Text Mining Project

17-01-2025

Andrea D'Amicis 869008
Gabriele Sormani 869217
Alisia Sallemi 866453

January 29, 2025

Contents

1 Introduction

This report presents a comprehensive exploration of text mining techniques applied to the Yelp Open Dataset, which includes extensive real-world data on businesses, reviews, and attributes. The text mining techniques used are text clustering and topic modeling, to extract meaningful patterns and insights from user reviews. Clustering is applied to check for the presence of groups based on semantic similarity of relationships, so as to identify meaningful groups that may contain within them knowledge concerning patterns of reviews. Topic modeling was employed to identify the underlying themes or topics present within the reviews. This extract meaningful word groups that define each topic, this makes it possible to identify which words provide the most insight into the content of the reviews. Representing reviews while maintaining context is not an easy task. The project therefore provides a comparison of different methods of representing text, starting with more basic methods such as Bag of Words or TF-IDF, and ending with the use of Sentence Transnformer models, designed specifically to provide a contextual representation of text. For each of the tasks, the performance of different state-of-the-art models such as K-Means, Latent Dirichlet Allocation, and BERTopic were then compared using different text representation techniques. This was done to understand which type of representation allows us to better describe the semantic structure of our data and consequently improve the results of the models.

2 Dataset

The dataset used is the Yelp Open Dataset. It provides real-world data related to businesses including reviews, photos, check-ins, and attributes like hours, parking availability, and ambience. The whole dataset contains over 6000000 reviews, related to over 150000 businesses. To carry out this project, the data related to reviews and business are considered. The textual data are represented by reviews that can be associated with the business information to which they refer.

2.1 Dataset preparation

A sample of the dataset was used in this project, as using the entire dataset of 6000000 instances was too computationally expensive. To select the data, some pre-processing operations were performed. For the purpose of carrying out all activities, two datasets were created, one to carry out the project tasks and one to perform fine tuning of a Sentence transformer. This distinction was made so that the contextual representation model is trained on different data than those used in the project so as to avoid excellent results due to overfitting. The initial operation involved selecting a sample from the dataset: the first 300000 rows for the dataset related to the project tasks, and rows 300001 to 600000 for the dataset related to the fine-tuning. The review data were then merged with the business data. The field of interest useful for extract information related to the review is the *categories* field, this have the following form: *"Active Life, Cycling Classes, Trainers, Gyms, Fitness & Instruction"*. To extract a single label for every review, the following operations are performed:

- The first category value of each review is considered.
- The frequency of the extracted categories are calculated, here are reported the top 6 most frequent categories.

Main category	**count**
Restaurants	51866
Food	19062
Bars	11629
American (Traditional)	8450
Nightlife	8395
American (New)	7635

Table 1: Most frequent categories

- Looking at the table 1, it's clear that the review related to the category *Restaurants* and *Food* are very unbalanced compared to the other. Another problem is that these two categories are very general and include within them all types of restaurants in the reviews. For these reasons these two categories are removed.

- The categories that received less than 2400 reviews were removed.
- 3 categories are removed in order to create some differences between the categories of the project dataset and the categories of the fine-tuning dataset.
- The new sampled dataset is created.

The same process was applied for the creation of both datasets, Below are provided the main characteristics of them:

- **Project Dataset**
  - **Length**: 92,376
  - **Number of Categories**: 19
  - **Categories**:
    - Bars
    - American (Traditional)
    - American (New)
    - Breakfast & Brunch
    - Mexican
    - Beauty & Spas
    - Shopping
    - Seafood
    - Pizza
    - Sandwiches
    - Italian
    - Automotive
    - Coffee & Tea
    - Burgers
    - Hotels & Travel
    - Home Services
    - Ice Cream & Frozen Yogurt
    - Sushi Bars
    - Health & Medical
- **Fine-Tuning Dataset**
  - **Length**: 86,522
  - **Number of Categories**: 19
  - **Categories**:
    - Bars
    - American (Traditional)

- American (New)
- Breakfast & Brunch
- Mexican
- Beauty & Spas
- Shopping
- Seafood
- Pizza
- Sandwiches
- Italian
- Automotive
- Coffee & Tea
- Burgers
- Hotels & Travel
- Home Services
- Ice Cream & Frozen Yogurt
- Sushi Bars
- Health & Medical

3 Sentence transformer fine tuning

In this project, contextual text representation methods will be used; the model used is the Sentence transformer all-MiniLM-L6-v2. This is a lightweight model to provide a representation of text, to be used as input for the various algorithms used. To adapt this model to the context of the Yelp dataset, fine-tuning was carried out, with the goal of improving the representation of reviews and improving performances particularly with regard to text clustering.

3.1 Training setting

The dataset used for fine tuning, is the one specially created in the dataset preparation phase. To do this, it is necessary to transform the dataset into a set of InputExamples, which are needed to perform the training. The structure of these is determined by the loss function used. The loss function used is the triplet loss. This loss is designed to assist training models to learn an embedding where similar data points are closer together and dissimilar ones are farther apart, enabling robust discrimination. This function is composed by 3 elements: the anchor, the positive and the negative instance, with the goal of minimizing the distance between an anchor and a positive instance. This allow to train the model in creating well separated and coherent groups of data point, in the reviews domain. Given the loss function, the data are organized so that a random text is set as anchor, another review belonging to the same category as positive, and a review belonging to a different category as negative. For training the model 40000 training instances are created. the details of the fine tuning are reported in the following table:

Train instances	40000
Loss	BatchTripletLoss
Learning rate	0.00002 (default one for W&B APIs)
Epochs	3
warmup steps	100

An example of the Triplet Loss applied to our data is reported below:

	Label	Text
Anchor	7	I'm one of their regulars and the food is pretty consistent. Their BBQ chicken pizza with thin crust and Chicken Alfredo pasta are amazing! We order from them weekly and delivery is pretty fast. I don't know why the reviews are so low because they never disappoint me.
Positive	7	Never have left a restaurant so upset and disappointed in a experience. Took over an hour for food on a Monday night with a half empty restaurant. No apology no excuses, just a simple lack of caring for the service of their guests. Food was good, but it's not worth the long wait, lack of service, and utter disregard for guests. I'd take my business elsewhere.
Negative	3	We stayed here for a long weekend last year. 2 kids, 1 a toddler and we had a great time. All the restaurants were good. My son and husband had fun at the arcade. Hubs enjoyed gambling. My favorite hotel in Reno.

4 Text Pre-Processing

Text preprocessing is a crucial step when working with textual data. Its primary purpose is to reduce dimensionality and eliminate noise in the text, while maintaining consistency and preserving the semantic relationships between words within phrases.

We applied a range of basic techniques to simplify subsequent tasks, such as text representation and model analysis. Since we employed multiple types of text representation, we used varying combinations of text preprocessing steps to optimize results.
As these techniques are language-dependent, we utilized specific libraries (e.g., ‘nltk‘) configured for the English language. Below is an overview of the preprocessing steps we applied:

Tokenization
This process involves breaking a stream of text into meaningful units, called tokens. Each token is a sequence of characters separated by spaces, punctuation marks, or other special characters. We implemented this step using the ‘word tokenize‘ function from ‘nltk.tokenize‘.

Normalization
Normalization ensures that the text is standardized by:

- *Removing non-alphabetic symbols*: Special characters, punctuation marks, and numbers were excluded using regular expressions.
- *Converting text to lowercase*: All words were transformed into lowercase to ensure uniformity.

Stop Words Removal
This step involves removing common words that contribute little to the semantic meaning of a sentence, such as ”the,” ”and,” or ”is,” as they dominate the text without providing significant context. Additionally, we created a custom list of words (e.g., ”could,” ”would,” ”us,” ”i’m,” ”you’d,” etc.) to further reduce noise during the creation of TF-IDF embeddings leading to more efficient memory usage . This step was implemented using ‘stopwords.words(’english’)‘ from ‘nltk.corpus‘.

Lemmatization
Lemmatization reduces words to their base or root forms, minimizing variation or inflections while retaining meaning. We implemented this step using the ‘WordNetLemmatizer‘ from ‘nltk.stem‘. Lemmatization process was guided by *Part-of-Speech* (POS) tags. Only key lexical categories—nouns, verbs, adjectives, and adverbs—were considered, as they provide the most context.

This structured and thorough preprocessing workflow allowed us to clean the text, optimize memory usage, and reduce computational workload, thereby ensuring that subsequent tasks were both efficient and meaningful.

5 Text Clustering

Text clustering is the process of grouping a set of objects (in this case, reviews) into classes of similar items. As clustering is an *unsupervised learning* task, there is no prior knowledge of the classes within the dataset. The goal is to uncover hidden patterns or structures and transform this unlabelled data into actionable knowledge.

Since the number of distinct document classes in our dataset is unknown, we adopted the *elbow method* —a data-driven approach— to estimate the optimal number of clusters.

We employed *K-means* as the flat clustering model for this hard clustering task. This means we assumed that: There is no explicit structure defining relationships between clusters and each document is assigned to exactly one cluster. This was further validated using multiple internal evaluation metrics, such as the Shilouette Score, the Inertia and the Davies-Bouldin Index, to assess the quality of the clustering and ensure effective separation among clusters.

