Master Thesis

in Business Administration, M.Sc.

University of Passau
Faculty of Business and Economics

Chair of Financial Data Analytics
Prof. Dr. Ralf Kellner

Topic: Evaluation of distributed representation of social network data in vector space

Submitted by: Fabian Dick
Matr. Nr. : 104936
E-Mail: dick16@ads.uni-passau.de

Submitted on: 9th June 2023

Supervisor: Prof. Dr. Ralf Kellner

Abstract

Alternative investments have gained popularity in recent years due to high return on investments. However, data sources to validate possible targets, e. g. in the Private Equity and Venture Capital industry, are incomplete and rare. Alternative data can fill the gap to some extent. One example are social networks. The creation of a distributed representation of documents and words resulting from social network scraping can be structured, analysed and used for deeper analysis. Either for topic modeling or for more specific applications. Top2Vec by Angelov, 2020 creates a vector space that semantically represents documents C and its associated words V in order to retrieve topics and corresponding topic vectors $\vec{t}$. With the additional help of the concepts of cosine similarity and sentiment analysis, it is possible to perform deeper analysis on the companies' product, associated language and competitor benchmarking leading to a supportive tool, called Customer2Vec. The creation of such an application in order to gain an overview from an investor's point of view is the topic of this thesis.

Contents

List of Tables

List of Figures

Symbols and Notations

Symbol	Meaning
d	Document
n	Number of documents
$\vec{d}$	Document vector
C	Corpus
$\vec{C}$	Document embedding matrix
C_t	Documents belonging to topic t
$C^{\lambda}_{w'}$	Documents with the property of $S_C(\vec{w'}, \vec{d}) \geq \lambda \; \forall \; w' \in V'$ and $d \in C$
λ	Threshold for S_C
S_C	Cosine similarity
$\vec{C}_t$	Document vectors belonging to topic vector $\vec{t}$
w	Word
m	Number of words
V	Vocabulary
$\vec{V}$	Vocabulary embedding matrix
$w*$	Topic word
$V*$	Set of topic words
w'	Keyword
V'	Set of keywords
t	Single topic
k	Number of topics
$\vec{t}$	Topic vector
T	Set of topics t
δ	Context window size used to create $\vec{w}$ and $\vec{d}$
X	Company
x	Arbitrary point in the reduced vector space via UMAP
e	Number of nearest neighbors within HDBSCAN
$\overrightarrow{brand}$	$\vec{w}$ with $w = X$
V_X	Vocabulary including only words with $S_C(\overrightarrow{brand}, \vec{w}) > \lambda \; \forall \; w \in V$
α	Dirichlet prior for per-document topic distribution parameter
θ	Dirichlet prior for topic distribution for a document
z	Selected topic chosen by the LSA model via the topic distribution θ
β	Topic-word matrix representing the word distribution for z within LDA
$\mathcal{M}$	Manifold representation of $\vec{C}$ or $\vec{V}$
ϕ	Number of used neighbors for approximating $\mathcal{M}$ in UMAP model
γ	Used dimensionality for approximating $\mathcal{M}$ in UMAP model
v	*Vector size* of $\vec{d}$, $\vec{w}$ and $\vec{t}$
U	Document-topic matrix
W	Topic-term matrix
P	Matrix created by dividing the r-th column of W by its L2-norm
Q	Matrix created by dividing the r-th column of U by its L2-norm
Σ	Matrix used for Singular Value Decomposition
$PWI(T)$	Probability weighted amount of information of a given set of topics T
$\vec{Y}$	Vector containing the unique numbers of years of a given UGTD sample
Θ	Vector containing the numbers zero to length of $\vec{Y}$
Γ	Degree of actuality
Y'	Unique years of given UGTD

