MASTER'S THESIS 2020

Representing and Grouping Technical Issues for Business Insights

Ola Westerlund

ISSN 1650-2884
LU-CS-EX 2020-32

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2020-32

Representing and Grouping Technical Issues for Business Insights

Ola Westerlund

Representing and Grouping Technical Issues for Business Insights

Ola Westerlund
ola@wlund.io

July 16, 2020

Master's thesis work carried out at Tetra Pak AB.

Supervisors: Pierre Nugues, pierre.nugues@cs.lth.se
Astrid Nielsen, astrid.nielsen@tetrapak.com

Examiner: Jacek Malec, jacek.malec@cs.lth.se

Abstract

In any product, errors are inevitable. In a large corporation with many products, prioritizing the correct errors is crucial but often non-trivial. Given an issue database consisting of hundreds of thousands of data points, all containing a mixture of data, including free text, this thesis presents an automatic solution for representing and grouping these issues to aid business analysts in their prioritization decisions.

Previous studies have shown the value of applying *natural language processing* and *machine learning* techniques in finding meaningful relationships between sentences and documents. In this thesis, these earlier findings are applied to the domain of machine errors. Embedding techniques and clustering algorithms are applied to error data. The results show that with sufficient data and state of the art sentence embeddings, meaningful clusters can be constructed, keywords can be extracted, and new issues can be successfully linked to existing clusters.

Keywords: clustering, NLP, embeddings, BERT, Doc2Vec

Acknowledgments

Many thanks to Pierre Nugues for his patience and valuable guidance and feedback throughout the process of writing this thesis, and to Fabio Sisi and Alessio Ronchini at Tetra Pak, for taking time to validate clusters.
Many thanks also go out to Hanna Johansson for proof reading, and to Angelica Westerlund without whom this thesis would never have been finished.

Contents

Chapter 1
Introduction

This chapter provides the background, context and aim of this thesis.

With the increasing amount of data available throughout industry and society, companies everywhere are in the middle of a data revolution. Machine learning is being applied to harness the power of big data to achieve data-driven decision making, resulting in valuable business insights and monetary gains (Provost and Fawcett, 2013). At Tetra Pak AB, this work is carried out at the Data Science Department. One of the fields Tetra Pak investigates closer is machine issues. Be it broken parts or unexpected behavior, every issue is logged with a plethora of information, including free text descriptions.

1.1 Tetra Pak AB

AB Tetra Pak was founded in 1951 by Dr. Ruben Rausing. The company was built around the idea of packaging fluids, mainly dairy products, in carton-based packages. Rausing's philosophy is summarized by this motto: "A package should save more than it costs" (Tetra Pak AB, 2019a,b). Tetra Pak has since grown to become an international giant, supplying customers worldwide with solutions for processing and packaging products such as ice cream, cheese, fruit, vegetables, pet food, and dairy.

1.1.1 Data Science Department

The Data Science Department at Tetra Pak was founded in 2017 and consists of ca. 20 data scientists working in smaller teams on different projects. The department acts as company-wide service provider, working on data science related projects in all branches of the company. The workflow in general requires collaboration with multiple stakeholders in a customer-style relationship, where the main beneficiary of a team's findings is not the team itself.

1.2 Problem Definition

In similarity to many modern corporations, Tetra Pak puts great effort into data collection throughout their business. This is a continuous process which has been ongoing at Tetra Pak for many years. One of the main motives behind this data collection is the allure of achieving data-driven processes and decision pathways. One area which has been highlighted as a potential candidate for leveraging the power of data is *technical issues* (TI) and service of Tetra Pak's machines in the field. For these purposes, Tetra Pak has developed an internal tool, *Quality and Technical Issue management* (QuTI-P), and subsequently deployed it to the Tetra Pak *Field Service Engineers* (FSE). The FSEs use QuTI-P to register information regarding every TI which arises in any Tetra Pak machine deployed to a customer.

In order to make correct business decisions regarding which TIs should be prioritized and solved, the issues have been grouped, consolidated, into *consolidated issues* (CI). When a new TI is created, a business analyst links the issue to a an existing CI. If an appropriate CI cannot be found, the analyst creates a new CI and then links the TI to the newly created CI. The hierarchy of data this creates is illustrated in the lower levels of Figure 1.1.

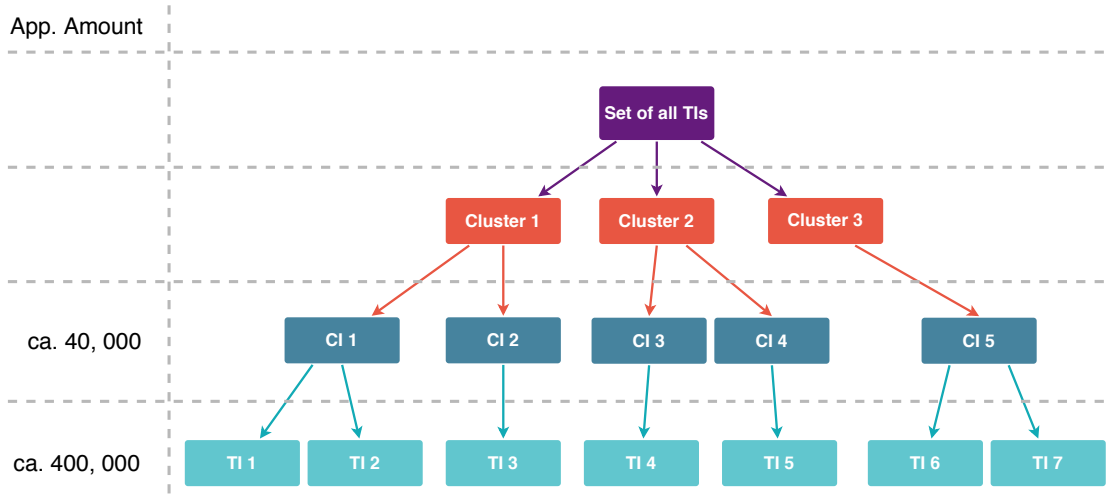


Figure 1.1: The hierarchy of data including the clustering level which is the focus of this thesis. The approximate number of instances is also shown for the CIs and TIs. This figure is simplified and only serves to illustrate the relationships between different data structures.

In the process of finding appropriate CIs for TIs, a plethora of CIs have been created over the lifetime of the QuTI-P app. Today the number of CIs exceeds 40,000, containing a total of almost 400,000 linked TIs. Each TI holds information about affected machine, involved parts, and status of the issue among other things. Each CI contains information entered by an analyst detailing and describing the group of issues, including affected systems, involved parts, as well as some data fields directly transferred from linked TIs.

The manual process of linking TIs to CIs is cumbersome and time consuming. The procedure is also considered error-prone by the business analysts and believed to be a prime target for an automated solution. Creating new CIs requires an analysts expertise, while the linking to existing CIs could be automate by utilizing e.g. clustering techniques.

It is the firm belief of the business analysts and the data science team that there is a large overlap between existing CIs, as well as inherent relations between many CIs. This means that in order to efficiently merge CIs and present better suggestions for prioritization, there is a need for an automated clustering of CIs. Not only will this lead to suggestions for merging CIs, but it will also give business analysts the power to identify groups of CIs which constitute larger categories of problems. This will free up business analysts to focus on relevant tasks and provide the means to prioritize resources to categories of issues, leaving singleton categories behind. The process from the occurrence/re-occurrence of an error to business insights is visualized in Figure 1.2. In this thesis emphasis is placed on the last three panels of this figure.

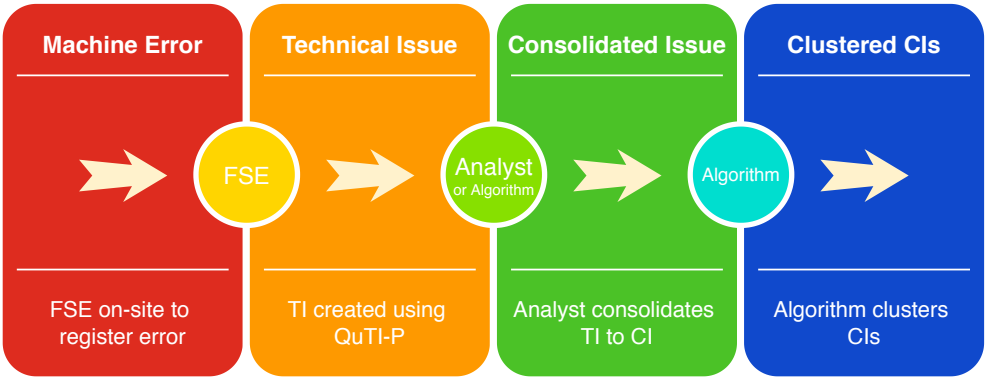


Figure 1.2: The process from machine error to business insights. In this thesis the focus is on the steps from TI to CI, and from CI to cluster.

A data science team at Tetra Pak has performed a pilot study to confirm the feasibility of the clustering of CIs. The first model uses a subset of the data available from each TI and CI in order to create vector space representations of each CI. These vectors are then clustered using k-means clustering (MacQueen, 1967). The results of this study were deemed promising by the business analysts, producing some meaningful clusters. The vast majority of clusters were, however, of low relevance. The main idea behind this thesis is that additional data, better models, and other clustering techniques, will increase the fraction of relevant clusters, and lead to valuable business insights.

1.3 Related Work

The field of *natural language processing* (NLP) has a long history dating back to the 1950's and publications by Alan Turing. Like so many other subfields of artificial intelligence, NLP has taken a leap forward with developments in the last decade concerning deep neural networks.

Building upon these advancements, Mikolov et al. (2013) introduced the Word2Vec framework. Given a word, Word2Vec predicts the words surrounding it. Given a short sequence of words, it can predict the words at the center of this sequence. The resulting hidden layers from the neural network used in the Word2Vec framework can be used as effective vector space representations, relating semantically similar words.

Building upon their previous work, Le and Mikolov (2014), introduced Doc2Vec which affords the same powerful representations as Word2Vec, but for sentences, paragraphs, and entire documents.

In their article *Automatic synonym extraction using Word2Vec and spectral clustering*, Zhang et al. (2017), use Word2Vec word embeddings to represent words in order to find synonyms. This is done by applying clustering to the word embeddings and thereby identifying groups of words with similar meaning.

Another publication applying clustering to embeddings is *Using Word2Vec to process big text data* (Ma and Zhang, 2015). Here, large data sets are reduced by representing large corpora using Word2Vec embeddings, and reducing the dimensionality of the data by grouping words of high similarity.

Devlin et al. (2018) introduced BERT, which compares two sentences and evaluates the similarity using bidirectional neural networks. this approach achieved new state-of-the-art results on multiple NLP tasks, including "semantic textual similarity".

Reimers and Gurevych (2019) extended upon the BERT framework in order to create *Sentence-BERT* (SBERT). SBERT is able to create sentence embeddings, and does so at a much lower computational cost than the original BERT implementation.

Guo and Berkhahn (2016) provided some insight into how to handle high-cardinality categorical data. In their article *Entity Embeddings of Categorical Variables*, they show that it is both feasible and meaningful to represent this data using NLP techniques for vector space representations.

In this thesis, the data at hand and appropriate tools and frameworks are utilized to seek meaningful clusters of business value. Building upon the ideas presented by Zhang et al. (2017) and Ma and Zhang (2015), the embedding frameworks and algorithms presented above are adapted to the domain specific purposes of this thesis. For a theoretical overview of the aforementioned models and algorithms, see section 2.1.

1.4 Research Questions

This thesis explores the possibility of clustering consolidated technical issues into groups. The aim is to find a method which is applicable to the large data set and high cardinality categorical values, as well as extensive free text fields, resulting in meaningful clusters from a business perspective. A secondary goal is to extend this method to finding existing consolidated issues to which new technical issues can be linked.

Based on the available data and previous work in the field, this thesis aims to:

- *Find a possible and meaningful way to represent the available data.*
- *Investigate and suggest a useful model for clustering the represented data.*
- *Automatically label clusters based on their contents*
- *Determine if it is possible to automatically suggest useful existing CIs when encountering a new TI.*

1.5 Scope and Delimitations of Thesis

In this thesis, focus is placed on linking TIs to CIs, as well as clustering CIs in order to allow business analysts to gain increased insights into the vast amount of available data.

The data is limited to the domain specific data found within the QuTI-P database at Tetra Pak. The resulting models are specific to the English language. The vector space representations which are considered are the Doc2Vec and SBERT frameworks. The clustering algorithm used is agglomerative hierarchical clustering. Evaluation of clusters is performed using domain expert validation.

1.6 Stages of this Thesis

The bulk of time of this thesis was spent iterating through different clusterings and receiving feedback from the business analysts for the next iteration. In a larger perspective, the thesis can be viewed as consisting of the following:

1. Understand available data and prepare it for use in the model training pipeline. The data is described further in section 3.2.3.
2. Create vector space representations of the CIs. For this the Doc2Vec framework, and pretrained SBERT embeddings are used. This is further described in section 2.1.
3. Cluster the CIs, using hierarchical clustering. This is detailed in section 2.2.
4. Create and implement measures for objectively evaluating cluster quality. See section 2.3 for more details on this process.
5. Receive feedback on clustering via business analysts (domain experts). This validation process is described in section 3.2.6.
6. Determine the main attributes upon which major clusters were created. This is further described in section 2.4.
7. Utilize vector space representations to link new TIs to existing CIs. Read more in section 3.2.5.
8. Evaluation of results, see section 3.2.6.

