Textual Clustering for Telecommunication Accident Recommendations

MICHAEL MALEK, University of Twente, The Netherlands

Over the last decades, a significant amount of people became dependent on sufficiently working communication networks. A lot of these networks are proven to be fairly stable and have backup solutions, but can trigger uttermost consequences in case of an accident due to faulty machines or human error. This can lead to a tremendous number of issues for example millions of money lost or even potential death cases. Researchers worldwide are trying to improve organisations' network safety and minimize the risk factor of network downtime and accident rates. Based on another research paper concerning accident analysis in telecommunication companies by Wienen et al. [24], this paper focuses on finding insights or new patterns in textual recommendations created by experts concerning telecommunication networks. To find these new patterns, sentence embedding methods in combination with unsupervised clustering algorithms had to be applied. Sentence Transformers were applied to convert the given recommendations into a quantitative embedding matrix. 8 clustering algorithms were found and trained, of which the most optimal model was selected with internal and external validation tools and cluster visualizations. To furthermore find insightful information for the most optimal model, topic visualization tools produced word clouds that formed topics based on the most frequent words in each cluster.

Additional Key Words and Phrases: text clustering, Natural language Processing, sentence embedding, unsupervised machine learning, validation

1 INTRODUCTION

1.1 Context and Relevance

Telecommunications networks form critical infrastructures, network outages due to incidents can have severe impacts on society and the organisations themselves. For example the average cost of unplanned application downtime in communication networks per year is estimated to be between $1.25 billion and $2.5 billion [6]. To be more specific, the 2019 Server OS Reliability Survey found out that 98% of companies loose at least $100,000 for one hour of downtime [9]. But the risk of accidents in telecommunication networks is not solely impacting companies negatively, but society as a whole. Failures of technologies or human error in telecommunication infrastructure can lead to preventable loss of life and damage to property, by causing delays and errors in emergency response and disaster relief efforts. Despite the increasing reliability of modern telecommunication networks to physical damage [14], the risk associated with communication outages remains serious because of further growing dependence upon these tools in emergency operations. To mitigate the risk of accidents and incidents within a company and society in the future, a research team from the University of Twente has prior to this thesis collected recommendations from different parties within one telecommunication operator. Before these recommendations can be reviewed, the recommendations itself must be separated into different clusters. Instead of filtering manually, the research team decided to distribute that part for this thesis and try to implement the clusters through the help of machine learning methods. This would give them the opportunity to prioritize their available time to other sections of their project and create a machine learning model that could potentially become handy for clustering new recommendations from other telecommunication operators.

1.2 Problem Statement

The primary purpose of this paper is to find meaningful insights from the recommendations and receive some context by clustering them into distinguishable groups. Furthermore, the clustering could have been conducted manually. However, this would take some additional time and the original researcher that produced the recommendations wanted the model to predict clusters for future recommendations in other telecommunication corporations as well. In essence, the task is to find an automatic way of clustering the recommendations through machine learning or artificial intelligence, to eventually minimize the effort of grouping the recommendations by human input. In addition, the script could find dependencies and patterns usually not recognized by human senses. In the end, the resulting models also have to be validated to compare the outcome with each other. For this, manual labeling of the recommendations was needed in combination with external validation methods to find how accurate the clusters became based on the manual labels.

Because the data consists of recommendations written in plain English text it first needs to be transformed from qualitative to quantitative data, thus this part will be conducted through the help of Natural Language Processing (NLP) to clean [20] and transform the recommendations into a vector matrix. Afterward, made use of different unsupervised machine learning models to eventually cluster the data. The data itself only consists of the unlabeled text recommendations, so the clustering approach has to be an unsupervised one. Because of this, there are no predictions if the clustering will become an automated success or if the data itself can not be distinguished into different clusters.

1.3 Research Questions

Based on the previous mentioned problem statement the following research questions were formulated:

(1) Can the recommendations for telecommunication organisations be sufficiently clustered into different groups using different machine learning methods?
(2) What is the optimal number of clusters for each machine learning method based on internal validation?
(3) What generic topics can be taken from the clusters of recommendations?

2 LITERATURE REVIEW

To answer the research questions, a theoretical background had to be formed first. For finding the corresponding research papers and

TScIT 37, July 8, 2022, Enschede, The Netherlands

informational books, databases like Scopus and Web of Science were primarily used. With search terms such as "sentence embedding" and "sentence transformer" the goal was to first find research papers about transforming the sentences into a matrix representing each sentence and search terms like "NLP clustering", "text clustering" and "text labeling" to eventually find literature about what clustering algorithms were already experimented with, what they resulted to and if they could be applied to this research as well. Additionally to that, a lot of papers were discovered through reading internet blogs in which a lot of modern and popular papers were referenced.

Unsupervised clustering has been a topic of research at least as far back as 1967 [11], in which for the first time Johnson tried to cluster data points with the help of a hierarchical clustering method. By measuring the quantitative distance between points in a coordinate system to ultimately compare the individual distances and group data points with small distances to each other. The number of clusters in hierarchical clustering has been decided based on human input and a dendrogram. This opened a door for many other researchers to continue based on his work and develop new and more complex algorithms and theories.

Over the next years, machine learning lost its relevance due to having no practicality in business models or missing accessibility. That was until around 2002 when Maulik et al. [12] and other papers investigated the efficiency of different unsupervised clustering approaches [26]. One of the most popular unsupervised clustering methods, apart from the traditional hierarchical clustering [11], came out to be the K-Means algorithm. It rather focuses on finding the optimal number of clusters automatically through the usage of for example the Elbow, Average silhouette or Gap statistic method [23] [22] instead of human input.

