming that when the user is opening the page of a certain restaurant, they are mainly interested on other restaurant with a similar cuisine or “feel” as the given restaurant.

CHAPTER 2

Problem Definition

Falter, has a column called "WIEN, WIE ES ISST"[1] (German wordplay for "Vienna, like it eats") in which it has compiled a list of more than 6000 restaurants, bars and cafés grouped by categories that are displayed to the user on Falter's website. The purpose of this column is to help the user choose a place to dine or drink at by providing them with a general description about the venue such as working hours, type of kitchen, capacity, location, images etc. For some of the places Falter has dedicated reviews, written by professional critics. When opening the page of one of this restaurants or cafés, the user also receives some recommendations for similar places. These recommendations are then displayed at the end of the page under "Diese Lokale könnten Sie auch interessieren" translated to "You might also be interested in these restaurants". For the rest of this thesis we will use the word "restaurant" to describe all of the venues.

For its recommendations Falter is using basic filtering of the restaurants' categories which usually comprise of type of cuisine (if given) and the type of venue. It then filters the restaurants further by the given restaurant's location (district). In the end the user receives recommendations based on the restaurant's categories and location. The problem arises in the fact that the categories sometimes fail to provide an actual similar restaurant to the one that the user is looking at. The task is to build a more advanced recommendation system which utilizes other given data like the restaurant's description or images to give users a better recommendation for their needs.

2.1 Example of Falter's Recommendations

The following example presents the problem with the categories and why these alone do not always provide meaningful information to the user.

[1] https://www.falter.at/lokalfuehrer (7 January 2023)

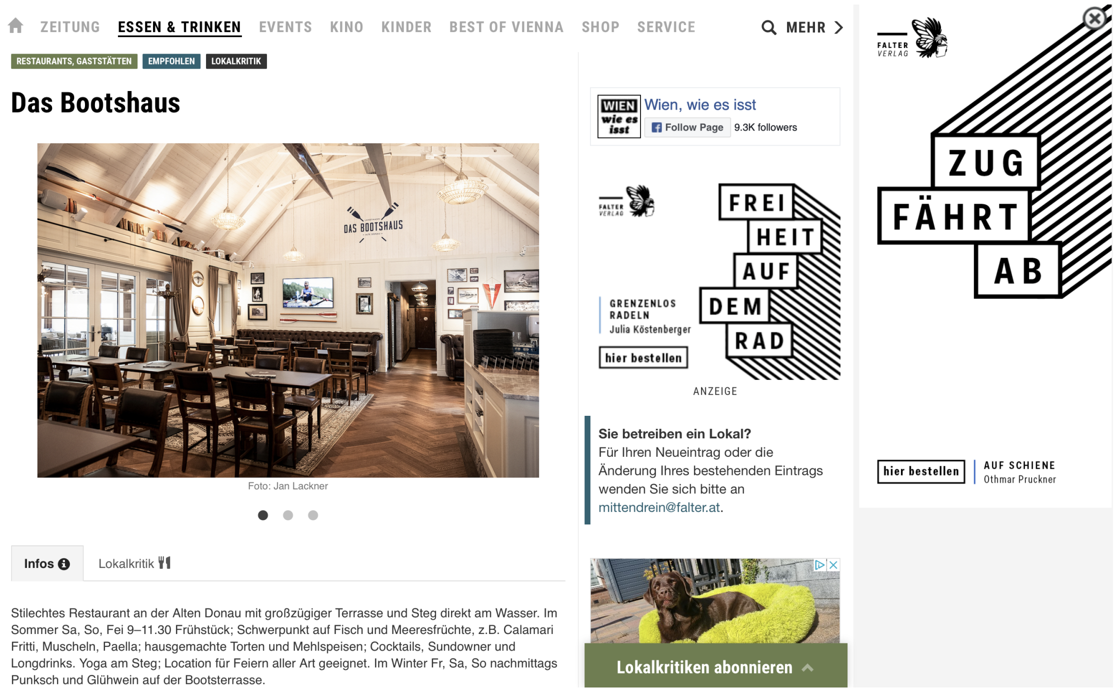
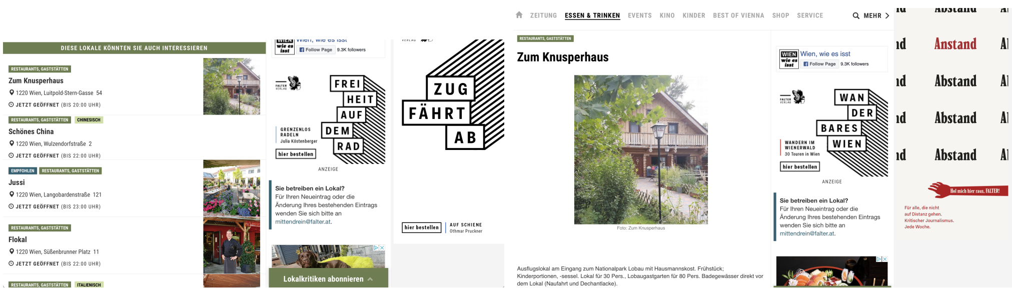


Figure 2.1: Restaurant Page (Das Boothaus)[2]

As we can see in figure 2.1, the top left corner shows some given tags (categories). Then looking further down we see the restaurant's images and a basic description. In the description it is said that this restaurant specializes in seafood. This however is not reflected by the restaurant's categories on the top left corner which only include the generic tag "Restaurants" and two other tags "Empfohlen" and "Lokalkritik" which mean that this restaurant was reviewed by a professional critic and is recommended. The images below will show the top recommended restaurants and the page of the first recommended restaurant.



(a) Recommendations for Boothaus (b) First recommendation [3]

[2]Das Bootshaus (7 January 2023): `https://www.falter.at/lokal/3239/das-bootshaus`
[3]Zum Knusperhaus (7 January 2023):
`https://www.falter.at/lokal/10824/zum-knusperhaus`

The first recommendation (figure 2.2a) is a restaurant, which according to its description, has access to a national park. Thus it appears to have nothing in common with our given restaurant apart from the tag (category) and the district. Figure 2.2b shows us the page of the first recommended restaurant. If we read the text of the first recommended restaurant, there is no indication that this restaurant specializes in seafood, which would logically be the first thing to look for when recommending similar restaurants to our given one.

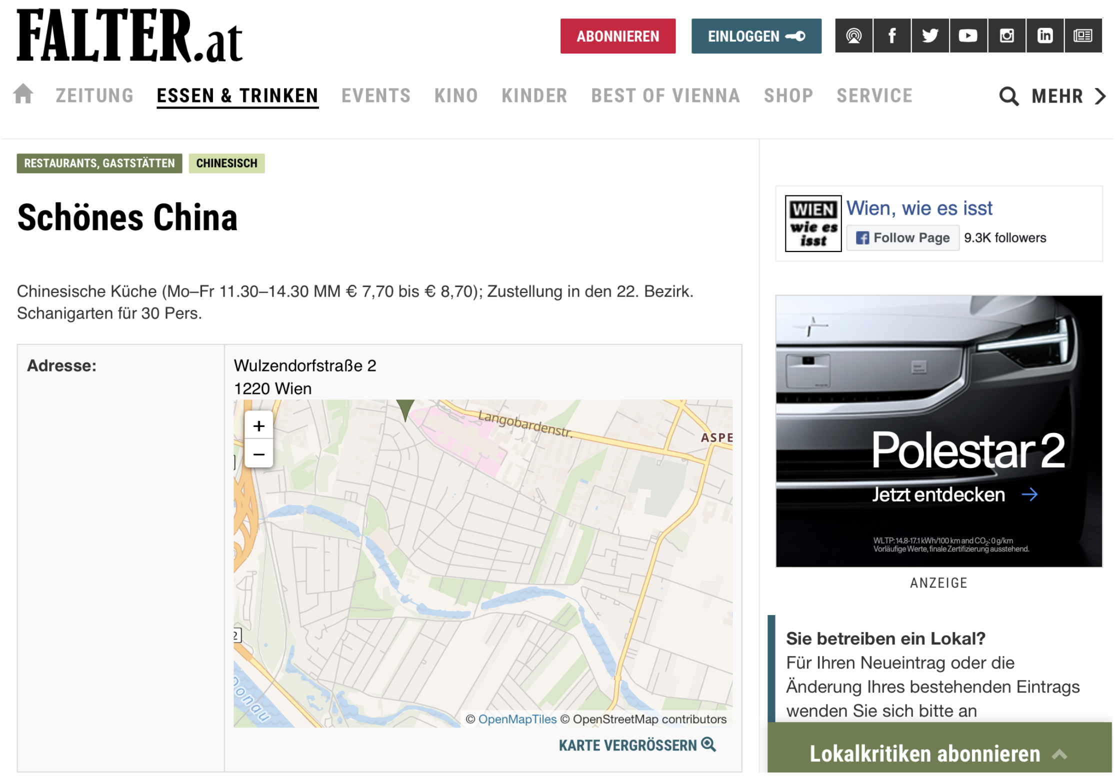


Figure 2.3: Restaurant Page (Schönes China)[4]