Chapter 2
Algorithms and Models

This chapter gives a brief theoretical overview of the methods employed in this thesis. For in-depth explanations the reader is referred to the cited material.

2.1 Vector Space Representation of Data

The first objective in this thesis is to find a way to represent CIs. For this two models are used: Doc2Vec and Sentence-BERT. Some basics of data types and representation are presented before introducing these frameworks in more detail.

2.1.1 Data Types

Data can be either *unstructured* or *structured*. There are different types of structured data, and an important step when approaching a data science task is analyzing and understanding what types of data are in your data set (Bruce and Bruce, 2018). The main distinction in the group of structured data is between *numeric* and *categorical* data. Numeric data can be further divided in:

1. *Continuous* – e.g. speeds, weight, any float representation
2. *Discrete* – integers, counts
3. *Ordinal* – the members of the set have an inherent order

Categorical data is restricted in that it only takes on values in a specific set:

1. *Multiple-category*
2. *Binary* – a special case where the set only contains two values

Numeric data has an inherent ordering which is often desirable to conserve when representing the data. When representing data types it is important to not impose an artificial ordering on data which has no natural ordering. This can lead to false results when applying algorithms to the data.

2.1.2 Word Embeddings

Basics of Language Representation

Data in the form of free text belongs to the category of *unstructured data* (Bruce and Bruce, 2018; Bengfort, 2018). Text is not easily and immediately understandable for a machine, but this does not mean it cannot be made useful. Language is unstructured, but it is not random. There are patterns to be found and exploited in order to represent texts in a manner which puts them in relation to other texts. This enables a machine to understand a given text's context (Bengfort, 2018).

Instead of representing texts or documents as strings, a *vector space representation* is used (Aggarwal, 2015). However, the naive implementations of such representations create an exceedingly sparse data structure. This is due to the relatively small number of words occurring in a single text compared to the number of words in the vocabulary of the data set. The simplest way to alleviate the issue of sparsity is the *bag-of-words* method, using a representation limited to the vocabulary of the document (Harris, 1954; Aggarwal, 2015).

There are two major drawbacks of methods like bag-of-words. The first is the loss of ordering of words. The second is the loss of the semantics of words, resulting in completely different sentences, albeit containing the same words. These completely different sentences will however, still have the same vector space representation (Le and Mikolov, 2014).

Embeddings

The concept of embeddings has a long history reaching back to the early 1960's and publications such as Salton (1962), which delved into the potential of vector space representations of language. The general idea is to reduce the dimensionality of data and represent all data points as fixed length vectors. Some properties of the data are encoded into the vectors. These vector representations then afford the possibility of measuring distances between data points in the resulting n-dimensional space.

Word2Vec

A major improvement in the the field of data mining regarding NLP was the advent of word and document embeddings as described by Le and Mikolov (2014). How these vector space representations of data points are calculated has varied historically. The principal behind the Word2Vec framework proposed in the aforementioned article, is learning representations of words using unsupervised learning. This is achieved with a shallow neural network using stochastic gradient descent and backpropagation in order to learn the weights of each node. The resulting weights of the hidden layer are then used as embeddings. The network design is described in more detail below. The training of the model is focused on predicting words in a paragraph, using the previous words to predict the following words.

Mikolov defines the *Skip-gram model* used in Word2Vec as:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t), \tag{2.1}$$

where w_t is the center word, and c is the size of the training context. The model finds the representation for a word with the highest log probability as per equation 2.1 (Mikolov et al.,

2013). Mikolov defines the softmax function $p(w_{t+j}|w_t)$ as:

$$p(w_O|w_I) = \frac{\exp(v'^{\top}_{w_O} v_{w_I})}{\sum_{w-1}^{W} \exp(v'^{\top}_{w} v_{w_I})}, \tag{2.2}$$

where v_w and v'_w are the vector representations of w which are input and output from the model. W is the number of words in the vocabulary. The softmax formulation in Eq. 2.2 is computationally inefficient and is therefore not used in practice, but instead approximated using the hierarchical softmax (Mikolov et al., 2013). The representations of words which are learned using this softmax function capture the semantics of the data.

Words with semantically similar meaning should be represented closer to each other in the vector space. This can be observed in Figure 2.1, where a similar transformation applied to any country will garner the capital city of said country. Countries which are "related" are also located spatially closer to each other in the vector space. This property of the skip-gram vectors, or Word2Vec framework, is useful for CIs since there are potentially similar relationships in this field. For this approach to translate to the domain of machine errors and give relevant clusters of CIs, representations of related CIs should be closer to each other in the n-dimensional space than those of unrelated issues.

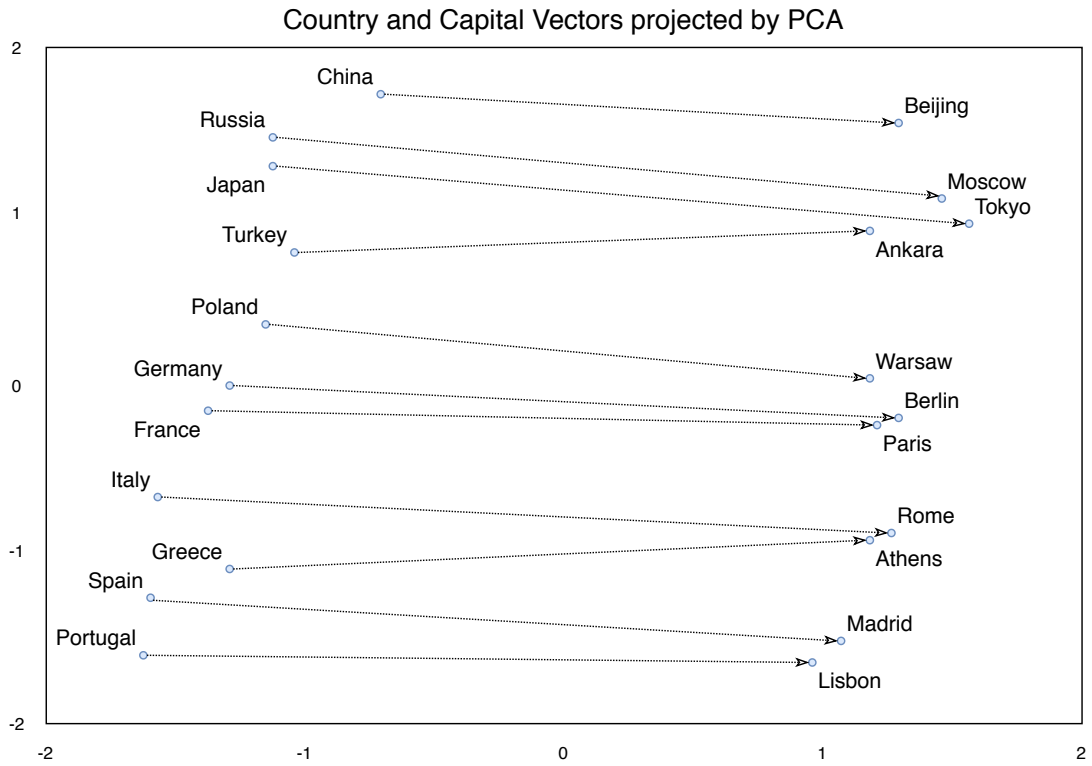


Figure 2.1: PCA of skip-gram vectors of countries and capitols. The figure represents the property of conserving semantics in the data. Performing similar transforms on any country will result in its capital. Adapted from Mikolov et al. (2013).

Word2Vec is based on a neural network with three layers: input, output, and a hidden layer, as is shown in Figure 2.2. The input and the output layers both have their dimensions determined by the vocabulary size. In the example shown in Figure 2.2, the vocabulary size

is 10,000. The size of the hidden layer determines the dimension of the vector space representations that will be learned for each word in the vocabulary. In the example, this size has been set to 300. One hidden layer representation will be learned for each word in the vocabulary, as the data set is traversed and the weights are adjusted for the hidden layer. The network takes as input a one-hot vector representing the current word in a text. The output is the probability of each word of the vocabulary being the next word in the text.

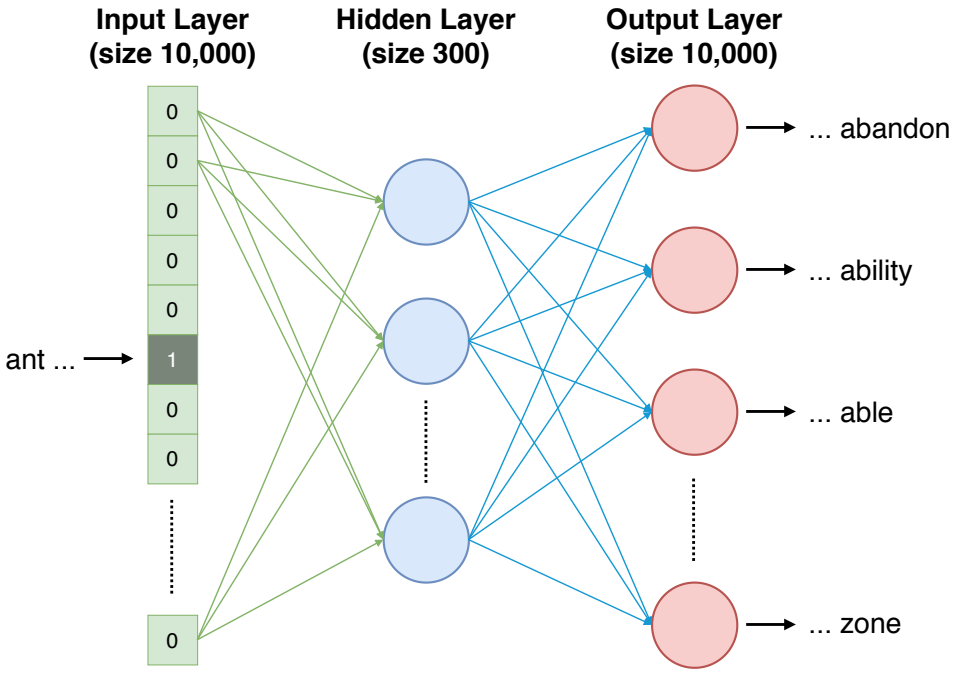


Figure 2.2: A three-layered neural network showcasing the network architecture used in Word2Vec. The input layer is a one-hot encoding representing each word in the chosen vocabulary. The output layer is the probability of each word of these words being the next word. The hidden layer becomes the word embeddings. The network is fully connected. The figure is simplified. Adapted from Nayak (2019).

The Word2Vec representation training must overcome one major hurdle: frequently used words which add little meaning to a sentence and are found in abundance throughout the data set. Word2Vec handles this by subsampling the frequent words. This, in practice, means that a word in the training set is discarded with a rising probability when it appears frequently in the data set (Mikolov et al., 2013).

2.1.3 Doc2Vec

Extending upon the same ideas used in Word2Vec, Le and Mikolov (2014) developed Doc2Vec. This framework first learns representations for the vocabulary of the data set. Then, it learns representations of each "paragraph". In the context of this thesis a paragraph corresponds to a CI. These representations result in vector space representations for entire documents. Ideally they place CIs with similar textual/semantic content in closer proximity to each other

than to those of different content. The word embeddings are shared across all the documents, while each document representation is unique.

Figure 2.3 shows the principal idea of how the Doc2Vec framework uses its vector representations of a series of words, and the paragraph vector, to predict the subsequent word. The process is identical to that of Word2Vec, except the inclusion of the paragraph vector, which is unique to Doc2Vec.

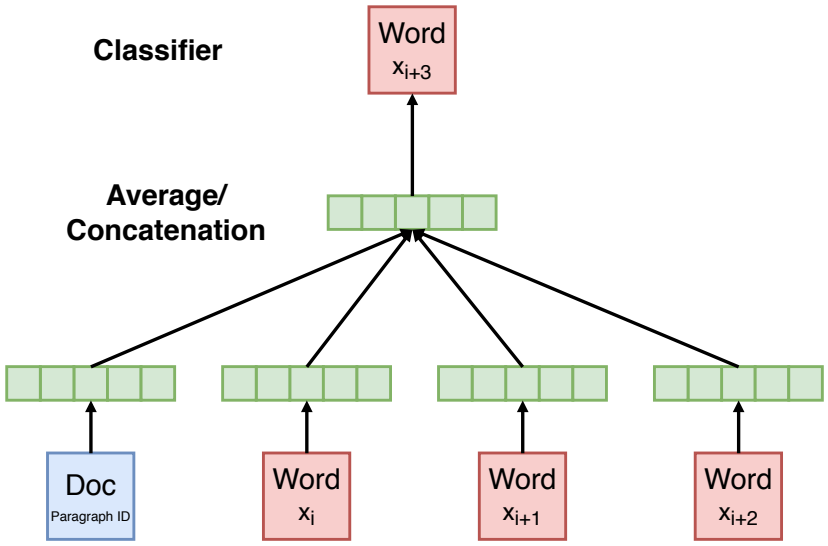


Figure 2.3: The process of predicting a word in Doc2Vec by averaging the preceding word representations and the paragraph vector. The paragraph vector is the left most vector in the figure. Adapted from Le and Mikolov (2014); Hui (2019).