These days K-Means and hierarchical clustering represent the most accessible methods of clustering [23] and are applied with different data types, as very recently for example with Natural Language Processing. Natural language processing clustering gets included in a lot of different fields like naming conventions in medicine [15] [16], Marketing analysis [10], but also accident analysis [4] which this research is also focusing on.

The recommendations have been received from the accident analysis written by Wienen et al. [24]. They got produced through multiple workshops with experts and ordinary workers of a specific telecommunication company to develop steps to minimize the risk of future accidents.

3 METHODOLOGY AND APPROACH

Natural Language Processing combines human language and semantic meaning of language with computational techniques and is used to quantify and analyze natural language or speech so that eventually a computer can make sense of a specific language. Before the computer could understand the qualitative textual recommendations, it initially needed to be transformed into a quantitative vector space filled with numbers representing the original sentences, to then become input for the machine learning algorithm to perform the clustering. Figure 1 illustrates the main phase of the research in which the central experiments have been prepared and conducted. The exact details and contents of those steps will be discussed from here on.

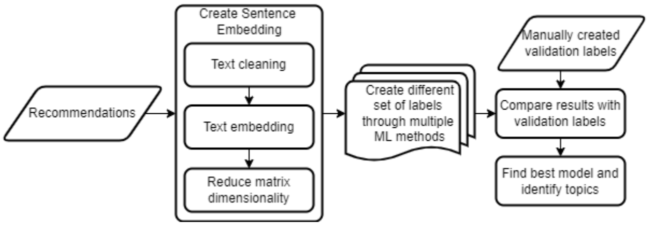
Recommendations
Create Sentence Embedding
Text cleaning
Text embedding
Reduce matrix dimensionality
Create different set of labels through multiple ML methods
Manually created validation labels
Compare results with validation labels
Find best model and identify topics

Fig. 1. Methodology of creating sentence embeddings and utilizing machine learning for clustering

The Flowchart in Figure 1 is based on the official Flowchart Symbols produced by the American National Standards Institute (ANSI) and is since 1970 also been an applied standard by the International Organization for Standardization (ISO).

3.1 Recommendation Preprocessing

The preprocessing phase intends to remove meaningless data from the recommendations and retrieve only relevant features from the raw text recommendations. In this research, NLP techniques have been used to first clean the data of unnecessary characters, generalize or lemmatize the words and lastly turn each recommendation into a separated list of words. This pre-processing ensured that each recommendation was normalized so they can be quantitatively compared later on. The following preprocessing steps have been proposed:

3.1.1 Clean Recommendation. Cleaning the recommendations meant removing stop words and punctuation from each recommendation. Stop words represent impractical words within the recommendations that do not add any meaning [16] for the analysis like *the, a, an, but, for, etc.* Punctuation includes spacing and special characters that are common in different languages as for standard English for example question marks, brackets, dots, commas, etc.

3.1.2 Stem or Lemmatize Recommendation. Stemming and Lemmatization are both techniques to simplify text for textual analysis but have different use cases. Stemming focuses on removing or replacing word suffixes and returns the common root form of a word e.g. "eating" is stemmed as "eat". Lemmatization, on the other hand, ignores any suffixes and directly returns the words base form e.g. "better" is lemmatized as "good" [10]. Generally, Stemming can lead to wrong base forms, for example, if you try to lemmatize the word "Caring", it would return "Care" but stemming it would return "Car", which of course are two completely different connotations. Furthermore, Lemmatization is considered to be computationally more expensive because it involves look-up tables and other resources to access the base forms, generally, Lemmatization is preferred for smaller datasets and if performance is not an issue. In our case, the recommendations only consist of 180 rows of text which allowed us to use Lemmatization.

3.2 Sentence Transformation

At this point, the recommendations have been cleaned and lemmatized for further processing. The next fundamental step was to create a so-called sentence embedding of the recommendations, meaning to transform each of the pre-processed recommendations into vectors of real numbers to ultimately receive an embedding matrix with each row representing each recommendation. Those embedding methods can vary in use-cases, efficiency, and effectiveness. One of the most basic embedding methods is called term frequency-inverse document frequency (TFIDF), a numerical statistic, based on the Bag Of Word (BOW) method, which calculates the importance of a word within a document of words and returns a constructed sparse matrix of word probabilities with each vector representing one recommendation. The TFIDF value increases proportionally to the number of times a word appears in a document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general [18]. Although TFIDF is still accessed in some fields of study and practical businesses, it can not distinguish the sentences semantically. TFIDF release already stretches some years and researchers have been able to construct overall more sufficient algorithms.

3.2.1 Sentence-BERT. One of the more cutting edge embedding methods is called BERT [5] which was constructed by Google researchers in 2019. BERT is a rather recent language model that uses deep neural networks to encode sentences into vectors and possibly decode or predict them back into for example another language. Because BERT makes usage of massively complex neural networks, its very computational heavy and can take multiple hours to train and evaluate similarities between sentences and words.

