Universität Zürich[UZH]

Master's Thesis
as part of the academic degree
Master of Science
in the Department of Informatics
of the University of Zurich

Unsupervised Text Clustering of Dental Patient Data

Author: Kevin Steijn
Matrikel-Nr: (19-770-429)

Advisor: Dr. Gerold Schneider

Supervisor: Prof. Dr. Martin Volk

Department of Computational Linguistics

Department of Informatics

Submission Date: (01.08.2022)

Abstract

The aim of this paper is to discover a method of finding semantically similar clusters from a text dataset in an unsupervised manner. An existing semantic text similarity benchmark will be used to substantiate the use of embeddings for this task. The embeddings will represent the entire text input using state of the art sentence transformers. These transformers will be combined with contrastive learning to further enhance the embeddings using state of the art research. By using transfer learning during this process this work can utilize the pre-trained models of previous research and retain their performance. These techniques will be applied to dental patient data. Resulting in visualizations that allow for exploration of the proposed clusters.

Zusammenfassung

Ziel dieser Arbeit ist es, eine Methode zu entwickeln, um semantisch ähnliche Cluster aus einem Textdatensatz auf unüberwachte Weise zu finden. Ein bestehender semantischer Textähnlichkeits-Benchmark wird verwendet, um die Verwendung von Einbettungen für diese Aufgabe zu motivieren. Die Einbettungen repräsentieren die gesamte Texteingabe unter Verwendung modernster Satztransformatoren. Diese Transformatoren werden mit kontrastivem Lernen kombiniert, um die Einbettungen auf der Grundlage des aktuellen Stands der Forschung weiter zu verbessern. Durch den Einsatz von Transfer-Lernen während dieses Prozesses kann diese Arbeit die vortrainierten Modelle aus früheren Forschungen nutzen und deren Leistung beibehalten. Diese Techniken werden auf zahnmedizinische Patientendaten angewandt. Das Ergebnis sind Visualisierungen, die eine Erkundung der vorgeschlagenen Cluster ermöglichen.

Contents

List of Figures

List of Tables

List of Acronyms

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
BERT	Bidirectional Encoder Representations from Transformers
CH	Calinski-Harabasz
DB	Davis-Bouldin
PCA	Principal Component Analysis
SBERT	Sentence BERT
SOTA	State of the art
NLP	Natural Language Processing
TSNE	T-distributed stochastic neighbort embedding

1 Introduction

This chapter features the motivation for this research, along with the research questions it aims to answer. At the end of the chapter you'll find the structure of this paper detailed.

1.1 Motivation

Natural language processing has been an ever growing field of research in recent years. This increase in attention has not been distributed equitably. In part this is due to the nature of each discipline, the discipline of mathematics will inherently have less text to analyze when comparing to a field such as history. In this case the field that is being addressed is the dental field, specifically the domain of orofacial pain. Which seemingly has been under served in research that explores natural language processing. This research aims to use the opportunity of obtaining a text dental dataset to research potential of NLP within the field and use state-of-the-art techniques to achieve this.

The dental patient dataset includes patients own description of symptoms via free text responses to questions. This research aims to explore whether patterns emerge between patients using their responses as data. These patterns would ideally correspond to the diseases that may be present in the patients. Which would lead to better understanding of the symptoms and patient profiles, with ideally the ability to detect new and previously unseen disease groups.

1.2 Research Questions

This paper is structured around a couple research questions which guide the work throughout. These questions will be answered to the extent that this work can substantiate the results. The answers will be evaluated in the discussion. The questions are ordered by the chronological order they were researched.

Which proposed metric is best to compare clustering results?

Due to the nature of the task being unsupervised it is important to research the best way to evaluate the quality of the results. To answer this question multiple metrics will be proposed to compare the clustering results. The metrics are fully described in the background section 2.3. These will be used in the results chapter to find the best performing method. The drawbacks of the metrics highlighted in the discussion.

How well do Word2Vec embeddings perform in the clustering task?

Word2Vec [Mikolov et al., 2013a,b] is a technique that was introduced in 2013. It is a well known technique that has featured in a large number of research papers. This paper will evaluate how well these word2vec embeddings capture the semantic information contained in the responses.

How well do sentence embeddings perform in the clustering task?

A proposal for capturing the semantic information in the text is to evaluate the entire sentence from each respondent. The hypothesis is that capturing the entire sentence will be better at capturing this information. To test this hypothesis state-of-the-art techniques in the form of sentence embeddings using transformers will be evaluated.

Which proposed method is best to explain the clustering results?

Due to the motivation of this thesis, it is important to address how explainable the results are. Explainable artificial intelligence is an ongoing research area and is quite expansive. Hence interpretability will be restricted to the visual communication of the findings.

1.3 Thesis Structure

This document is structured in the following manner. Chapter 2 highlights and explains relevant topics to the thesis work and brings in context by detailing state-of-the-art research relating to this thesis. Chapter 3 describes the data used, mainly the custom dataset obtained from the orofacial unit Zurich. Chapter 4 documents the methods that produce the results. Chapter 5 details results found. Chapter 6 critically discusses the results in relation to the research questions. Chapter 7 proposes avenues for future research and concludes the paper.

2 Background

This section introduces the terminology and concepts used in this paper.

2.1 Natural Language Processing

Natural language processing concerns all research into understanding text and is a massive field. Therefore this section covers only background knowledge relevant to this paper.

Embeddings

Embeddings are latent space representations of the input. To obtain this representation from a sentence or word, it is passed through an encoder. The output of the encoder is a vector which represents the sentence or word in that model. This will be referred to as an embedding in this paper.

Detailed below is background information for the various techniques this paper will feature to obtain embeddings for text.

Word2Vec

Word2Vec describes the model published in 2013 [Mikolov et al., 2013a]. It introduced several concepts that allowed for more computationally efficient generation of word embeddings. They further improved the model by subsampling the frequent words during training and using a simplified variant of noise contrastive estimation during training [Mikolov et al., 2013b].

Their main innovation was two architectures, namely continuous bag-of-words model and continuous skip-gram model [Mikolov et al., 2013a]. Continuous bag-of-words attempts to predict the current word based on the context that it is given (see Figure 1). Continuous skip-gram is similar, but it tries to predict the context based on the

word that it is given. The continous skip-gram architecture is used by part of this paper and will be referred to as skip-gram for readability.

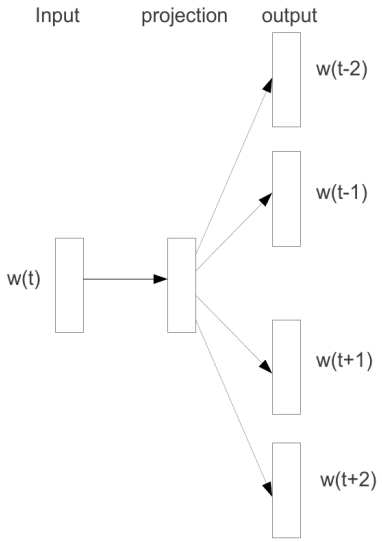

Figure 1: Skip-gram Architecture from [Mikolov et al., 2013b, 2]

Bidirectional Encoder Representations from Transformers (BERT)

Natural language processing found renewed interest after a paper came out about transformers [Devlin et al., 2019]. In the following years research compounded on that work, in this subsection this concept of a transformer will be explained.

Introduced by BERT, transformers use multi-head attention to process their inputs. They are heavily researched due to the mechanics of the attention part. Where previous research relied largely on sequential inputs, the attention aspect of transformers allowed the transformer itself to decide which parts of the input are important for the results.

Sentence Transformers

Sentence BERT (SBERT) [Reimers and Gurevych, 2019] is an adaptation on the BERT transformer which handles sentences. This step was taken because the BERT transformer had no clear path to contextualizing entire sentences. Standard approaches were to take all the word embeddings of a sentence and average them. This however resulted in suboptimal performance, hence Sentence BERT was proposed and accepted as a solution.

2.2 Contrastive Learning

With the original ideas of contrastive learning being from the 1980s, this technique came back when it was found to be quite useful in self-supervised use cases. This concept has several state-of-the-art models using it to great effect. Recently the SimCLR [Chen et al., 2020] played a large role in inspiring new research into the area. In the context of this paper, the implementation of SimCSE [Gao et al., 2021] is relevant. For the unsupervised part of their research, they used a self-contrastive measure combined with dropout to train a model. Their paper which produced SOTA results is described further in related work (see 2.4).

2.3 Clustering

Clustering involves sorting data into groups dependant on maximizing or minimizing certain criteria. This is highly relevant for this work as there aren't labels provided with the data. Since we don't have the ground truth, external criteria aren't applicable to this work. Hence this paper relies on criteria that are known as internal validation criteria. Below are the criteria that this paper relies on to determine the quality of the clusters. Along with some additional concepts that pertain to this paper.

KMeans

KMeans is an algorithm that clusters data into k groups, where k is passed to the algorithm before running. The algorithm minimizes a critereon called inertia, which represents the within-cluster-sum-of-squares. It is a fast algorithm and well known.

Gaussian Mixture Models

Gaussian mixture models are probabalistic models which assumes all points are generated from a mixture of gaussian distributions. In this work just the gaussian mixture is used, which implements the expectation-maximization algorithm. The only hyperparameter that a gaussian mixture needs to fit the data is the number of components, which will be represented as n components in this work. This paper will refer to the gaussian mixture model as GMM.