The second recommendation (figure 2.3) also seems to be unrelated to our given restaurant. One of the categories "Restaurants, Gaststätten" matches that of the given restaurant but the description shows us that the cuisines are not at all similar. This restaurant offers a Chinese cuisine which is very different from general seafood cuisine. The other recommendations on the list are also unrelated to our given restaurant.

[4]Schönes China (7 January 2023): `https://www.falter.at/lokal/9929/schones-china`

The problem becomes now apparent, which leads us to the following research questions:

- **RQ1:** Which models are suitable for efficiently using textual and visual features in the field of restaurant recommendation systems?
- **RQ2:** How well does the selected restaurant recommender system perform for recommending similar food and cuisines?

CHAPTER 3

The Dataset

The provided dataset is a JSON file with website data that contains information about all the restaurants displayed on the Falter website. It has 6860 instances with 23 columns. The columns contain information like the restaurant's name and id, it's address (street, house number, zip-code) together with the coordinates. Meta-data that won't be required for our use-case are the restaurants' social media pages or other links to websites and phone numbers. The most important information is contained in the following columns:

- **id:** Restaurant's id.
- **name:** Restaurant's name.
- **category_text:** This is the basic description also displayed to the user on the website.
- **address:** This contains address meta-data (street, house number, zip-code, city) in JSON format.
- **coordinates:** This column contains the atitude and longitude data of the restaurant.
- **attributes:** These are tags that summarize a restaurant's amenities, facilities and other information, e.g. whether you can eat breakfast or brunch, whether the restaurant is open on Sundays, whether the restaurant has a garden area where people can sit and eat etc. These can be seen better in figure 3.1.
- **categories:** This column contains the tags which are also displayed in the Falter's website and are used for their recommendations.
- **kitchen:** The kitchen consists of a list containing the type of cuisines for each restaurant.

- **price:** This column indicates the prices of the restaurant by using the Euro symbol "€". The more "€", the more expensive a restaurant is. Ranges from "€" to a maximum of "€€€€"
- **opening_hours:** This displays the restaurant's opening hours.
- **pictures:** This is a column which has links to the pictures used on the website.
- **link:** This column stores a link path to locate the restaurant page in Falter's website.
- **review:** This is the dedicated review written by a professional.

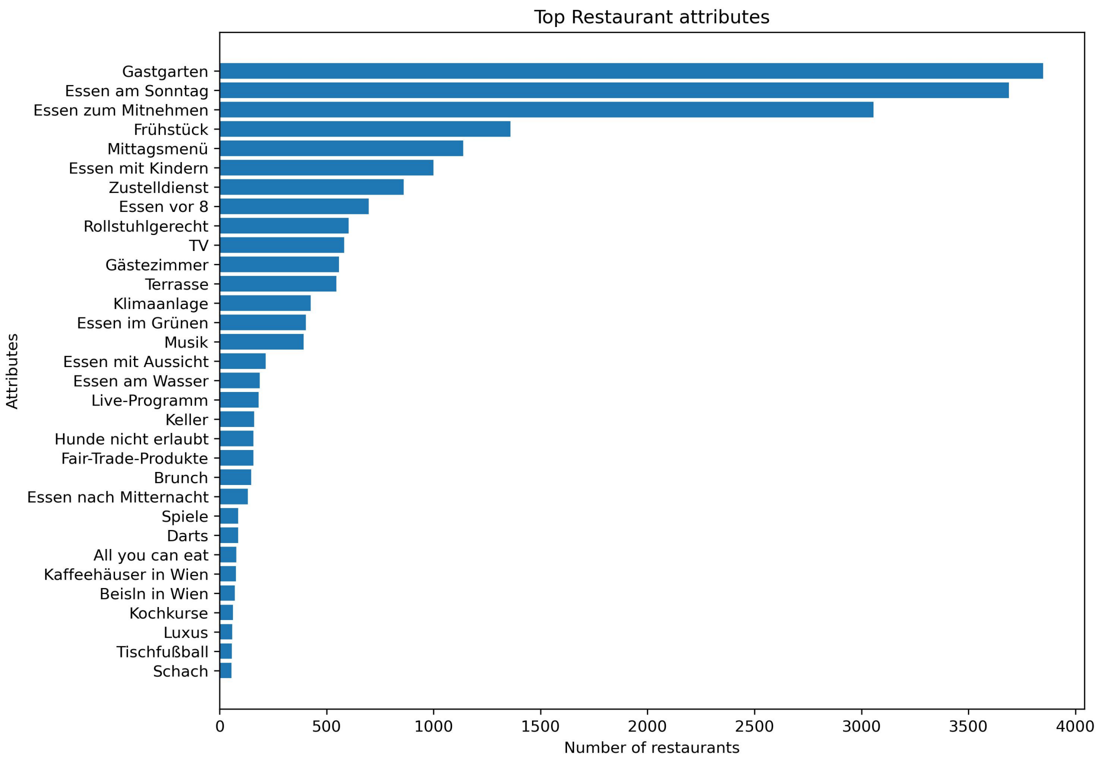


Figure 3.1: Attributes distribution

3.1 Challenges Faced From the Data

Of the 6860 restaurants only 312 have dedicated reviews, 6855 have a basic description and 3508 have images. Although the website is called "Wien, Wie Es Isst"[1] only 4579 restaurants are based in Vienna. While the basic description remains a reliable source to

[1] https://www.falter.at/lokalfuehrer (7 January 2023)

use for a recommender system, the dedicated reviews are too few to be used on their own for a recommender.

3.1.1 Category Imbalance

As mentioned in chapter 2 there exists a lack of information within the categories which makes the filtering by category (Falter's current recommendations) inaccurate. On top of this, even the less generic categories which give us information about the restaurant's kitchen are too little in quantity in order to provide any meaningful recommendation.