That is where Sentence-BERT (S-BERT) comes into place; Established by Reimers et al. [17], a Python framework that extends on the BERT architecture, but includes pre-trained models for a variety of use-cases and computationally faster processing than the original BERT model. The usage of a pre-trained S-BERT model becomes very effective in the case of small data sets because a small data set should not train the entire corpus of words for a model. Instead, the pre-trained models were already trained on vast amounts of text data for different instances e.g. text classification, similarity or semantic search, and more. For this research, the pre-trained model 'sentence-t5-xl' was chosen. Although its slightly lower average performance compared to the all-rounder and best performing model 'all-mpnet-base-v2', sentence-t5-xl was primarily tuned for sentence similarity tasks, suiting the recommendations to be embedded in this research. Picking the specialized model for a certain use-case rather than the model tuned for a lot of different use-cases can return higher accuracy results.

3.2.2 Reduce Dimensionality of Embedding. The before-produced embedding is constructed by a lot of numbers and dimensions, already too sparse for machine learning to cluster effectively. Dimensionality reduction techniques are often used for data visualization, nevertheless, these techniques can also be used in applied machine learning to simplify a classification data set to better fit a predictive model. The performance of unsupervised machine learning algorithms can deteriorate with too many input variables; having a large number of dimensions in the feature space can imply a high range of values between all data points, therefore often representing a small and non-representative sample. To achieve a simpler embedding for further machine learning processing, the dimensionality reduction tool UMAP was used. Uniform Manifold Approximation and Projection is a dimension reduction technique that can be used for transforming multi-dimensional data points into two-dimensional space. The particular strength of UMAP comes from its option of general non-linear dimension reduction and its speed [13], scaling well in terms of both data set size and dimensionality compared to alternative methods.

3.3 Clustering

After the recommendations were successfully transformed and reduced, the next phase included trying to cluster the data. Two classification approaches were suitable for this scenario; Either decide on a most effective machine learning model or let a deep learning model classify the recommendations. Eventually the decision was put on machine learning because deep learning is still considered to be a black box [1], meaning there is almost no valuable knowledge about how all the individual neurons in a deep learning model work together to arrive at the final output, ending up not as controllable as a machine learning model. Additionally to that, machine learning generally has a better application when used in combination with less data, deep learning is rather used in the context of big data, concerning massive amounts of data. Furthermore, the recommendations are completely unlabeled and do not give any instructions for a supervised approach, instead making use of unsupervised clustering. Generally, unsupervised clustering techniques apply when there are no predictions of labels, but a division into natural groups instead [25]. This entailed the search for unsupervised clustering methods; One of the most popular and referenced algorithms for unsupervised clustering is K-Means [21], nevertheless it should not be the sole algorithm for the experiments, because especially unsupervised clustering approaches handle the clustering very different from each other and every algorithm fits better for different use cases. Starting from here, the number of clusters that were considered optimal input in machine learning algorithms will be called K. Based on the accessible algorithms available through Python libraries and the comparison of algorithm performances [26] especially in text clustering where not all algorithms are applicable [3] the following clustering methods were chosen:

3.3.1 K-Means Clustering. As mentioned above, the K-Means algorithm is one of the most accessed machine learning concepts in unsupervised clustering. That is because of its common use to naturally partition a data set into K clusters where K first needs to be estimated [8]. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum. The number of K was unknown at the point of the experiments, hence a commonly used method for finding an optimal K value is the Elbow Method [22]. It goes through an iterative set of numbers for example the numbers 0 to 100 and estimates the average distances of each number in the set and graphs those in a coordinate system. Analyzing this graph will rapidly change at a point and thus creating an elbow shape.

The elbow method can unfortunately only be used for the K-Means algorithm.

3.3.2 Spectral Clustering. A clustering algorithm with roots from mathematical graph theory, which applies clustering to a projection of the normalized Laplacian [19]. It uses information from all the available eigenvalues of a matrix, also called spectrum, to compare each vector within its matrix. Similar to the K-Means algorithm, Spectral Clustering requires a pre-defined number of clusters K as input to eventually cluster the recommendations.

3.3.3 Agglomerative Clustering. Agglomerative Clustering is the most common type of hierarchical clustering. Hierarchical cluster analysis [11] measures distances between data points to distinguish them from each other and merge data points if their distance is shortest. It includes building clusters that have a preliminary order from top to bottom. The Agglomerative Clustering method extends on this and can cluster data again based on a number of clusters K, but can also produce a Dendrogram to make usage of its visual aspect of picking the correct number of clusters.

The clustering methods mentioned so far all require input parameters about the number of clusters K. Besides that, other clustering methods do not require K, instead, they have internal methods to assume an optimal number of clusters but still need a set of input parameters to achieve this. Ultimately, the difference between the following methods is their unique approach of not being dependent on the given number of K but instead can produce their interpretations of how the clusters should look, therefore potentially finding insights into the clusters human senses would not be able to spot intuitively. This was the reason to include the following methods that do not require an initial number of clusters K:

3.3.4 HDBSCAN. Instead of using a distance measure, an older method of clustering called DBSCAN forms clusters based on how many data points fall within a given radius [3]. This algorithm is especially effective for data containing clusters of similar density. HDBSCAN extends on this but instead converts the DBSCAN into a hierarchical clustering algorithm, and then uses a technique to extract a flat clustering based on the stability of clusters. This method does not require any input about the number of clusters K.

3.3.5 Mean Shift. a distance centroid-based algorithm, which works by updating candidates for centroids to be the mean of the points within a given region [19]. The main difference between K-Means is; Mean Shift does not require an input about the optimal number of clusters K but automatically sets the number of clusters by relying on a single parameter bandwidth, which dictates the size of the region to search through.