In the context of this thesis, the Doc2Vec framework is used. Each CI contains multiple sentences of texts which together can be viewed as an individual document.

2.1.4 BERT

Bidirectional Encoder Representations from Transformers (BERT), as introduced by Devlin et al. (2018), is an approach to understanding natural language. BERT uses a neural network approach with the objective of predicting masked words in sentences by context.

Transformers

BERT is based on *transformers* as described by Vaswani et al. (2017). The principle behind transformers is using *encoders* and *decoders* combined with the concept of *attention*. The network is trained on sentence pairs and learns to translate between the two by encoding to, and decoding from, a high dimensional representational space. This makes transformers useful for e.g. translation and question answering tasks. When decoding a specific position of a series of words the encoder is fed the entire input text, while the decoder is fed the decoded text generated up to that point. To allow the decoder greater insight into the most relevant parts of the input text, attention is used. Attention is a method of giving the decoder access to information from the most relevant part of the input text for the current position of the decoding.

BERT Neural Network Architecture

The main strength of BERT compared to previous language models is its ability to use both left and right context simultaneously, in its predictions. This is achieved by a network architecture consisting of multiple layers of fully connected transformers - a deep bidirectional transformer Devlin et al. (2018), which can be seen in Figure 2.4. The input text, split into tokens, can be seen at the bottom of the figure, while the predicted output is seen at the top.

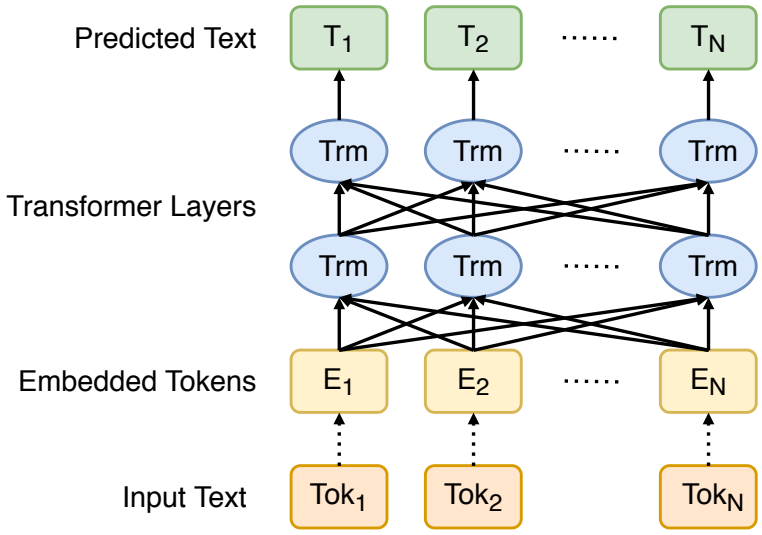


Figure 2.4: The schematics of a BERT neural network, consisting of multiple layers of transformers, fully connected, allowing for leveraging context in both directions. Adapted from Devlin et al. (2018)

Creating a BERT model is divided into two steps, pretraining and fine-tuning:

1. The pretraining is performed on unlabeled data on a multitude of tasks. This is done on sentence pairs with masked-out words.

2. Fine-tuning is then performed using labeled data from relevant domains. Fine-tuning is repeated from the pretrained parameters for each new domain.

The ability to differentiate homographs is a feature which sets BERT apart from other embedding methods such as Word2Vec (Devlin et al., 2018). The BERT representations of *close*, as in "close a door", and *close*, as in "that was close", would be different.

Sentence-BERT

Using BERT to derive sentence level embeddings is described by Reimers and Gurevych (2019), who introduced *sentence-BERT* (SBERT). Embeddings are calculated by modifying the network layout of BERT. Among other changes, a pooling layer is added on top of the BERT network. This network is then pretrained and fine-tuned as described for BERT above.

In this thesis, a BERT model implemented by Reimers and Gurevych (2019) is used. This model has been fine-tuned on specific data, optimizing for *semantic textual similarity*.

2.1.5 Principle Component Analysis

In order to plot and explore high-dimensional vector space representations, it is necessary to project the data onto a two-dimensional space. This can be achieved using principle component analysis (PCA). This representation affords the ability of visually inspecting and comparing vector space representations before clustering.

The theory behind PCA is to find the dimensions in which the data has the greatest variance, e.g. where the most significant differences arise. In the two-dimensional case this means projecting the vector space representations onto the two-dimensions with the greatest variance. For details, please see Goodfellow et al. (2016).

2.2 Clustering

At this point, a way to represent the technical issue data in an n-dimensional space has been found. The next challenge is to determine which data points have common properties or could be considered to be of the same class. This is a *clustering* problem. Clustering algorithms are unsupervised learning algorithms for identifying classes (Ester et al., 1996). determining if a clustering is correcdt is a non-trivial problem inherent to unsupervised learning: there is no gold standard to use for validation. There is no way of testing the results to objectively determine if the rendered clusters actually make sense and add any value. In this thesis this is solved by the use of domain experts who manually evaluate clusters.

2.2.1 K-means Clustering

The baseline pilot model uses k-means clustering, which is a very common clustering technique. Therefore, it is described briefly.

The clustering method k-means was first described by MacQueen (1967). It is a method of "partitioning an n-dimensional population into k sets". The algorithm is initialized by selecting k initial centroids at random, and then iteratively updating these cluster centers according to the steps:

1. Each example is assigned to the nearest of the k centroids using the Euclidean distance as distance measure;
2. Each centroids position is updated to be the mean of its constituent examples.

This process repeats until convergence and the resulting clusters are determined. An example of the converged result of a two-dimensional K-means clustering, $k = 3$ can be seen in Figure 2.5.

2.2.2 Hierarchical Clustering

Hierarchical clustering is a recursive approach to grouping examples. It can function in either a top-down, or a bottom-up fashion, i.e. it either begins from one large cluster, splitting it into smaller clusters to a desired level, or it does the reverse, beginning from singleton clusters and merging successively (Maimon and Rokach, 2010). The former strategy is called *divisive*, while the latter is termed *agglomerative*.

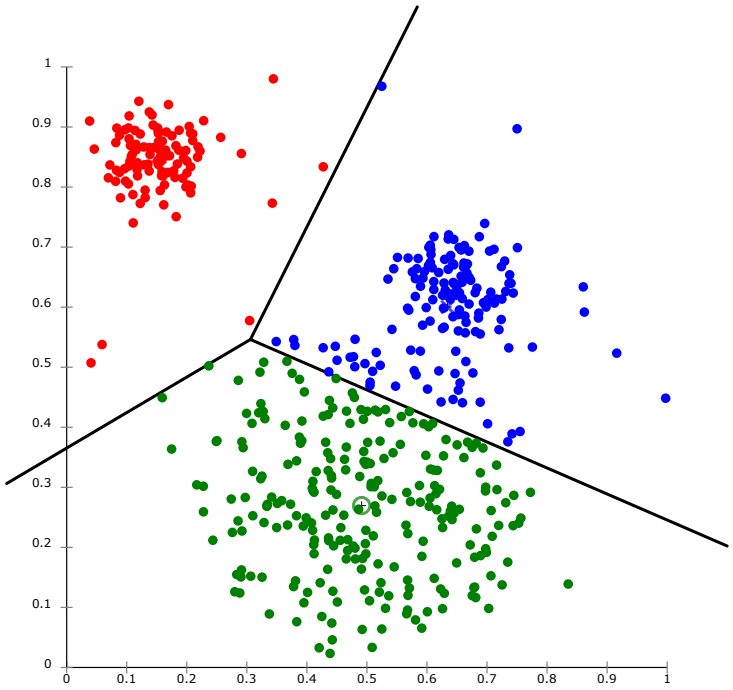


Figure 2.5: K-means clustering performed on two-dimensional data with $k = 3$ (Wikimedia Commons, 2011)

After performing the clustering, a *dendrogram* can be obtained. It shows the clusters at different levels of clustering. A horizontal cut through the dendrogram garners a complete clustering of the data set. Figure 2.6 shows a small example of a dendrogram when applying hierarchical clustering to cardinal directions. The horizontal cut results in the following set of clusters: $clusterset = \{\{W\}, \{SW, NW\}, \{S\}, \{N, SE, NE, E\}\}$

Hierarchical clustering is a general concept of clustering and it can use different metrics of distance, similarity measure, for determining which clusters to merge/split.

2.3 Evaluation of Clusters

Once a clustering has been accomplished it is necessary to determine the optimum number of clusters for this the silhouette score is a common choice. It is unlikely that all clusters generated will be useful from a business perspective, and therefor it is of interest to correlate the subjective cluster usefulness with some objective cluster measure. For this task we will introduce the intra-cluster distance below.

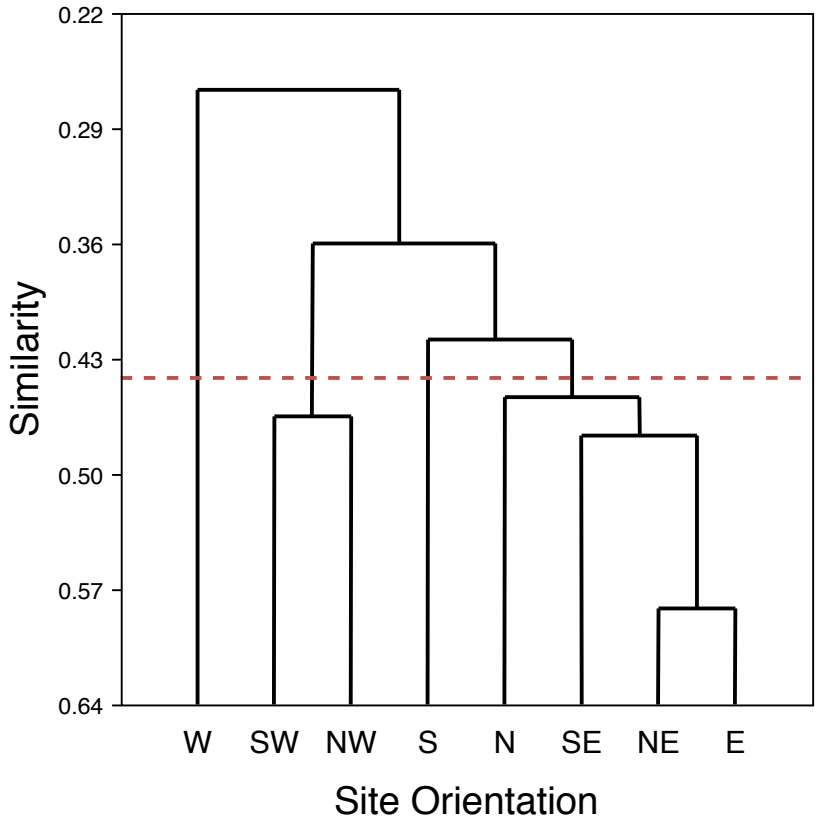


Figure 2.6: Example of hierarchical clustering of cardinal directions showing the typical dendrogram structure. The red dashed line garners four clusters. Adapted from Menasria et al. (2015).

2.3.1 Silhouette Score

The silhouette score was proposed as a metric for clustering by Rousseeuw (1987). It is a measure which takes into account the tightness and the separation of all the clusters of a data set and produces a single value as output. The general principal of silhouette score is described below.

Given that i is any object in the data set, A is the cluster which i belongs to, A is not a singleton, and C is any cluster $C \neq A$. The following expressions are given (Kaufman and Rousseeuw, 1990):

$$a(i) = \text{average dissimilarity of } i \text{ to all other objects of } A, \tag{2.3}$$

$$d(i, C) = \text{average dissimilarity of } i \text{ to all objects of } C, \tag{2.4}$$

$$b(i) = \min_{C \neq A} d(i, C), \tag{2.5}$$

where the dissimilarity of an object i to a set of objects B is calculated as the average distance between i and the constituent objects of B. Using these equations, the silhouette score $s(i)$

for object i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{2.6}$$

Obtaining the silhouette score for the entire data set is then a simple matter of calculating the average of $s(i)$ for all data points i in the data set. A lower silhouette score indicates better, more well separated clusters.

2.3.2 Intra-Cluster Distance

In order to objectively rank individual clusters we introduce the intra-cluster distance (ICD). The ICD can then be used to sample clusters for validation by domain experts. The experts can then define threshold values for the ICD which indicate at what point clusters lose meaning. The ICD for a cluster C is defined as:

$$\mathrm{ICD}(C) = \left(\sum_{i \in C}(i - \mathrm{cent}(C))\right)/|C|, \tag{2.7}$$

where i is a data point, C is a cluster and $\mathrm{cent}(C)$ is the centroid of the cluster, i.e. the average vector space representation of the data points in cluster C.

2.4 Keyword Extraction

With clusters in place, the next objective to address is that of cluster labelling. In order to automatically generate labels for clusters the term-frequency - inverse document frequency is employed.