To enable k-means to capture *semantic similarity*, we experimented with different text representations:

- *Bag of Words (BoW)*: TF-IDF.
- *Word Embeddings*: GloVe as a count based model and Doc2Vec as a predictive model.
- *Contextualised Word Embeddings*: sentence transformer BERT based.

This exploration allowed us to determine which representation best captured the relationships between words and the general context of each review.

5.1 Text Representation

As previously mentioned, we employed various text representation techniques to capture both the semantic meaning and the context of each review.

Each method represents words in the text using vectors, with distinct approaches to encoding their meaning.

5.1.1 BoW - TF-IDF

TF-IDF is a numerical statistic used to assess the importance of a term within a collection of documents. It combines two factors: the term's frequency within a specific document (TF) and its inverse frequency across the entire document corpus (IDF). By weighing terms based on their frequency in individual documents and their rarity in the broader collection, TF-IDF values help highlight the most relevant features, making it a valuable tool for feature extraction.
It's formulated as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

The pre-processing steps for this type of vectorization include tokenization, normalization, stop words removal and lemmatization.

5.1.2 Word Embedding - GloVe and Doc2Vec

Word embeddings are dense vector representations of words in a continuous vector space. They capture semantic and syntactic relationships between words based on their distribution in a corpus. Unlike traditional methods like TF-IDF, which assign a static weight to each word, embeddings map words into a low-dimensional space where similar words are closer.

GloVe

It is a count-based model that calculates the frequency with which each word co-occurs with its neighboring words in a large text corpus. These co-occurrence statistics are then mapped to a small, dense vector for each word. Specifically, the model leverages statistical information by training exclusively on the non-zero elements of a window-based co-occurrence matrix. It's formulated as follows:

$$J = \sum_{i,j=1}^{V} f(P_{ij}) \left(w_i^\top w_j + b_i + b_j - \log P_{ij} \right)^2$$

Where:

$$\begin{aligned} P_{ij} &: \text{Probability of words } i \text{ and } j \text{ co-occurring.} \\ w_i, w_j &: \text{Word embeddings for } i \text{ and } j. \\ b_i, b_j &: \text{Bias terms.} \\ f(P_{ij}) &: \text{Weighting function to scale co-occurrence counts.} \end{aligned}$$

The preprocessing steps for this type of word embedding include tokenization, normalization, and the removal of stop words.

Doc2Vec

Doc2Vec is an extention of Word2Vec, a predictive model, to represent entire documents as dense vectors. Unlike typical word embeddings, it generates embeddings for documents by considering word order and the document's unique identity. The Word2Vec model aims to maximize the predicted log-probability for a sequence of words in context as follows:

$$\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t | w_{t-k}, \dots, w_{t+k}),$$

where k is the window size for preserving contextual information. The probability is computed using the softmax function, in which each y_i, is the i-th output value of the model and is calculated as follows:

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W).$$

where b denotes the bias terms between the hidden and output layer, U denotes the weight matrix between the hidden and output layer, h denotes the average or concatenation for context words, and W denotes the word embedding matrix. In the Doc2Vec model, each document is associated with a unique vector, represented as a column in the matrix D. The updated formulation for Doc2Vec incorporates the document vector as follows:

$$y = b + Uh(w_{t-k}, \ldots, w_{t+k}; W, D).$$

The preprocessing steps for this type of word embedding include tokenization, normalization, and the removal of stop words.

5.1.3 Contextualised Word Embeddings - BERT

Contextualised word embeddings dynamically generate different vector representations for a word depending on its context. These embeddings are typically derived from Neural Networks and transformer-based models like BERT, GPT, or Sentence Transformers. Sentence Transformers are advanced contextualised embedding models designed to generate meaningful sentence-level representations. These embeddings are derived from transformer-based models like BERT and fine-tuned for specific tasks such as sentence similarity.

The main idea is to use BERT as a base model and develop a fine-tuned model on our datasets to create embeddings that preserve semantic meaning at the sentence level. In this way we obtain embeddings where the same word can have different representations depending on its context within the sentence, leveraging the transformer architecture.

The pre-processing step for this type of contextualised word embedding include only text normalization.

5.2 Elbow Method

The elbow method is a technique used to determine the optimal number of clusters (k) in a clustering algorithm, most commonly with K-means. It helps identify the point at which adding more clusters no longer provides significant improvement in the clustering results. This method evaluates clustering performance across different values of k using various metrics to identify the best balance between cluster compactness and separation and the ”elbow” in the curve is the point where the rate of improvement slows significantly. This point represents the optimal trade-off between the number of clusters and the clustering quality. In our case, we decided to set $k = 15$ as the maximum number of clusters for each type of document representation.

This is a general overview of how it work practically the elbow method:

1. The dataset is clustered multiple times using K-means, with the number of clusters varying across a range of k values.
2. For each k, metrics are calculated to assess the clustering performance.
3. The results for each metric are plotted against k to visualize their trends.

5.3 Clustering Evaluation methods

Evaluation by the elbow method was carried out taking into account three internal metrics explained below.

5.3.1 Silhouette Score

The Silhouette is a metric used to evaluate the quality of clustering. It measures assesses how closely clusters are separated from each other. The Silhouette value ranges from -1 to 1, where:

- A value close to 1 indicates that the point is well assigned to its cluster and far from neighboring clusters.
- A value close to 0 suggests that the point is on or near the boundary between clusters.
- A negative value (< 0) indicates that the point might be assigned to the wrong cluster.

Formula
The Silhouette for a single data point i is calculated as follows:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:

- $a(i)$ is the average distance between the point i and all other points in the same cluster (cohesion).
- $b(i)$ is the average distance between the point i and the points in the nearest neighboring cluster (separation).

The overall Silhouette score for the clustering is obtained by averaging $s(i)$ for all data points.

Interpretation
A high Silhouette value indicates well-defined clusters, with data points well separated between different clusters and tightly grouped within the same cluster. This makes the Silhouette score a useful metric for comparing different clustering configurations and selecting the optimal number of clusters k.

5.3.2 Inertia

Inertia is a metric used to evaluate the compactness of clusters in a clustering algorithm. It measures the sum of squared distances between each data point and the centroid of the cluster it belongs to. Lower inertia values indicate tighter and more compact clusters.

Formula
The inertia for a clustering result is calculated as follows:

$$\text{Inertia} = \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - \mu_k\|^2$$

Where:

- K is the number of clusters.
- C_k is the set of points assigned to cluster k.
- x_i is a data point in cluster k.
- μ_k is the centroid of cluster k.

Interpretation
Inertia provides an indication of how well the data points are grouped around their respective centroids. A lower inertia value suggests that the data points are closely packed within their clusters. However, inertia tends to decrease as the number of clusters increases, which is why it is often used in combination with other metrics, such as the Silhouette Score as in our case, to determine the optimal number of clusters k.

5.3.3 Davies-Bouldin Index

The Davies-Bouldin Index (DBI) is a metric used to evaluate the quality of clustering by analyzing both the compactness of clusters and their separation. Lower DBI values indicate better clustering, as they represent more compact and well-separated clusters.

Formula
The Davies-Bouldin Index is calculated as follows:

$$DBI = \frac{1}{K} \sum_{k=1}^{K} \max_{j \neq k} \left(\frac{\sigma_k + \sigma_j}{d_{k,j}} \right)$$

Where:

- K is the number of clusters.
- σ_k is the average distance between each point in cluster k and its centroid (compactness of cluster k).
- $d_{k,j}$ is the distance between the centroids of clusters k and j (separation between clusters k and j).

- The term $\max_{j \neq k}$ identifies the cluster j that has the worst separation relative to cluster k.

Interpretation
The Davies-Bouldin Index considers both intra-cluster compactness and inter-cluster separation:

- A **lower DBI** indicates that clusters are compact and well-separated, which is desirable in clustering.
- A **higher DBI** suggests that clusters are overlapping or not well-defined.

This metric is particularly useful for comparing different clustering configurations and selecting the optimal number of clusters k. It provides an objective measure that balances the trade-off between compactness and separation.

5.3.4 Rand Index

The Rand Index is a metric used to evaluate the agreement between two clustering solutions. It measures the similarity between the predicted clustering and the ground truth labels by comparing pairs of data points and determining whether they are assigned to the same or different clusters in both solutions. The Rand Index value ranges from 0 to 1, where:

- A value of 1 indicates perfect agreement, meaning the clustering perfectly matches the ground truth.
- A value of 0 indicates no agreement, meaning the clustering is entirely independent of the ground truth.

Formula
The Rand Index is computed as follows:

$$\text{Rand Index} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP (True Positives): The number of pairs of points that are in the same cluster in both the predicted clustering and the ground truth.
- TN (True Negatives): The number of pairs of points that are in different clusters in both the predicted clustering and the ground truth.
- FP (False Positives): The number of pairs of points that are in the same cluster in the predicted clustering but in different clusters in the ground truth.
- FN (False Negatives): The number of pairs of points that are in different clusters in the predicted clustering but in the same cluster in the ground truth.

Interpretation
A high Rand Index indicates that the clustering solution aligns closely with the ground truth, reflecting good clustering performance.

5.4 Results obtained

In this section are provided the results obtained for each type of document representation and evaluated with the metrics defined above expect for Rand Index.