1 Introduction

Companies have gained an abundance of channels to communicate with their customer base or users with the rise of social networks. They can market new product launches, ask for customer opinions or share news about general company developments. While social media is an enabler for marketing campaigns, many users also monitor the interaction between companies and their customers. 77% of users read online reviews and 75% trust these reviews more than personal recommendations (Choi et al., 2020). One source of reviews and opinions about a company is its social media profile, which is often used by smaller private companies. The question is whether these social media comments can be of value to different communities, such as private investors. Indeed, private investments like Private Equity or Venture Capital are limited in their information gathering because information on private, unlisted companies is either unavailable or incomplete. Social network content may be able to reduce the resulting information asymmetry (Retterath, 2020). Social media platforms have become an increasingly important source of information and networking opportunities for professionals in various industries, including finance. Facebook and Instagram for example can provide real-time insights into customer opinion and sentiment, which can be valuable to investors assessing potential investment opportunities. Social media can add additional data points for evaluating investment opportunities. Monitoring a target company's internet presence provides insights into customer engagement, brand perception, market trends and the competitive landscape. Sentiment analysis can help measure customer satisfaction, brand reputation and potential risks associated with the investment, such as bad publicity or help to predict revenue (Asur and Huberman, 2010). However, social media data, especially user or customer comments, are unstructured, noisy and have no claim to accuracy. Text analysis, known as Natural Language Processing (NLP), can be used to expose the additional value of these texts while also seperating informative documents from noise. In this thesis this is done by structuring comments via clustering their underlying themes, numerically represent customer opinions via sentiment, and reveal symmetries across companies. To do so a fully distributed vector space of different companies from different social media platforms is generated via word and document embeddings.

The theoretical background, similarities and differences towards other NLP concepts are drawn in chapter 4. The derived methodology is then used to apply the concept of vector spaces to social network data in chapter 5 for latent company specific topic detection via a combination of dimension reduction and dense clustering. Finally the distributed vector space is used to form Customer2Vec - a fully automatic framework for detecting social media inherent topics for a given company. Further the vector space is used in combination with sentiment analysis to develop further applications based on cosine similarities. These applications make use of the

semantic embeddings for presenting different analyses for competitor benchmarking, product mining and brand context analysis. Finally, the question whether a distributed vector space can be used to provide intuitive and meaningful analysis, from an investors point of view, of noisy social network data is answered.

2 Alternative data for alternative investments

Investors, both Private Equity and Venture Capital funds, suffer from information asymmetry regarding potential investments in private companies. Since most corporate information of unlisted companies is not published until a certain stage of the bidding process, investors suffer from this information asymmetry and have to rely on external data sources. In addition, companies tend to publish only success stories, which help them to maintain access to funding, customers and employees. This unbalanced knowledge transfer leads to problems in the study of private companies in both the academic and business ecosystem and information becomes scarce (Retterath, 2020). Other data sources, often referred to as alternative data, can therefore fill this gap by providing analysts with information about the company that it does not publish. While the typical investment sourcing process relies mainly on financial, operational and market data, such as financial metrics or customer churn rates, it is possible to gain a deeper insight by using alternative data sources alongside traditional databases (Scharfman, 2012). One example of alternative data is social media commentary. In recent years, social networks such as Facebook, Instagram, Twitter or Tik Tok have become a popular online place for sharing photos, opinions or stories (Farzindar and Inkpen, 2020). Especially business-to-customers or direct-to-customer businesses use social networks to share information about product campaigns, company milestones or other marketing related news. This offers the opportunity to keep customers up to date and allows third parties to directly investigate the business-customer relationship and patterns. While several years ago marketing research shifted attention toward share-of-wallet or purchase frequency, more and more budget is relocated to social media platforms (Kirtiş and Karahan, 2011). Especially Facebook and Instagram are used from marketers to communicate with their customers as can be seen in figure 2.1 (Stelzner, 2023). As a result, companies have expanded their social media presence for product marketing, customer relationship management or recruitment purposes. As companies increase their postings, users begin to comment on those posts, sharing their experiences with the brand, the company as a whole or specific products. This results in a large volume of unstructured *User Generated Textual Data* (UGTD). On social networks, consumers have the opportunity to interact directly with *Company Generated Textual Data* (CGTD) through likes, views and comments. This interaction creates a company-specific online ecosystem that reveals certain strategies, behaviours and opinions depending on the level of customer engagement on the platform. While conservative investment sourcing relies on the investors own network or other marketplaces, social media can enhance these channels by adding a customer perspective. Of course, not every comment is useful, but customers tend to use social media comments to tell