3.3.6 Affinity Propagation. In Affinity Propagation, the data points can be seen as a network, where all the data points send messages to all other points until the algorithm finds the optimal clusters based on its internal parameters. This will return a description of the data set using a small number of exemplars, which are identified as those most representative of other samples. This algorithm generally differs from the aforementioned algorithms but very recently has achieved an increase in usage and performance. Again, this algorithm does not require K as input, instead, it chooses the number based on the data provided.

3.4 Cluster Validation

Each of the before-mentioned clustering algorithms has different use cases and performances, indicating some kind of numerical comparison between all the produced results. This will be done through two well-established methodologies:

3.4.1 Internal Validation. Internal validation tries to measure the goodness of a clustering structure without any reference to external information. Because it does not need any external resources, it can also be used for estimating the optimal number of clusters K and the appropriate clustering algorithm without any external data. Exactly this approach was used within the clustering experiments, especially for algorithms that require K as a parameter, for example with K-Means, Spectral and Agglomerative Clustering. To find K, indices like Silhouette Score, Davies-Bouldin, and Calinski-Harabasz [19] were looped within a certain iteration of clusters and then evaluated per loop to find the adequate number of clusters.

3.4.2 External Validation. External validation is a measure of agreement between two produced labels where the first partition is a set of clusters that represent the true labels of the data, and the second result comes from the clustering experiments themselves. Originally this research did not contain any true labels, but after validity considerations with the supervisor, the manually clustered labels with interpretation were provided. Those true labels of the recommendations were used to have a reference of validation for the machine learning predictions, although it is important to mention that the goal was not to find an algorithm that represents the true labels perfectly but to find an algorithm that achieves sufficient labels with own insights that are not solely based on true labels. Since we know the true cluster number in advance, this approach is mainly used for selecting the right clustering algorithm for a specific data set.

Ultimately external validation was applied in combination with the true labels for all algorithms to evaluate how close the predictions are to the true data without missing new insightful data. To achieve this, indices like Rand Index, Homogeneity, Completeness, and V-Measure Scores [2] were applied. Additionally, a Contingency Matrix, or Confusion Matrix, was constructed to evaluate another important index, namely the Purity of two partitions. All these mentioned indices produce a number on a scale between zero and one, zero representing no similarity to the true labels and one representing full similarity to the true labels.

3.5 Topic Modelling

Topic Modelling, also called topic analysis, is another unsupervised machine learning technique scanning a set of documents and detects semantic word patterns to eventually cluster word groups and similar expressions that best characterize a set of documents. The particular advantage of topic modelling is the utilization of a more advanced form of the already mentioned TF-IDF method, namely C-TFIDF [7], to find the most important and frequent terms within the each cluster of recommendations to eventually form logical topics.

Algorithm	**K-Means**	**Agglomerative**	**Spectral**	**HDBSCAN**	**Affinity Propagation**	**Mean Shift**
Optimal K	7/11	7/9/12	7 or 8	2	11	2
Silhouette Score	0.248/0.248	0.295/0.301/0.253	0.345/0.315	0.066	0.328	-

Table 1. Optimal K and corresponding internal validation indices per algorithm

Model	**Rand Index**	**Homogeneity Score**	**Completeness Score**	**V-Measure**	**Purity**
affinity_11	0.734	0.354	0.229	0.278	0.592
agglomerative_12	0.731	0.318	0.199	0.245	0.575
agglomerative_9	**0.734**	**0.326**	**0.231**	**0.271**	**0.598**
agglomerative_7	**0.719**	**0.288**	**0.228**	**0.254**	**0.570**
hdbscan_2	0.466	0.042	0.106	0.061	0.380
kmeans_11	**0.736**	**0.344**	**0.219**	**0.267**	**0.603**
kmeans_7	0.705	0.227	0.180	0.201	0.531
meanshift_2	0.257	0.000	1.000	0.000	0.380
spectral_7	**0.730**	**0.347**	**0.224**	**0.272**	**0.587**
spectral_8	**0.717**	**0.276**	**0.204**	**0.235**	**0.542**

Table 2. External validation indices per trained clustering model

Following this presented the opportunity to further discover each cluster produced in more detailed dimensions.

4 RESULTS

Following the proposed Methodology by first preprocessing, then transforming the sentences into vector space, internal validation was used to find the optimal number of clusters for the clustering algorithms K-Means, Spectral Clustering, and Agglomerative Clustering. The results of the experiments will be presented sequentially:

4.1 Evaluate Clustering Models

The clustering algorithms that require K as input were iterated through from 1 to 50 clusters, for each cluster evaluating the internal validation indices stated before, to eventually receive the most optimal numbers of clusters. Those algorithms can also produce multiple numbers of optimal clusters K if the probability is the same or similar to other optimal clusters. Furthermore, the other three clustering algorithms that do not require K as input just estimated their parameters for clustering. The results of the internal validation experiments for the best-performing models are all summarized in Table 1. Based on the results of the internal validation, K-Means produced an optimal number of 7 or 11 clusters, Agglomerative Clustering an optimal number of 9 or 12 clusters, and Spectral Clustering an optimal number of 7 or 8. Other than that the algorithms that did not require an input number of clusters estimated an optimal number of 2 clusters for HDBSCAN, 11 clusters for Affinity Propagation, and 2 clusters for Mean Shift.