5.4.1 TF-IDF

The clustering results of document representation with TF-IDF seem rather poor. Probably BoW, disregarding context and semantics, is not a sufficient type of document encoding for our purposes. The silhoutte values continue to grow without reaching an elbow point and the same for the inertia values. DBI reaches a minimum peak with 4 clusters, but from silhoutte is understood that clusters are, probably, quite overlapped and not very separable. Alternative and more sophisticated representation could help clustering to achieve better results.

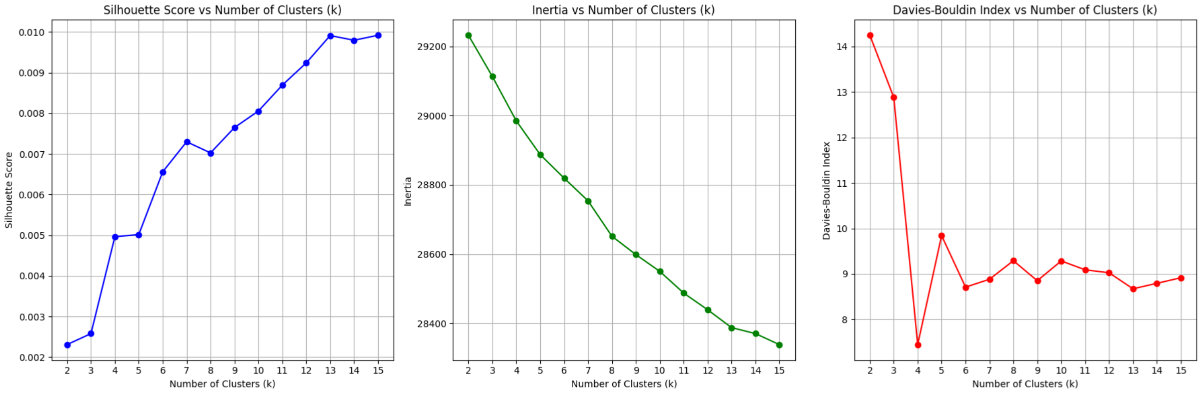

Figure 1: Plots of Silhoutte, Inertia and DBI related to TF-IDF

5.4.2 GloVe

The clustering results with the Glove-based document representation still seem to fall short. The silhoutte values continue to decrease reaching an elbow point that is orrible. The other metrics have pretty bad results as in previous case and don't show clear indications of well-defined clusters. Although there is some variation between different k-values, the overall performance suggests that the data may not naturally form distinct clusters and that alternative and more sophisticated solutions are needed.

5.4.3 Doc2vec

The clustering results with the current approach still appear suboptimal. The Silhouette values not only decrease but also reach negative values, indicating significant misclassifications and poorly formed clusters. This represents the worst performance among all the metrics analyzed. The other metrics also yield disappointing results, failing to provide any clear indication of well-defined clusters. While there is some variability across different k-values, the overall outcome strongly suggests that the data lacks a natural

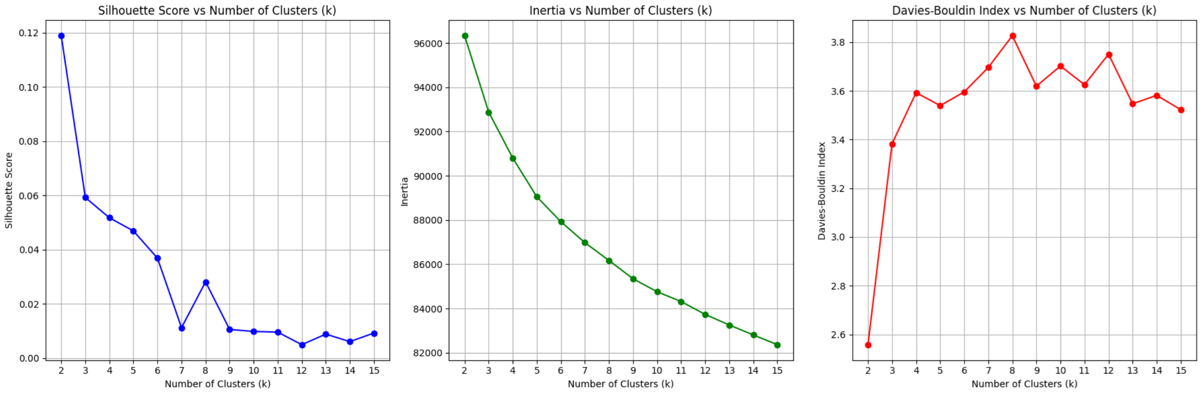

Figure 2: Plots of Silhoutte, Inertia and DBI related to GloVe

clustering structure, and alternative or more advanced methods should be explored to address these challenges.

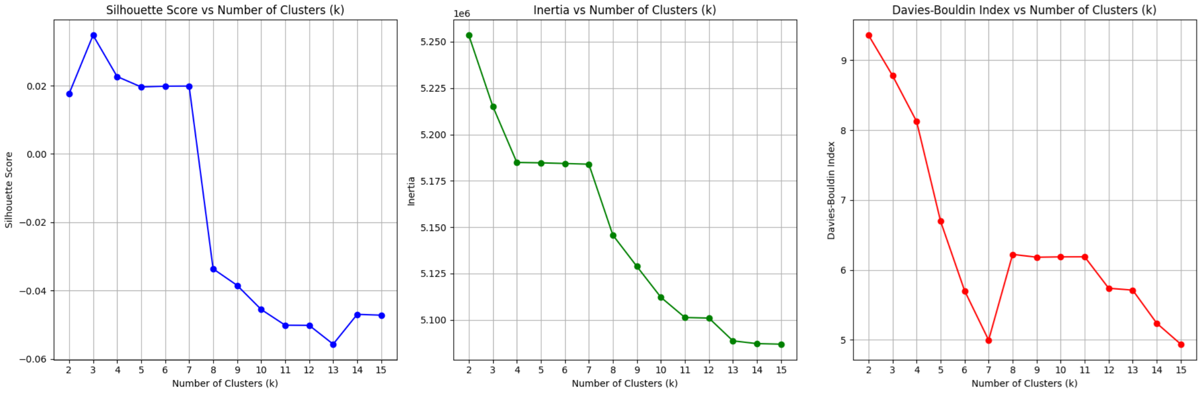

Figure 3: Plots of Silhoutte, Inertia and DBI related to Doc2vec

5.4.4 BERT fine-tuned

As an additional test, we decided to fine-tune BERT on our dataset to evaluate whether the results could be further improved. With this final version, the metric outcomes are remarkably clear, showing a well-defined elbow point at $k = 5$, where the Silhouette Score reaches an impressive 0.9372 achieving a good result for Davies-Bouldin Index. The fine-tuning of the model provided the crucial enhancement needed to define the clusters even more clearly, resulting in well-separated, highly compact, and well-defined groupings.

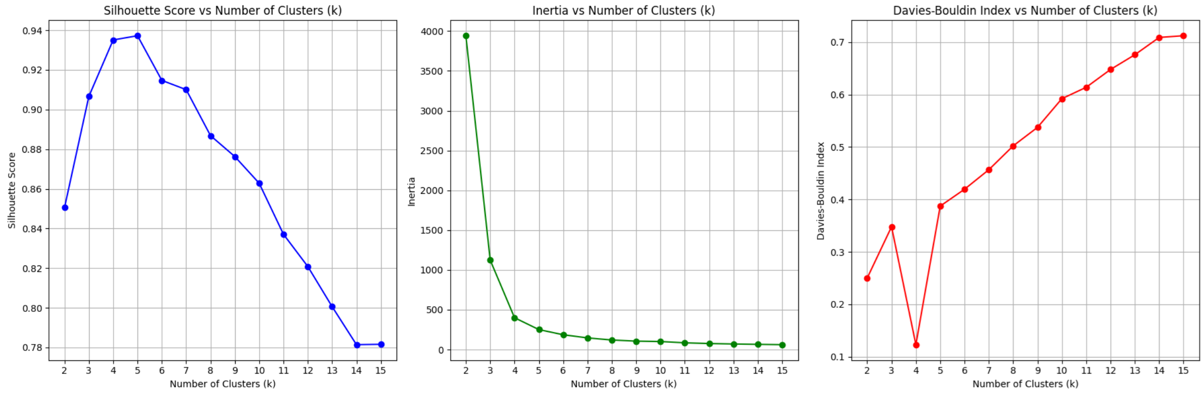

Figure 4: Plots of Silhoutte, Inertia and DBI related to BERT fine-tuned

5.4.5 Elbow method summary

To sum up the results achieved with elbow method is provided below a table that take the best results for each model.

Table 2: Clustering results

Model	**k**	**Silhouette**	**Inertia**	**DBI**
TF-IDF	13	0.0099	28387.7595	8.6726
Glove	2	0.1188	96329.2997	2.5578
Doc2Vec	3	0.0347	5215163.6658	8.7797
BERT fine-tuned	5	0.9372	250.9741	0.3873

5.5 Further Investigations

Once the clustering was done with K-Means and the optimal number of clusters was found, $k = 5$, we wondered what these clusters could represent on a discourse level. We wondered if they could somehow represent the stars associated with reviews or trivially the division of review categories. This final section then included a further in-depth analysis applying K-Means with 5 and 19 clusters using Rand's index with respect to 5 classes of stars associated with the reviews and with respect to the 19 categories.