Principal Component Analysis

Principal component analysis (PCA) was introduced in 1999 [Tipping and Bishop, 1999] and serves to reduce dimensionality while preserving as much variance as possible. This is important in a situation where you have computational constraints and therefore have to reduce the dimensions of your data. It can also be used for visual representations, where the data is reduced to 2 or 3 dimensions for visualization in a graph.

T-distributed stochastic neighbor embedding (TSNE)

T-distributed stochastic neighbor embedding (TSNE) is a technique that aids with visualizing high dimensional data. Its goal is to model points closer together that are more similar based on a probability distribution calculated on the higher dimensional data. [van der Maaten and Hinton, 2008]

Clustering Metrics

There are various methods to evaluate clustering performance. As the data doesn't provide ground truth, none of the metrics will require any knowledge of it. These are known as internal clustering metrics, which will be used to determine the quality of the clusters. Crucially the metrics will motivate the amount of clusters that are deemed optimal. An important task which is further elaborated in section 4.4. Table 1 shows the metrics and their associated optimal values.

Metric	Optimum approaches
Akaike Information Criterion	$-\infty$
Bayesian Information Criterion	$-\infty$
Calinski-Harabasz Index	$+\infty$
Davis-Bouldin Index	0
Silhouette Coefficient	1

Table 1: Optimal values for respective clustering metrics

Calinski-Harabasz Index

The Calinski-Harabasz Index (CH Index) was introduced in 1974 [Caliński and Harabasz, 1974]. This index calculates dispersion, which is the sum of distances squared, of elements within clusters and the dispersion between all clusters. A higher CH Index value indicates better clustering.

Mathematical description

Given:
E as the set of data, n_E as the size of the dataset, k as the amount of clusters, C_q as the set of points in cluster q, c_q as the center of cluster q, c_E as the center of E, n_q as the number of points in cluster q, $tr(X)$ as the trace of Xb.
Then the index is computed as follows:

Calculate the between group dispersion matrix:

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T \tag{2.1}$$

And the within-cluster dispersion matrix:

$$W_k = \sum_{q=1}^{k} \sum_{x \epsilon C_q} (x - c_q)(x - c_q)^T \tag{2.2}$$

Then the Calinski-Harabasz Index equates to:

$$CH - index = \frac{tr(B_k)}{tr(W_k)} * \frac{n_E - k}{k - 1} \tag{2.3}$$

Davis-Bouldin Index

This Davis-Bouldin Index (DB Index) was introduced in 1979 [Davies and Bouldin, 1979]. This index calculates the size of the clusters and their distances between each other. It then averages this into a similarity value which is indicative of the separation between the clusters. A DB index value of 0 indicates optimal clustering, hence positive values closer to 0 imply better clustering.

Mathematical description

Given:
s_i as the average distance between the centroid of the cluster i and each point in cluster i, d_{ij} as the distance between cluster centroids i and j.
Then the index is computed as follows:

Calculate the similarity measure:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}} \tag{2.4}$$

Then the Davis Bouldin Index equates to:

$$DB - Index = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij} \tag{2.5}$$

Silhouette Coefficient

The Silhouette coefficient was introduced in 1987 [Rousseeuw, 1987]. It describes how well the clusters are defined. A Silhouette coefficient has values between $[-1, 1]$ with values closer to 1 indicating better clustering.

Mathematical description

Given:
a as the average distance between a point and all other points in the same cluster, b as the average distance between a point and all other points in the closest neighboring cluster. Then the silhoutte coefficient (sil_cof) equates to:

$$sil\_cof = \frac{b - a}{\max(a, b)} \tag{2.6}$$

Akaike Information Criterion

The Akaike Information Criterion (AIC) is an estimator of prediction error. A lower AIC indicates that the model has a better fit to the data.

Mathematical description

Given:
$\hat{L}$ as the maximum likelihood of the model, d as the number of parameters. Then the AIC equates to:

$$AIC = -2\log(\hat{L}) + 2d \tag{2.7}$$

Bayesian Information Criterion

The Bayesian Information Criterion (BIC) is similar to AIC, but it increases the scale of the penalty based on the amount of parameters. A lower BIC indicates that the model has a better fit to the data.

Mathematical description

Given:
$\hat{L}$ as the maximum likelihood of the model, d as the number of parameters, N as the number of samples. Then the BIC equates to:

$$BIC = -2\log(\hat{L}) + \log(N)d \tag{2.8}$$

2.4 Related Work

This section aims to contextualize the current state of research into this area. It features research this paper builds on and research that proposes different methods to the current topic.

Competitions

In the NLP field there are countless competitions that offer their data sets to achieve SOTA results. Mentioned are competitions similar and relevant to this work. Due to the fact that most competitions have ground truths, it's not a perfect comparison. Some of the featured competitions offer avenues for future research.

Semeval-2022 Task 1 [Mickus et al., 2022]
Comparing dictionaries and word embeddings

This competition aims to research the creation of dictionaries and their mapping to word embeddings. It is a possible extension by using dental dictionaries to further verify whether the symptoms are caputed well.

MedSTS [Wang et al., 2020]
A resource for clinical semantic textual similarity

This paper introduces a semantic textual similarity dataset for the medical domain. They are planning on releasing a task using their MedSTS_ann corpus.

Unsupervised Deep Embedding

The "Deep Embedding Clustering" method introduced in [Xie et al., 2016] resulted in a significant SOTA improvement. This work was important in establishing using learned representations within the context of a clustering objective.

SimCSE

This research innovates on the sentence BERT research and is highly relevant to this work. [Gao et al., 2021] builds on previous research into the nature of the embeddings of BERT models. This previous research, such as [Li et al., 2020], formulates an anisotropy problem concerning the learned embeddings of BERT models. To address this problem they propose two methods, one for unsupervised and supervised tasks. The unsupervised method is relevant for this research and its visual representation is present in figure 2. As this work only refers to the unsupervised method of the SimCSE paper, SimCSE will be synonymous with unsupervised SimCSE onwards.

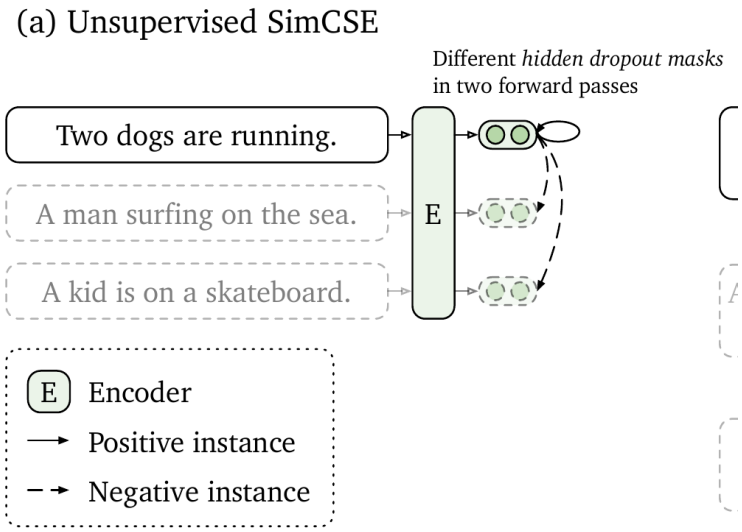

Figure 2: SimCSE unsupervised process, from [Gao et al., 2021, 2]

The unsupervised method of SimCSE relies on using dropout, which is a well known technique to prevent overfitting of neural networks [Srivastava et al., 2014]. Whereas a supervised approach would take predetermined positives and negatives to train, an unsupervised approach doesn't have access to this ground truth. The innovation of SimCSE is to make the sentence pointing to itself a positive pair while using dropout as noise to ensure a different embedding. The negative pairs are other sentences in the corpora. They found SOTA results using this technique and state that it " *acts as minimal "data augmentation" of hidden representations* " [Gao et al., 2021]

3 Data

This section elaborates on the considerations and decisions made regarding the data used by this project. As the data led the research direction this section will provide more context for the methods chosen.

3.1 Dataset selection

The selection of the dataset was done by obtaining the patient dental data. After which the second dataset was selected based on similarity of the task at hand and to the dental data.

Dental data

The dataset provided for the thesis corresponds to self-reports from the orofacial pain unit of Zurich. These were collected via a web-based interdisciplinary symptom evaluation (WISE) [Ettlin et al., 2016].

Data format

The data was presented in *.csv* format, a table with each column representing user data and the row being the relevant patient. The part of the data that this work focusses on has the structure seen in table 2.

Response ID	Date submitted	Q_1	...	Q_n
1	2019-11	Ja	...	Kiefer
2	2020-04	Ja	...	Zahn/oberkiefer
...	...	...	...	...
2236	2020-07	Nein	...	Kopfschmerzen

Table 2: Example data for dental dataset (*.csv*)

Data types

Figure 3 shows the distribution of data types in the dental dataset. What is clear from this figure is that text is the most informative of the data types. Which is part of the reason this work focusses on the free text responses from patients.

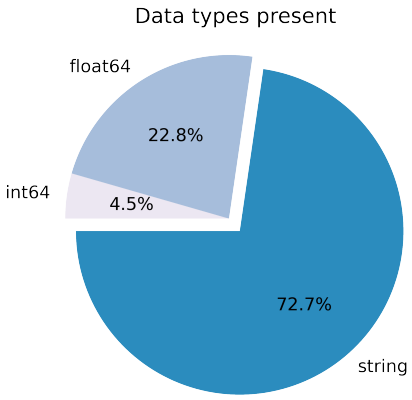

Figure 3: Data types in the dental dataset

Languages

Table 3 shows the percentage of start languages. These 'start languages' correspond to the language of the web page when filling out the questionnaire. Noteworthy is that these languages don't fully align with the language of the responses.