2.4.1 Term Frequency - Inverse Document Frequency

In order to extract the most important and representative words of a text, the well-known term frequency-inverse document frequency (tf-idf) representation is employed. In the context of this thesis tf-idf is used for keyword extraction individual clusters. The concepts behind tf-idf have evolved over time with major contributions being made by Sparck Jones (1972). The concept is divided into two parts: term frequency, and inverse document frequency. The former is defined as follows:

$$\mathrm{tf}(t, d) = \frac{\text{count of } t \text{ in } d}{\text{number of } \textit{words} \text{ in } d}, \tag{2.8}$$

where t is the term being examined, and d is the document. The latter, the idf, is defined as follows:

$$\mathrm{idf}(t, D) = \log \frac{N}{|\max\{d \in D : t \in d\}|}, \tag{2.9}$$

where t is the term being examined, d is a document, D is the corpus, and N is the total number of words in the corpus. The tf-idf is then defined as follows:

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D). \tag{2.10}$$

The resulting formula will produce high tf-idf values for terms which are frequent in a document but uncommon in the corpus, avoiding highlighting terms which are abundant in all of the corpus and instead highlighting the terms that set a document apart. In this thesis each cluster is equivalent to a document, and the set of clusters is the corpus.

2.5 Automatic linking of TIs

The process of automatically linking TIs to CIs is based on the same vector space representations as the clustering. To determine how close two points are to each other in an n-dimensional space the cosine-similarity is a useful measure.

Cosine similarity, as explained by Tan et al. (2005), is a measure of the angle between two vectors in an n-dimensional space. Given the properties of the cosine function, two identical vectors will have a cosine similarity of 1, while two orthogonal vectors will have a cosine similarity of 0. The cosine-similarity is defined as follows:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \cdot \|\mathbf{B}\|}. \tag{2.11}$$

Chapter 3

Approach

This chapter serves to emphasize the main sub-fields of machine learning and natural language processing relevant to this thesis, as well as a description of the methodology employed.

3.1 Machine Learning and Natural Language Processing Themes of the Thesis

Prior to this thesis, the data science team at Tetra Pak performed a pilot study, highlighting the potential of clustering CIs, as well as the possible benefits of extending the vector space representation training model with more data. Using this pilot study as a basis, new vector space representations of documents were trained. Clustering was then performed and fine-tuned using objective measures in conjunction with validation by domain experts. From the resulting models and clusters, keywords could then be extracted and methods of linking TIs to CIs could be explored and evaluated.

3.1.1 Representing Data

Before attempting to create any structure in the available data, the data scientist is faced with the challenge of representing data in a meaningful way. Large, sparse data sets are not trivial to approach and represent using common methods such as *one-hot encoding* (OHE). OHE quickly becomes intractable for high cardinality categories, as an extra column is added for every value of the category. Based on the findings of Guo and Berkhahn (2016) and Devlin et al. (2018), this thesis utilizes the Doc2Vec and SBERT frameworks respectively, to create *embeddings* (see section 2.1.2) to represent CIs in a high-dimensional vector space.

3.1.2 Clustering

It is the aim of this thesis to explore and bring structure to the existing CI and TI data available at Tetra Pak. The structuring in this case is focused on attempting to cluster the CI data into groups which have meaning and value from a business perspective. Hierarchical clustering is the main clustering algorithm used.

3.1.3 Evaluate Clusters

It is also an ambition to find a clustering and corresponding quality measure which shows correlation between improved quality of clusters and more meaningful clusters from a business perspective. This process requires domain experts to determine validity of clusters. When this is achieved, it is possible to use objective measures in conjunction with the domain experts' hand annotation to gauge clustering success.

3.1.4 Labeling Clusters with Keywords

As a means of quickly understanding the contents and themes of clusters this thesis attempts to automatically extract keywords from created clusters. This is accomplished using the *term frequency - inverse document frequency* measure, applied to existing clusters.

3.1.5 Automatic Linking of TIs

The business analysts would like to automatically link new TIs to an appropriate existing CI. Building upon the vector space representations explored for clustering, automatic linking suggestions are achieved by representing TIs in the same vector space as the CIs. Recommendations of CIs to which to link the TIs are then made choosing from the closest CIs in the vector space, using e.g. the cosine similarity.

3.2 Methodology

Like almost all modern computer science projects, this project is approached in an iterative fashion. The overarching method is described first, followed by sections describing the individual steps.

3.2.1 Large scale method

The overarching method of this thesis is CRoss-Industry Standard Process for Data Mining (CRISP-DM) (Chapman et al., 2000). The model was originally introduced as an efficient way of conduction data mining projects, and it is well suited for an exploratory data science project such as this thesis. Figure 3.1 shows an overview of the main steps of the iterative loop of CRISP-DM. The cycle is performed by the team as a whole, with different individuals contributing to different steps. The steps of the cycle, and how they are performed within the scope of this thesis is detailed in the following sections.

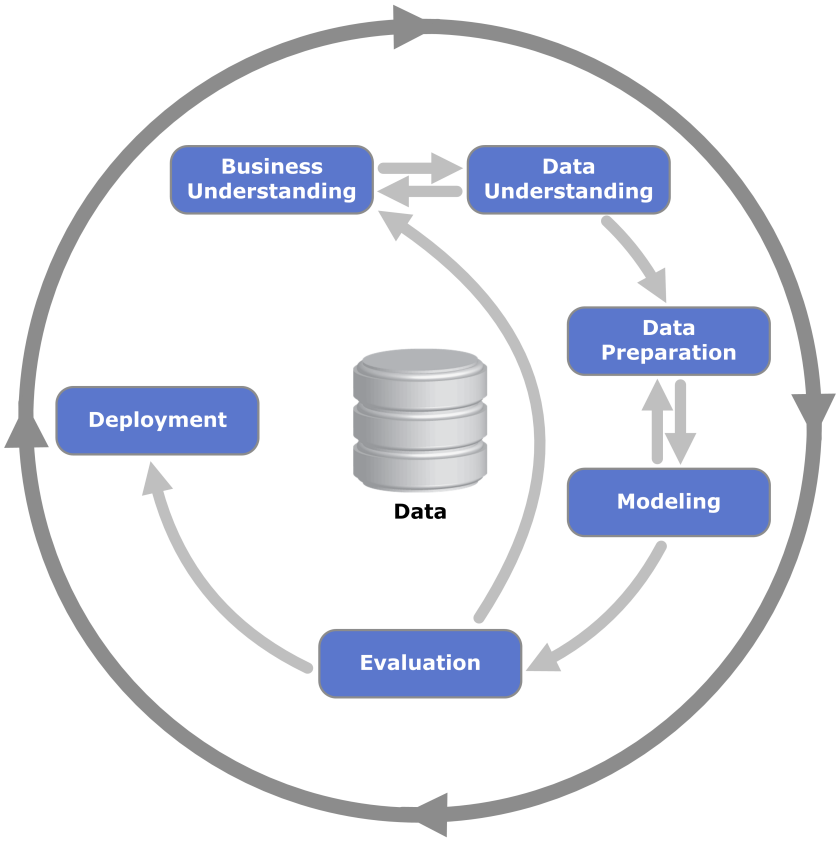


Figure 3.1: Flow chart describing the iterative process of CRISP-DM (Wikimedia Commons, 2012)

3.2.2 Business Understanding

Business understanding is possibly the most important step of the process of data mining (Shearer, 2000). The point of business understanding is to grasp the potential gains of the data mining, and to understand the objectives from a business perspective. During the first month of thesis work, a considerable amount of time was spent understanding the value of the project. Before attempting to create a first clustering, a workshop was held with domain experts/business analysts to discuss expectations and value of the project.

Data Mining Goals

In collaboration with the data analysts and the team at Tetra Pak, a set of business goals for the project was conceived. These focused on automating the process of finding the most important groups of issues. From these broader goals, data mining goals which needed to be achieved in order to reach the business goals were defined.

- *Find an efficient vector space representation of the CIs*
- *Find structure and meaningful clusters in the set of > 40 000 CIs*
- *Find the defining characteristics of each cluster*
- *Find a model for linking new TIs to CIs*

Fulfilment of these goals is not exclusively objective. In fact, the first two bullet points are highly subjective, depending on the business insights of the domain experts at Tetra Pak to

evaluate if the clusters are meaningful. Meaningful clusters, by extension, validate the vector space representation. The third bullet is simply a matter of extracting the most significant features of a given cluster. By choosing and applying an appropriate algorithm for this task, the goal is met. The final bullet, analogous to the former, is a matter of finding an algorithm, applying, and tuning it.

3.2.3 Data Understanding

Data Understanding, according to Chapman et al. (2000), consists of collection, description, exploration, and quality verification. During the initial phase of the project, great effort was put into understanding similar projects being run in parallel within the team. These projects used much of the same data and were therefore useful as a starting point to understanding what the data looked like, what quality it had, and what type of data cleaning and selection could be necessary.

Data Collection

With the aid of business analysts, the potentially relevant data fields available in QuTI-P were identified. The relevant data was readily available through QuTI-P's underlying databases. The data collection process consisted of creating the relevant database views with the help of the data engineers at Tetra Pak. To create the final working data set, these views were joined in order to render the complete data set.

An important note is that the QuTI-P application is used on a daily basis and new data points are added every day. Many of the deployed models at Tetra Pak are trained on a daily basis on the updated data set. For the purposes of this thesis, a data set was extracted on a set date and then remained static for the duration of the project.

Data Description

The raw data set consists of 371,793 technical issues, grouped in 43,898 different consolidated issues. Each data point has 39 fields shown in Table 3.1. Fields in Table 3.1 which start with *ti_* are derived from the technical issue that the entry refers to, in this case TI *1000278602*. Fields which start with *ci_*, conversely, are derived from the linked CI. As can be seen, some fields have the value *#*. This is simply an indicator that the value for this field was missing for this TI. Due to corporate privacy restrictions, the nature of the data fields cannot be further disclosed in this thesis.

Data Exploration

In order to get a better understanding of the data, some inquiries into the properties of the data set were made. Of the 39 fields, 30 can be interpreted as categorical, eight are purely free text fields, and the last field is the unique *ti_id* field. To determine if there were any candidate fields suitable for distinguishing CIs into clusters, the cardinalities of the available categorical fields were investigated. A high cardinality, approaching the size of the data set, implies a poor clustering feature since there are very few examples which have common values. On the other hand, the opposite case, a very low cardinality approaching one, implies very few clusters for that feature. As can be seen in Table 3.2 and Figure 3.2, there is a fair amount of spread in cardinality.

Table 3.1: The data fields for each TI used when constructing the document embeddings. The "Example Value" column shows field values taken from a specific TI. Data is not available for all fields.

Feature Name	Example Value
ti_id	1000278602
ti_failure_mode_id	2xx
ti_failure_mode_descr	mechanical assembly
ti_failure_type_id	211
ti_failure_type_descr	211 - loose (disconnected, untight)
ti_material_id	6485750400
ti_material_descr	filling machine tetra pak a3/flex
ti_package_type_volume_shape	tga 1000 lf
ti_machine_system	tp a3/f
ti_abc_cat	c1: customer operational costs increase
ci_material_vendor_name	#
ci_abc_category	c1: customer operational costs increase
ti_part_number_id	29409670000
ti_collateral_damage_part_no	29409670000.0
ti_b_group	648584-0400
ti_b_group_description	final folder
ti_c_group	2926900-0400
ti_c_group_descr	infeed
ci_b_group	648584-0400
ci_b_group_description	final folder
ti_title	a3/f,tga,infeed belt damage-2,mitsui
ci_title	a3/f tga ffu belt too long
ti_problem_description	i changed it approximately two months ago, but...
ci_problem_description	during maintenance phase at 1100 hours the in ...
ti_collateral_damage_short_text	timing belt
ti_possible_reason	short life
ti_action_and_result	part orderd
ci_possible_reason	#
ci_symptoms_and_impact	#
ti_status	documentation verified
ci_status	on hold
flag_generic_ci_descr	no
ti_flag_confirmed_unconfirmed_descr	confirmed
ci_module_object_part_description	final folder unit
ti_module_object_part_description	#
ti_area_id	ca_23
ti_portfolio	cve
flag_responsible_organisation_key	serv-gq&ts
ci_id	2000007385

Table 3.2: The cardinality of each categorical field prior to cleaning the data set, but after removing duplicate TIs

Feature Name	Value
ti_flag_confirmed_unconfirmed_descr	4
flag_generic_ci_descr	4
ti_status	8
ti_area_id	10
ti_abc_cat	15
ci_status	16
ci_abc_category	16
ti_portfolio	18
flag_responsible_organisation_key	30
ti_failure_mode_descr	101
ci_module_object_part_description	256
ti_failure_mode_id	286
ti_module_object_part_description	288
ci_b_group_description	289
ti_b_group_description	316
ti_package_type_volume_shape	324
ti_machine_system	386
ti_failure_type_id	442
ti_failure_type_descr	483
ci_material_vendor_name	627
ti_material_descr	904
ti_material_id	1162
ci_b_group	1183
ti_b_group	1409
ti_c_group_descr	2465
ti_c_group	9413
ti_part_number_id	27211
ti_collateral_damage_part_no	28297
ci_title	43439
ci_id	43887

Data Quality Verification

As stated above, the initial data set consisted of more than 370,000 data points. The data quality, however, was very poor. This became evident from the high number of missing values throughout the data set. It became apparent upon further inspection that QuTI-P is a highly organic application, and the number of fields available to the field service engineers had increased over time, leaving these values missing for older entries in the database. There were also a number of examples which were not relevant to the task at hand for different reasons. The domain experts provided valuable input in determining which examples to exclude completely. The exact steps of this process is detailed in Section 3.2.4.