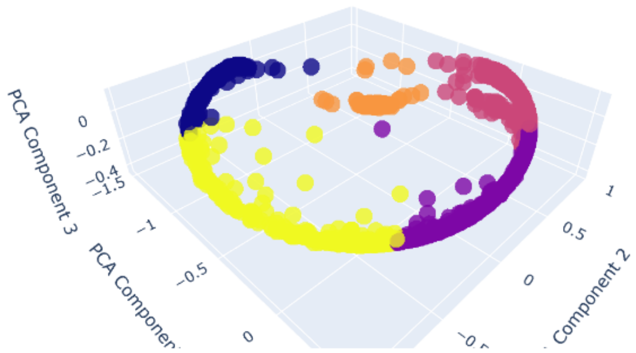

Figure 5: 3D plot of 5 clusters related to stars of reviews

To visualise the clusters, we used the PCA algorithm while keeping only the 3 main dimensions of the embeddings, allowing us to create a 3-dimensional plot. The result shows a large circular cluster divided into various points by the various clusters created by K-Means. Although it is not particularly suitable for the detection of spherical clusters, it nevertheless seems to achieve a good level of accuracy when compared ideally with the 5 evaluation classes of the reviews, reaching a Rand Index value of 0.5.

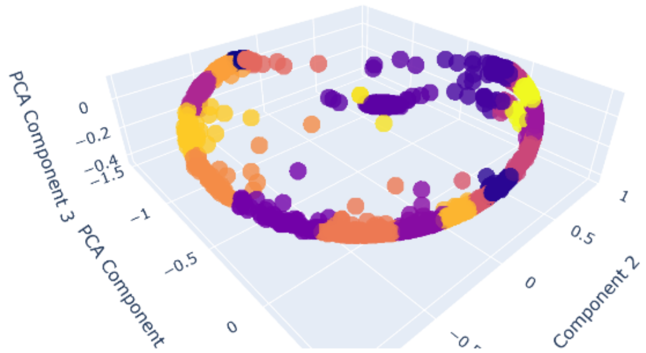

Figure 6: 3D plot of 19 clusters related to stars of reviews

An additional analysis was conducted by considering the 19 review categories available in our dataset. Similar to the previous approach, we utilized the PCA algorithm to reduce the dimensionality of the embeddings, retaining only the top 3 components to enable visualization in a 3D plot. The resulting visualization reveals a large circular structure, segmented into distinct regions corresponding to the clusters formed by K-Means. Despite K-Means not being inherently ideal for identifying non-spherical clusters, it appears to perform reasonably well in this case. When compared to the 19 review categories, the method achieves a Rand Index score of 0.61, indicating a decent level of clustering accuracy.

5.6 Conclusions

The best clustering with KMeans was achieved by representing sentences through a BERT model fine-tuned on our dataset, reaching an impressive Silhouette score of 0.9372. This result highlights the effectiveness of BERT in capturing the semantic relationships within the text. Further analysis revealed a large circular cluster in the 3D space, which likely indicates that the BERT representation in the multidimensional space is capable of capturing the underlying structure of the data. This structure, probably, may correspond to the various categories and star ratings associated with the reviews, suggesting that the model has successfully learned to distinguish between different review types and topics. These findings underline the potential of BERT for semantic clustering tasks and its ability to provide meaningful insights from textual data.

6 Topic modeling

Topic modeling is an unsupervised machine learning technique used to group groups of words that make sense together, which are interpreted as topics. This task is based on the idea that each document is a combination of one or more topics, and that each topic is characterized by a set of words. In this project the goal is to identify which words within the reviews of the dataset can be grouped within the same topic and which groups are created.

This project consists of applying different algorithms to perform topic extraction, then these will be evaluated and compared with each other to identify which model best identifies topics in reviews. The text representation used, the models and the evaluation metrics are explained in the next chapters.

6.1 Methods

The following chapter will cover the text representation techniques employed, the models utilized, and the evaluation metrics applied.

6.1.1 BoW

To create a representation of the text, it is necessary to focus on the representation of individual words, which will be subject to grouping in topic. The word representation technique that is employed is Bag-of-Words. The goal is to associate every document in the dataset to the list of its words. This means that the features passed to the topic modeling models are the unique words contained in the documents, without considering their order and context. This kind of word representation is used by the LDA algorithm.

6.1.2 Sentence transformer

In order to use BERTopic, another text representation technique is used. In this case, the sentence transformer model all-MiniLM-L6-v2 without any kind of fine tuning is employed. This allow to provide a contextual representation of the documents in the dataset. Using such a representation is expected to have improvements in performance as the words are represented within the context of the review. This type of representation is used for the BERTopic model.

6.1.3 LDA

Latent Dirichlet Allocation (LDA) is a Bayesian network, meaning it's a generative statistical model, it assumes that documents are produced from a mixture of topics. The idea behind LDA is that: each document has a particular probability of using particular topics to generate a given word. The input of LDA is a corpus containing the BoW representation of various documents. This model needs to select a fixed number of topics, which will be extracted from the text. The outputs of the model are: the topic model, and the documents expressed as a combination of the topics.

6.1.4 BERTopic

BERTopic is an advanced topic modeling algorithm that uses Transformers-based embedding models to generate semantic representations of texts and group them into topics. After the documents have been represented using all-MiniLM-L6-v2, BERTopic uses a dimension reduction algorithm called UMAP, to reduce the complexity of the vectors. After size reduction, BERTopic applies a clustering algorithm called HDBSCAN, which finds clusters in the data and automatically identifies the optimal number of clusters, while also handling outliers. For each cluster generated by HDBSCAN, BERTopic determines a set of representative keywords.

6.1.5 Evaluation metrics

To evaluate the models' abilities in extracting topics in documents and to compare them with each other, several topic modelling performance metrics were considered.

Coherence
The coherence measure the degree of semantic similarity between high scoring words in the topic. A good topic should be interpretable and have representative words that "make sense" together. To calculate this metric, special models are defined that take into account co-occurrences i.e. how often these words appear together in documents and semantic similarity, which indicates how close the words are semantically. The higher this metric the better.

Perplexity
Perplexity is a statistical measure of how well a probability model predicts a sample. It indicates how much uncertainty the model has when assigning probabilities to documents. The perplexity is an output of a probabilistic model like LDA. Since this is related to probabilistic models, is not using to evaluate BERTopic.

Diversity
Topic diversity assesses how distinct the topics are from each other. A good topic model should produce topics that do not overlap too much. This metric is calculated measuring the cosine similarity between topic vectors or quantifying the spread of topics across documents.

6.2 LDA results

To test the LDA algorithms, the documents in the dataset were represented using the Bag-of-Words technique. To identify the model with the best performance, 4 different models were evaluated, the difference between them being the number of topics to be identified. One model has 5 topics, one has 10, another has 15, and the last has 20. For each of these models, all evaluation metrics were evaluated and the word cloud showing the most relevant words for each topic was displayed, as well as the distribution of topics among the documents. All the results of the experiment are reported below.

6.2.1 LDA with 5 topic

The first model to be tested is the LDA with only five topic, the topics identified in this experiment are showed below.