After calculating what K would be optimal for the corresponding algorithm by utilizing internal validation, the subsequent step was to calculate the external validation of each of the trained models. This meant training all the clustering models with the corresponding K values found in Table 1, to then calculate the external validation indices for each trained model, which results are stated in Table 2. The format of the aforementioned models in Table 2 is equal to '<model-name>_<K>', for example, the HDBSCAN algorithm prefers a K of 2, so the model description is hdbscan_2. In conclusion, the models that received the lowest ratings in all external validation indices were, similar to finding K, obvious discrepancies; The Mean Shift and HDBSCAN models fell very short in all scales of external validation indices, even reaching below 0.5 on each scale. Based on this, it was assumed those models would not represent the true labels of data and the decision was to eliminate them from the list. Looking at the best results, we found models like Affinity Clustering with 11 clusters, Agglomerative Clustering with 9 clusters, K-Means with 11 clusters, and Spectral Clustering with 7 or 8 clusters producing the highest results in the external validation scales. Unfortunately, following the reproduction of models for the sake of reversibility, it was noticed that the Affinity Propagation can not always produce the same results. In fact, in one run it produced 9 clusters, in another 10 clusters, and just another even 11. This makes the algorithm unpredictable and of course not reproducible in other scenarios, therefore also eliminating the algorithm.

4.2 Accuracy and Precision Comparison

As seen in Table 2, different models produced identical predictions about how many clusters would be optimal, for example, Agglomerative Clustering, K-Means, and Spectral Clustering all indicated one model with 7 clusters as the potential optimal number of clusters K. The Affinity Propagation also predicted the same number of clusters as one model of K-Means with 11 clusters, although being ignored because of the unpredictable nature of Affinity Propagation. So how

Models to Compare	**Accuracy**	**Precision**
kmeans_7 & agglomerative_7	0.017	0.018
kmeans_7 & spectral_7	0.274	0.334
spectral_7 & agglomerative_7	0	0

Table 3. Models with matching K compared

close are the resulting labels from each other, do they correlate to the same degree, or have the models predicted the same number of clusters but with completely distinct labels and data points? To justify this question, all labels with the same number of clusters have been compared with the observational error scales for machine learning, Accuracy, and Precision. Accuracy describes how close or far-off a given set of measurements are to their true value, while precision is how close or dispersed the measurements are to each other. In this case, it does not matter what labels are true and predicted, the only importance is how close one model's resulting labels are compared to another with the same K. Based on the outcome presented in Table 3, it became apparent that neither Accuracy nor Precision scores was substantially high, revealing minimal to no similarity between each of the produced labels. This furthermore confirms the unique approach applied by each of the clustering methods utilized in this research, as discussed in the Clustering section. Nonetheless, the Spectral Clustering and K-Means models with 7 clusters achieved partially interesting results, because some of their labels appear to overlap with each other, indicating a minor similarity between both models. Ultimately it can be said that none of the models just discussed have significantly similar labels and should consequently be seen as separate units.

4.3 Cluster Visualizations

Based on the results of the external and internal validation, Agglomerative Clustering with 7 or 9 clusters, K-Means with 7 or 11 clusters, and Spectral Clustering with 8 or 7 clusters were found to be further explored. For this, UMAP was utilized again, because while reducing the dimensionality of the embedding into a two-dimensional space, UMAP also offers a set of visualization tools that use the two-dimensional space to indicate how spread each cluster is. From there on, the approach was to inspect all the generated UMAP illustrations from the four most optimal clustering models and compare them visually, eventually deciding on one optimal model with the best-chosen clusters.

For example Figure 2 represents the two-dimensional UMAP representation of the sentence embedding for the K-Means algorithm with 11 clusters. It becomes obvious that the data is not optimally spread between all points to form perfect clusters, but despite that, the K-Means model was still able to form reasonable clusters. Generally, it appeared that the K-Means model sometimes splits one general cluster into sub-clusters like between the clusters on the top right or clusters on the top left, furthermore indicating why the model produced the highest number of clusters out of all optimal models. It also produced some tiny clusters with only some samples included, suggesting that those clusters could be more generalized into again main cluster instead of tiny sub-clusters. Moreover, a lot of data points interfere with other labels, making the model seemingly less accurate.

Figure 3 represents the same concept as Figure 2, but labeling the two-dimensional space with the results of the Agglomerative Clustering. The Agglomerative Clustering model, for once, estimated 9 clusters, so 2 less compared to the K-Means results. This illustration has fewer, but generally more unified clusters, yet also some interfering data points with different labels for example in the center. Overall the model appeared to more accurately suit the data structure than K-Means, originating the indication that less then 11 clusters are preferable for this data set. The Agglomerative Clustering algorithm also produced high external validation scores for 12 clusters, but after manual inspection of the UMAP illustration, it became obvious that 12 clusters are not represented by this data set, therefore eliminating almost all Agglomerative Clustering models, further investigating one more later in this section.