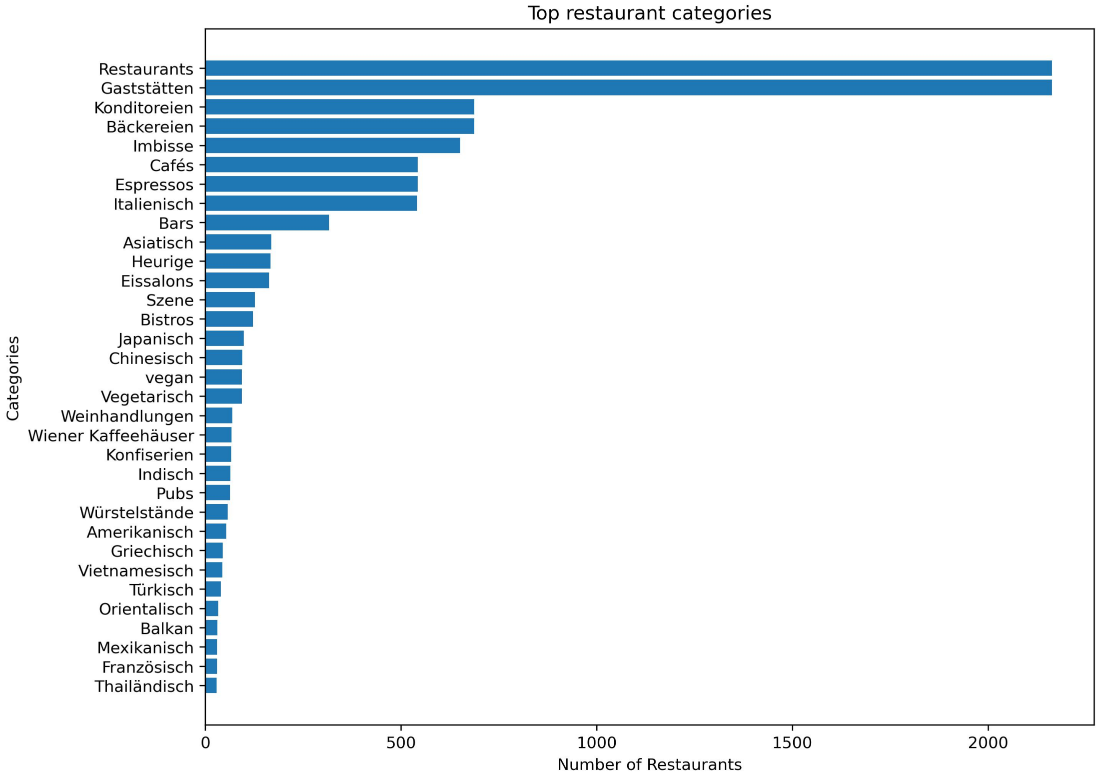


Figure 3.2: Category distribution

Figure 3.2 shows the distribution of categories with more than a certain amount of instances in them. The generic categories which classify venues as restaurants, cafes, snacks, bakeries etc. comprise more than 66% of all categories. This could be due to several reasons such as there actually being very few restaurants in Falter's website that actually have different types of cuisines. Another reason could also be the insufficient facilities for labeling all of the restaurants.

3.1.2 Attribute Redundancy

A similar attribute to “categories” is “kitchen” which displays the types of cuisines for a given restaurant.

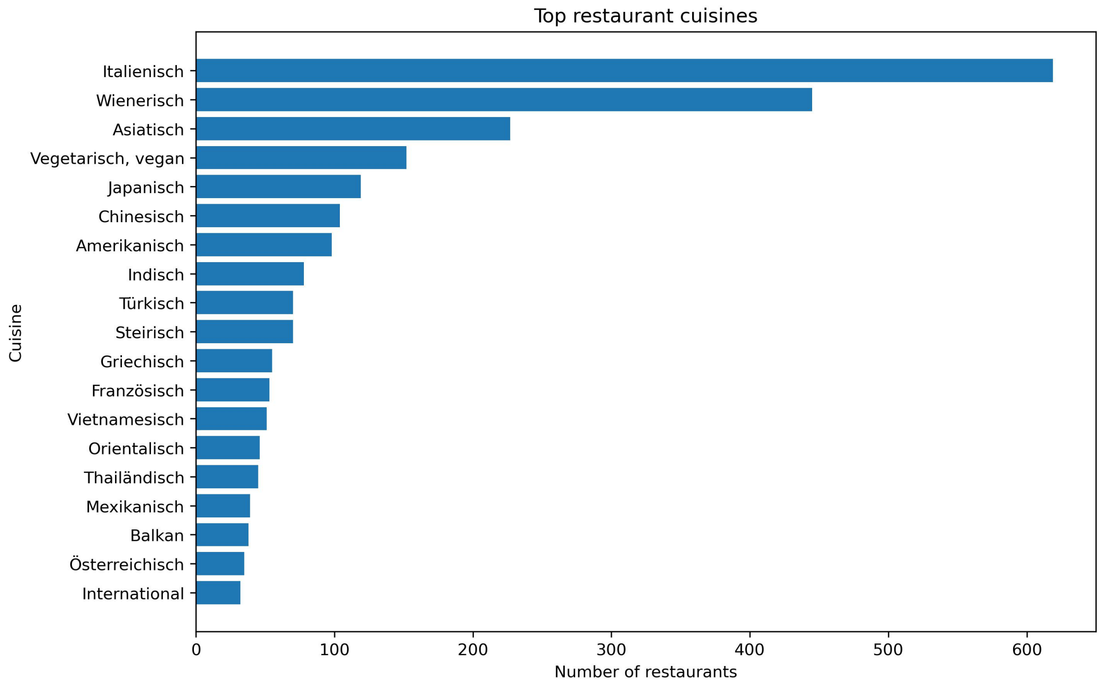


Figure 3.3: Kitchen distribution

Figure 3.3 shows that the “kitchen” attribute provides almost the same information as the “categories” attribute. These can be observed by removing the generic categories like “Restaurants”, “Gaststätten”, “Cafes”, “Bars” etc. which leaves us with the type of cuisine. However we decide to keep this attribute since the type of cuisine is already filtered out from the categories and also there are some restaurants which have a “kitchen” attribute but don’t appear on “categories”. For example there 619 restaurants with the attribute “kitchen” set to “Italienisch” (Italian), but there are only 541 restaurants with the “categories” attribute set to “Italienisch”. It should also be mentioned that only 2300 restaurants have a “kitchen” attribute to begin with.

3.1.3 A Closer Look at the Critiques

The reviews are rather a critique written by a professional in the same fashion as a news article. For the remainder of this thesis the terms review and critique will be used interchangeably. Taking a closer look at an excerpt from one of these reviews also reveals the critic’s writing style:

"Rudern ist ein toller Sport. Elegant, lautlos, schnell, effizient – quasi das Rennradfahren unter den Wassersportarten (wenngleich man mit dem Ruderboot umkippen und sodann ertrinken kann; da dann doch lieber Rennrad ...). Und weil zwei Söhne von Berndt und Irmgard Querfeld, Wiens unangefochtenem Kaffeehaus-Kaiserpaar, diesen Sport auf wettkampflichem Niveau betreiben, sahen sich die Querfelds veranlasst, das Schutzhaus Neu Brasilien an der Alten Donau zu übernehmen..."[2]

A translation with Google Translate gives us the following:

"Rowing is a great sport. Elegant, silent, fast, efficient - the racing bike among the water sports, so to speak (although you can tip over with the rowing boat and then drown; then it's better to have a racing bike ...). And because the two sons of Berndt and Irmgard Querfeld, Vienna's undisputed royal couple in the coffee house, practice this sport at a competitive level, the Querfelds feel compelled to take over the Neu Brazil refuge on the Old Danube..."

It can be clearly seen that the opening paragraph of the critique provides no relevant information regarding the restaurant's cuisine or food. The following paragraphs of the critique go on to describe the history of how the restaurant came to be and only one small paragraph really describes the food, its taste and the price. It can be said that the review focuses more on the history of the restaurant and sets a nice tone for the reader. The problem with using these reviews for text feature extraction is that they would provide a lot of noise and no relevant information for the task at hand. The use of pre-trained language models to extract meaningful information from such a text becomes even harder due to the fact that the critiques are written in German. A possible solution could be to feed the text in ChatGPT[3] and ask it to rate the review as a negative or positive one, however that is not the focus of this thesis.

[2] Restaurant critique (review): `https://www.falter.at/lokal/3239/das-bootshaus/lokalkritik` (7 January 2023)

[3] `https://chat.openai.com` (23 April 2023)

CHAPTER 4

A Look at Software Frameworks

Frameworks provide a wide range of pre-implemented algorithms, which can be used to build a recommender system for various applications, such as product recommendation, content recommendation, and personalized search. These algorithms are often well-optimized and tested, and their implementation has already been taken care of. By using these pre-implemented algorithms, developers can focus on other essential aspects of the system, such as data preprocessing and evaluation.