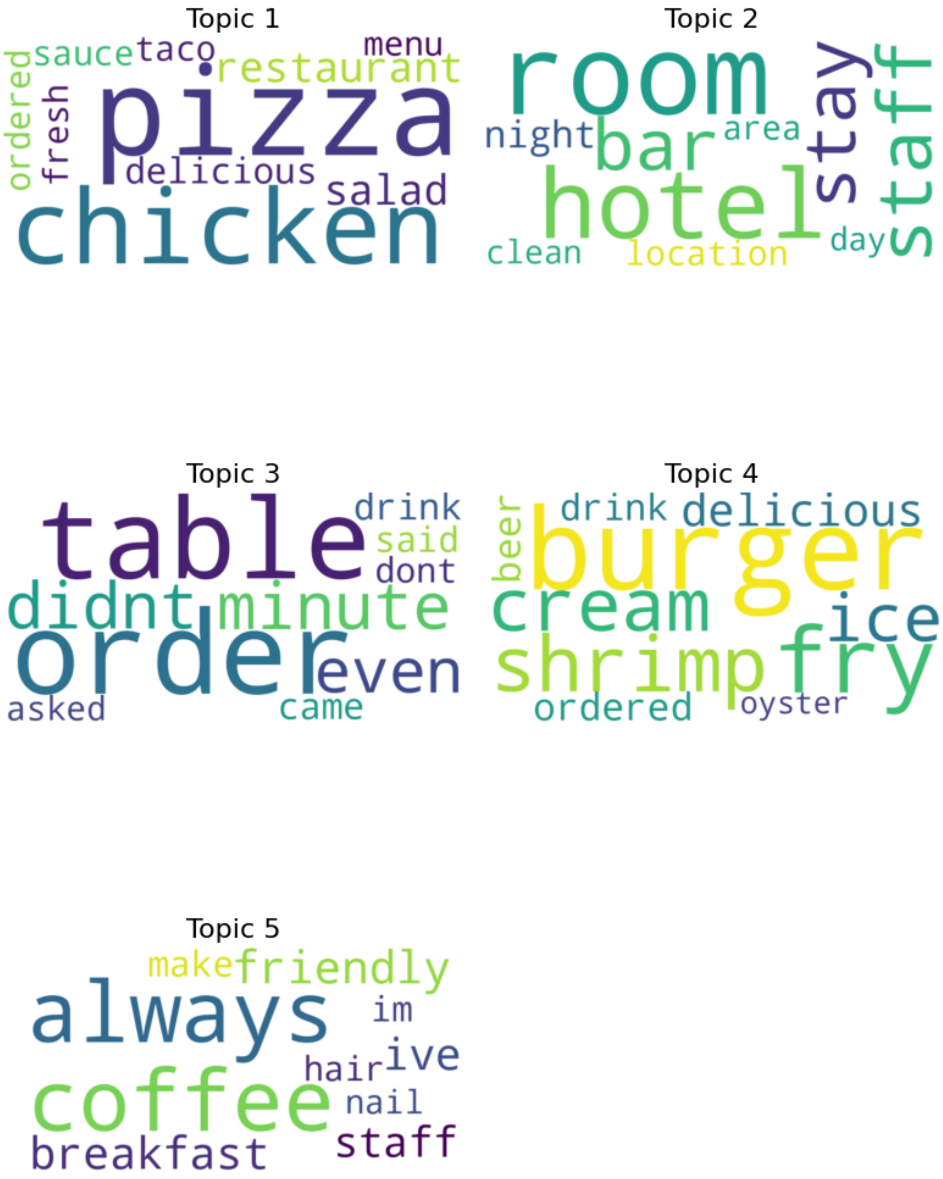

Figure 7: Word cloud for LDA with 5 topics

The distribution of topics among documents is as follows.

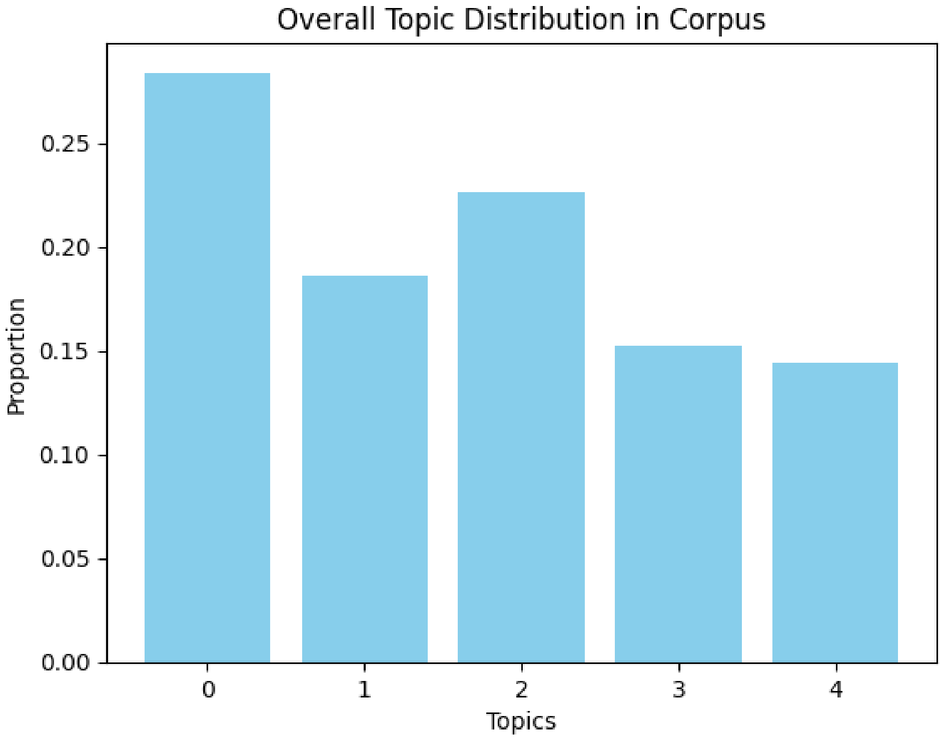

Figure 8: Word cloud for LDA with 5 topics

The evaluation metrics for this experiment are:

- Coherence = 0.43
- Perplexity = -7.97
- Diversity = 0.77

Looking at the results, it can be seen that more heterogeneous terms are present in the 5 identified topics, particularly in topic 5 and 4. Words referring both to reviews related to beauty centers and to bars and places to have breakfast are included in the same topic. The first topic contains words all related to food, but fails to distinguish between the various types of restaurants considered in the dataset. The other topics contain generic terms that are probably shared among multiple reviews. These results suggest the possibility of improving topic modeling by increasing the number of topics, as the values of the metrics can also be improved especially the 0.77 of diversity. Looking at the distribution of topics in the documents, we see that topic 0 related to food is the predominant one, next is topic 2 which contains generic terms so again the result is reasonable but as mentioned above to be improved.

6.2.2 LDA with 10 topic

The second experiment is carried out using 10 topics in the model. the results are reported below.

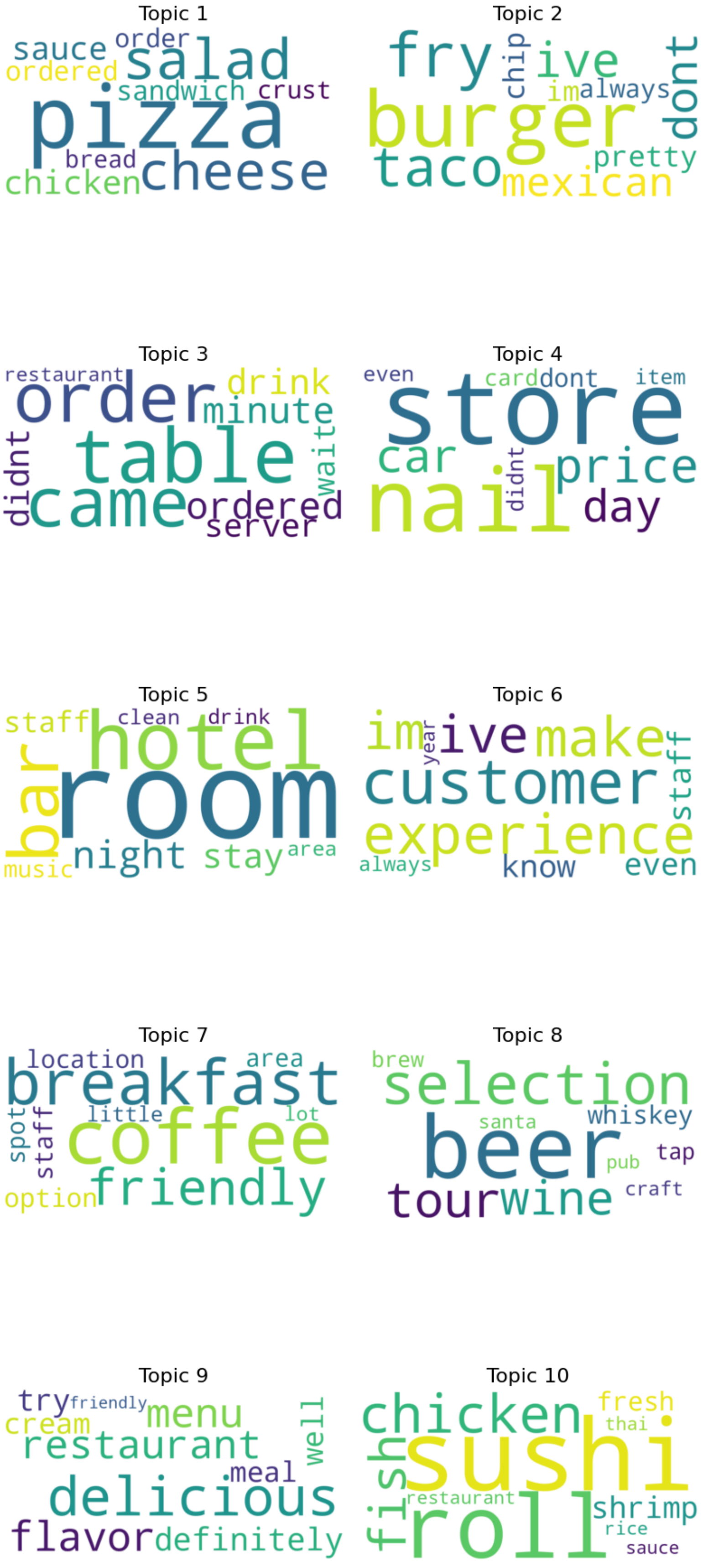

Figure 9: Word cloud for LDA with 10 topics

The distribution of topics among documents is as follows.