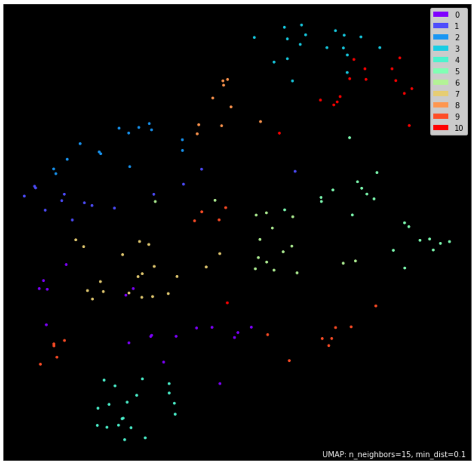

Fig. 2. K-Means with 11 clusters based on UMAP

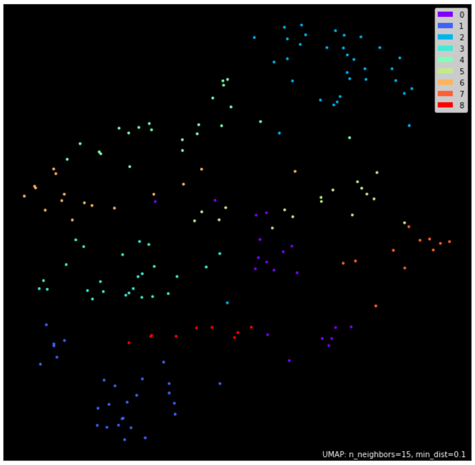

Fig. 3. Agglomerative Clustering with 9 Clusters based on UMAP

The next UMAP illustration is generated based on the Spectral Clustering model with 8 clusters and is represented in Figure 4. Principally the clusters appear the finest so far, although also still providing interference between different labels especially in and around the center. It looked like there was one more cluster in the center that could have been unified, so there was one more model to further evaluate. After evaluation of the same model with 7 clusters, it was found that none of the models represent the data set, eliminating all Spectral Clustering models.

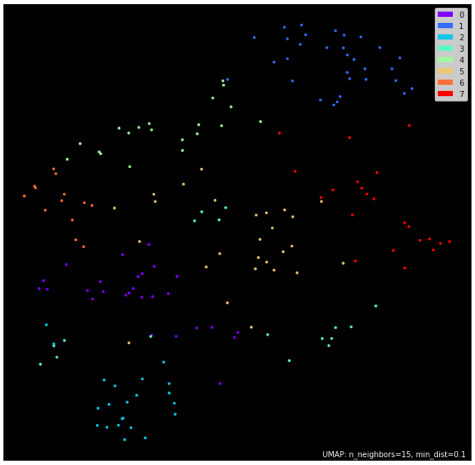

Fig. 4. Spectral Clustering with 8 Clusters based on UMAP

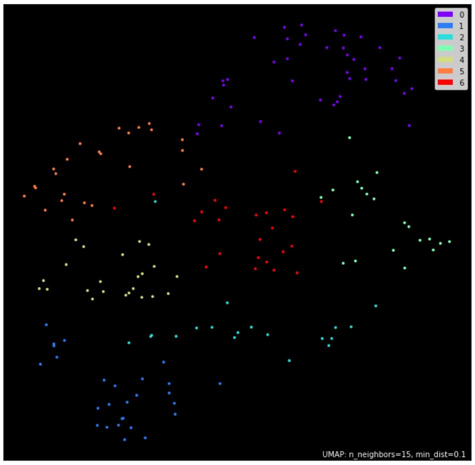

Fig. 5. Agglomerative Clustering with 7 Clusters based on UMAP

The last UMAP illustration that has been observed is once more the Agglomerative Clustering model, but trained with 7 clusters. This is displayed in Figure 5 and surprisingly demonstrates almost perfect clustering of all data points in the UMAP space, showing almost no signs of outliers and interference between the different labels. The outliers detected could even realistically be out-masked over more advanced parameter tuning, further explained in the Limitations section. Every area in the UMAP space generally predicts main clusters with no sub-clusters, thus this model was chosen as the final model to further investigate throughout Topic Visualization.

4.4 Topic Modelling

To continue, each of the 7 clusters created by the Agglomerative clustering model had to be inspected more profoundly. This was supposed to be handled throughout BERTtopic, which uses a special form of the TFIDF embedding method to extract and find the most important and frequent words within each label. Unfortunately, BERTopic was suspected to generate the topics not specifically based on the clusters created by the Agglomerative Clustering model but instead utilizes their clustering mechanism so that the original clusters could potentially not match with the most frequent words originating from BERTopic. Because of that, the process BERTopic uses to extract the most frequent word in each cluster had to be recreated manually to fit our specific scenario. Following this, the most frequent words of each cluster have been transformed into word clouds, the result of this can be seen in Figure 6. Sometimes words appear that have not been within the original set of recommendations, this is because the words have been cleaned, or rather lemmatized before processed, meaning that some very specific words have been generalized into for example 'failure' in Topic 1, or "train" in Topic 2, although they have not been included in the original data set. The size of a word in the word cloud also represents the frequency of this word, the bigger the word in size the higher its importance within the cluster.