Language	Amount of rows	Percentage
German	2035	95.6%
English	63	3%
Portuguese	29	1.35%
Thai	1	0.05%

Table 3: 'Start languages' present in the dental dataset

Unique responses

It is clear why previous research on this dataset took only the first 3 columns, as they have low nan value counts compared to the rest of the columns, as seen in Figure 4.

However there are columns that are important to not filter out, as these may be crucial for the clustering part.

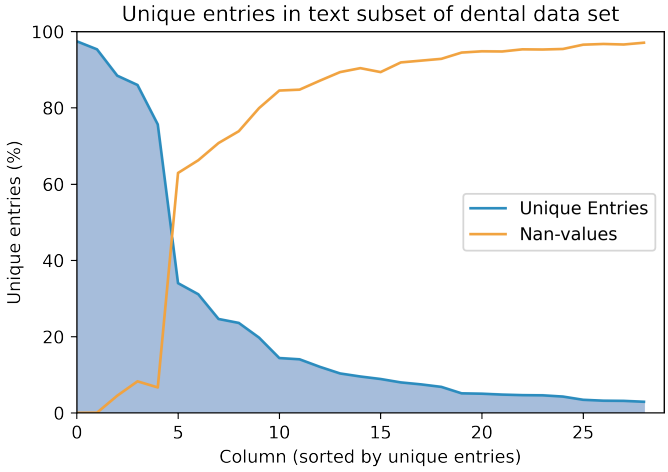

Figure 4: Unique entries in the text subset of the dental dataset

Top 3 questions

As seen in Figure 4 there are a few columns of which most patients responded. These 3 questions which got the most responses:

1. **"Please describe your chief complaint for which you seek consultation:"**
   Responses: 2128

   Example: *"Kieferschmerzen links, beim Kauen."*

2. **"What do you expect as the result of the examinations and treatments in our clinic?"**
   Responses: 2127

   Example: *"Ansatz für die Milderung der Schmerzen finden."*

3. **"Which factors aggravate your complaints? (e.g. chewing hard / soft food, biting, drinking, mouth opening (e.g. yawning), talking, physical / emotional stress, playing a musicial instrument, ...)"**
   Responses: 2032

   Example: *"physical activity, stress"*

These questions will be referred to later in this work as they hypothetically capture the underlying semantic information the best.

SemEval STS-b

[Cer et al., 2017] is a dataset corresponding to SemEval which is a competition that hopes to incentivize research into text processing. In this paper it will also be referred to as sts. The dataset will be used to support the methods this paper presents.

Data format

The dataset is available online for download as open access. The data itself is contained in *.csv* files with the two sentences that are compared in similarity on the same row. The similarity score is from 0 to 5, where 5 represents the sentences being very similar. The general structure can be seen in table 4

Genre	Filename	Year	Score	Sentence1	Sentence2
main-captions	MSRvid	2012	0.5	A woman is writing.	A woman is swimming.
main-forum	images	2014	1.6	Two horses ...	A group of horses ...
...	...	...	...	...	...
main-news	headlines	2014	3	Prime Minister to ...	investments during ...

Table 4: Example data for STS-b dataset (*.csv*)

3.2 Preprocessing

This section on preprocessing will predominately feature the private dental dataset, as the sts-b dataset was already processed. Any steps that occur to both datasets are detailed below.

Common preprocessing steps

These actions were performed to both datasets, mainly for the word2vec model. The actions are also performed in the order presented.

Lower case

The text is all transformed to lower case to avoid cases where respondents miss-capitalized words. Which would be counted as separate words, which this step fixes.

Tokenization

Tokenization describes taking a sentence and producing tokens which each contain a word. This is a common technique and is done for faster processing later. For this step the function *word_tokenize* is used from the nltk library [Bird et al., 2009].

Punctuation removal

Since punctuation is present in almost all grammatically correct sentences it is important to remove before training the word2vec model.

Stopword removal

In addition to punctuation, stopwords are commonly used throughout language. Hence removing them is important since not doing so could result in the model presenting sentences as similar due to overlap in stopwords.

In this implemenation stopwords are sourced from *nltk.stopwords* [Bird et al., 2009]. Then just the english and german words are used for filtering. With a minor modification being removing 'man' from the german stopwords, due to its relevance for determining similarity in downstream tasks.

Dental Data

Removing newline characters

As some of the responses are very long, they naturally include the newline character "\n". To avoid the models missinterpretting this symbol it is removed from the dental dataset.

Handling of nan-values

A lot of the response fields had nan values, which meant they were empty. Figure 5 depicts the amount of nan values percentage wise for the dentist dataset. Not all questions pertain to each individual, hence not everyone felt the need to respond to all questions. Therefore its important to state the method of handling these nan-values. In some research nan-values are treated as a signal, however for the purposes of this research they are not further used in the later stages. The preprocessing filters them out to the best capacity.

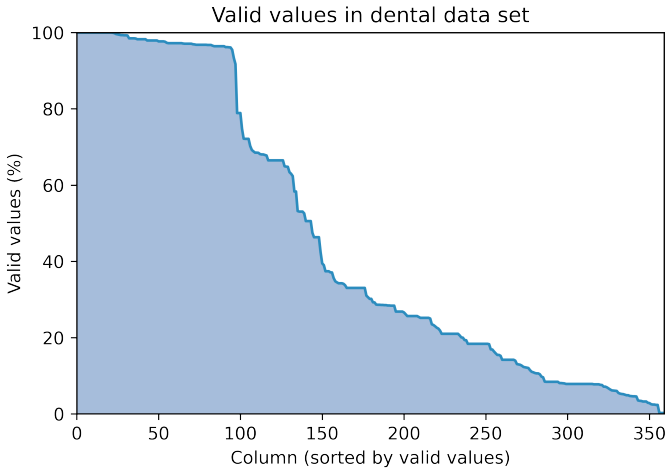

Figure 5: Valid values in the dental dataset

Privacy preservation

Due to the nature of the data privacy has been considered throughout the work. In addition to the already anonymized dataset, some columns were removed during preprocessing that may lead to reidentification of individuals. Patients in this work will only be referred to using their IDs as no knowledge of their identity has been given.

Response threshold

To maintain the quality for the downstream stages, only the columns with more than 16 unique responses were selected. The reason being that below this number there were either too few responses to the question or the text responses were variants of yes/no or similar likert scale answers.

4 Methods

Given the provided background information, this section details the methods used to answer the research questions. The sections are divided in a similar manner as in the code, hence the implementation of each method can be found in the corresponding folder at GitHub [0].

4.1 Overview

This section will present a short overview to help understand the roles of each component better.

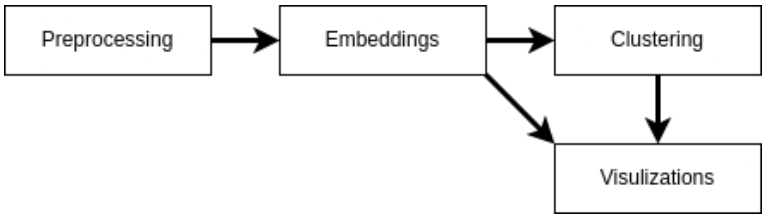

Figure 6: Overview of the process

The above figure Figure 6 gives a brief visual overview of the pipeline that is used. Initially the datasets are passed through the preprocessing pipeline, which you'll find more information on in chapter 3, along with some additional information present in section 4.2. Next, the cleaned dataset is passed to the embeddings stage, described below in section 4.3. The embeddings that are generated are sent to their respective visualizations and importantly to the clustering stage. In the clustering stage, which will be elaborated in section 4.4, the embeddings are clustered using various techniques. Those results will then be sent to the visualization stage so they can be further explored. The visualization stage aims to communicate the findings and allow for exploration of the computed dataset, details of which are found in section 4.6

[0] github.com/TemporalData/UDC

4.2 Preprocessing

This component takes in the datasets and processes them to the correct format for the remaining components. The main actions are to extract specific features from the dental dataset and to tokenize for the word2vec embedding generation. For a more detailed description, please refer to section 3.2. All the preprocessing code is in its own virtual environment managed by poetry. The execution is mainly done thourgh jupyter notebooks.

4.3 Embeddings

Embeddings are generated in two ways, Word2vec and SimCSE. Both have their respective python libraries and virtual environments, importantly for the SimCSE part a GPU was used to train the model.

Word2Vec

This embedding component takes in a processed input (see 3.2) and generates an embedding for each word in the dataset. Our implementation of word2vec uses the skipgram model, which is further detailed in section 2.1. The hyperparameters chosen for our word2vec models were the following. All word2vec models had *min_count* set to 1 which means it should include all words in the corpus. They also had *vector_size* set to 256 and *epochs* set to 20.

SimCSE

This embedding component takes in the raw sentences and generates a representation of the sentence. It is first trained on all the sentences in a manner described in section 2.4. The hyperparameters chosen follow those in the SimCSE paper (found in the appendix [Gao et al., 2021]). Namely, batch size was equal to 64 and the learning rate was equal to $3e-5$. The sentence embeddings generated are a vector of length 768.