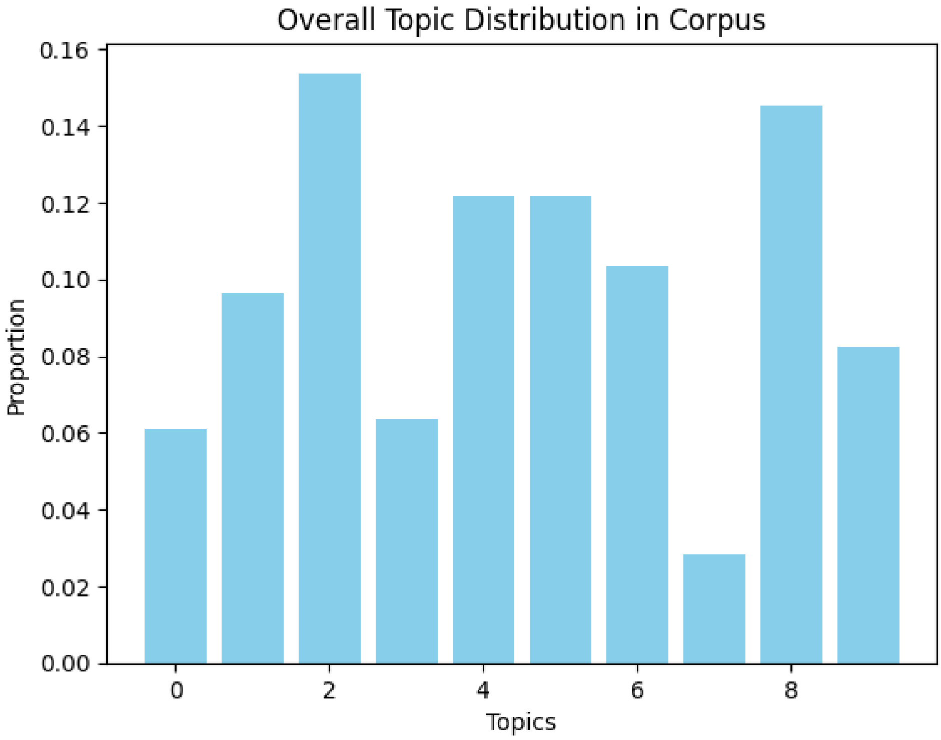

Figure 10: topic distribution for LDA with 10 topics

The performance metrics of this mdoel are reported below:

- Coherence = 0.44
- Perplexity = -8.3
- Diversity = 0.87

Looking at the results we can see that in this case the algorithm was able to divide different types of restaurants, this is particularly noticeable in topics 1,2 and 10. Also in this case, there are topics that contain generic words, probably contained in more reviews. Coherence and diversity improved compared to the previous test, especially diversity, this was confirmed by the graphs showing topics capturing groups of words that are less correlated with each other, managing to differentiate more between the different activities reviewed. Looking at the distribution of topics we see that topic 8 related to Bars is the most frequent, consistent with the labels of the reviews.

6.2.3 LDA with 15 topic

The next experiment is made using LDA with 15 topics.

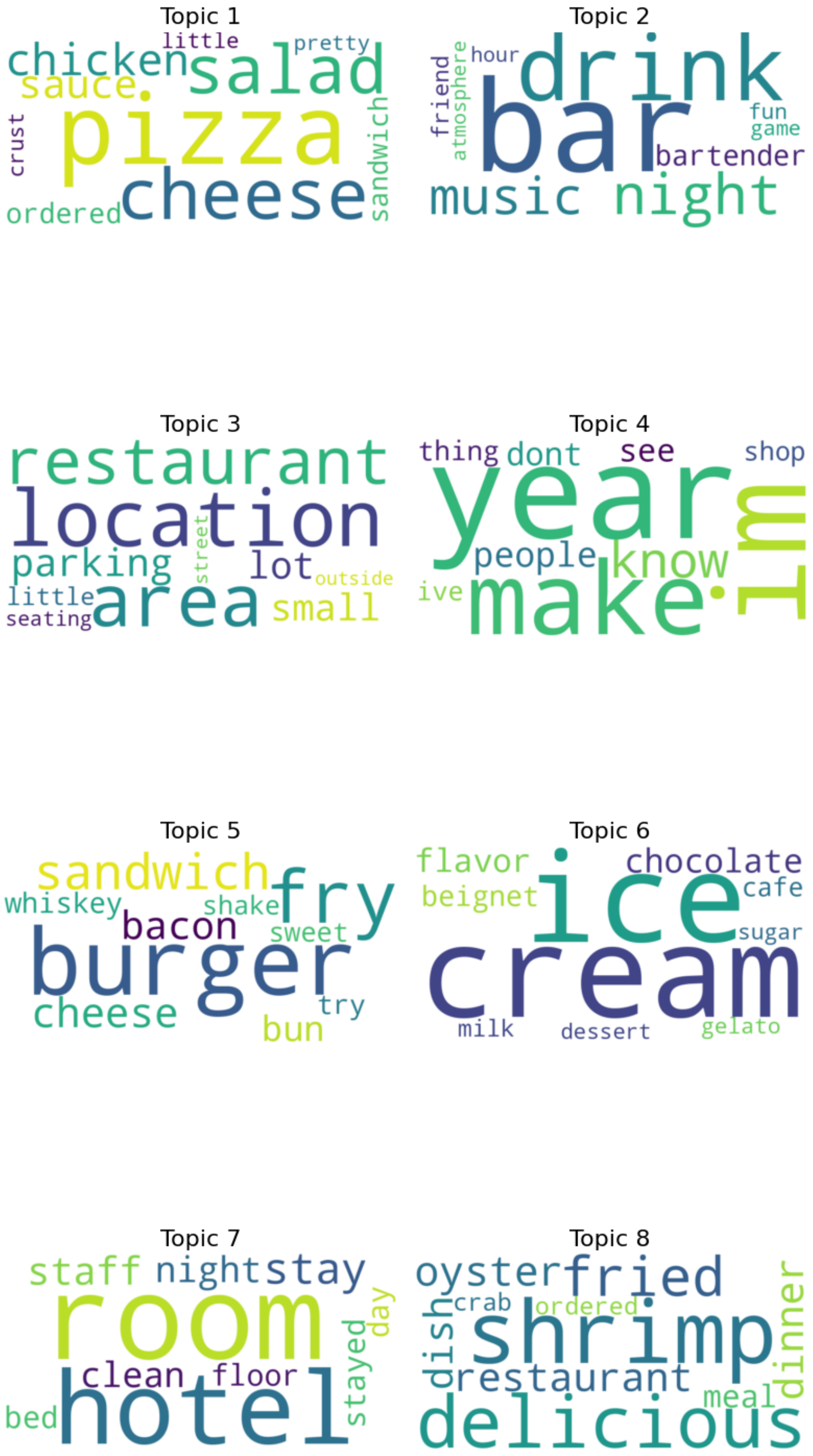

Figure 11: Word cloud for LDA with 15 topics

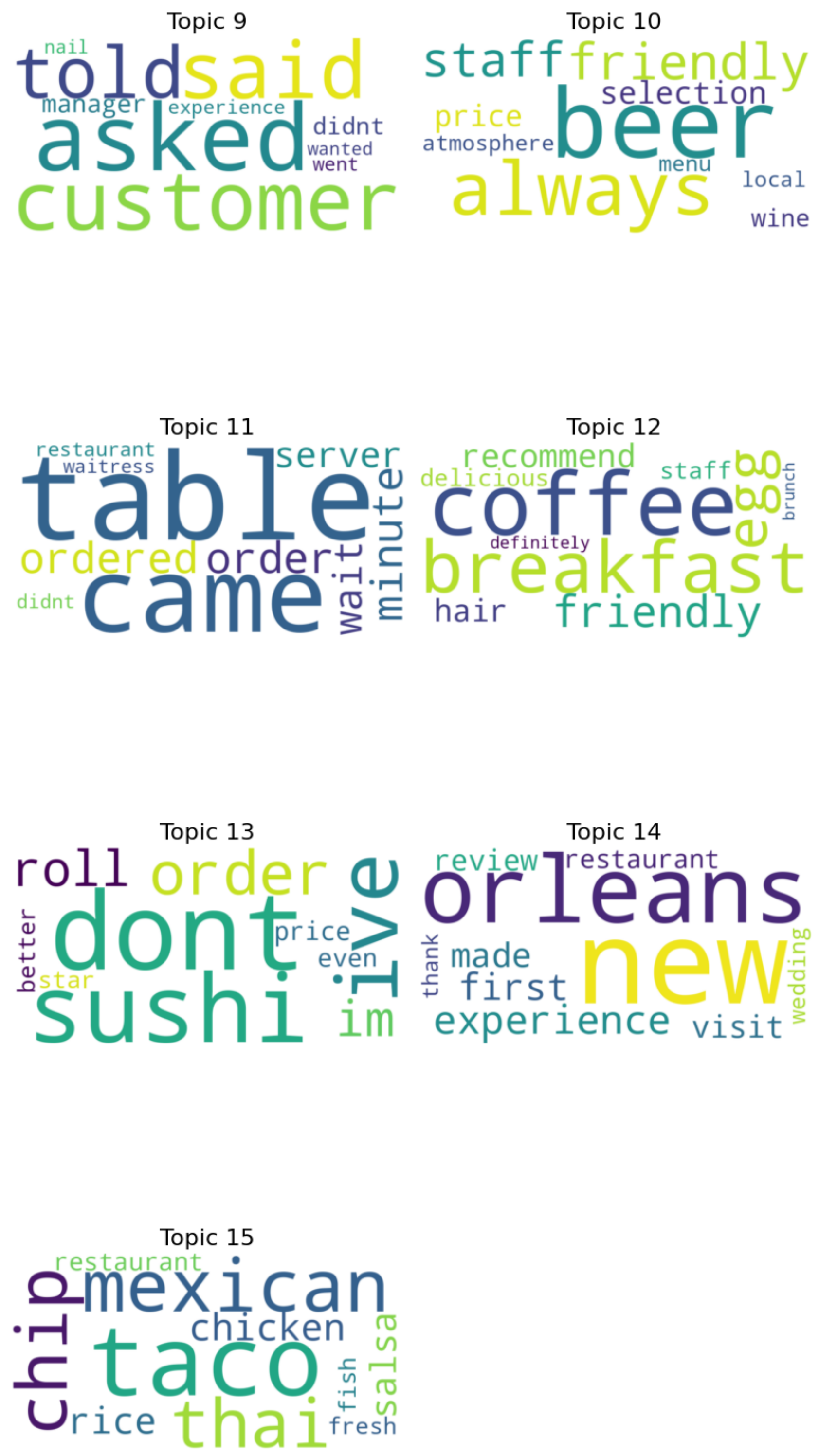

Figure 12: Word cloud for LDA with 15 topics

The distribution of topics among documents is as follows.