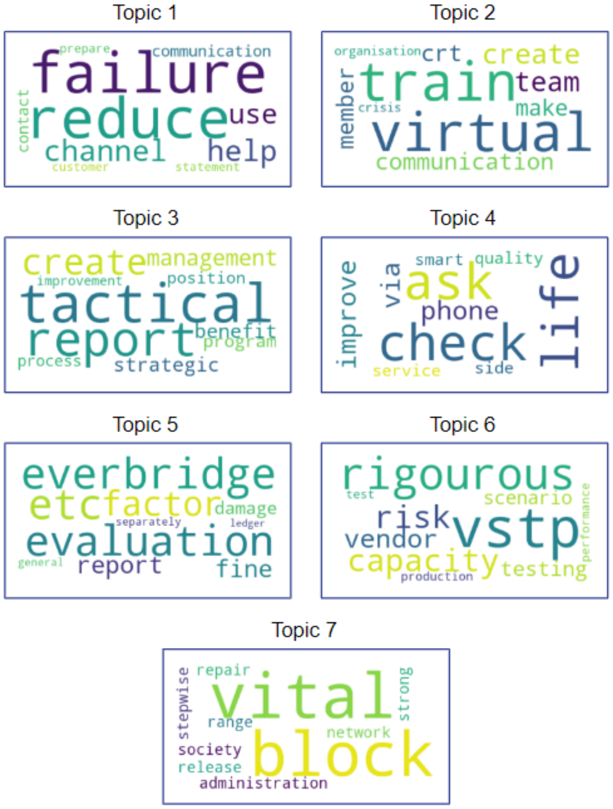

Fig. 6. Word Cloud based on clusters from Agglomerative Clustering with K=7

From here on, the most frequent generalized words within each cluster of the Agglomerative algorithm with 7 clusters are known and can be evaluated to form vague logical topics. It should be noted, that the following results are interpreted by the limited knowledge of the researcher of this paper about the recommendation context, they should rather be inspected and interpreted by professionals with detailed domain knowledge. Nevertheless, the following topic descriptions were created in communication with the supervisor of this research. Inspecting the first cluster with main words like 'failure', 'reduce', 'channel', and 'communication', recommendations in the first cluster appeared to focus on recommending reducing the failing communications to the customer, possibly generalized as Reduce communication failure. The second topic includes words like 'train', 'virtual', communication', 'create', 'team', 'members', and 'organization', concerning recommendations that advocate training or creating virtual communication between members of the organization. The third topic includes 'tactical', 'report', 'create', 'benefit', 'strategic' and 'management', potentially about creating strategic reports for the management. On the other hand, Topic 4 includes 'life', 'ask', 'check', 'improve', 'smart', 'phone', 'quality', and 'service', which was harder to identify the exact topic for, because it was unknown if the words 'smart' and 'phone' belong together to form the word 'smartphone' or are supposed to be separated. But assuming 'smart' does not correlate to any other word in the cluster, topic 4 could summarize recommendations about checking the life of smartphones, especially quality and services. Topic 5 includes 'everbridge', 'evaluation', 'factor', 'report', and 'damage', where after minor research, 'everbridge' turned out to be a company that provides information about critical events to help with personal safety and business continuity. This can be interpreted as an evaluation of Everbridge reports and damage factors. Topic 6 includes 'rigorous', 'risk', 'vendor', 'vstp', 'capacity', 'scenario', and 'test', in which vSTP exemplifies the vital method by which mobile networks set up connections and send messages. Therefore this topic could be described as testing risks and capacity of the vSTP rigorously between vendors. The last topic includes the terms 'vital', 'block', 'repair', 'network', 'administration', and 'stepwise', potentially summarized as administrating vital networks stepwise by blocking or repairing. This concludes all the insights found within each cluster.

5 DISCUSSION

5.1 Resources Required

To conduct the cluster experiments a set of Resources was needed. For once the main data consists of the primary text recommendations received by the research supervisor. Those recommendations were completely anonymized beforehand. Additionally to that, an experimental research tool had to be picked to construct and configure the text pre-processing and machine learning experiments, for that Python was chosen. This is based on the fact that Python is widely used in data science and machine learning applications and covers an extensive range of documentation and libraries. The libraries used for this research consist of NumPy and pandas for data manipulation and scikit-learn as access to machine learning resources. Furthermore, Jupyter Notebook was utilized for data exploration, code preparation, and generating the UMAP visualizations.

5.2 Limitations

During this research, some segments could have been conducted differently or extended with other methods, but due to time or out-of-scope limitations were not able to be considered.

For example, during the transformation from pure text to sentence embedding, the embedding could have included more custom features to help the machine learning algorithm with its clustering task. For example character length, word count, or even more advanced text indices like sentence complexity could have been concatenated to the embedding for additional distinguished attributes.

Another very important consideration in applied machine learning is parameter tuning. In the majority of cases the standard parameters in most machine learning algorithms are sufficient enough to solve the task, but in reality, could achieve more accurate results only with small changes in the parameters. Unfortunately, this can take a lot of time depending on the hardware capabilities of the system, because during the parameter tuning process the machine learning algorithm needs a lot of iteration with a lot of different sets of parameters to eventually find the optimal parameters. Considering all clustering methods would have needed their own parameter tuning experiments, this task would have required more advanced computational resources.

Additionally, the data set of recommendations is comparatively very low on information to pick up on for the machine learning algorithms. For example, the recommendations themselves are often written in a shortened style instead of long descriptions. They still include a lot of keywords to evaluate clusters from, which makes the general clustering possible but could again be more accurate with more elaborative recommendations. The shallow amount of data is also a reason why deep learning was not applied, it usually requires a tremendous amount of data to produce accurate results.

The last important consideration to mention is the assumptions about the UMAP illustration; The entirety of the results is based on the multi-dimensional transformation to two-dimensional space, so if UMAP somehow produced a mistake in the transformation, the results could be falsely labeled. Although in contrast, in recent years UMAP produced one of the best dimensionality reduction outputs available in the domain, generally providing reliable results combined with fast processing. There are semi-validation indices available to test the transformation, but they do not consider transformations applied for machine learning tasks yet.