Similarities and differences

There are some important differences in the implementation of word2vec and simcse that are relevant. Both models were fairly quick to train, but the word2vec model was faster. The word2vec model is trained from the ground up while the simcse model uses a pretrained BERT model. Taking SOTA performance as a starting point will likely skew the simcse model towards performing better. The simcse model also has a larger vector for the sentence embeddings, which may also favor the simcse model.

4.4 Clustering

The clustering metrics are calculated using the python package scikit-learn [Pedregosa et al., 2011].

K-means

K-means models are trained using the scikit-learn library. The amount of clusters k is iterated over the range $[2, 10]$ to find the best performing (see table 8 for details).

Gaussian Mixture Models

Gaussian mixture models (GMM) are trained using the scikit-learn library. The amount of components of the gmm are iterated over the range $[2, 10]$ and the best performing is chosen according to the details below. For GMM the metrics AIC and BIC are also considered.

Clustering Metrics

The clustering metrics described in 2.3 will be combined into new indicies. These indicies represent the performance of the model and will be used to determine the optimal k or n for a given model. The two main branches of clustering models are kmeans and gaussian mixture models. Each branch has their respective index which will both be set up so that the maximalization of the index indicates better performance of the model. The metrics chosen have been widely used in research, the

reasoning and drawbacks behind their combination are further discussed in section 6.1.

Scaling indicies

As the indicies presented have different ranges of potential values, they are all scaled to $[0, 1]$. The sole exception to this is the sillhoutte coefficient, which has a range of $[-1, 1]$. The other inidicies are scaled using the *minmax_scale* function from the sklearn library [Pedregosa et al., 2011].

The computation of the scaled indicies occurs as follows: First the index is computed for models trained on differing $k \in [2, 10]$. Then the vector of length 9 is passed to the *minmax_scale* function. This function transforms the values to a range of $[0, 1]$ where 0 corresponds to the min of the recorded indicies, 1 refers to the max. They are then stored, e.g. the DB index is scaled and stored as "db_scaled". After which the scaled values are used in the performance indicies described below.

KMeans performance index

The aim of the KMeans performance index (KM_perf 4.1) is to represent the clustering performance for a given k clusters. This is achieved by combining the scaled indicies introduced in 2.3. For KMeans these indicies are the CH index, DB index, and the Silhouette coefficient. The scaled version of the CH index is summed together with the Silhouette coefficient, and then the scaled DB index is subtracted from the result (see 4.1). The reason that the DB index is subtracted is because this index represents better performance the closer to 0 it gets. Hence *db_scaled* with a value of 1 represents the worst performing k in the sampled range. KM_perf has a range of $[-2, 2]$ and maximizing this combined index indicates a better clustering result. The validity and signficance of this KMeans performance index is discussed in 6.1.

$$KM\_perf = ch\_scaled - db\_scaled + sil\_cof \tag{4.1}$$

GMM performance index

The aim of the GMM performance index (GMM_perf 4.2) is to represent the clustering performance for a given n components. The GMM performance index is equivalent to the KMeans performance index in addition to GMM specific metrics

AIC and BIC (see 2.3, 2.3). The AIC and BIC are scaled using *minmax_scale* as with the other indicies. Since the minimization of both AIC and BIC indicates better clustering performance, they are subtracted from the summation (see 4.2). GMM_perf has a range of $[-4, 2]$ and maximizing this combined index indicates a better clustering result. The validity and signficance of this GMM performance index is discussed in 6.1.

$$GMM\_perf = ch\_scaled - db\_scaled + sil\_cof - aic\_scaled - bic\_scaled \quad (4.2)$$

Example

In table 5 the process of selecting the best amount of clusters for a given question is shown. The bold numbers are those that the indicies described above indicate they are optimal.

k	DB	CH	Silhouette	db_scaled	ch_scaled	combined_index
2	4.30	**41.95**	0.051	1.00	**1.00**	0.05
3	3.89	34.60	0.052	0.61	0.64	0.08
4	3.33	33.33	0.061	0.08	0.58	**0.56**
5	3.47	30.68	0.061	0.22	0.44	0.29
6	3.40	28.70	0.066	0.15	0.35	0.26
7	3.37	26.75	0.069	0.12	0.25	0.20
8	3.39	24.61	0.072	0.14	0.15	0.08
9	3.31	22.72	0.072	0.06	0.05	0.06
10	**3.25**	21.65	**0.077**	**0.00**	0.00	0.08

Table 5: KMeans clustering metrics of question 5 from dental dataset (k = [2,10])

$$KME\_perf(k = 4) = 0.58 - 0.08 + 0.061 = 0.56 \quad (4.3)$$

The equation 4.3 above shows how 4 was found to be the optimal amount of clusters. It is simply equation 4.1 with the values substituted in.

4.5 Aggregate Clusters

The aim of aggregate clusters is to combine the findings of the per question clustering into an overall cluster assignment. In an ideal scenario these aggregate clusters contain, per cluster, all patients that share an underlying ground truth. The method of finding these aggregate clusters is detailed below.

In this work these overall clusters are generated based on a criteria which enforces that the smallest cluster can't be smaller than a certain size. The minimum cluster sizes of 25 and 50 are used. Those will be referred to as 'Min group of 25' and 'Min group of 50'. The procedure of finding the aggregate clusters given a minimum amount is as follows:

Method to find aggregate clusters

1. Start with the question that has most responses.
2. Divide the data into groups based on the cluster assignment of this question.
3. Check if this division results in clusters with size lower than the minimum amount.
4. If true, revert the division and finish the loop
5. Otherwise, continue with the question that has most responses and has not been seen, then go to step 2 using subsets based on current groups.

This method uses the questions with the most responses first to maximize the chance that people are placed into clusters based on their responses. Sometimes people did not respond to a question that is used during this method, these individuals are then placed to the closest found cluster assignment.

4.6 Visualization

All interactive visualization of this work use tensorboard [Abadi et al., 2016], specifically the projector tool included in the library. This tool allows for exploration of the embeddings as well as the clusters. It is also fit for use by both dentist and patients, however some introductory material such as the background chapter (2) may be required before they can use it effectively.

Embeddings

The embeddings are parsed as .tsv files and uploaded in to the correct structure. The projector then performs PCA or the chosen dimension reduction technique to visualize the embeddings. With this visualization one can determine whether or not similar concepts or words are close together. Since you reduce the dimensions the visualization isn't fully generalizable, but this drawback is in line with the exploratory purpose of this work.

Clustering

The clustering visualization is identical to the embedding visualization with added labels. The labels of the clusters are added as metadata, and can therefore be selected as the user wishes. One limitation is that there only exist up to 10 unique colors for the visualization. Which users should be aware of when visualizing greater than 10 classes as it could be misinterpreted. Further one can select the type of cluster one wants to visualize between the following options:

Question specific

To allow further exploration, for each question the user can select to visualize the optimal lusters found by KMeans and GMM. This corresponds to the KMeans model with the best KM_perf and the featured GMM model has the best GMM_perf.

Aggregate cluster

The overall clusters detailed in section 4.5 can be selected with two options. *"Min group of 25"* shows the clusters found by grouping with at least 25 in a cluster, identical with *"Min group of 50"* aside from it being 50 people. These overall clusters can be selected on all the visualization from the simcse model.

Tensorboard projector

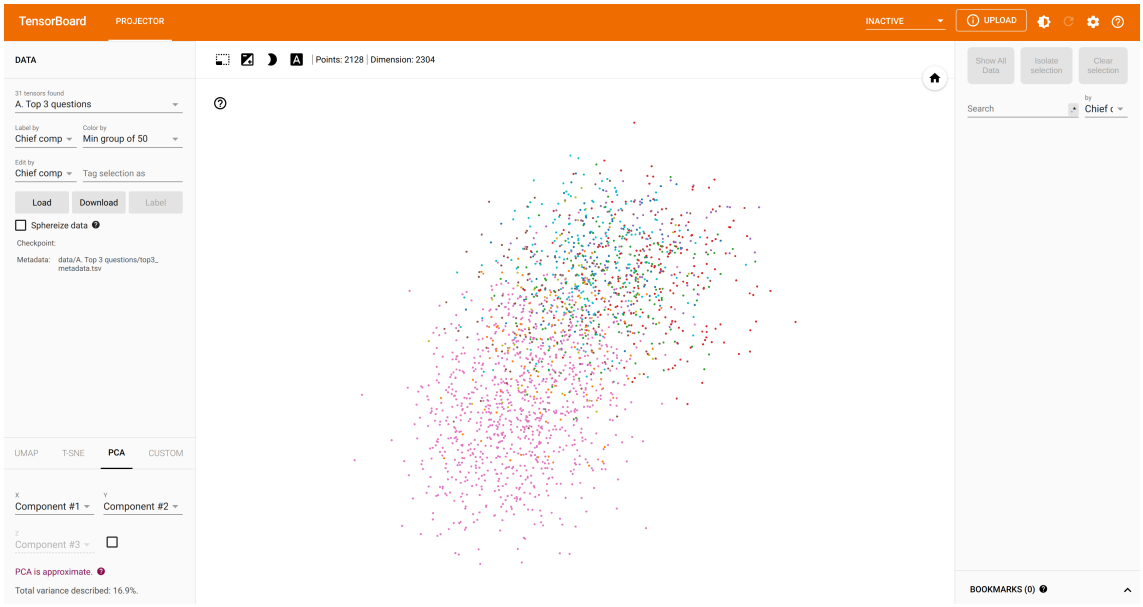

Figure 7: Tensorboard projector for dental dataset with 'Top 3 questions' selected

The projector is a part of the tensorboard feature set, it is used to explore the embeddings of encoder decoder architectures. Above is Figure 7 of the projector with the dental dataset loaded. The main parts of the projector will be explained below.