Another benefit of using a recommender system framework is their ability to support rapid prototyping. These frameworks provide a flexible and customizable platform, which allows developers to experiment with different algorithms, parameters, and data sources. Developers can quickly prototype and test their ideas using a variety of algorithms and data sources, which can help them find the best approach for their specific application. Once the optimal approach is identified, the developer can then refine the system's implementation and improve its performance.

A recent state-of-the-art framework is RecBole [19] which provides 73 recommendation models on 28 benchmark datasets, covering the categories of general recommendation, sequential recommendation, context-aware recommendation and knowledge-based recommendation. The problem with using such a framework, however, is its focus on collaborative-filtering which is based upon the assumption that we are provided with user data. Another disadvantage is the limited documentation[1] provided by RecBole, which mostly covers quick starts for the provided models without giving a more in-depth look at how to prepare the dataset for model use. In order to be able to use the models, the data needs to be prepared into atomic format but the documentation for preparing a custom dataset is insufficient. For these reasons we refrained from using such frameworks for our task.

[1] `https://recbole.io/docs/get_started/quick_start.html` (14 January 2023)

CHAPTER 5

A Look at Label Extraction

Having a set of labels that are not ambiguous would be very helpful in order to increase the scope of evaluation for the recommenders. In order to overcome the problem stated in sections 3.1.1 and 3.1.2, label extraction through natural language processing (NLP) was considered. The library spaCy[1] [9] was used in order to attempt extracting noun-adjective pairs or nouns relating to food from the description of the restaurants. The descriptions are usually not more than a paragraph long and provide meaningful information about the restaurants. Oftentimes the type of cuisine is revealed in the form of a noun-adjective pair. To demonstrate this we provide the following sentence which is an excerpt from a restaurant's description:

"Offene Sushi-Küche, offene Hauptküche und eine offene Wok-Station, bei der man sich die Zutaten seines Pfannengerichts selbst zusammenstellen kann; die Speisekarte bietet ein fein selektiertes Potpourri aus asiatischer, mediterraner und wienerischer Küche..."[2]

and the translation:

"Open sushi kitchen, open main kitchen and an open wok station where you can put together the ingredients of your stir-fry yourself; the menu offers a finely selected potpourri of Asian, Mediterranean and Viennese cuisine..."

The idea would be to extract the following noun-adjective:

- offene Sushi-Küche
- offene Hauptküche
- offene Wok-Station

[1] `https://spacy.io` (22 January 2023)
[2] `https://www.falter.at/lokal/10/do-co` (22 January 2023)

- asiatischer Küche
- mediterraner Küche
- wienerischer Küche

To achieve this a pre-trained pipeline on German texts that can recognize such entities has to be used in combination with spaCy. The general pipeline offered by spaCy for German texts is 'de_core_news_sm' which was trained on the TIGER Corpus [1, 3]. The TIGER Corpus consists of approximately 900,000 tokens (50,000 sentences) of German newspaper text, taken from the Frankfurter Rundschau[3]. The corpus was semi-automatically POS-tagged (Part-of-speech) and annotated with syntactic structure. Moreover, it contains morphological and lemma information for terminal nodes[4].

5.1 Syntactic Analysis With spaCy

With the help of this pipeline spaCy divides the text into tokens where each token has specific properties:

'head' is the property that provides the syntactic head of the token, i.e., the word that the current token is syntactically dependent on. In the case of *"Offene Sushi-Küche"* *"Sushi-Küche"* is the head of the token *"Offene"* since the noun gives meaning to the adjective. This would be the logic for all noun-adjective pairs.

'pos_' is the property that provides the part-of-speech (POS) tag of the token. A POS tag is a label assigned to a word in a sentence that indicates the word's syntactic category and its grammatical function in the sentence. This basically tells us wheter a token is a noun, adjective, verb etc.

'dep_' represents the syntactic dependency relation between the current token and its head, i.e., the word that the current token is syntactically dependent on.

Let us consider the first noun-adjective pair *"Offene Sushi-Küche"*:

[3] `https://www.fr.de` (22 January 2023)

[4] `https://www.ims.uni-stuttgart.de/en/research/resources/corpora/tiger/` (22 Janurary 2023)

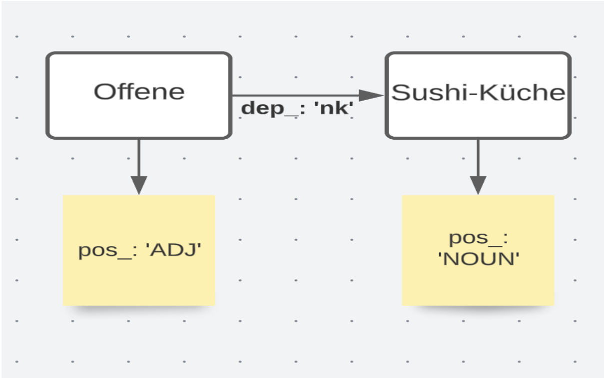


Figure 5.1: A case of trivial syntax

Figure 5.1 shows how spaCy analyzes this case. *"Offene"* is an adjective dependent on a noun kernel element (hilighted through 'dep_: nk') and *"Sushi-Küche"* is a noun. The filled from *"Offene"* to *"Sushi-Küche"* signifies that the noun is the head of the adjective token. In such a trivial case it is clear how to extract the pair since we can see the same pattern emerge also for the other noun-adjective pairs. The following figure will demonstrate a non-trivial case where the adjectives are bundled and connected through conjunctons.

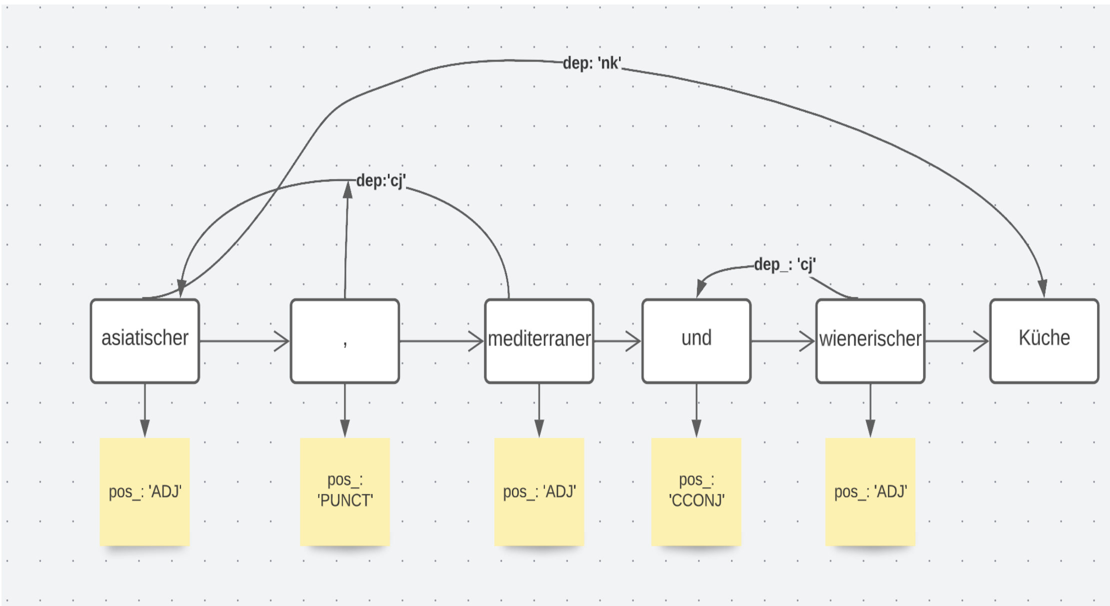


Figure 5.2: A case of non-trivial syntax

Figure 5.2 shows how the relationships change when we have more than one adjective. The filled arrows once again point from the token to their respective heads. For the first adjective the pipeline recognizes correctly the head (*"Küche"*) but for the other adjectives the pattern is not clear anymore. The second adjective's head is the first adjective, but for the third adjective the head is the conjunction *"und"*. We can assume that in the case of multiple adjectives this pattern will be consistent and apply the rules to extract the noun-adjective pairs as shown in algorithm 1.