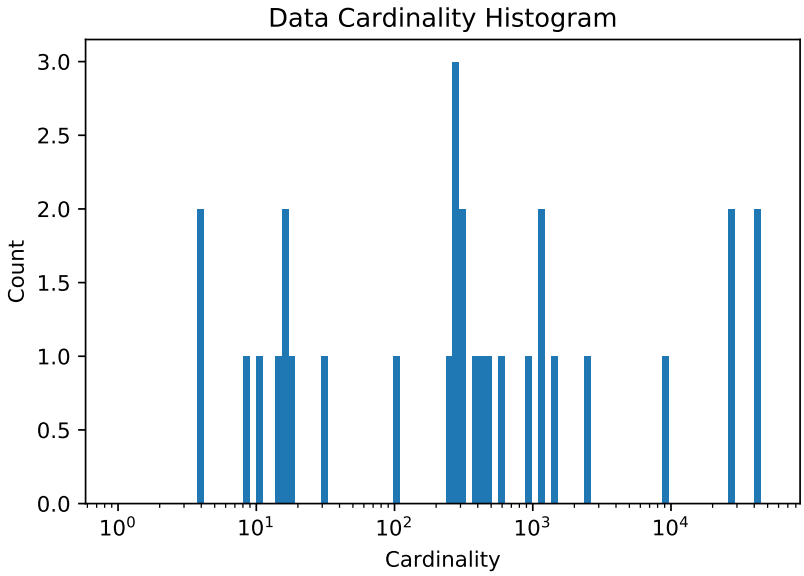


Figure 3.2: Histogram of field cardinalities of the raw data set, on a logarithmic scale.

3.2.4 Data Preparation

Data preparation is the phase of the data mining process which results in the final data set: selecting and cleaning data from the initial raw data (Chapman et al., 2000).

Data Selection and Cleaning

After input from domain experts, the final data was selected through a process of exclusion, including, but not limited to, the following steps:

1. *Exclude TIs with a given naming convention, as they were not deemed relevant*
2. *Exclude CIs with a given naming convention, as they were not deemed relevant*
3. *Exclude generic CIs*
4. *Exclude "unconfirmed" TIs*
5. *Exclude "cancelled" TIs*
6. *Exclude "cancelled" CIs*
7. *Exclude TIs not belonging to relevant organizations within Tetra Pak*

After deciding to encode all data, including categorical fields, as free text, we opted against removing fields, despite the sparsity of some fields, instead representing missing data as "#". Cleaning of free text fields was then performed by removing e.g. stop words, words inside brackets and all non-alphanumeric characters. This process is further described in section 3.2.5.

The final data set consisted of 85,000 TIs grouped in 16,000 CIs. Each data point in the data set consisted of 39 features as can be seen in Table 3.1. This is in stark contrast to the available data from the pilot study which consisted of only eleven data fields. These original fields can be seen in Table 3.3.

Table 3.3: The data fields for each TI used in the pilot. The Values column shows example values taken from a specific TI.

Feature Name	Value
ti_id	1000185941
ti_part_number	15381090000
ti_failure_mode	mechanical
ti_failure_type	218 - separated (broken into pieces, detached)
ti_material	6481600300
ti_package_type_volume_shape	tba 125 s
ti_machine_system	tba/19
mob_word	jaw system
ti_title	$tba/19 line04 spring 1538109-0000 broke
ti_problem_description	customer reported broken spring weekly care re...
ci_id	2000043793

3.2.5 Modelling

This thesis consists of multiple steps requiring modeling: vector space representations of the data set, clustering, and keyword extraction from the clusters. The first step uses neural networks to create embeddings, the second uses unsupervised clustering algorithms, and the final step makes use of a statistical method to extract keywords.

Vector Space Representations

The first consideration pertained to vector space representations. The pilot performed at Tetra Pak prior to this thesis employed an NLP approach to the representations, training Doc2Vec embeddings on the limited initial data set. As a first step, a new Doc2Vec model was implemented for the new, enhanced, data set. This way the benefit of the added fields could be seen. As a second step, pretrained SBERT embeddings were used for CI representation. This was done as an attempt to improve upon the results obtained by the Doc2Vec model.

The vector space representations were created using the Gensim (Řehůřek and Sojka, 2010) and Sentence-Transformer (Reimers and Gurevych, 2019) libraries. The SBERT embeddings provided in Sentence-Transformer have been trained using "Natural Language Inference" and then fine-tuned using the STS Benchmark data set which results in a model excelling in detecting semantic similarity between sentence pairs.

Clustering

The second consideration was that of clustering algorithm. In this domain, there are many choices. The pilot used the classic k-means clustering algorithm. A hierarchical clustering algorithm was chosen for this thesis. This choice was made to offer the business analysts a better view of how different clusters related to each other, and what larger clusters could potentially be created.

The implementation of agglomerative hierarchical clustering used for this thesis is part of the scikit-learn framework. This implementation afforded the possibility of selecting the number of clusters, as well as the specific measures to be used when calculating distances between clusters. The required measures were *linkage* and *affinity*. The linkage defines the

criterion used for deciding which clusters to link or "merge". The affinity defines the metric to be used when calculating the linkage, i.e. which distance measure (scikit-learn). In this thesis *Ward's method* for linkage was used. Ward's method minimizes the variance of the merged clusters (Murtagh and Legendre, 2014). Using Ward's method requires using the Euclidean distance as the affinity measure (scikit-learn).

The number of clusters can be altered manually when applying agglomerative clustering. Determining the number of clusters was first approached ad-hoc in order to create many smaller clusters, rather than few giant clusters. Eventually the number of clusters was determined using the silhouette score as described in section 2.3 and below.

Keyword Extraction

For the keyword extraction, a couple of approaches were discussed before deciding on tf-idf. This technique garnered acceptable results and therefore no additional approaches were employed. The tf-idf implementation found in scikit-learn was used for all keyword extraction.

Automatic Linking

For the automatic linking of TIs to CIs, a subset of the TIs from the data set was removed and used as a test set. These TIs were then represented in the same vector space as the CIs, by removing all CI specific values from the data points and using the SBERT embeddings. By using the cosine similarity, it was then possible to get a ranking of the most similar CIs for each TI and compare these findings to their originally linked CIs.

3.2.6 Evaluation

Evaluation of the different steps was one of the major hurdles of this thesis as it required the aid of the domain experts at Tetra Pak. A projection of document vectors onto a two-dimensional space using PCA was performed in order to observe any high-level differences between the vector space representations. Further evaluation of the representations were difficult to perform directly on the embeddings. Instead the evaluation of the representations became a part of the evaluation of the clusters.

Evaluation of the clusters was twofold. The first was the objective measures, in the form of silhouette scores and intra-cluster distances (described in section 2.3). The second was conducted by the business analysts and consisted of a manual effort of examining a subset of the created clusters to determine if the created clusters were reasonable and of business value. In order to utilize the analysts' time in a useful way, while still getting a useful evaluation, clusters were sampled for validation so that a wide spread of cluster sizes would be considered. The analysts would then designate a cluster as being either "meaningful" or not with a "yes" or a "no". In some cases, the analysts also left a comment to aid business understanding and further clustering efforts.

The keyword extraction was likewise evaluated by hand by the business analysts who would look at the cluster, its constituent CIs, and the extracted keywords. Only then, could they determine if the keywords were actually representative of the cluster. For evaluation the highest ranked clusters according to ICD were used. This choice was made in order to evaluate on clusters which were all useful from a business perspective.

Evaluation of the automatic linking of TIs to CIs was conducted by using a subset of the existing TIs as test set. By calculating the similarity of these TIs to the set of CIs and exam-

ining how similar the TIs were to their originally linked CI. This similarity was calculated by the cosine similarity. By counting how many CIs were more similar to the TI than the originally linked CI a measure of the performance of the autolinking was calculated.

3.2.7 Deployment

The final implementation and deployment of a tool incorporating the models and findings of this report is outside of the scope of this thesis.

3.2.8 Tools and Libraries

The following are the major tools and libraries that were utilized for this thesis:

- **PyCharm** – a Python IDE which was used for the implementation.
- **Jupyter Lab** – a sandboxing environment for Python used for prototyping.
- **Overleaf** – a collaborative LaTeX editor used for writing the thesis.
- **QuTI-P** – a Tetra Pak internal tool used for accessing and understanding the data.
- **Github** – a configuration management tool used for version control.
- **scikit-Learn** – a machine learning library for Python used for e.g. clustering.
- **Gensim** – a machine learning library for Python focusing on NLP including Doc2Vec (Řehůřek and Sojka, 2010).
- **Sentence-Transformer** – a machine learning library for Python including pretrained SBERT embeddings (Reimers and Gurevych, 2019).
- **draw.io** – a handy tool for making charts and diagrams.

Chapter 4
Results

This chapter serves to present the results of the thesis. It covers the results of clustering for the different vector space representations, as well as clustering metrics, automatic linking results, and keyword analysis.

4.1 Vector Space Representations

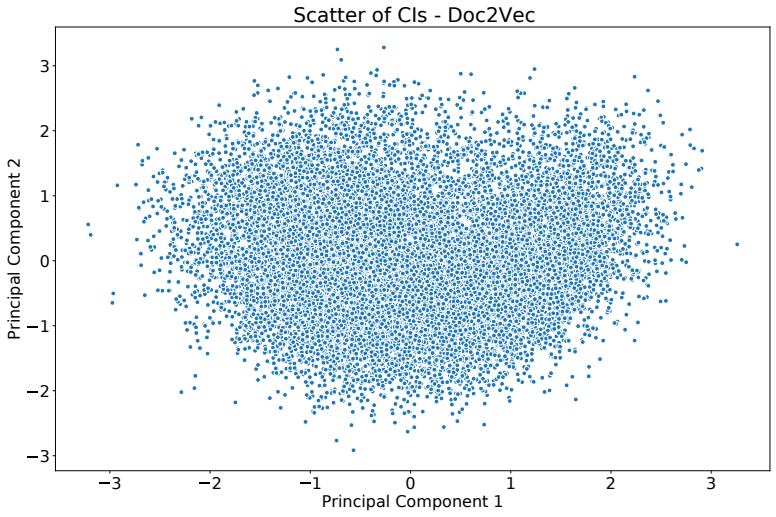
The initial results of visualizations of the two vector space representations Doc2Vec and SBERT were performed using principle component analysis (PCA). The results of this can be seen in Figures 4.1a and 4.1b for Doc2Vec and SBERT, respectively. As is evident from these two scatter plots, there were no distinguishable, larger clusters, visible in this low-dimensional space. The separation along both x and y axis was larger using SBERT with a factor two.

4.2 Clustering

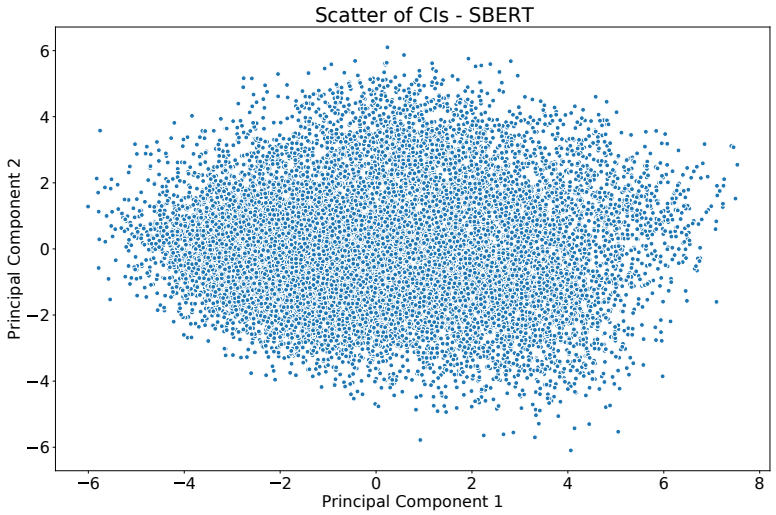
This section presents the results of the clustering. From our findings, it becomes clear that the two vector space representations garnered comparable results, both being able to deliver meaningful clusters from a business perspective. Upon closer inspection of the intra-cluster distances and the amount of usable clusters, the SBERT model gave better results.

4.2.1 Number of Clusters

The results of iteratively calculating the silhouette score of a clustering using embeddings can be seen in Figures 4.2a and 4.2b, for Doc2Vec and SBERT respectively. The results show that the optimal number of clusters n was very close to $n = \frac{N}{2}$, where N is the number of CIs in the data set. $n = \frac{N}{2}$ was used for all clusterings throughout the thesis.