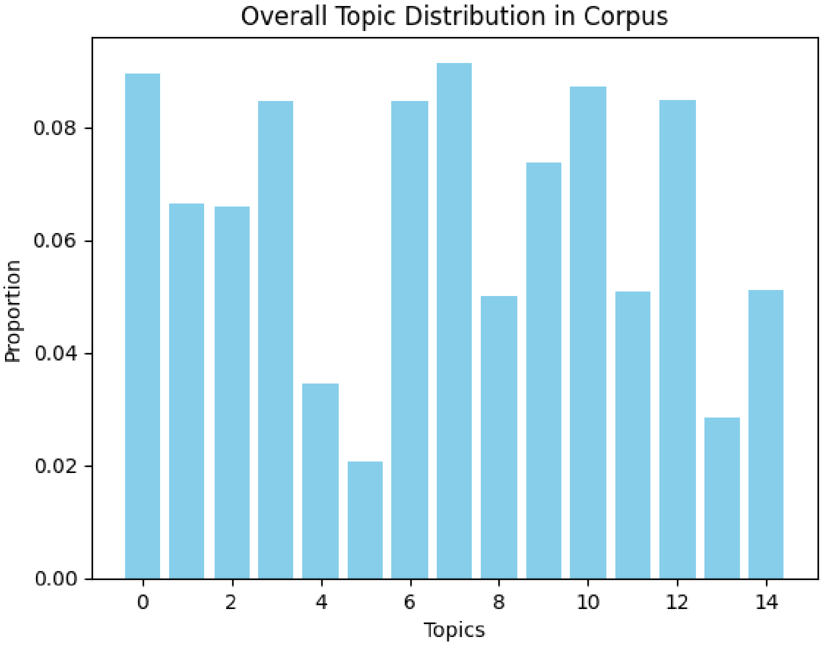

Figure 13: topic distribution for LDA with 15 topics

The evaluation metrics of this experiment are:

- Coherence = 0.49
- Perplexity = -8.9
- Diversity = 0.88

The results reveal clear distinctions between the topics identified, as reflected in the differences between the words displayed on the WordCloud graph. These variations also highlight the diversity of business types reviewed within the dataset. The evaluation metrics further support these findings, showcasing the highest values observed so far. This indicates that the identified word groups are both distinct one from another and internally consistent in their syntax, demonstrating an effective and meaningful breakdown of the topics.

6.2.4 LDA with 20 topic

The results of the application of LDA with 20 topics are shown below:

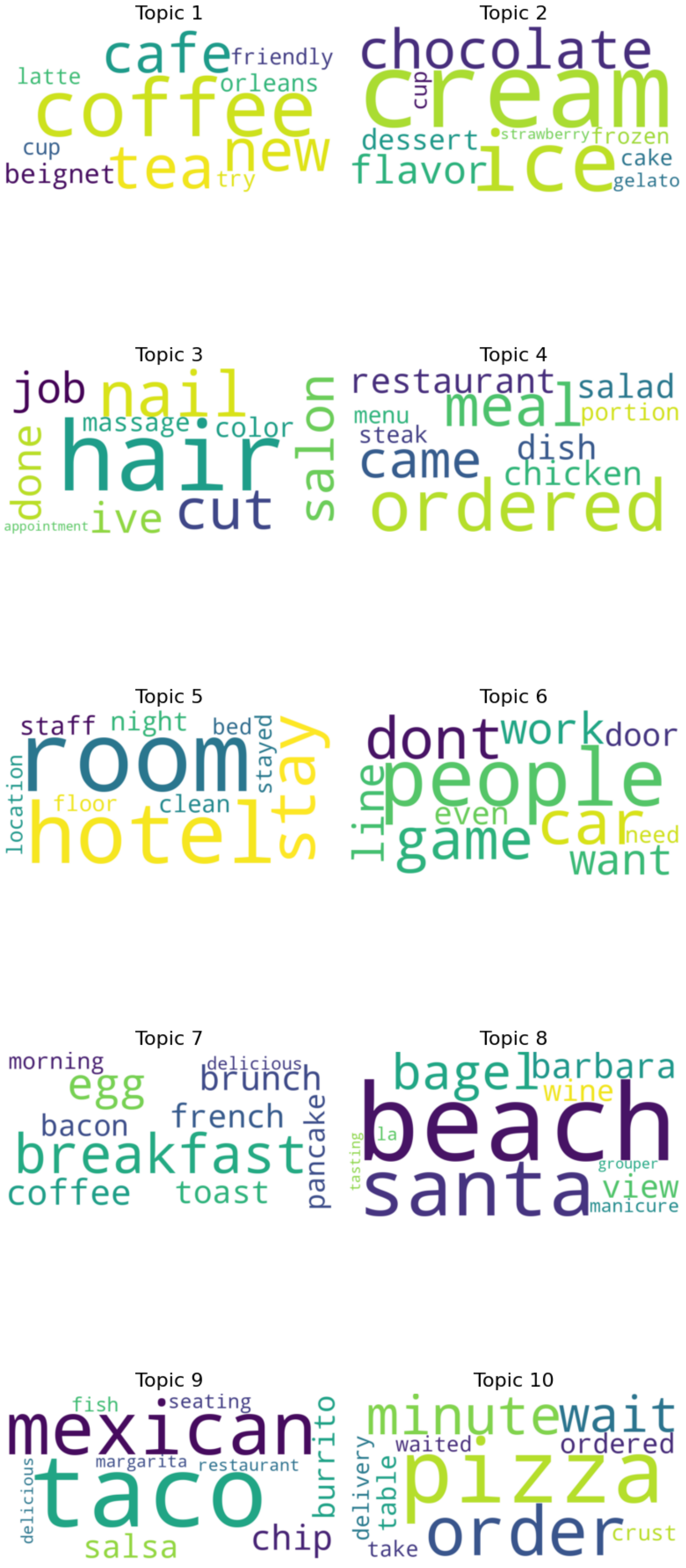

Figure 14: Word cloud for LDA with 20 topics

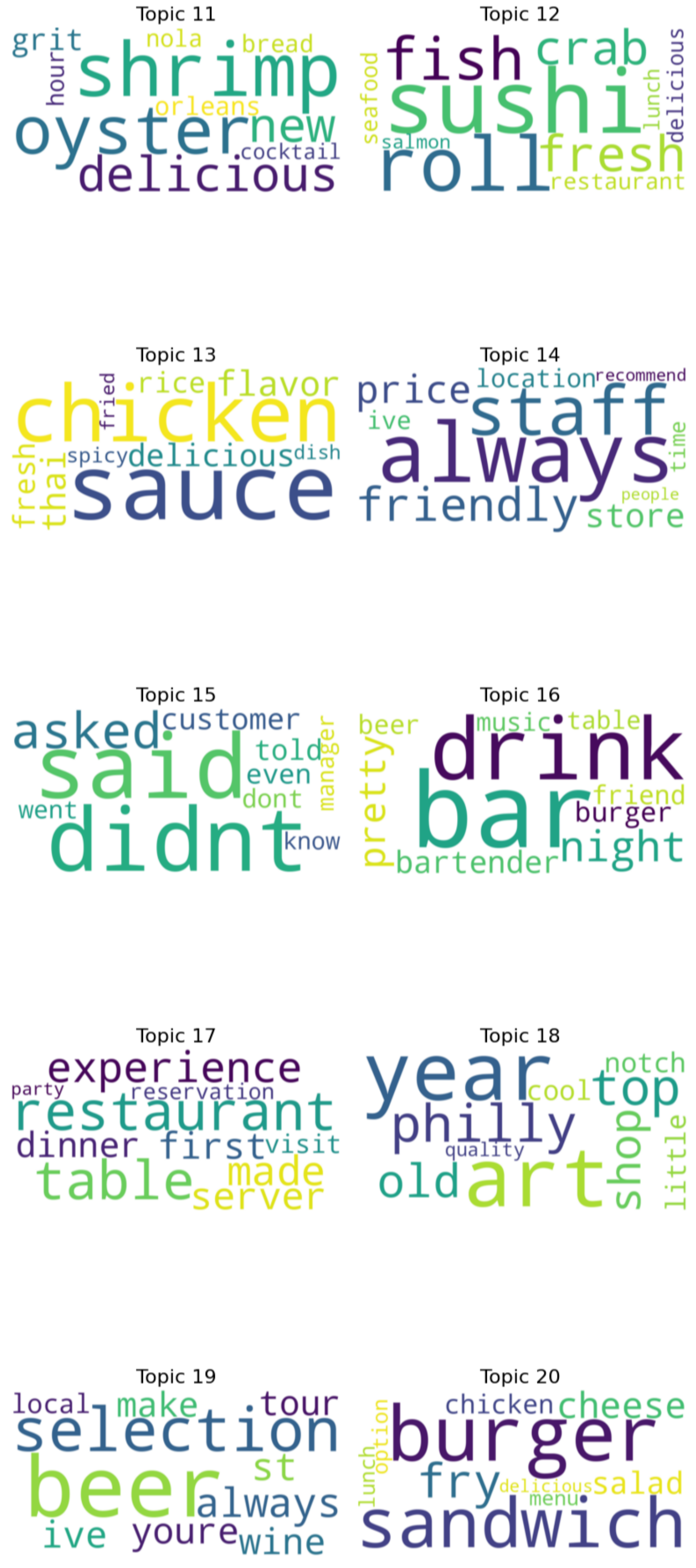

Figure 15: Word cloud for LDA with 20 topics

The distribution of topics among documents is as follows.