Algorithm 1 Noun-adjective pair extraction

```
hanging ← initialize empty list
pairs ← initialize as empty list
doc ← text description
for token in doc do
    if token.pos_ = ADJ and token.dep_ = nk then
        pairs.append((token.text, token.head.text))
    else if token.pos_=ADJ and token.dep_=cj then
        hanging.append(token)
    else if token.pos_=NOUN and hanging.length > 0 then
        for adjective in hanging do
            pairs.append((adjective, token))
        end for
        hanging.clear()
    end if
end for
```

5.1.1 Unrecognized Tokens

There are cases where spaCy cannot recognize or misclassifies tokens and their properties. An example would be a sentence that starts like this:

"Regionale und saisonale Küche..."

In such a case spaCy misclassifies the token *"Regionale"* as a noun instead of an adjective. This could be because of misclassifications in the used corpus (TIGER). Another problem is the use of English words. The following restaurant description demonstrates this case:

"Große Auswahl an Pizzen, auch individuell zusammenstellbar, verschiedene Pizzaböden (Italien Style, Domino´s Original, Cheesy Crust, Pan Pizza) und zahlreiche Toppings, div. Vorspeisen und Beilagen wie Chicken Wings und -Strips, Stuffed Cheesy Bread uvm.; Desserts (Cinnabites, Apple oder Choco Pie). Spezielle Aktionen über die Domino's App. Abholung, Zustellung (innerhalb von 30 Minuten nach Bestellung) oder Essen vor Ort möglich."[5]

Words such as *Style, Cheesy, Crust, Pan, Toppings, Bread, Chicken, Wings* etc. can simply not be recognized at all by the pipeline. This is important information that we could use for label extraction, but it is impossible to be extracted. Due to the mentioned

[5] `https://www.falter.at/lokal/11308/dominos`(22 January 2023)

obstacles faced while using spaCy it is impossible to extract meaningful information and use it as a label for the data.

CHAPTER 6

Baseline Models

Since there is no real label available for model use, an unsupervised learning approach has to be taken in order to perform recommendations. Text and image features will be used in order to recommend restaurants.

6.1 Text Feature Extraction and Recommendation

In order to extract text features from the restaurant's description we consider using TF-IDF as our baseline. The formula for the term frequency in a given document $tf(t, d)$ is defined in formula 6.1:

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{6.1}$$

Where $f_{t,d}$ is the raw count of the given term t within the document d and the denominator is the sum of the raw count of all terms within the document.

For the inverse document frequency we use the sklearn's smoothed inverse document frequency[1] as shown in formula 6.2:

$$idf(t, D) = log(\frac{|D| + 1}{|d \in D : t \in D| + 1}) + 1 \tag{6.2}$$

$|D|$ is the number of all documents in the corpus and $|d \in D : t \in D|$ is the number of documents that contain the given term. Adding one to both the numerator and the denominator means that there is an extra document where the term appears. The

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html` (15 February 2023)

one that is added outside the fraction avoids the ignoring of terms that appear in all documents. TF-IDF is then calculated as in formula 6.3:

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \tag{6.3}$$

In order to calculate a feature vector we take the TF-IDF of every word in a given description (formula 6.4).

$$v_{d_1} = (tfidf(t_1,d_1,D), tfidf(t_2,d_1,D), ..., tfidf(t_n,d_1,D)) \tag{6.4}$$

This is done for every description in order to obtain all feature vectors for all descriptions $v_{d_1}, v_{d_2}, ..., v_{d_n}$.

6.1.1 Preprocessing and Lemmatization

In order to be able to use TF-IDF we first need to preprocess the descriptions in a meaningful way. A lot of the descriptions contain the food price in them in the following format "(€15,25)" thus we can remove the prices, since they don't give an important information for the text feature extraction. We can use the price attribute of the dataset later in the post-processing. The next step is to take the text apart into single tokens (words) and lemmatize them (put the token in its root form).

Lemmatization is an important step of the preprocessing because without it we would lose important information. For example if there are two restaurants of asian cuisine and in their descriptions this is apparent as follows:

For the first restaurant the description contains *"...asiatische Küche..."* and for the second *"...asiatischer Küche..."*. If not brought to its root form TF-IDF will count two different words (*asiatische vs asiatischer*) even though it is the same adjective declinated differently depending on the case. By using the lemma of the word we transform both versions into their root form which is *"asiatisch"*. This will then be recognized as the same word by TF-IDF and give the restaurants a higher similarity. We use HannoverTagger [18] for the lemmatization. The last step for the preprocessing is removing the stop-words such as "*und, oder, ein, einer etc.*" which provide no meaningful information for the recommendation.

6.1.2 Further Cleaning

Removing stop-words is already a good idea for noise reduction. Stop-words are however generic words used in any kind of domain. Because the task at hand only processess restaurant data, there are still some words which can be removed. A lot of the restaurant descriptions contain information about amenities, facilities and days when the restaurant is opened which again are only needed for post-processing. Such words are e.g. the abbreviations for the weekdays (*mo., di., mi., etc.*) or information about amenities/facilities

(*Schanigarten 20. Pers., TV, Kindersessel, Klimaanlage etc.*). These words appear in a lot of restaurants but have nothing to do with the cuisine that the restaurant offers. Removing these words will improve recommendations.

6.1.3 Computing Similarities

After calculating the feature vectors for each description we can compare them to each other by using cosine similarity, which is defined with formula 6.5:

$$cos(\theta) = \frac{\boldsymbol{A} * \boldsymbol{B}}{||\boldsymbol{A}|| \, ||\boldsymbol{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{6.5}$$

$\boldsymbol{A}$ and $\boldsymbol{B}$ are two feature vectors that are being compared to each other. This metric produces a result between -1 and 1 but since there are no negative TF-IDF scores then the value range becomes 0 to 1. The higher the result the more similar the documents are. By computing the cosine similarity we get the nearest neighbors (the restaurants with the highest cosine similarity for the given restaurant).

6.1.4 Example

We provide the following example to demonstrate TF-IDF's recommendations:

Let's return to the example from the second chapter in section 2.1. The restaurant "Das Bootshaus" has a focus on seafood and fish.

Table 6.1 shows how TF-IDF with only generic stop-word removal recommends restaurants:

Top-7 Recommendations
Das Bootshaus (Given Restaurant)
Blaustern
Ufertaverne
Golfstüberl
Gasthaus Hansi
Servus Du
Pizzeria Adamo
Gästhaus Käpt'n Otto

Table 6.1: Generic TF-IDF Recommendations

Disregarding the first result which is the restaurant itself, only a few restaurants match the criteria of seafood and fish. The first recommendation "Blaustern" is just a coffee shop. "Ufertaverne", "Golfstüberl", "Standgasthaus Birner", "Gasthaus Käpt'n Otto" and "Pizzeria Adamo" have Fish and Seafood mentioned in their descriptions and they are not even on the top of the list. The other restaurant from the recommendations is similar just because of the common restaurant words that include facilities and open hours.

Table 6.2 shows recommendations from an enhanced TF-IDF, which uses cleaning discussed in section 6.1.2:

Top-7 Recommendations
Das Bootshaus (Given Restaurant)
Ufertaverne
Pizzeria Adamo
Neuzeit
Gästhaus Käpt'n Otto
Cafe Restaurant Denito
Selbstverständlich Strandbeisl
Strandgasthaus Birner

Table 6.2: Enhanced TF-IDF Recommendations

Looking at only the top-7 recommendations we observe that all but one recommendation (Neuzeit) has seafood or fish in their description. With the generic example 5/7 of the top-7 had fish and seafood in their descriptions whereas this number improves to 6/7 with the improved model.