Points graph (center)

In the center of the projector is the important part of the tool. Here you can see each higher dimensional vector represented in 2D or 3D depending on your setting toggle in the bottom right. Each point represents a sentence in our case, with the color being determined by the selected 'Color by' labels. These labels are the aggregate groups min 25 and min 50, which are further detailed in section 4.5. Aside from those labels, for the per question visualizations you can also chose the best performing KMeans and GMM clustering result. Clicking on a point will show its characteristics as well as highlight its nearest neighbors on the graph.

Dataset selector (top left)

This selector controls which data is passed to the dimension reduction method. In the case of the dental data, the options are as follows. First you have 'A. Top 3

questions', which combines the sentence embeddings of the top 3 most answered question for each patient. Then you have 'B. All responses' which shows all the sentences from the dataset. The goal of this option is to explore the relation between sentences without siloing them by patient or question. Lastly you have each question that can be selected. Selecting a question will show the embeddings of the available responses, lower amount of points mean that fewer people responded to that question.

'Label by' selector (top left)

This selector determines what you see when you hover over the individual points. It can be used to generate insights by varying over the options for a selected dataset.

'Color by' selector (top left)

This selector can be used to alter the colors of the points in the points graph. The important option is that you can select the aggregate clusters which can be used to explore the dataset more efficiently.

Dimensionality reduction techniques (bottom left)

At the bottom left you'll find the dimensionality reduction techniques that can be applied to explore the dataset. The projector supports UMAP, TSNE, and PCA out of the box. For this work the most relevant is PCA, however TSNE was also used. For PCA, importantly, you can select whether you want a 3d visualization. This option is preferable when exploring the dataset, however this paper was limited to using the 2d visualizations for its figures. It also tells the user the explained variance.

Search function (top right)

With this field one can explore the dataset. The user can select the field it wishes to search and when doing so the visualization will highlight the relevant points in the center points graph.

5 Results

This chapter covers the results from the embedding and clustering stage. Then it documents the resulting visualizations.

5.1 Embeddings

In this section the results from the embeddings generated by the multiple variations are presented.

Word2Vec

In this section the focus is on the word2vec models. For each model a word is selected and then the top 5 most similar words are presented.

Dental word2vec model

In the table 6 below there are the top 5 most similar words to *"kopfschmerzen"*. These were found using a word2vec model trained on the dental dataset.

Word	Similarity score
"kopfschmerzen"	
"nackenschmerzen"	0.92
"migräne"	0.91
"kieferschmerzen"	0.89
"kopf-"	0.89
"rückenschmerzen"	0.88

Table 6: Top 5 words similar to *"kopfschmerzen"*
using Dental word2vec model

STS word2vec model

In the table 7 below there are the top 5 most similar words to *"water"*. These were found using a word2vec model trained on the STS dataset.

Word	Similarity score
"water"	
"animal"	0.990
"dock"	0.989
"swimming"	0.988
"backyard"	0.988
"pool"	0.987

Table 7: Top 5 words similar to *"water"*
using STS word2vec model

Simcse

In this section there are results from the two simcse models on both datasets.

Spearman metric

As simcse uses the STS dataset during training by computing the Spearman metric, reporting those gives an indication of performance. For reference, the spearman reported by the simcse paper was 76.85 [Gao et al., 2021, 7].

Dental model

For the dental model the spearman metric was 0.605.

STS model

For the sts model the spearman metric was 0.755.

Similar sentences from dental model

Taking the simcse dental model and finding similar sentences is comprised of finding the neareset neighbors of the embedded version of a sentence. For this task we take all the sentences together, not dividing by question. Due to the absence of supervised labels qualiatative analysis of similar sentences is presented.

Simcse similarity example

Taking the sentence *"Penizilin ,"* and finding the top 5 similar sentences determined by cosine distance:

1. "Penizillin"
2. "Alkohol, Rauchen"
3. "Entspannen, Botox-behandlung"
4. "Pelicilin"
5. "Penicln"

The model performs well in associating incorrect spellings together, however it has also suggested two sentences that may be questionable. Suffice it to say that some caution may be required when interpretting similar sentences from the 'all sentences' data.

5.2 Clustering

The dental dataset that has been encoded using simcse was taken to be clustered. The simcse was chosen due to its innovative approach and promising application to the dataset.

Clustering example of Question 11

Clustering the responses of question 11 resulted in $k = 3$ being optimal for KMeans and $n = 4$ being optimal for GMM. KMeans came out as the best variation based on their metrics. This result can be seen visually at Figure 8. Below is a brief description of each cluster:

1. (Blue) Characterized by various types of treatment ("behandlung")
   Most central sentence: *Wurzelbehandlung*
2. (Red) Characterized by various types of teeth procedures
   Most central sentence: *Nach Zahnspange*
3. (Pink) Characterized by longer explanations of dental procedures
   Most central sentence: *Als ich zahnstein zweimal gemacht hatte ging es ...*

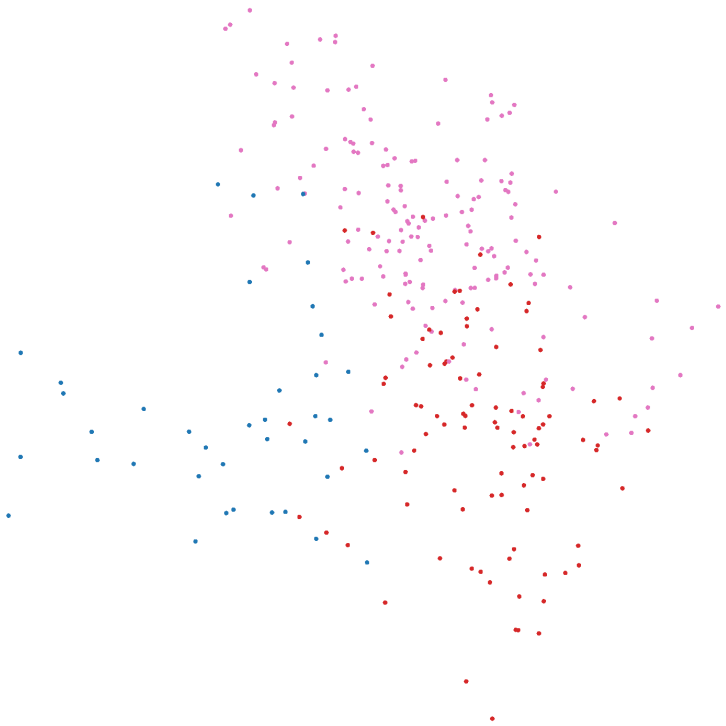

Figure 8: Question 11 responses with 3 kmeans clusters colored

Best cluster model per question

The table 8 shows the results of finding the best cluster per question using the techniques described in the methods chapter (4). The numbers in bold represent the chosen amount of clusters for each question. It is clear from the table that KMeans seems to be favored, hence for the cases where both models had equal scores the KMeans variant was chosen. Of note is that having the same score is indicative of having the same assignments, so this choice should not affect the end results.

Question Number	Shortened text	KMeans best k	GMM best n	Chosen model
1	Chief complaint	**2**	**2**	Both
2	Expected result	**2**	2	KMeans
3	Aggravating factors	**4**	3	KMeans
4	Stops you from	**2**	2	KMeans
5	Alleviates complaints	**2**	3	KMeans
6	Eating habits	**4**	5	KMeans
7	Diagnosis	**2**	2	KMeans
8	Personal diagnosis	**2**	2	KMeans
9	Complaint attacks triggered by	**3**	4	KMeans
10	Bothersome life events	**2**	2	KMeans
11	After which treatment	**3**	4	KMeans
12	Pain remarks about head or face	2	**2**	GMM
13	Allergies	**2**	3	KMeans
14	Pain remarks about body	**2**	2	KMeans
15	Other problems	**2**	**2**	Both
16	Tell us anything else	**2**	2	KMeans
17	How chief complaint started	**2**	2	KMeans
18	Other quality of chief complaint	**2**	2	KMeans
19	Which other treatment	**3**	2	KMeans
20	I suffer from	**3**	3	KMeans
21	Other mouth/jaw related habits	**2**	2	KMeans
22	After which accident	**2**	2	KMeans
23	Other quality at onset of illness	**2**	10	KMeans
24	After which illness	**2**	2	KMeans
25	After which emotional stress	**2**	10	KMeans
26	Pain remarks about torso	**10**	2	KMeans
27	Other performed treatments	**7**	3	KMeans
28	After which operation	**5**	3	KMeans
29	After which physical stress	**2**	2	KMeans

Table 8: Best clustering method per question for dental dataset(k = [2,10]) Questions orderd by amount of unique values

Aggregate clusters

Using the cluster assignment from the results shown in table 8 the aggregate clusters were generated according to the method described in section 4.5. After testing a range of numbers, the numbers 25 and 50 were chosen for the minimum group sizes. The main criteria for selection was interpretability and visual clarity on the projector.

Min group 25

20 clusters were found after running the procedure to find aggregate clusters with 25 as the minimum cluster size.