(a)

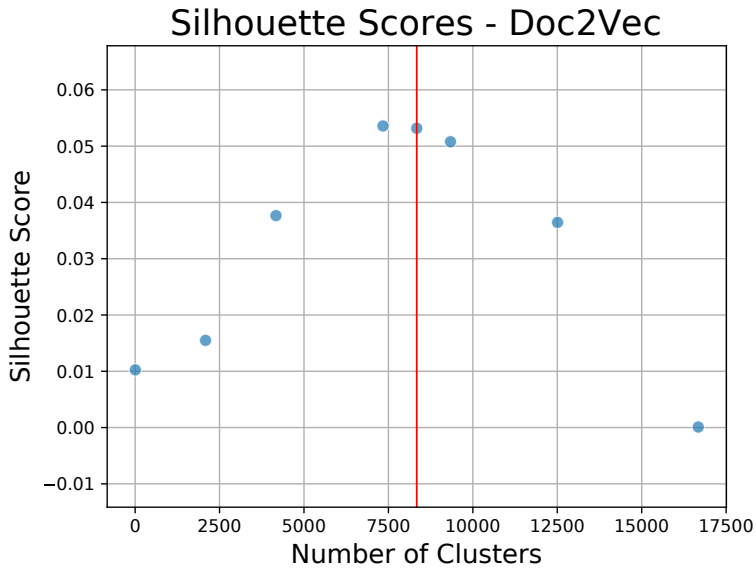


(b)

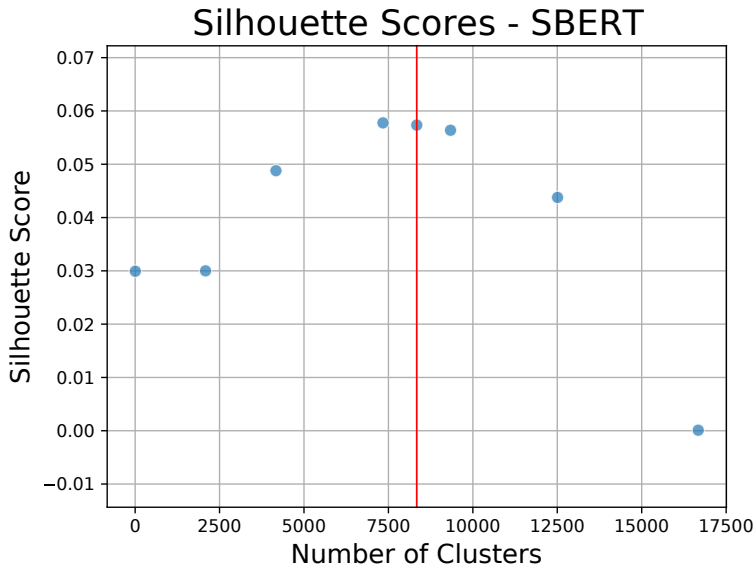
Figure 4.1: PCA analysis of CI vector space representations. (a) shows the CIs represented by Doc2Vec embeddings. (b) shows the CIs represented by SBERT embeddings. Notice the different scales on the axes of the figures.

4.2.2 Doc2Vec

Clustering using Doc2Vec embeddings and agglomerative clustering with $n = \frac{N}{2}$ as per above, resulted in a spread in the cluster sizes as shown in Figure 4.3a. The plurality of the clusters have cluster size two with few clusters larger than four, and no clusters larger than twelve. The cluster validation by the business analysts showed that ICD $\gtrsim$ **1.5** results in generally poor clusters. Looking at the entire set of clusters, including singleton clusters, this means that 36.0% of the clusters were within the "acceptable" spectrum. See A.1 for the validation results.



(a)

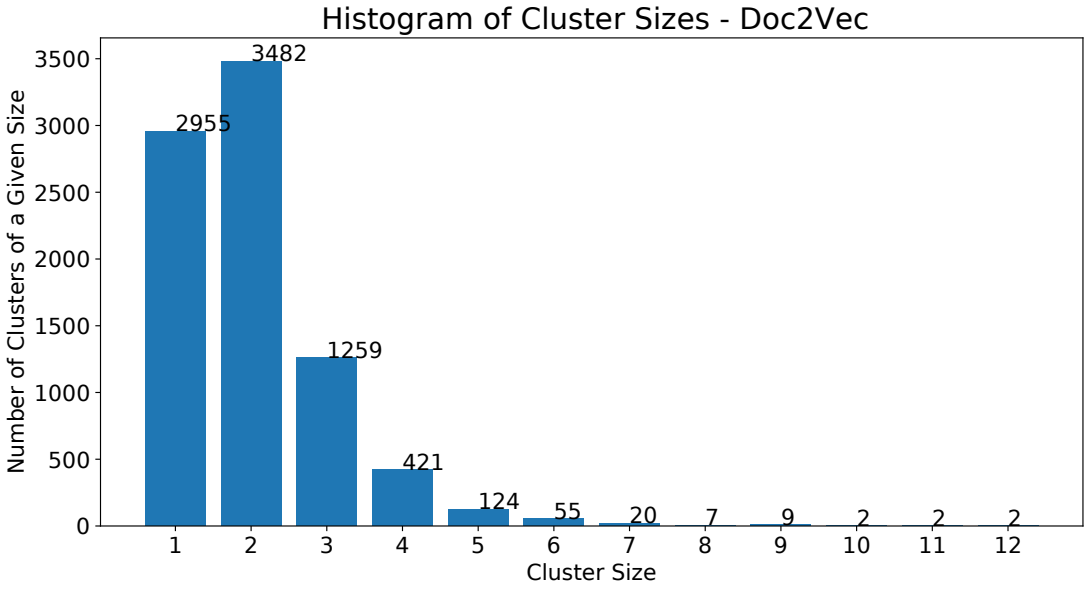


(b)

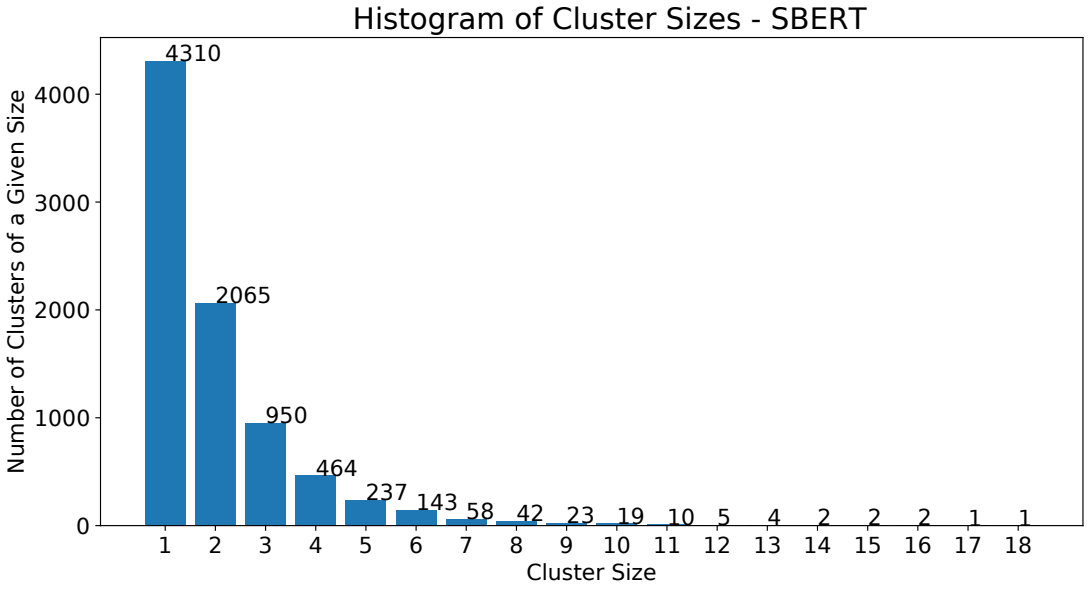
Figure 4.2: The silhouette scores obtained for different number of clusters. (a) represents the scores calculated on the clusters made using Doc2Vec embeddings. (b) represents the scores calculated on the clusters made using SBERT embeddings. The vertical lines denote *number of clusters*= $\frac{N}{2}$.

4.2.3 Sentence-BERT

Clustering using SBERT embeddings and agglomerative clustering with number of clusters $n = \frac{N}{2}$ as per above, resulted in a spread in the cluster sizes as shown in Figure 4.3b. The plurality of clusters are singleton sets, with the second highest count being couples. There



(a)



(b)

Figure 4.3: Hierarchical Clustering with number of cluster set to $n = \frac{N}{2}$, where N = *number of CIs*. (a) shows the results of using Doc2Vec embeddings, (b) shows the results of using SBERT pre-trained embeddings.

are few clusters larger than five, and no clusters larger than 18.

The results of the domain expert evaluation of a sampling of the clusters, made using the SBERT embeddings, show that ICD $\gtrsim 2$ resulted in generally poor clusters. Looking at the entire data set of clusters, including the singleton clusters, this means that 53.3% of the clus-

ters were in the “acceptable” spectrum. More clusters, in a wider range of cluster sizes, were evaluated for SBERT as it was proving more promising. As can be seen in the aforementioned table, also clusters of cluster size larger than two resulted in acceptable clusters. The larger clusters which were acceptable also had a low ICD value, close to 2. See A.2 for the validation results.

4.2.4 Comparing Vector Space Representations

Figure 4.4 shows a comparison of clusters created by the two different clusterings using the two different vector space representation. Here the intersection between the the clusters created by Doc2Vec and those created by SBERT is shown. The number of identical clusters created by the two representations is counted. The clusters are then separated by cluster size and a histogram is created. As is evident from the figure, the overlap is very small for all cluster sizes except singletons. Upon closer inspection of the underlying clusters it becomes apparent that some clusters created using one embedding exist as subsets of other clusters using the other embedding. I.e. a cluster containing two CIs created using e.g. Doc2Vec is a subset of a cluster of size three created using SBERT.

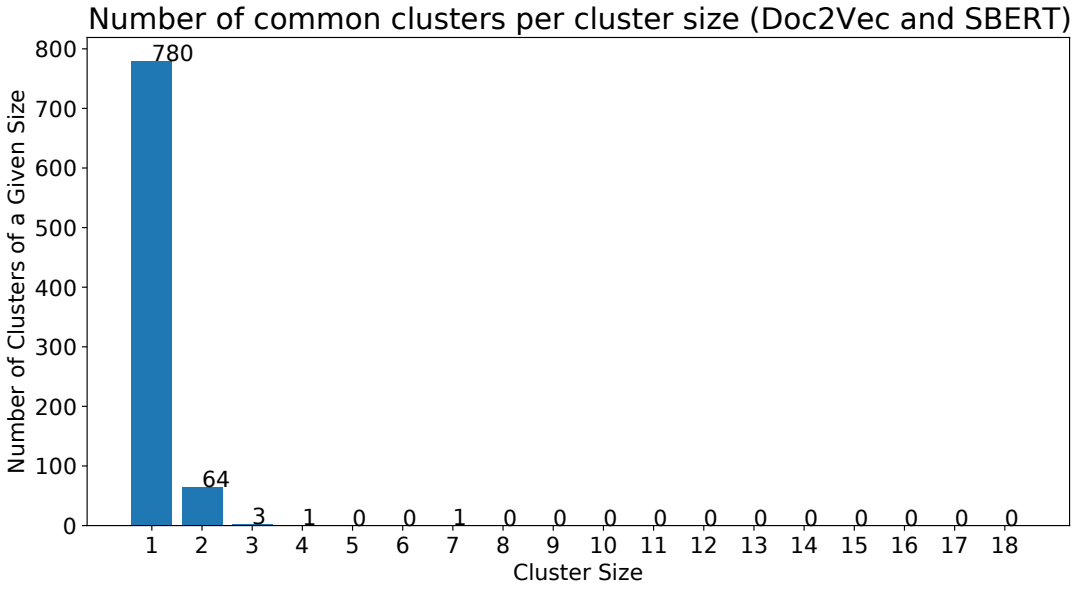


Figure 4.4: Intersection of clusters, for Doc2Vec and SBERT clusterings, counting exact matches per cluster size.

4.3 Keyword Extraction

Keyword extraction was performed using tf-idf on clusters created based on Doc2Vec as well as SBERT embeddings. The clusters chosen for evaluation of keyword extraction were the top ranked clusters according to the intra-cluster distance defined in 2.3. The results of the keyword extraction and the business analyst evaluation can be seen in tables 4.1 and 4.2 for

Doc2Vec and SBERT respectively. The column "Analyst Ranking" in these figure represents the quality of the keyword extraction for each cluster. In general the approach garnered representative keywords, with a majority of clusters having "good" keywords.

Table 4.1: tf-idf results for best clusters using Doc2Vec. Clusters ranked based on intra-cluster distance

ICD Rank	Cluster Size	Top 3 Terms	Analyst Ranking
1	2	width, rail, station	good
2	2	cau, nihon, xhic	poor
3	2	rail, excessive, presents	partial
4	2	inserts, helicoil, guide	good
5	2	link, acid, electrolitics	good
6	2	capps, finger, gripper	good
7	2	level, probe, floater	good
8	2	block, occurred, helicoil	partial
9	2	unnormal, alcip, rod	partial
10	2	brazil, assessment, nr12	good

Table 4.2: tf-idf results for best clusters using SBERT. Clusters ranked based on intra-cluster distance.