6.2 Image Feature Extraction and Recommendation

6.2.1 Color Similarity

An interesting approach to consider for recommendations would be the similarity in color palettes between the restaurant images. The idea behind this is to explore whether the images of a restaurant convey a certain "feeling" through the dominant colors in the given image. One way to check the color similarity between different images is by calculating the Earth Mover's Distance (EMD also known as Wasserstein Distance) [15] which has shown to be a reliable metric for such a task. EMD shows how much "work" is needed to move a histogram P to a histogram Q so that they overlap and check how similar their distributions are.

EMD is expressed through formula 6.6:

$$EMD(P,Q) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} d_{ij} f_{ij}}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}} \tag{6.6}$$

P and Q are two given color distributions. The paper recommends to use the CIE-Lab color space of images. d_{ij} is the ground distance which in our case is the Euclidean distance in the CIE-Lab color space. f_{ij} is the flow that minimizes the overall cost of “moving”. In order to calculate this metric we use scipy’s implementation[2].

6.2.2 Example of Color Similarity Through EMD

To demonstrate how EMD works we use image 6.1a as the given restaurant from which we want recommendations and show its top-3 most similar restaurant images in terms of color (figures 6.1b, 6.1c and 6.1d):

(a) Given Image

(b) 1st recommendation

(c) 2nd recommendation

(d) 3rd recommendation

Figure 6.1: Recommended restaurants based on color similarity

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html` (15 February 2023)

As can be seen EMD does a good job at showing images with similar color palettes. The given image's 6.1a main colors are white, brown, purple and dark gray. This can be clearly seen to be the case for the top-3 images as well.

6.2.3 CNN (VGG-16)

The approach of using a Convolutional Neural Network (CNN) to extract image features is also discussed in other works [5]. The proposed CNN in for such a task is the Visual Geometry Group model (VGG-16) [16] which is a CNN that is 16 layers deep. The model was trained on the ImageNet Dataset and can classify up to 1000 objects. For the current task we don't require object classification, rather we use the features that are computed from this network as extra information for our recommendation.

6.2.4 Example of VGG-16 Reommendations

Given the same restaurant as in 6.1a we use cosine similarity on the feature vectors extracted by VGG-16 for a Nearest Neighbors approach:

(a) Given Image

(b) 1st recommendation

(c) 2nd recommendation

(d) 3rd recommendation

Figure 6.2: Recommended restaurants based on VGG-16

The recommendations don't have any hard relations to each other, rather common objects like chairs, tables and glasses. In conclusion VGG-16 is better used in combination with other features if we want to have meaningful recommendations.

CHAPTER 7

SBERT

7.1 Introduction to BERT and SBERT

BERT, which stands for Bidirectional Encoder Representations from Transformers [6], is a neural network-based language model that was introduced by researchers at Google in 2018. It is designed to learn contextualized word embeddings by training on large amounts of text data in an unsupervised manner. BERT has achieved state-of-the-art performance on a variety of natural language processing tasks, such as question answering, sentiment analysis, and language translation.

The BERT architecture consists of a stack of transformer encoder layers. Each encoder layer has a multi-head self-attention mechanism that allows the model to attend to different parts of the input sequence while accounting for dependencies between them. This self-attention mechanism is based on the concept of attention, which was first introduced in 2017 [17] by the same group of researchers at Google.

In addition to the self-attention mechanism, each encoder layer also includes a feedforward neural network that processes the outputs of the self-attention layer. The output of the final encoder layer is used as input to a task-specific neural network that is trained on a downstream task, such as sentiment analysis or named entity recognition. The task-specific network is fine-tuned using supervised learning, where labeled examples are used to update the weights of the model.

One of the key features of BERT is its bidirectional training approach, which allows it to take into account both the left and right context of a word. This is in contrast to previous models, such as the popular Word2Vec model [13], which only considered the context to the left or right of a word. BERT also uses a masked language modeling objective during training, where a random subset of the input tokens are masked and the model is trained to predict the original values of these masked tokens based on the context of the other tokens. This encourages the model to learn contextualized representations of the input tokens, which can be used for a wide range of downstream tasks.

Sentence-BERT (SBERT) [14] is a variant of BERT that is specifically designed to generate sentence-level embeddings. While BERT is a word-level language model that generates contextualized embeddings for individual words, SBERT takes entire sentences as input and generates embeddings for the sentences as a whole. It achieves this by training a siamese neural network architecture, where two identical BERT networks are used to encode two different sentences that are then compared using a similarity function.

The main difference between SBERT and BERT lies in the way they process input sentences. While BERT processes input text as a sequence of individual words, SBERT first tokenizes each input sentence and then adds special tokens to indicate the start and end of the sentence. These modified input sequences are then fed into the siamese network, which generates embeddings for each sentence. The siamese network is trained using a contrastive loss function that encourages similar sentences to have similar embeddings, while dissimilar sentences have dissimilar embeddings.

7.2 Example of SBERT Recommendations

We can compute SBERT Recommendations in similar fashion as with TF-IDF, namely, by using Cosine Similarities. For the same restaurant considered in TF-IDF's case we generate the following recommendations using SBERT (table 7.1):

Top-7 Recommendations
Das Bootshaus (Given Restaurant)
Landtmann's Jausen Station
Mühlwasser Platz'l
Klyo
Zur Alten Kaisermühle
propeller
Fischerie
Landtmann's Original Café & Tortenshop

Table 7.1: SBERT Recommendations

In this case SBERT remains less on-topic (recommends less restaurants with fish on the menu) than the enhanced TF-IDF recommender. However the generated recommendations are not completely inexplicable. Both recommended restaurants with *Landtmann* in their names are similar to our given restaurant because they come from the same “chain” (Landtmann). Some of the restaurants have their locations near the river, just like our given restaurant. We can say SBERT discovers semantical meanings rather than looking for the same word like TF-IDF does.

CHAPTER 8

Fused Models

8.1 Motivation

Given the fact that the dataset provides pictures for half of the restaurants, we decided to explore the potential of using them to enhance our recommendations. The idea is to retrieve more features by extracting them from images using the VGG-16 model which could potentially contain more information about the restaurants. This idea was also explored on another paper [5] mentioned in section 1.2.

8.2 Approach

Since we have two text-based (TF-IDF, SBERT) recommenders we decided to use each of them in combination with VGG-16 by scaling, weighting and concatenating their feature vectors. Suppose that for a given restaurant r_1 we have the following feature vectors:

$$v_{SBERT}(r_1) = (x_1, x_2, ..., x_n)$$
$$v_{TF-IDF}(r_1) = (y_1, y_2, ..., y_n)$$
$$v_{VGG-16}(r_1) = (z_1, z_2, ..., z_n)$$

where x, y, z are the features generated by SBERT, TF-IDF and VGG-16 respectively. To have a combined model between a text-based recommender and an image-based recommender we concatenate their feature vectors as follows:

$$v_{TF-IDF+VGG-16}(r_1) = (y_1, ..., y_n, z_1, ..., z_n) \tag{8.1}$$

The same idea as in formula 8.1 could be used for SBERT + VGG-16.

8.3 Challenges

Concatenating the feature vectors of both models is not enough to build working recommender. While implementing the feature extraction for VGG-16, we noticed a high difference in scale between the value ranges of VGG-16 and TF-IDF. TF-IDF vectors have a scale of zero to one, as shown in section 6.1.3, whereas VGG-16 vectors have a scale of 0 to 521, which can also vary depending on the input image. Concatenating these two vectors will weigh the recommendations heavily towards VGG-16's features and the text features' influence might not be observed at all. One solution would be to scale down the image features between zero and one value range however the resulting recommendations are not very explainable.

Looking back at VGG-16's recommendations in section 6.2.4 we can see why this is the case. VGG-16 is a model that can classify objects from 1000 different categories. It is not a model that can tell much of the ambient of the restaurants and since restaurants have almost the same objects (glasses, tables, chairs, spoons etc.) the extracted features will not have particularly meaningful information for the task at hand. Other papers [5] used VGG in combination with Support Vector Machines (SVM) in order to classify pictures to categories (indoor, outdoor, food etc.) and not for recommendations. This would be impossible in our case since we have no provided labels for the pictures. A possible solution would be to concatenate the scaled down image vector but have it contribute less to the recommendation by introducing weights to the features, however this still does not achieve promising results.