Min group 50

9 clusters were found after running the procedure to find aggregate clusters with 50 as the minimum cluster size.

5.3 Visualization

In this section visualizations are presented from the tensorboard projector. The steps to recreate the visualization are also present to aid exploration of the tool and verify results. All the code for the projector are available on the github, what is not present on the GitHub is the dental dataset which is not publicly available.

Top 3 questions

To recreate the visualization (Figure 9) for the top 3 questions follow these steps:

1. Select 'A. Top 3 questions' as the dataset
2. Select 'Min group by 50' on the 'color by' selector
3. Untick the third PCA component

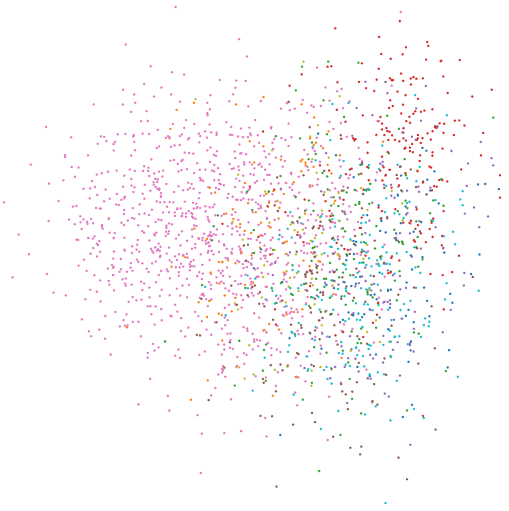

Figure 9: Top 3 questions embedded and colored by "Min group of 50" clusters

In Figure 9 there are 9 clusters present, each identified by a color. For each patient, the sentences for the top 3 questions have been parsed to PCA and reduced to 2 dimensions.

All sentences

To recreate the visualization (Figure 10) for all sentences follow these steps:

1. Select 'B. All Responses' as the dataset
2. Set neighbors slider to 10
3. Search "Chronische Kopfschmerzen"

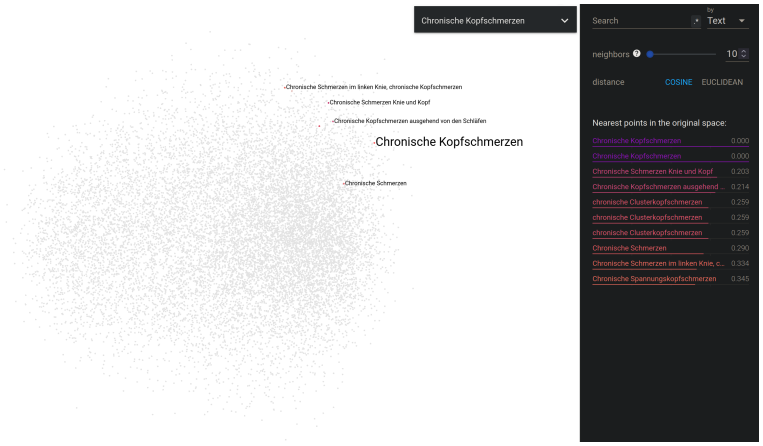

Figure 10: All responses graph with 10 nearest neighbors selected

In Figure 10 the sentence "Chronische Kophfschmerzen" has been selected in the all responses graph. Shown are the 5 closest neighbors based on cosine distance. All neighbors have similar semantic meaning present in their sentences, which the model seems to capture well.

Question based

To recreate the visualization (Figure 11) for question 2 follow these steps:

1. Select 'Q 2: Expected result' as the dataset
2. Select 'k_means' on the 'color by' selector

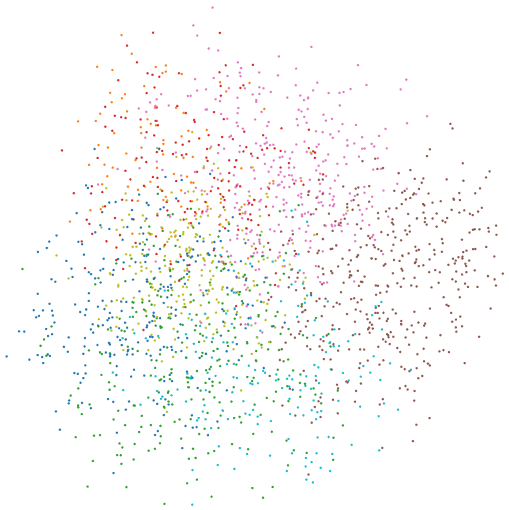

Figure 11: Question 2 with the 8 kmeans clusters shown colored

In Figure 11 the clusters resulting from running KMeans with $k = 8$ are shown. This k was found to be optimal based on the clustering metrics. In the center there seems to be some overlap of clusters, whereas there is more distinction the further the point is from the center.

5.4 Use cases

In this section there are two cases that are relevant to the end users of this research. Those end users, aside from researchers, are dentists and patients. In showing the process a user may go through, it is possible to analyze from the interpretability perspective. This perspective is discussed in section 6.3.

Dentist

The goal of the dentist is to explore whether there is a pattern among patients relating to gums ("Zahnfleisch"). Below is one way this dentist could use the tool this work presents to search for such patterns.

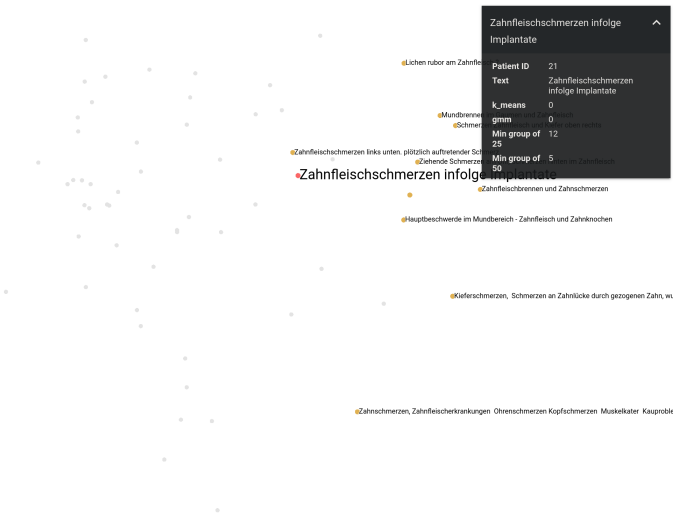

Figure 12: For the question 1 the dentist searches 'Zahnfleisch' and selects patient *(Question 1: Chief complaint)*

As the first step, the dentist searches for complaints relating to gums. After searching, the dentist has selected a patient which is seen in Figure 12. They note the patient ID and the clusters the patient belongs to. Then they move to check the aggravating factor of this patient in the question 3 dataset.

In Figure 13 the dentist has searched for the patient ID 21 and can read the aggravating factors of the patient. It also shows the patients that have similar factors, which the dentist notes. Finally, they want to see if the patient has been diagnosed and what the diagnosis is. So they move to question 7, which asks for the diagnosis of the patient.

Searching the question 7 dataset, the dentist finds that patient 21 did not submit

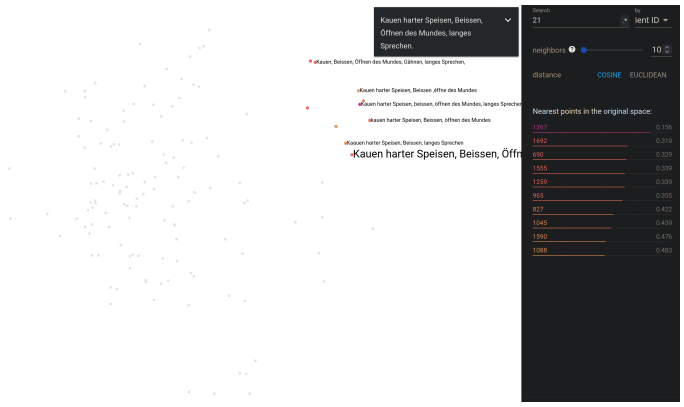

Figure 13: Dentist searches patient 21 in question 3 responses *(Question 3: Aggravating factors)*

Figure 14: Dentist searches similar patients diagnoses in question 7 reponses *(Question 7: Diagnosis)*

any diagnosis. They decide to search for patients that have similar factors as patient 21, namely patient with id 1267. The result of the search is seen in Figure 14, the dentist turns on the text display and reads the diagnosis of patients similar to 1267. Since patient 1267 had similar factors as patient 21, the dentist can hypothesize that patient 21 would likely have a similar diagnosis as those seen on Figure 14. They have quickly found a pattern that they may use as they see fit.

There are of course considerations about this use case that are elaborated in section 6.3. The process detailed above can be replicated using the code on the GitHub and the dental dataset, the process can naturally also be done for other search queries. The aim was to present a potential flow for a dentist so that this can be discussed by evaluating the advantages and drawbacks later in the discussion.

Patient

The goal of the patient is to explore what similar patients may be experiencing. Below is a potential sequence that such a patient may use, the patient with ID 2084 was selected.

Figure 15: Patient 2084 finds similar complaints of other patients
(Question 1: Chief complaint)

The patient starts with finding themselves in the question 1 dataset, this can be seen in Figure 15. They can see there are others that have similar complaints, they also note the clusters they belong to. They then continue to question 4 to find out what it stops others from doing.