6 CONCLUSION

To conclude the research questions, the selection of the most optimal clustering algorithm for this specific set of recommendations, based on the methodology of this paper, appeared to be Agglomerative Clustering with 7 clusters. It generated one of the highest scores in all external validation indices and proved to be well separated and clustered through the UMAP illustration in Figure 5. The model also predicted consistent and logical topics, each topic representing a different type of recommendation with almost no interference between the most frequent words. In the case of applying the same methodology for a different set of recommendations, the same algorithm could be used as long as the recommendations have similar context and words. Nevertheless, subject to new data, the methodology should be applied again in the same way it was applied in this paper, to confirm the suitability of all available clustering methods. This is especially true if the new data differs completely from the data used in this paper.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. https://doi.org/10.1109/access.2018.2870052

[2] J Alboukadel. 2018. https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/

[3] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. *A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques* (Jul 2017). https://doi.org/10.48550/arXiv.1707.02919

[4] Abbas Chokor, Hariharan Naganathan, Wai K. Chong, and Mounir El Asmar. 2016. Analyzing Arizona Osha Injury Reports using unsupervised machine learning. *Procedia Engineering* 145 (2016), 1588–1593. https://doi.org/10.1016/j.proeng.2016.04.200

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805

[6] Stephen Elliot. 2014. DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. http://info.appdynamics.com/rs/appdynamics/images/DevOps-metrics-Fortune1K.pdf

[7] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. https://doi.org/10.48550/ARXIV.2203.05794

[8] Peihuang Huang, Pei Yao, Zhendong Hao, Huihong Peng, and Longkun Guo. 2021. Improved constrained K-means algorithm for clustering with domain knowledge. *Mathematics* 9, 19 (2021), 2390. https://doi.org/10.3390/math9192390

[9] IBM. 2020. ITIC 2020 global server hardware, server OS reliability report - IBM. https://www.ibm.com/downloads/cas/DV0XZV6R

[10] Nikita Jain, Pooja Agarwal, and Juhi Pruthi. 2015. Hashjacker- detection and analysis of hashtag hijacking on Twitter. *International Journal of Computer Applications* 114, 19 (2015), 17–20. https://doi.org/10.5120/20085-2111

[11] Stephen C. Johnson. 1967. Hierarchical clustering schemes. *Psychometrika* 32, 3 (1967), 241–254. https://doi.org/10.1007/bf02289588

[12] U. Maulik and S. Bandyopadhyay. 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 12 (2002), 1650–1654. https://doi.org/10.1109/tpami.2002.1114856

[13] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* 3, 29 (2018), 861. https://doi.org/10.21105/joss.00861

[14] William J. Mitchell and Anthony M. Townsend. 2005. Cyborg Agonistes: Disaster and reconstruction in the Digital Electronic Era. *The Resilient City* (2005). https://doi.org/10.1093/oso/9780195175844.003.0021

[15] Azadeh Nikfarjam, Abeed Sarker, Karen O'Connor, Rachel Ginn, and Graciela Gonzalez. 2015. Pharmacovigilance from social media: Mining Adverse Drug Reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association* 22, 3 (2015), 671–681. https://doi.org/10.1093/jamia/ocu041

[16] Porkodi R and Shivakumar B.L. 2012. Rule based approach for constructing gene/protein names dictionary from Medline Abstract. *International Journal of Advances in Computing and Information Technology* 1, 4 (2012), 457–468. https://doi.org/10.6088/ijacit.12.14014

[17] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using Siamese Bert-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (2019). https://doi.org/10.18653/v1/d19-1410

[18] G. Salton and C. Buckley. 1988. On the use of spreading activation methods in automatic information. *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '88* (1988). https://doi.org/10.1145/62437.62447

[19] scikit-learn developers. 2022. 2.3. clustering. https://scikit-learn.org/stable/modules/clustering.html

[20] C. Silva and B. Ribeiro. 2003. The importance of stop word removal on recall values in text categorization. *Proceedings of the International Joint Conference on Neural Networks, 2003.* (2003). https://doi.org/10.1109/ijcnn.2003.1223656

[21] Kristina P. Sinaga and Miin-Shen Yang. 2020. Unsupervised K-means clustering algorithm. *IEEE Access* 8 (2020), 80716–80727. https://doi.org/10.1109/access.2020.2988796

[22] M A Syakur, B K Khotimah, E M Rochman, and B D Satoto. 2018. Integration K-means Clustering method and elbow method for identification of the Best Customer Profile Cluster. *IOP Conference Series: Materials Science and Engineering* 336 (2018), 012017. https://doi.org/10.1088/1757-899x/336/1/012017

[23] Robert Tibshirani, Guenther Walther, and Trevor Hastie. 2001. Estimating the number of clusters in a data set via the Gap Statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63, 2 (2001), 411–423. https://doi.org/10.1111/1467-9868.00293

[24] Hans Christian Augustijn Wienen, Faiza Allah Bukhsh, Eelco Vriezekolk, and Roel J Wieringa. 2019. Applying Generic AcciMap to a DDOS Attack on a Western-European Telecom Operator. (2019).

[25] Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques.* Elsevier.

[26] Jia Wo, Chongliang Zhang, Binduo Xu, Ying Xue, and Yiping Ren. 2020. Performances of clustering methods considering data transformation and sample size: An evaluation with fisheries survey data. *Journal of Ocean University of China* 19, 3 (2020), 659–668. https://doi.org/10.1007/s11802-020-4200-3