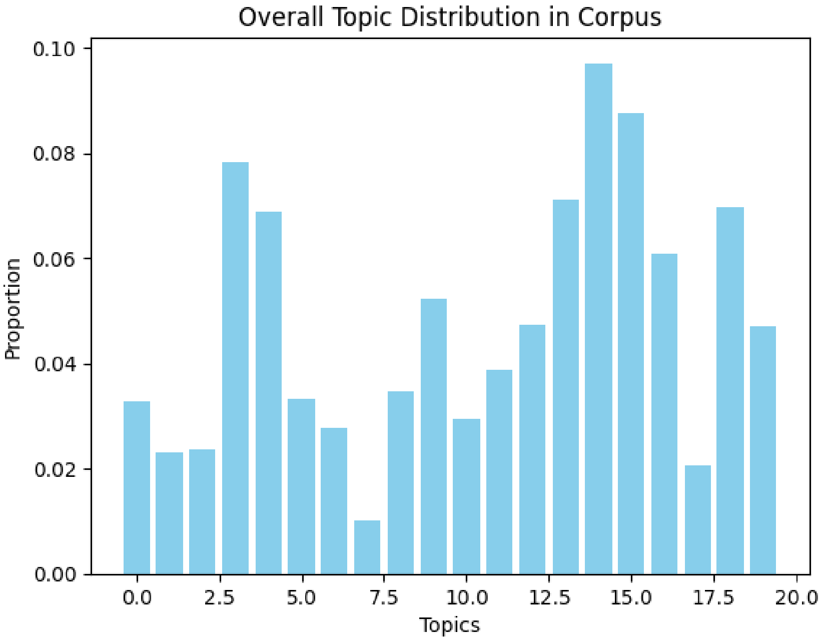

Figure 16: topic distribution for LDA with 20 topics

The evaluation metrics of this experiment are:

- Coherence = 0.48
- Perplexity = -9.2
- Diversity = 0.89

The results show a slight deterioration in the consistency of the identified topics compared to the 15-topic LDA experiment and a slight improvement in diversity. However, the results are in line with the previous ones, and the graphs show that again, the words of the different topics represent different elements present within the reviews of the dataset.

6.3 BERTopic results

After performing experiments using LDA, these were compared with an additional transformer-based model: BERTopic. This model does not require choosing the number of topics, but to compare it with the previous ones, the word distribution of only 15 topics will be shown, the number of topics is choosen based on the model with the highest coherence identified earlier. The metrics used to evaluate and compare this model are consistency and diversity; perplexity is not applicable since this is not a probabilistic model. The goal of this comparison is to see if using a model that provides a contextual representation of documents, can extract topics more accurately. The results of the topic extraction using BERTopic are shown below:

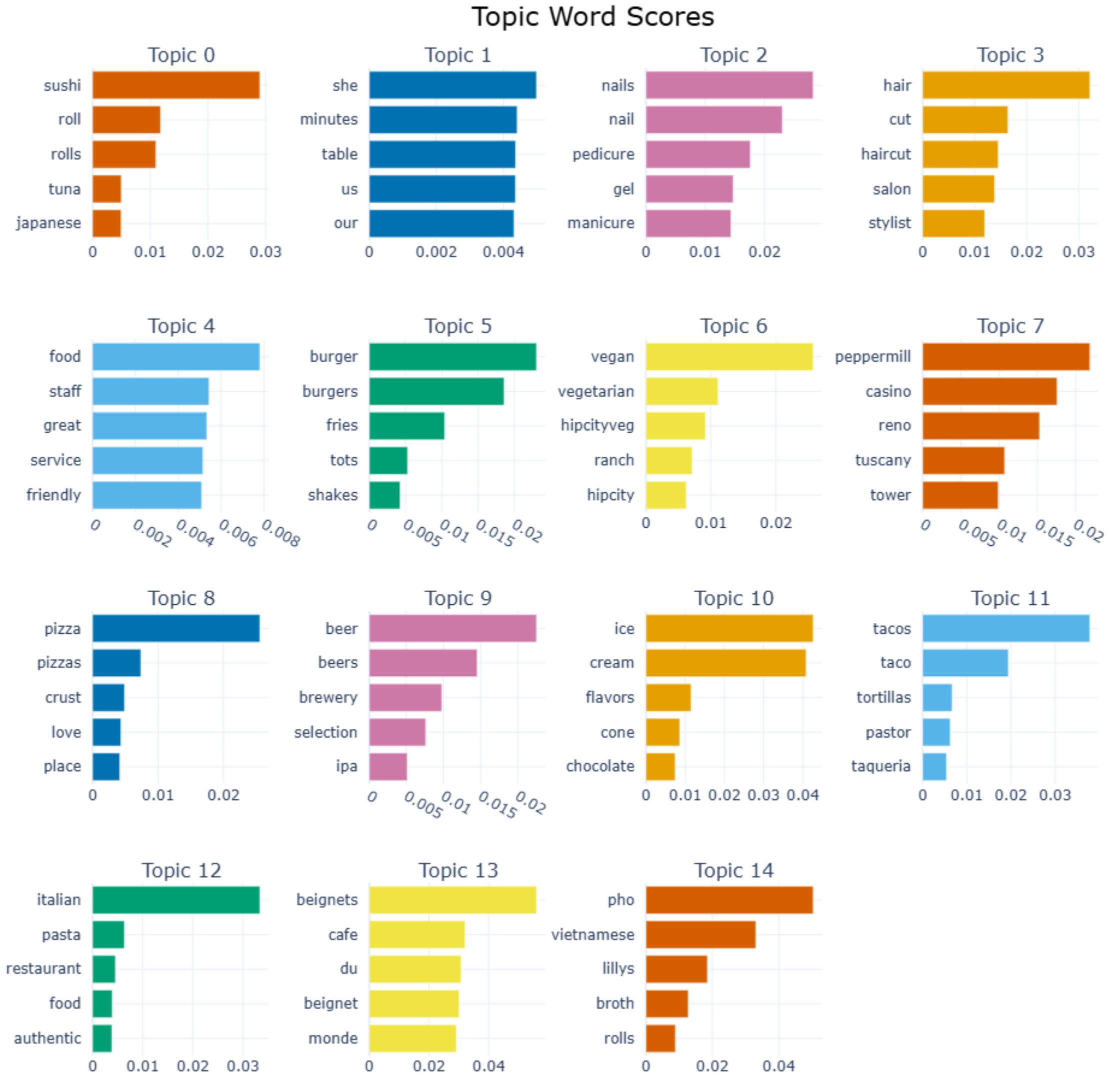

Figure 17: Top 15 topics of BERTopic

Looking at the results, it can be seen that the identified topics clearly describe the different businesses in the reviews of the dataset. Different types of restaurants and other types of businesses such as beauty salons and hair salons can be distinguished. The words within the different topics are related to each other and refer to the same business activity. At first glance, therefore, the breakdown into topics using BERTopic is more effective. Below to confirm this are the metrics of all the models seen so far:

Model	Coherence	Perplexity	Diversity
LDA 5 topics	0.43	-7.9	0.77
LDA 10 topics	0.44	-8.3	0.87
LDA 15 topics	0.49	-8.9	0.88
LDA 20 topics	0.48	-9.2	0.89
BERTopic	0.63	-	0.99

Looking at the evaluation metrics, it can be seen that the BERTopic model performed significantly better from both coherence and diversity perspectives. It thus confirms that it is the best model among those tested, as hypothesized earlier by observing the words within the topics. It also confirms that providing a contextual representation of the text succeeds in better capturing the relationships between words and consequently improving the results.

7 Conclusions

This project explored advanced text mining techniques and semantic analysis applied to Yelp Open Dataset reviews. Through a structured approach, some techniques of text clustering and topic modeling have been used and compared in relation to the different text representation techniques used. The clustering results highlighted how the use of contextualized representations, in particular with a fine tuned Sentence Transformer model, allows for cohesive and well distinct groups,which upon further analysis are confirmed to represent the business categories of the dataset. Also in the context of topic modeling the BERTopic algorithm outperformed traditional methods like LDA, which has made it possible to distinguish between different business types. The results therefore prove the importance of providing an effective text representation for all text mining operations performed. They show that by using contextual text representation models, better results can be achieved in comparison to more basic representation techniques.