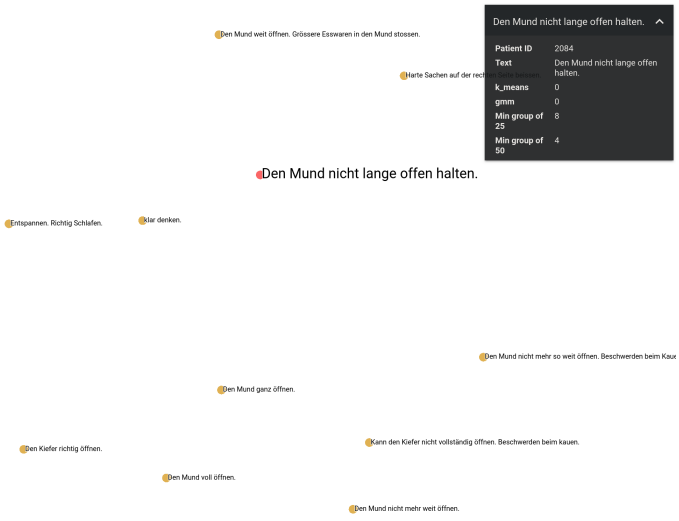

Figure 16: Patient 2084 finds similar responses to question 4
(Question 4: Stops you from)

In Figure 16 the patient has searched their ID and observes what the others in the dataset are experiencing. Finally the patient is curious whether there may be any triggers that they may not know about yet, so they move to question 9.

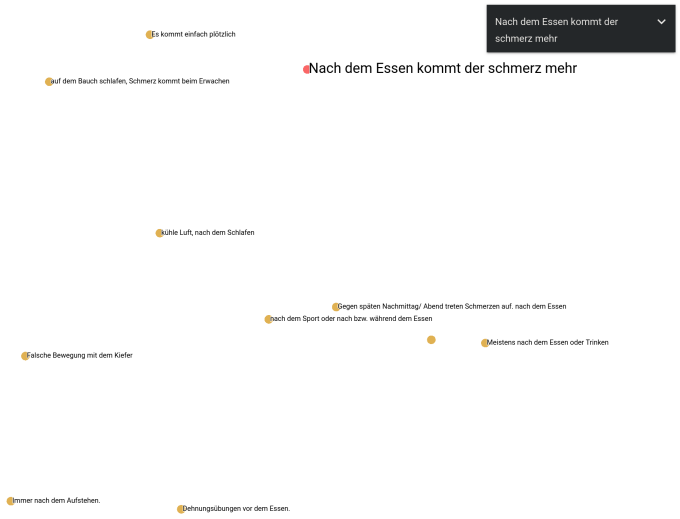

Figure 17: Patient 2084 finds similar patients for question 9
(Question 9: Complaint attacks triggered by)

In searching the question 9 dataset the patient notes what triggers attacks in others so they can reflect on whether those may also be the case for themselves.

This use case has considerations that are detailed in section 6.3. The use case was an example of what a patient may learn from exploring the dental dataset using the tool featured in this paper. This process was not exhaustive and many more comparisons could be made by the patient, but additional visuals would be similar to those presented. They also suffice for highlighting the advantages and drawbacks which are described in the discussion.

6 Discussion

This chapter will discuss the results detailed above in the context of the research questions we posed earlier (see 1.2).

6.1 Research Questions

Which proposed metric is best to compare clustering results?

The proposed metrics KM_perf (4.1) and GMM_perf (4.2) are the best metrics to compare clustering results. They do have considerable limitations which are discussed below.

Limits of KM_perf and GMM_perf

Due to the fact that the clusters had to be combined, this work decided to restrict the per question clustering to a max of 10 clusters. Various experiments also showed the weakness of using KM_perf and GMM_perf as indicators for performance. When k was tested for values to 200, the following was observed. The DB index was negatively correlated with k, as was the CH index, while the silhoutte coefficient was weakly negatively corerlated as it had minimal variance over the range. For the GMM models the AIB and BIC were both positively correlated with n.

All these observations point to the counter acting forces between a CH index that always favors a lower k while the DB index always favors a higher k. In the end KM_perf ended up recommending 2 quite often due to this imbalance, as the CH index was always highest. Immediately followed by a number at the end of the range, when testing with $k > 50$. This issue may be resolved by reimplementing a better performance indicator, but it may also be more fundamental.

There are other metrics that perform well that could have been considered, such as S_Dbw from [Liu et al., 2010]. Yet [Arbelaitz et al., 2013, 254] recommends the three

metrics chosen as the best among those they tested. They however state that there is no significant difference between the performance of those three, which invites the reconsideration of using the combination of metrics as a decider for best clustering performance. As [van Craenendonck and Blockeel, 2015] found that none of the measures they proposed could be used to compare the metrics. They importantly highlight the concerns, *"All measures exhibit some undesired properties: sensitivity to points identified as noise, a preference for highly imbalanced solutions, or a bias towards spherical clusterings."* [van Craenendonck and Blockeel, 2015, 7]

How well do Word2Vec embeddings perform in the clustering task?

One of the areas of the word2vec method that has received attention is how to effectively transition from word embeddings to sentence embeddings. Research such as [Arora et al., 2017] proposes a method for this. As the simcse method in this work use more recent research and more promising techniques, generating sentences from the word2vec models wasn't further developed.

For the task of similarity, [Marcinczuk et al., 2021] found that word2vec can compete with BERT models as a measure of similarity. Hence both methods are featured in this paper.

How well do sentence embeddings perform in the clustering task?

SimCSE

Based on all the examples in this paper it is clear that simcse performs well in generating quality embeddings. This is further confirmed by [Wang and Isola, 2020], who found that contrastive learning methods optimize for alignment and uniformity.

Aggregate clusters

The performance of aggregate clusters are hard to judge. They fulfil the task of finding clusters that span all the questions, but the methodology has room for improvement. One potential direction is to change order which the questions are selected for division. Instead of most responses, the distance between clusters in a

question could be used to pick the questions which feature well defined clusters first. Overall, further research in the best way to combine all the clusters found for each question could be beneficial.

Which proposed method is best to explain the clustering results?

Of the various visualization presented by this work, the best is the visualizations based on each question individually. All of the methods are evaluated below, they are ordered by quality with the last being the best.

All responses

The advantage of having all sentences is that one can explore patterns that may exist irrespective of which question the sentence answered. However removing this information is also a massive drawback as there are 16315 sentences in the all responses dataset. This makes traversing the graph fascinating but not very clear, which is why this visualization was not chosen.

Top 3 questions

The main advantage of the top 3 questions visualization is that it summarizes the most information dense questions. The reason this wasn't the best visualization because after the dimensional reduction only 13.4% of the variance is explained. It is lower than the other visualizations due to the fact that the embeddings of three questions were combined. Since interpretability is an important aspect of the work, this low explained variance could lead to incorrect conclusions so it wasn't chosen.

Question based

The ability to explore every question and its responses solidifies this visualization as the best of those presented in this work. Being able to select the optimal clusters that were identified also allows for further insight.

6.2 Dataset analysis

In this section the datasets used for this paper will be evaluated to give the reader a better understanding as to the limitations of this research.

Dental dataset

The dataset is unique and is being researched to its fullest extent [Schneider et al.]. There were some considerations regarding the dataset that are discussed below.

Purely textual data

Taking just the free text from the dataset is a choice that was made to focus the work into a more innovative direction. The clear drawback is that there may have been informative columns that were ignored by this decision. However the non-textual data had a mix of data collected by the dentist added to it. Which meant that some of the columns weren't really the patient answers, just the dentist measurements that the patient filled in. Along with various calculations, this made these additional columns less interesting to include in the entire pipeline. Therefore taking only textual data was a preferred route.

Response threshold

After extracting the textual data from the dental dataset the next consideration was taking only the columns which had more than 16 unique responses. The main reason this measure was taken was due to the amount of yes/no columns, along with the likert scale responses. Having a yes/no response in the pipeline would not add more information unless the question was additionally encoded somehow. Since this research did not attempt to also encode the question into the embedding the aforementioned measure was taken.

Language

The decision not to split the data for training or inference based on language, comes from the fact that the column *'Start language'* in the dental dataset wasn't reliable. The distribution of these languages can be seen in table 3. There were some occurrences of german text being present in the responses of the *'en'* tag and vice

versa with the german tag *'de'* having english responses. Further encouraged by recent research in multi-language transformers this paper did not differentiate between languages during training and inference.

This does however result in practical issues due to the fact that there were far more german responses than any other language. Visually this can be seen on the TSNE plot shown at figure Figure 18. On the right of this figure two dense clustering of points can be identified (both colored red). These clustering of points all include sentences from english and portuguese respectively. The results on the TSNE plot can't be generalized due to the randomness of TSNE. However it can be stated that these sentences clearly fall closer together in the embedding space than sentences of german language based on the objective of TSNE.