ICD Rank	Cluster Size	Top 3 Terms	Analyst Ranking
1	2	cau, nihon, xhic	poor
2	2	width, rail, station	good
3	2	flange, workshop, insufficiency	partial
4	2	oil, seal, leakage	good
5	2	steering, locked, device	good
6	2	rail, excessive, presents	partial
7	2	hepa, filter, saturation	good
8	2	chute, drop, welding	good
9	2	inductor, r1, sealing	good
10	2	654128, sca, cap	poor

4.4 Automatically Linking TIs to CIs

As described in section 2.3, the automatic linking was evaluated by calculating the number of CIs which were more similar to the test TI than the originally linked CI. The results of this evaluation on the test set of 100 randomly chosen TIs can be seen in 4.5. From this bar chart it is evident that in almost half of the test cases, i.e. the left-most bar, the correct CI was found in the top five most similar CIs to the test TI. Similarity in this context is equivalent to proximity calculated using cosine similarity in the SBERT embeddings vector space. If this examination is extended to the top 25 ranking CIs, i.e. the third bar from the left, then the correct CI was present in around 70% of the cases. In practice these means that

when searching for an appropriate CI for a TI the originally linked CI is in the top five most similar CIs in 50% of cases, and in the top 25 most similar CIs in 70% of cases. This narros the number of CIs for a business analyst to consider for linking from tens of thousands to under 100 in most cases.

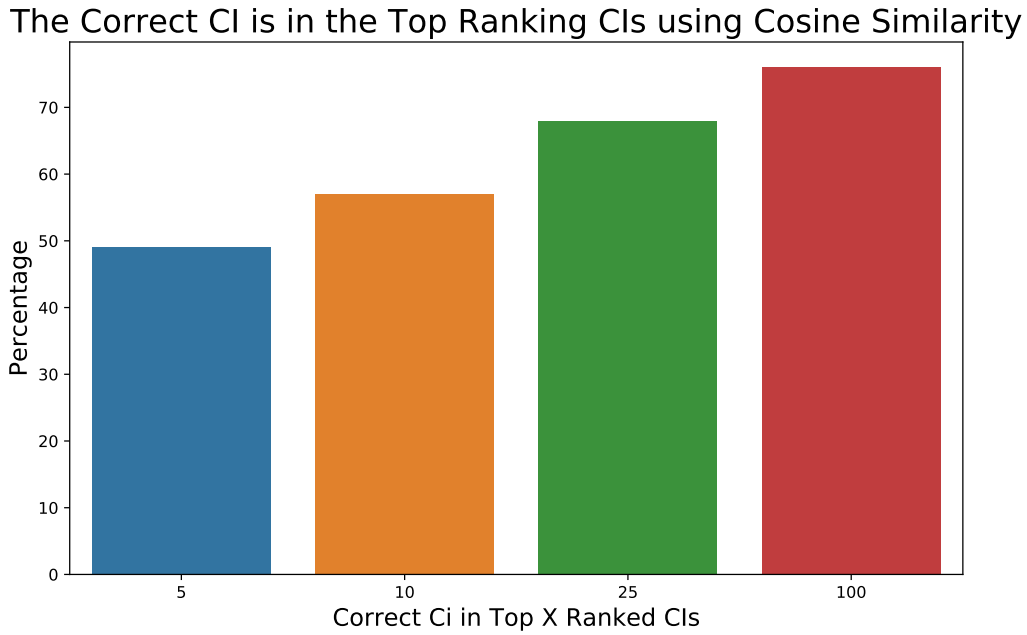


Figure 4.5: Percentage of instances where the correct CI was among the top ranking CIs when ranking CIs for a given TI

Chapter 5

Discussion

This chapter presents an in-depth discussion and interpretation of the results and findings from Chapter 4, as well as a the methods used, detailed in Chapter 3. It also includes some general reflections on the thesis work and some highlights of what the next steps forward could be. Throughout this chapter "we" refers to the author.

5.1 Methodology and Results

This section will discuss some of the choices made regarding workflow and models for working with a data science project. How these choices impacted the results will also be touched upon.

5.1.1 CRISP-DM

Given the authors' background in computer science, and agile workflows, the prospect of an agile approach to data science was highly appealing. At the outset of this thesis, we hoped to conclude multiple iterations of the CRISP-DM process. A process which is seen in Figure 3.1. However, the time consumption of data understanding, data preparation, as well as evaluation proved to exceed expectations, resulting in only two complete iterations.

Naturally more iterations mean the opportunity of better models and better results, but the iterative nature also means that results are reached early, multiple times, and offers an opt-out of continued work. Despite few completed iterations during this thesis, we still view CRISP-DM as a powerful structural tool when pursuing a data mining or data-science related project. It provided a solid structure for the thesis, clearly highlighting the necessary high-level actions at each step of the process.

5.1.2 Vector Space Representations

The clustering of words into groups of synonyms by using Word2Vec was shown by Zhang et al. (2017) and we therefore had high hopes of similar success using Doc2Vec applied to CIs. As can be seen in the expert evaluations in Table A.1, a majority of clusters with a significantly low intra-cluster distance (ICD) were meaningful when using the Doc2Vec embeddings.

Sentence-BERT, as described by Reimers and Gurevych (2019) is a state-of-the-art model for sentence-pair regression, excelling in identifying e.g. semantic textual similarity. The neural network used for training sentence embeddings in SBERT allows for the distinction between homographs (see Chapter 1). This ability which Doc2Vec lacks, would suggest that SBERT should achieve comparable, or better results than Doc2Vec when creating document embeddings. Upon PCA analysis and scatter-plotting, it would seem that SBERT is a better choice already at this stage, achieving a better spread of CIs in the vector space, using the two principle components for each representation, see Figures 4.1a and 4.2b.

The decision to represent all data as free text was based on the previous work done at Tetra Pak and this choice of modelling technique restricted the modelling process. This restriction still allowed for the use of all relevant fields. However, this choice constrains the ability to decipher how different categorical fields affect the clustering, as the vector space dimensions no longer represent these fields and their sets of values. This choice obscures the original field values and how they affect representation, but also comes with the benefit of a dimensionality reduction which decreases the amount of memory required. The tradeoff is acceptable as the results of clustering were meaningful, and the keyword extraction still served as a means to understanding the underlying reasons for clustering the CIs of a given cluster.

The process of creating SBERT embeddings, as is detailed in section 2.1.2, consists of two steps. The first step is to train the sentence embeddings on an appropriate corpus. The second step is fine-tuning the embeddings by using domain-specific data. In this thesis, we use pretrained embeddings described in 3.2.5. These embeddings are optimized for semantic similarity tasks and suit our purposes well. In this thesis we performed no further fine-tuning of these embeddings, but rather used them "as-is". The alternative scenario is continued fine-tuning using domain specific data. An interesting topic for further study would be to fine-tune these embeddings on the data sets available from QuTI-P and compare the resulting cluster quality with that of the current model.

5.1.3 Clustering

Clustering was performed using agglomerative hierarchical clustering. This choice was made in conjunction with the business analysts and data scientists at Tetra Pak who wished to demonstrate the relationship between different clusters, and the levels at which different clusters related to each other. The choice of hierarchical clustering naturally places some decisions in the hand of the engineer. Unlike an algorithm like DBSCAN, which determines the number of clusters independently of the user, agglomerative clustering takes the number of clusters as a parameter.

As described in section 3.2.5, the scikit-learn implementation of agglomerative clustering requires a choice of linkage and affinity measure. The implementation defines default settings for these measures. The default choice for the linkage criterion is *Ward's method*. This approach minimizes the variance of the clusters being merged. The default affinity measure

when using Ward's method is the Euclidean distance scikit-learn. As is noted in the above referenced section, these defaults were used. The decision to use the default values was made in an attempt to alter as few parameters as possible for the initial clustering attempt, while at the same time choosing measures which would have a good chance of producing acceptable results. Ward's method stems from the 1960's (Murtagh and Legendre, 2014) and is a popular choice of linking criterion in hierarchical clustering. The default values proved to garner acceptable clusters in the context of this thesis and emphasis was placed on comparing vector space representations rather than an in-depth exploration of different clustering criteria. A different choice of linkage criterion would very likely have resulted in a different set of clusters. One choice of affinity measure which we suggest may result in useful clusters is *cosine* (scikit-learn), given that the cosine similarity gave good results in the automatic linking. The exploration of this and other methods of linking and affinity measurement is left as potential future work.

It is worth noting that we did not know beforehand how many, if any, useful clusters existed in the data set. The notion of defining a number of clusters for the algorithm to create then becomes almost baroque. This could potentially lead to large numbers of clusters which contain CIs with no discernible connection to each other, but which have rather been grouped in order to fulfill the externally mandated number of clusters. In the Doc2Vec clustering results, what speaks against this being the case, is that the plurality of clusters are actually of size two (Figure 4.3a). In the case of SBERT, however, we see that the majority of clusters are singletons, which rather points to that there perhaps are fewer meaningful clusters in this set than we are imposing on the algorithm. This is contradicted by the silhouette score which objectively determined the optimal number of clusters, discussed further below.

The clusters created by a simple k-means clustering algorithm were comparable in composition to those created by the agglomerative algorithm, given that the number of clusters was the same. The k-means clustering however does not afford the same tree of clusters as hierarchical clustering.

Cluster size was a topic of great interest to the business analysts at Tetra Pak. There was not a strong desire to create larger clusters of high cardinalities since these offered little overview and were very tedious to validate. Instead the approach of many clusters and a means to gain insight into how these might relate on a higher level was more desirable. Nevertheless, we made sure to validate some of the larger clusters as well, to determine if these clusters should rather be broken up into multiple smaller clusters.

Looking at the evaluations made by the business analysts, the cut-offs were approximated to ICD $= 1.5$ for Doc2Vec and ICD $= 2$ for SBERT. Looking at the full clusterings, we can determine that for Doc2Vec, for ICD $\lesssim 1.5$, we found almost no clusters of $size > 2$. For SBERT only 16% of the clusters with ICD $\lesssim 2.0$ had a size > 2. This lends credit to the analysts desire for smaller clusters. However, upon validation of larger clusters, we do see that some of these clusters were also regarded as having some value, even though the ICD in these cases are outside of the range which the analysts decided. This leads us to believe that the ICD boundaries might be somewhat conservative, which is discussed further below.

Silhouette score was used to determine the optimum number of clusters. Silhouette score, as is detailed in section 2.3, is an objective measure for determining the number of clusters. The silhouette score takes into account the tightness and separation of all the clusters, so given a set vector space representation, the silhouette score will find the optimal number of clusters. It however says nothing about the vector space representation. If the representation

of the data is poor, then the clusters will also be poor. That being said, the number of clusters was set to $n = \frac{N}{2}$ (see Chapter 4) after silhouette score evaluation. There are alternative ways of calculating the optimum number of clusters, such as the *Elbow method*. Choosing a clustering algorithm which does not require a manual input of number of clusters removes the need of calculating the optimum number of clusters all together. As in the choice of parameters for agglomerative clustering, we chose not to investigate alternatives further.

The manual cluster evaluation method used, requiring business analysts with limited time to validate clusters is possibly the greatest limitation of this thesis. It was necessary to choose subsets of clusters for validation and estimate a cut-off for the ICD. With further validation efforts, it is possible that the ICD cut-off values could have been increased. As is seen in A.1, there are larger clusters which were deemed useful, but which had ICD-values exceeding the limits. With further validation efforts we might have been able to determine variable ICD-values for different cluster sizes, allowing greater tolerance for larger clusters.

The pilot study performed at Tetra Pak was based on a smaller subset of the data fields used in this thesis. The pilot used the same type of Doc2Vec embeddings as we have used here. The subjective view of the business analysts was that the results obtained in this thesis was an improvement on the results obtained in the pilot. This means that adding the additional fields added relevant information which improved the clusters, even when using the same vector space representations as the pilot. Using SBERT embeddings then improved the clusters further.

5.1.4 Keyword Extraction

Keyword extraction can be performed in a number of different ways. The choice of tf-idf was based on its widespread use, as well as the fact that the team had previously attempted to cluster using tf-idf in order to get the added benefit of keyword analysis, without managing to do so due to memory constraints. In our solution, we did not employ tf-idf as a means of clustering, but rather applied it to precalculated clusters. This resulted in reduced memory usage and no new clusterings were created. A first iteration of keyword extraction was done extracting only the top ranked word. This often gave unsatisfactory results as there was no context. In a second iteration, the top three ranked words were extracted instead, and this turned out to be more useful to the analysts. We performed no particular pre- or postprocessing of the data to and from the tf-idf algorithm, and this resulted in numbers, abbreviations and other technical terms appearing in the output. If this is desirable or not is dependent on who will use the results.

As a next step, it could be interesting to attempt automatic text summarizing based on the text fields of each cluster. It is however uncertain how successful this would be as much of the free text in the system is already written in a very brief format.

5.1.5 Automatic Linking of TIs to CIs

Automatically linking TIs to CIs is a classification problem. There is a plethora of more or less advanced approaches to classifying data into different categories, ranging from decision trees to deep neural networks. Given that we already had a vector space model for our CIs, we thought this would be the most obvious starting point, trying to find the CIs closest to our test TIs in the same vector space.