CHAPTER 9

Evaluation

After considering different approaches to build a RS (text only, image only, text and image), the text-based RSs (TF-IDF, SBERT) offered the most promising results at first sight. Evaluating this hypothesis proved however to be a challenging task because of insufficient labels for the restaurants. There are only about 2300 restaurants out of 6800 in total that have a kitchen attribute which could be used as a label for evaluation. In order to offer a more comprehensive understanding of the RSs' functionalities we have to evaluate them using two different approaches, a quantitative and qualitative one.

9.1 Quantiative Evaluation

For this approach we consider only the 2300 restaurants with an existing kitchen attribute. Referring back to figure 3.3 we will use the three biggest kitchen types and their subtypes for a quantitative evaluation: the Italian kitchen, Asian together with its subtypes Japanese, Chinese, Korean, Thai, Vietnamese and lastly the Viennese kitchen. For the Viennese kitchen we also include Austrian, Tyrolean and Styrian as subtypes. These kitchens cover more than 70% of all restaurants that have a kitchen attribute. To demonstrate the performance with restaurants from smaller kitchen groups, we will also choose the Indian kitchen.

The evaluation metric used is the Hit-Rate and is defined as the total number of recommendations (from a top-10 recommendation list) with the same kitchen type (or subtype) as the given restaurant divided by the total number of recommendations (ten). Formula 9.1 is used to calculate the hit-rate:

$$hitrate = \frac{\sum_{i=1}^{n} I_i}{n} \tag{9.1}$$

$$I_i = \left\{ \begin{array}{ll} 1, & \text{if } i \in P \\ 0, & \text{if } i \notin P \end{array} \right\} \tag{9.2}$$

Where n is the total number of recommendations, P is the set of recommendations that have a matching kitchen or subkitchen to the given restaurant and I is an indicator function that is one if the current recommendation has a matching kitchen or subkitchen to that of the given restaurant and zero otherwise. If we consider the top left table from table 9.2 we could say that the hit rate is 0.8 since all but one restaurant (Mani im Vierten) have a matching kitchen/subkitchen to the given restaurant thus $\frac{4}{5} = 0.8$.

The following performance was observed:

Kitchen	**TF-IDF**	**SBERT**
Italian	84.05%	81.35%
Asian	84.84%	77.66%
Viennese	68.20%	80.53%
Indian	49.49%	42.95%
Average	**71.65%**	**70.62%**

Table 9.1: Hit Rate of TF-IDF and SBERT

The results from table 9.1 show that TF-IDF outperforms its counterpart on Italian and Asian kitchens but greatly underperforms when it comes to the Viennese kitchen. A possible explanation for this could be the use of more common words for the first two cases (Italian: pizza, pasta etc.) (Asian: sushi, maki, bento etc.) whereas for the Viennese case it could only be characterised by the adjective “Viennese” and less common food names giving it a lower similarity score for TF-IDF. SBERT however picks up semantic similarities and thus performs better in this case. When looking at the hit-rate for Indian kitchen, we see a significant drop in accuracy. This could be due to the low number of Indian cuisine restaurants, which comprise only 78 out of 2300 restaurants with an existing kitchen attribute.

9.2 Qualitative Evaluation

The text-based RSs show satisfying results when it comes to pure text-content, however other factors such as atmosphere or location are unaccounted for. This is also reflected during the qualitative interview. For this interview we had the chance to show eleven recommendation lists to an expert from Falter that compiles the restaurant dataset and get their opinion on the matter.

9.2.1 Experiment Design

In order to test the performance of both models we first devided the recommendation lists in three main topics:

- Restaurants with a given kitchen attribute
- Restaurants with focus on certain food but no specified kitchen attribute in the dataset
- Restaurants with no particular focus on any food and no specified kitchen attribute in the dataset

Dividing the lists into these topics provides good coverage for all cases provided in the dataset. A list is comprised of the top-5 recommendations for a given restaurant from one of the two text-based algorithms. For the first two topics there are two given restaurants and for the last topic only one given restaurant. For each of these restaurants we generate recommendations from both of the models, giving us four lists for each of the first two topics and two lists for the last topic summing up to ten lists and 50 restaurants in total. The algorithms that compiled the lists were not revealed until the end of the interview and the lists were shuffled within the topic in order to prevent bias towards a certain model.

For each of the lists the expert was asked to rate each recommendation on a scale of one to ten (ten being very good and one being very bad). At the end of each list the expert was asked three questions based on the diversity, serendipity, novelty, and coverage [10] of the RS. The questions were the following:

- How diverse are the recommendations? Do they cover different types of restaurants or just similar restaurants?
- What else would you have liked to see in the list that wasn't accounted for by the current recommendations?
- List recommendations that you found particularly useful or interesting and explain why.

The expert was also asked to identify main differences between the recommenders before these were revealed. The following tables show the ratings for each recommandation divided by topics.

Restaurants With a Given Kitchen Attribute

TF-IDF recommendations for Apadana	**Rating**
Pars	9
1001 Nacht	9
Kook	9
Hafes	10
Mani im Vierten	9

SBERT recommendations for Apadana	**Rating**
Pars	10
Il Melograno	1
Lord Of S	10
Retsina	7
Noodle House	1

TF-IDF recommendations for Onyx	**Rating**
Sajado	2
Kitcha Restaurant - Taco & Grill	7
Zano	unsure[1]
Green Express	3
Ya Sushi	2

SBERT recommendations for Onyx	**Rating**
Green Express	1
Sajado	3
Som Kitchen	3
Orangery	3
Selleny's	8

Table 9.2: Topic 1 Ratings for each restaurant and model

Expert's feedback to table 9.2:

The expert suggested that since Apadana is a Persian restaurant, it may be worthwhile to consider including restaurants with oriental or Arabic, Syrian cuisine in the recommendations. They also pointed out that Il Melograno is an expensive restaurant and Noodle House is a cheap restaurant. Such a high difference in prices may not be preferred by some users. Therefore, it is important to consider the price range of recommended restaurants and aim for a more balanced selection. Onyx is an expensive restaurant. To provide more value to users who are interested in high-end dining, it would make sense to include other expensive restaurants in the recommendations.

[1] no rating (expert couldn't decide on a rating)

Restaurants With Focus on Certain Food but No Specified Kitchen Attribute in the Dataset

TF-IDF recommendations for Das Bootshaus	Rating
Ufertaverne	7
Pizzeria Adamo	5
Neuzeit	6
Gasthaus Käpt'n Otto	6
Cafe Restaurant Denito	7

SBERT recommendations for Das Bootshaus	Rating
Landtmann's Jausen Station	6
Mühlwasser Platz'l	6
Klyo	9
Zur Alten Kaisermühle	10
propeller	6

TF-IDF recommendations for Do & Co	Rating
Mikado Sushi Style	4
Onyx Restaurant	10
Gastwirtschaft Wolfsberger	6
Wok House	3
Wienerwald	1

SBERT recommendations for Do & Co	Rating
Restaurant ON	10
Stuwer – Neues Wiener Beisl	9
Rochus	8
Gasthaus Alt Wien	6
Konditorei Housecafe	3

Table 9.3: Topic 2 Ratings for each restaurant and model

Expert's feedback to table 9.3:

The expert noticed that the recommendations for this topic are more diverse in a good way. This can be explained because of the lack of descriptive text for the kitchen or more general foods mentioned. For Das Bootshaus it was suggested that more recommendations based on location would be meaningful. Since this restaurant is located near the Danube it would make sense to have more restaurants near the Danube area. It should be said however that there are already recommendations in the list which satisfy this criterion like "Gasthaus Käpt'n Otto".