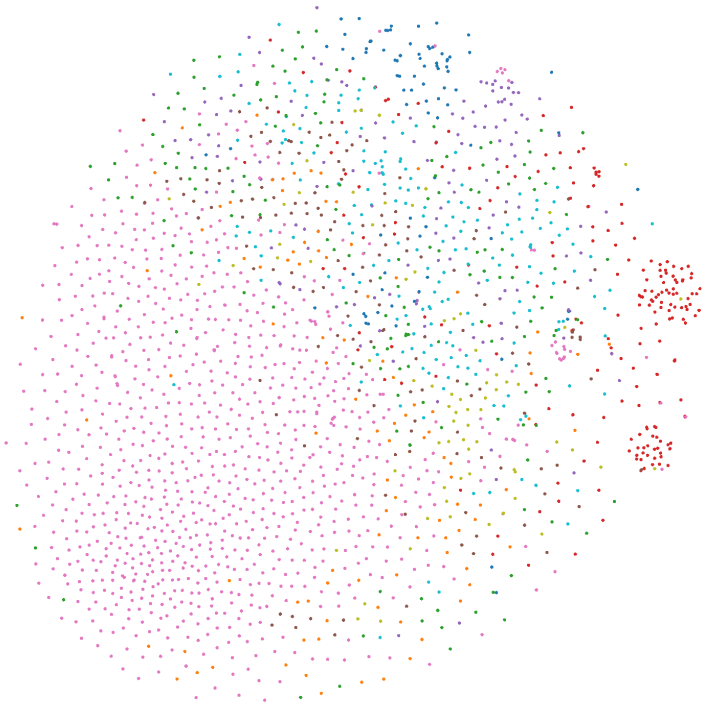

Figure 18: TSNE visualization with hyperparameters:
perplexity: 50, learning rate: 1, supervision: 20

Various solutions exist to this problem, yet fundamentally its constrained by the imbalance of the dataset. As other research has done, to correct this imbalance one could use third party translating services to create datasets in a specific language by translating all responses to the desired language. This strategy is however reliant on the quality of those third party translation services. It may also introduce artifacts that can't be reproduced and hence fell out of the scope of this research. In an ideal case the BERT model used as a starting point for simcse could have been trained on multiple languages. This is a more recent innovation in the field and hence not

much research has been done in performing a technique similar to simcse on these models.

STS-b dataset

The STS dataset was used for training a word2vec model and during training by the simcse model. It is a well researched dataset that has stood the test of time. For further research, [Faruqui et al., 2016] highlights some issues with word similarity tasks that factor into this paper.

6.3 Discussion of Use Cases

The use cases presented gave insight into how the tool may be used by the end users. Yet the shortcomings of the visualization should be understood in these contexts so that they can be used appropriately.

Dentist use case

Overall the process resulted in success for the dentist, however there are some shortcomings that should be highlighted.

Long sentence responses

For all the visualizations (Figure 12, Figure 13, Figure 14) there are long sentence responses present. These longer sentences could affect the embedding in two clear ways.

First there is the fact that there may be parts of the embedding that signal longer sentences. Then you would be merely finding sentences that are longer to be similar, not actually properly considering the semantic meaning. Then there is the question of how the model handles multiple concepts within one sentence. An average of three concepts would not adequately represent the concepts. Further research is needed to explore how the models encode these longer sentences, in this context this implies caution before generalizing from such longer sentence groups.

Variations in spelling

In the visualizations ((Figure 12, Figure 13) there are cases of neighbors being almost identical sentences aside from spelling. The fact that these show up as neighbors is

an encouraging sign. What should be approached carefully is increasing the neighbor count until you see different sentences. Doing so may be beneficial when a lot of similar spelled sentences exist in a dataset, however it is very difficult to generalize from the distances of neighbors as shown in Figure 13 on the right middle side. Even with these concerns, these variations in spelling do not present a hurdle for exploratory purposes, yet should be noted for future work.

Patient use case

The case of the patient is different from the dentist as there is no inherent searching for a pattern. The patient may use the tool to cope with and better understand their predicament. Even for these goals the visualizations have some shortcomings that should be addressed.

Lack of other responses

Perhaps a patient was one of the few who responded to a certain question. In the presented visualizations question 29 has the lowest response count, with only 62 responses. Searching these low response count questions will likely result in not so similar sentences being shown as close neighbors. This could result in confusion or stress if these neighbors contain responses from patients with more serious conditions. As long as the patients are informed before then this problem can be alleviated somewhat but the core problem remains. The potential solutions would be to remove the lower response count questions or feed more data into the model.

Misinterpreting distance

In the visualizations (Figure 15, Figure 16, Figure 17) the higher dimensional data is reduced to two dimensions. The risk with this reduction is that patients fail to understand that not all the variance is captured by the visualization. Such as with Figure 16, where *"klar denken"* is closer than *"Den Mund ganz öffenen."*. The patients should rely on the distances shown on the far right of the visualization, as seen in the dental use case Figure 13. Yet the chance for misinterpretation still exist due to the strong visual signal present in the graph itself.

7 Conclusion

In this section this paper explores future avenues for research and concludes the paper.

7.1 Future work

Proposed in this section are various research directions that the author sees as natural extensions to the current work.

Data

As highlighted in the chapter 6, the data was from a highly pre-selected group of individuals hence it is a candidate for future research. Expanding the dataset or testing the methods with another dataset will be a worthwhile endeavour. One potential direction is for the methods presented in this paper to be used on larger datasets. By increasing the dataset one can determine better if the results found in this paper are generalizable. Also, there is the potential that the methods may suit larger datasets even better, however researching this aspect fell outside the scope of the current thesis.

Embeddings

Crucial to the success of this work was the embeddings extracted from the text. Naturally this part of the work could be expanded to include different approaches to the embedding techniques. One proposal is to use all the layers of the BERT transformer to compute the embeddings. Another proposal may be to use a larger multi-language BERT model or a specific BERT model trained on the german language such as [Scheible et al., 2020]. A specific model may perform better since a lot of the multi-language models use third party translation services to create their translations.

Graph Neural Network

Graph neural networks (GNN) are an active research field [Zhou et al., 2020]. A future direction for this research is to explore applying a graph neural network on the sentence embeddings. Create a graph by adding edges that represent the patient between the sentences, creating a chain of sentences. Then potentially infer better clusters by using a GNN to interpret the high dimensional space filled with chains of sentences. The scalability of GNNs are not a direct benefit due to the size of the current dataset, yet this technique may be a consideration if the dataset size increases significantly.

Generation from clusters

Once the clusters have been obtained, a next step could be to generate likely sentences from that cluster. [Jiang et al., 2017] proposes a method to generate these sentences. The generation of sentences may help all parties to better understand the nature of the proposed clusters. The interpretability would be of concern, but in an idealistic case these sentences could then be used by further models as training data to create even more detailed models. However text data augmentation of unsupervised data for supervised models is complex and remains a question without a clear answer.

In a more realistic scenario, patients could use these generated sentences to cope with their diseases. Recent advancements AI chatbots could spawn a field of therapeutic conversations for patients. Feeding these chatbots generated sentences may offer an alternative to feeding actual patient responses. This would bridge the uncomfortable gap between an AI being fed your medical data, where the solution would be generalized data from a sample. It is left open for future research whether these generated sentences offer viable solutions.

7.2 Conclusion

This paper sought to explore the best approach of clustering based on an unknown semantic meaning. In researching this approach the word2vec model was used to find similar concepts in the data based on words in the sentence. Then the entire sentence was considered by using the SimCSE model. This model was trained in an unsupervised manner on the dental dataset.

The SimCSE model was then used to create multiple visualizations of the embeddings. These visualizations can be used by dentists, patients, and researchers to further explore the dataset. The pipeline presented could also be used on similar datasets for exploratory purposes.

The visualizations also include cluster assignments. These cluster assignments were found by running KMeans and GMM on the dataset, then using custom metrics the best clusters were found. The interpretability of these cluster assignments was analyzed along with evaluating the methods that generated them. These tools and the insights provided by them can be a building block for future research.

References

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognit.*, 46: 243–256, 2013.

S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR*, 2017.

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3:1–27, 1974.

D. M. Cer, M. T. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *SemEval@ACL*, 2017.

T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

D. A. Ettlin, I. Sommer, B. V. E. Brönnimann, S. Maffioletti, J. Scheidt, M.-Y. Hou, N. Lukic, and B. Steiger. Design, construction, and technical implementation of a web-based interdisciplinary symptom evaluation (wise) - a heuristic proposal for orofacial pain and temporomandibular disorders. *The Journal of Headache and Pain*, 17, 2016.

M. Faruqui, Y. Tsvetkov, P. Rastogi, and C. Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, 2016.

T. Gao, X. Yao, and D. Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 2017.

B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. On the sentence embeddings from pre-trained language models. In *EMNLP*, 2020.

Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. *2010 IEEE International Conference on Data Mining*, pages 911–916, 2010.

M. Marcinczuk, M. Gniewkowski, T. Walkowiak, and M. Bedkowski. Text document clustering: Wordnet vs. tf-idf vs. word embeddings. In *GWC*, 2021.

T. Mickus, K. van Deemter, M. Constant, and D. Paperno. Semeval-2022 task 1: Codwoe – comparing dictionaries and word embeddings. *ArXiv*, abs/2205.13858, 2022.

T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013b.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20: 53–65, 1987.

R. Scheible, F. Thomczyk, P. Tippmann, V. Jaravine, and M. Boeker. Gottbert: a pure german language model. *ArXiv*, abs/2012.02110, 2020.

G. Schneider, D. Ettlin, M. Wolf, and S. Wildermuth. Text-mining of patients' self-reports from an orofacial pain unit. *Forthcoming*.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61, 1999.

T. van Craenendonck and H. Blockeel. Using internal validity measures to compare clustering algorithms. In *ICML 2015*, 2015.

L. van der Maaten and G. E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

Y. Wang, N. Afzal, S. Fu, L. Wang, F. Shen, M. Rastegar-Mojarad, and H. Liu. Medsts: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, 54:57–72, 2020.

J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81, 2020.