This brings us to the question of the existing CIs. These have been created by business analysts over many years in order to bring order to technical issues. However, one of the main motivations behind this thesis was that this system of consolidating TIs had grown to be unmaintainable in this manner. Many CIs were regarded as more or less duplicates of already existing CIs, or at the very least overlapping, due to be clustered in this thesis. This means that the quality of the "clustering" that CIs inherently are, is flawed and poor. This means that the suggestions we get by employing cosine similarly to TIs and CIs in this vector space are affected by this same flaw. This in turn means that in the list of recommended CIs for a new TI, some CIs should actually have been merged, thereby moving the correct CI further up in the ranking and improving the results of the automatic linking.

It can be argued that the existing CIs should be removed altogether as they are in some way all biased by the analyst who created them and decided to group TIs in a particular way. Instead a new clustering could be made based solely on TIs, and this would then allow for an objective hierarchy of technical issues. These clusters could then be appended with relevant new TIs which are added to the system. This would effectively make these clusters equivalent to the current system of CIs.

Chapter 6
Conclusions

With the increasing amount of data available throughout industry and society, companies everywhere are in the middle of a data revolution. At Tetra Pak, the data set of technical issues (TI) of deployed machines, has been continually appended for many years. At the outset of this thesis three research goals were defined based on this data. The first goal was to find a useful vector space representation of the data which consisted, among other things, of multiple free text fields. This was achieved using Doc2Vec and pretrained Sentence-BERT embeddings. The second goal was to suggest a useful clustering model for the vector space representations. This was successfully achieved using agglomerative hierarchical clustering. The third goal was to determine if it was possible to automatically link new TIs to existing consolidated issues (CI), given that a good match existed. This was achieved by representing TIs in the same vector space as the CIs and then suggesting CIs based on the cosine similarity. Additionally, we found that we could successfully extract meaningful keywords from CI clusters by applying tf-idf, thereby giving business analysts insight into cluster content at a glance.

The field of Natural Language Processing (NLP) has taken great leaps forward with the rise of deep learning, pushing the boundaries of what machine learning can achieve. Combining ever growing and maturing data sets available in modern industry, with modern document embeddings and clustering techniques, NLP is poised to change the role of the business analyst. Fewer hours will be spent on tedious day-to-day tasks, such as classifying and grouping data. Instead time can be spent drawing conclusions from higher level data structures and aiding the development of stronger models requiring domain expertise.

These concepts can be generalized. The idea of finding and grouping similar objects is universal. Being able to find these similarities efficiently in natural language is instrumental to many tasks that data scientists are faced with today. Powerful document embeddings, such as SBERT, enable better search engines, recommendation systems, question answering systems, and more. The domain specific power of these concepts showcased in this thesis, gives a taste of the insights that can be automatically generated from the vast amounts of data being generated across the industrial landscape today.

Acronyms

TI	Technical Issue
CI	Consolidated Issue
FSE	Field Service Engineer
QuTI-P	Quality and Technical Issues Platform
OHE	One-Hot Encoding
ICD	Intra-Cluster Distance
BERT	Bidirectional Encoder Representations from Transformers
SBERT	Sentence Bidirectional Encoder Representations from Transformers
ML	Machine Learning
NLP	Natural Language Processing
tf-idf	Term Frequency - Inverse Document Frequency
CRISP-DM	CRoss-Industry Standard Process for Data Mining

Glossary

Doc2Vec	framework used to train document embeddings
Word2Vec	framework used to train word embeddings
gensim	library including Doc2Vec for Python
embedding	vector representation of e.g. words
BERT	model for training embeddings
SBERT	more efficient version of BERT
scikit-learn	Python library for ML
Python	programming language which is well suited for ML/NLP
silhouette score	scoring system for cluster quantity
k-means	basic clustering algorithm
hierarchical clustering	clustering algorithm which gives a tree of clusters
agglomerative clustering	version of hierarchical clustering
data cleaning	improving data quality by removing data
data selection	improving results by choosing appropriate data
homograph	words with the same spelling but different meaning
cosine similarity	use the angle between vectors to determine similarity
vector space representation	creating a vector corresponding to a data point
cluster	group of data points which are said to belong together
CRISP-DM	model for working iteratively

Bibliography

C. C. Aggarwal. *Data Mining: The Textbook*. Springer, apr 2015. ISBN 9783319141411. URL https://www.xarg.org/ref/a/3319141414/.

B. Bengfort. *Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning*. O'Reilly Media, jul 2018. ISBN 1491963042. URL https://www.xarg.org/ref/a/1491963042/.

P. C. Bruce and A. Bruce. *Practical statistics for data scientists: 50 essential concepts*. OReilly Media, 2018.

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. R. H. Shearer, and R. Wirth. *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS, 2000.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996. URL http://dl.acm.org/citation.cfm?id=3001460.3001507.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, pages 45–49. MIT Press, 2016. http://www.deeplearningbook.org.

C. Guo and F. Berkhahn. Entity Embeddings of Categorical Variables. *arXiv e-prints*, art. arXiv:1604.06737, Apr 2016.

Z. S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954. doi: 10.1080/00437956.1954.11659520. URL https://doi.org/10.1080/00437956.1954.11659520.

I. Hui, 2019. URL https://irenelizihui.files.wordpress.com/2016/07/13871742_837560163046220_343227247_n.jpg. Accessed: 2019-12-10.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Wiley, 1990.

Q. Le and T. Mikolov. Distributed representations of sentences and documents. *31st International Conference on Machine Learning, ICML 2014*, 4, 05 2014.

L. Ma and Y. Zhang. Using word2vec to process big text data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2895–2897, Oct 2015. doi: 10.1109/BigData.2015.7364114.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press. URL `https://projecteuclid.org/euclid.bsmsp/1200512992`.

O. Maimon and L. Rokach. *Data mining and knowledge discovery handbook*. Springer, 2010.

T. Menasria, S. Neffar, S. Chafaa, L. Bradai, R. Chaibi, M. Mekahlia, D. Bendjoudi, and A. Si Bachir. Spatiotemporal diversity, structure and trophic guilds of insect assemblages in a semi-arid sabkha ecosystem. *PeerJ*, 3:e860, 03 2015. doi: 10.7717/peerj.860.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`.

F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, Oct 2014. ISSN 1432-1343. doi: 10.1007/s00357-014-9161-z. URL `http://dx.doi.org/10.1007/s00357-014-9161-z`.

M. Nayak. https://miro.medium.com/max/1352/1*d0jwmf36suey7as8bva-dw.jpeg, 2019. URL `https://miro.medium.com/max/737/1*d0JWmF36SUey7aS8bvA-dw.jpeg`. Accessed: 2019-10-23.

F. Provost and T. Fawcett. Data science and its relationship to big data and data-driven decision making. *Big Data*, 1(1):51–59, 2013. doi: 10.1089/big.2013.1508.

R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 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. doi: 10.1016/0377-0427(87)90125-7.

G. Salton. Some experiments in the generation of word and document associations. In *Managing Requirements Knowledge, International Workshop on*, volume 1, page 234, Los Alamitos, CA, USA, dec 1962. IEEE Computer Society. doi: 10.1109/AFIPS.1962.61. URL https://doi.ieeecomputersociety.org/10.1109/AFIPS.1962.61.

scikit-learn. sklearn.cluster.AgglomerativeClustering scikit-learn documentation. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering. Accessed: 2020-11-14.

C. Shearer. The crisp-dm model: the new blueprint for data mining. *J Data Warehousing*, 5 (4):13.22, 2000.

K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*, pages 74–76. Addison-Wesley, first edition, 2005.

Tetra Pak AB, 2019a. URL https://www.tetrapak.com/us/about. Accessed: 2019-12-15.

Tetra Pak AB, 2019b. URL https://www.tetrapak.com/about/history. Accessed: 2019-12-15.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wikimedia Commons. Kmeans-gaussian-data.svg, 2011. URL https://upload.wikimedia.org/wikipedia/commons/e/e5/KMeans-Gaussian-data.svg. Accessed: 2019-12-15.

Wikimedia Commons. Crisp-dm process diagram, 2012. URL https://upload.wikimedia.org/wikipedia/commons/b/b9/CRISP-DM_Process_Diagram.png. Accessed: 2019-11-22.

L. Zhang, J. Li, and C. Wang. Automatic synonym extraction using word2vec and spectral clustering. In *2017 36th Chinese Control Conference (CCC)*, pages 5629–5632, July 2017. doi: 10.23919/ChiCC.2017.8028251.

Appendices

Appendix A
Additional Tables

Table A.1: Evaluation of Select Clusters - Doc2Vec

ICD	Clust. Size	Useful Clust.	Analyst Comment
0.421	2	Yes	
0.486	2	Yes	
0.658	2	Yes	Both CIs with PS. Solution is the same but two different PS IDs
0.723	2	Yes	Both CIs Solved, part number different but belonging to same group
0.744	2	Yes	
0.773	2	Yes	
0.852	2	Yes	
0.951	2	Yes	
0.975	2	Yes	
1.028	2	Yes	Different Stakeholders since Platforms were different
1.118	2	No	Different issues
1.281	2	Yes	
1.448	2	Yes	
1.474	2	Yes	
1.908	10	No	all related to Mu, but different problems
2.301	10	No	subsystem of CIs are similar
2.291	11	Yes	one CI is borderline, but overall suggetion is OK
2.444	11	No	
2.318	13	No	subsystem of CIs are similar
2.368	13	No	

Table A.2: Evaluation of Select Clusters - BERT

ICD	Clust. Size	Useful Clust.	Analyst Comment
0.416	2	Yes	
0.482	2	Yes	
0.599	2	Yes	
0.832	2	Yes	
0.934	2	Yes	
0.995	2	No	Both CIs with PS. Solution is the sam but two different PS IDs
1.134	2	No	
1.171	2	Yes	
1.191	2	Yes	
1.206	2	Yes	
1.513	3	Yes	
1.732	4	Yes	
1.782	5	Yes	different faults but component involved belongs to the same family
1.993	4	Yes	
1.998	2	Yes	different FMs not all of the TIs have strong commonalities
2.001	3	No	3 CIs 1to1- same b-group but different components
2.122	6	No	different problems
2.142	5	No	Some CIs have an OK matching
2.170	13	No	Two smaller subsets within the cluster are meaningful
2.181	7	Yes	Somewhat similar problems)
2.220	8	No	Subsets which could possibly be relevant clusters
2.255	10	Yes	Not all CIs similar problems. but clustering gives meaningful overview
2.326	15	Yes	Good, All related to same family of commercial component
2.363	15	No	All same B-Group but too different problems
2.399	12	No	Overall cluster not ok, but subsystems with similarities are found
2.448	11	No	Oveall cluster not ok, but subset of Cis could turn into a cluster
2.500	2	No	different problems
2.507	6	No	All same B-Group but too different problems
3.000	2	No	Very different issues
3.000	4	No	Different problemss
3.001	3	No	All 3 CIs refers to different problems with different B/C Groups, p/n7
3.001	6	No	Some CIs similar from B-Group point of view, but too broad problemss
3.001	5	No	Different problems
3.002	8	No	Different problems
3.009	9	No	Oveall cluster not ok, but subset of Cis could turn into a cluster

INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2020-06-11

EXAMENSARBETE Representing and Grouping Technical Issues for Business Insights
STUDENT Ola Westerlund
HANDLEDARE Pierre Nugues(LTH), Astrid Nielsen (Tetra Pak)
EXAMINATOR Jacek Malec (LTH)

AI-modell för beslutsstöd vid prioritering av maskinfel

POPULÄRVETENSKAPLIG SAMMANFATTNING **Ola Westerlund**

Tack vare de senaste årens explosion i tillgång på data, har idag många större företag en egen avdelning dedikerad till AI och utveckling av denna typ av algoritmer. Inom industrin är det vanligt att samla in data när olika fel uppstår för att kunna följa upp och dra nytta av kunskaper om liknande fel dyker upp igen.

På Tetra Pak har man sedan många år samlat in data vid maskinfel. Servicetekniker har med hjälp av en app kunnat registrera informationen. Bland annat har teknikerna skrivit fritext där de beskriver olika aspekter av ett fel. Det har länge funnits en idé på Tetra Pak om att använda all den data som samlats in för att förstå vilka problem som är snarlika, som hör ihop, och på så vis kunna se vilka grupper av fel som finns. Om man kunde lyckas skapa grupper av fel skulle man kunna dra slutsatser om vilka fel som borde prioriteras för mer permanenta lösningar ifrån företaget.

I detta examensarbete har vi utvecklat en metod för att automatiskt gruppera olika maskinfel. Maskinfelen jämförs med varandra, baserat på bl.a. de fritextsvar som teknikerna matat in, och slås sedan ihop till grupper. I arbetet har vi provat ut olika metorder för att representera maskinfelen, och optimerat antalet grupperingar.

En av svårigheterna med att gruppera denna data är just fritext-elementet. Med hjälp av moderna språkteknologi-metoder, som använder sig av neurala nät, kan man skapa matematiska analoger för varje datapunkt, inklusive dess text-innehåll. Dessa kan sedan jämföras på ett systematiskt vis.

För att underlätta för företagets analytiker som ska använda grupperingsmjukvaran, har vi även tagit fram ett system som automatiskt märker varje gruppering med några nyckelord som är representativa för innehållet i gruppen.

För att framtidssäkra systemet, utvecklades en algoritm för att länka nya maskinfel till befintliga grupper av fel, på så vis behöver inte längre analytiker göra manuella grupperingar.