Restaurants With No Particular Focus on Any Food and No Specified Kitchen Attribute in the Dataset

TF-IDF recommendations for Dialog	Rating
Dal Maestro	4
Café zum 1. Stock	6
Café Kaiser	7
Café Kairo	9
Schani's Beisl	9

SBERT recommendations for Dialog	Rating
Café Kairo	9
Mimis Stüberl	8
Kleines Café	9
Café Simon	10
Zum Schü	9

Table 9.4: Topic 3 Ratings for each restaurant and model

Expert's feedback to table 9.4:

The expert found most of the recommendations appropriate. The only odd occurrence was Dal Maestro which is a restaurant and not a coffee shop like the given venue. No further feedback was provided.

9.2.2 Key Results

By talking to an expert from Falter and having them answer questions based on on the recommendations (section 9.2.1) the following conclusions were established:

- TF-IDF offers less diverse recommendations whereas SBERT offers novel and diverse recommendations. This is because TF-IDF searches for texts that contain the same words and SBERT can understand the semantic meaning of words.
- TF-IDF performs better when a kitchen attribute is given but SBERT performs better on general cases.
- The same restaurant can have a different rating depending on the context in which it was presented, meaning that if the same restaurant is presented among other good suggestions it might have a lower rating than it would have if it was presented among worse recommendations e.g. Restaurant Pars in table 9.2.
- A very important point for the expert was the atmosphere of the restaurant. This also explains why some recommendations had low scores even though the contents of the restaurants' descriptions were similar. We can show this by looking at table 9.3. According to the expert from Falter Mikado Sushi Style has a low rating because Do & Co appeals to higher paying customers and thus offers a different atmosphere, even though the type of food offered in both restaurants is similar.
- The above point leads us to the other remark made by the expert that is the price range, which was not used as a filtering option for the recommendations. According to the expert, the price range offers meaningful information about a restaurant and its clientele. People that dine at less-expensive restaurants may not want to dine in more expensive ones and vice versa. Even though the cuisines might be similar, clients might be looking for a certain kind of atmosphere dictated only by a certain price range. The expert's suggestion was that for restaurants with the price ranges on both the highest end ("€€€€") and the lowest end ("€"), restaurants with identical price ranges should be offered or one level less or more (For price range "€€€€" we could also consider suggesting price range "€€€" and for price range "€" we could consider suggesting price range "€€"). For restaurants in the mid ranges the expert suggested to recommend all price ranges.

CHAPTER 10

Conclusion

10.1 Summary and Discussion

In this thesis we explored different approaches for making a content-based restaurant RS given just a short description, images and some other meta-data useful for post-processing. We managed to show the importance of case-specific text preprocessing by removing words that normally aren't considered as stop-words, but could be considered as such for certain cases.

Apart from using text-based recommendation we explored possible ideas to derive a restaurant's atmosphere by looking at images' color similarities or by using a pre-trained model like VGG-16 to extract features from them.

We showed that simple baseline models such as TF-IDF with proper preprocessing can sometimes outperform state-of-the-art language models like SBERT [14] and that a robust recommender has to offer both models for recommendations based on the case (general or specific). Most importantly we managed to derive from an interview with an expert that further post-processing by price-range is needed in order to have more meaningful recommendations. We can thus answer the research questions:

- **RQ1:** Which models are suitable for efficiently using textual and visual features in the field of restaurant recommendation systems?

We presented work that shows a content-based RS which recommends restaurants based on their descriptions by using text feature extraction with TF-IDF and SBERT while also discussing an image-based approach by using CNNs or analyzing color similarities. Both methods offered interesting insights to restaurant recommendations, however text extraction offered more promising results.

- **RQ2:** How well does the selected restaurant recommender system perform for recommending similar food and cuisines?

As shown in table 9.1, the developed recommender systems that use only text data achieve high hit-rates when recommending restaurants from the most frequent cuisines. This could imply that the description of the restaurant offers sufficient information for suggesting other restaurants with similar cuisines.

The feedback from the qualitative evaluation shows that the recommendations are useful for the user and an improvement over the current available suggestions from Falter's website. This is argued through the use of the restaurants' descriptions for the proposed RS as opposed to the lack of this feature in the current RS. However it was pointed out that the post-processing should include the price range and location as filtering options for the recommendations.

10.2 Limitations and Future Work

Throughout this thesis we came across a certain amount of challenges that limited further work on certain approaches. Such a limitation was for example the lack of proper frameworks for the development of a purely content-based RS as discussed in chapter 4. The dataset itself proved to also have its limitations such as the verbosity of the critiques which made them not usable for our case or category ambiguity and imbalance that left us with unlabelled or ambiguously labelled data. This was discussed more in detail in section 3.1. We tried to overcome this by attempting to develop an algorithm which can extract elaborate labels from the description of the restaurants, however this was also not possible due to a lack of pipelines for token recognition in the German and English language (section 5.1.1).

In chapter 8 we discussed approaches on developing a fused model that uses both text and image data for recommendations, however challenges were faced when coming up with a meaningful way to combine both features. Future research could focus in this area and explore strategies for combining such features. Another interesting topic discussed in the thesis was the color similarity (section 6.2.2) and how this could be used to filter restaurants based on their atmosphere. Research has been done on colors and perceived venue luxury, emotions etc. [4] and similar ideas could also be explored for restaurants.

Eventually, a content-based RS could be developed that takes into consideration all aboved mentioned limitations and suggestions on future work and improves on them offering the user better experience when looking for their next dining option.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] Albert, S., Anderssen, J., Bader, R., Becker, S., Bracht, T., Brants, S., Brants, T., Demberg, V., Dipper, S., Eisenberg, P., et al. Tiger annotationsschema. *Universität des Saarlandes and Universität Stuttgart and Universität Potsdam* (2003), 1–148.

[2] Asani, E., Vahdat-Nejad, H., and Sadri, J. Restaurant recommender system based on sentiment analysis. *Machine Learning with Applications 6* (2021), 100114.

[3] Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. Tiger: Linguistic interpretation of a german corpus. *Research on language and computation 2* (2004), 597–620.

[4] Cho, J. Y., and Lee, E.-J. Impact of interior colors in retail store atmosphere on consumers' perceived store luxury, emotions, and preference. *Clothing and Textiles Research Journal 35*, 1 (2017), 33–48.

[5] Chu, W.-T., and Tsai, Y.-L. A hybrid recommendation system considering visual information for predicting favorite restaurants. *World Wide Web 20*, 6 (nov 2017), 1313–1331.

[6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Gomathi, R., Ajitha, P., Krishna, G. H. S., and Pranay, I. H. Restaurant recommendation system for user preference and services based on rating and amenities. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)* (2019), pp. 1–6.

[8] Gupta, A., and Singh, K. Location based personalized restaurant recommendation system for mobile environments. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2013), pp. 507–511.

[9] Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. Spacy: Industrial-strength natural language processing in python.

[10] KAMINSKAS, M., AND BRIDGE, D. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst. 7*, 1 (dec 2016).

[11] KIM, W. G., LI, J. J., AND BRYMER, R. A. The impact of social media reviews on restaurant performance: The moderating role of excellence certificate. *International Journal of Hospitality Management 55* (2016), 41–51.

[12] LOPS, P., DE GEMMIS, M., AND SEMERARO, G. *Content-based Recommender Systems: State of the Art and Trends*. Springer US, Boston, MA, 2011, pp. 73–105.

[13] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[14] REIMERS, N., AND GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (11 2019), Association for Computational Linguistics.

[15] RUBNER, Y., TOMASI, C., AND GUIBAS, L. J. The earth mover's distance as a metric for image retrieval. *International journal of computer vision 40*, 2 (2000), 99.

[16] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[17] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).

[18] WARTENA, C. A probabilistic morphology model for german lemmatization. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)* (2019), pp. 40–49.

[19] ZHAO, W. X., MU, S., HOU, Y., LIN, Z., CHEN, Y., PAN, X., LI, K., LU, Y., WANG, H., TIAN, C., ET AL. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (2021), pp. 4653–4664.