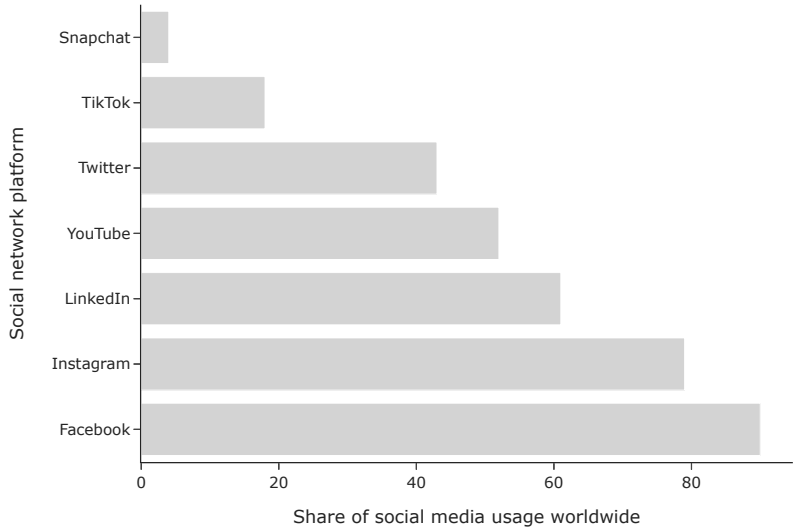


Figure 2.1: *Leading social media platforms used by marketers worldwide as of January 2023 (in %) (Stelzner, 2023).*

others or the company directly about their experiences, such as customer service, packaging, sustainability or requests for specific features. Harvesting this data can be useful for company-specific analysis, not only of social media performance, but also of operational performance, such as customer service, product quality or employee satisfaction. It is argued that knowing what is being said about the company or product and when, can be useful to either a potential investor or management. Therefore, a prospected tool that can be used to analyse this type of data seems to be beneficial. The use of this information is generally recognised as *social listening*, e.g. in Ballestar et al., 2020. The advantage of social media as an alternative data source is that it enables real-time insights into (potential) consumer opinions, wishes, complaints or suggestions, on demand and in real time. Based on this, various analyses are possible and can generate meaningful insights either for the investor or the management team for the company or even in relation to other possible targets or competitors. However, this data is highly unstructured in terms of content and quality. Nevertheless, it may be desirable to structure and analyse the data. Mining information from social media data is not new. For example, Choi et al., 2020 have developed an algorithm to support product development based on time-evolving product opportunities. Their work combines semantic analysis with an opportunity algorithm developed by Ulwick in 2009. Ulwick argues that an opportunity level can be represented numerically at a certain point in time by $Opportunity_p = Importance_p + \max(Importance_p - Satisfaction_p, 0)$, while p represents a point in time, satisfaction is calculated based on sentiment analysis and importance is measured by calculating the frequency of mentioned products. Then Choi et. al represent the product events in three clusters: products with high importance but low satisfaction (under-served), low importance but high satisfaction (over-served) and balanced

satisfaction and importance (appropriately served). Kolajo et al., 2022 also build on event detection with social media data and develop a way to encounter noisy terms, such as slangs or typos, in the pre-processing stage by introducing semantic analysis of slangs, abbreviations and acronyms. They argue that not dropping noisy terms leads to better results. The authors suggest transforming noisy terms using a local vocabulary. The translation of noisy terms via the vocabulary is performed within a context window for which a combination score is calculated and the translated term is then represented by the maximum score. Other literature looks at different modelling approaches to cluster customer opinions about events, products or companies. Ramamonjisoa, 2014, for example, examined the performance of different systematic approaches to create themes from different data sets. The literature suggests that customer feedback is essential for both sales and market growth. Since what the general public thinks of a product plays a key role in what the public perceives as potential sales success, it is important that the thoughts and feelings expressed in subsequent reviews are recognised. These reviews collectively represent the *wisdom of the crowd* and can be a very reliable indicator of potential sales success, as argued by Usmani et al., 2021. However, it may not be advisable to (i) use every comment on a post and (ii) use them at one point in time. Therefore, Customer2Vec tracks the contribution over time (horizontal) and the amount of similar contributions (vertical) at a point in time. This leads to attention and time-weighted value creation opportunities regarding either a product or the company in general, and can be used for further discussion in the investment process. The collection of UGTD and the derivation of the methodology used in Customer2Vec is explained in the following chapters.

3 Process for the acquisition of data from social network profiles

The UGTD of this thesis is obtained through a large-scale web scraping on two Lenovo machines, with each having an Intel(R)Core(TM) i5 processor and 8GB RAM. Web scraping describes the process of extracting relevant data from a website, in this thesis Facebook and Instagram, in order to store the information for further processing. Since the necessary company information is on two different websites, Facebook (facebook.com) and Instagram (instagram.com), two different web scrapers are needed to collect the data. In addition, in order to collect as much data as possible per entity, companies that are present on both social networks were selected whenever it was possible. The platforms address a unique key to each company profile. Therefore, it is necessary to identify the social network key of the respective target. The key does not necessarily have to be the company name, but can be collected via the URL of the respective company on its social media website. In the dataset, however, the company key is renamed to the actual company name, regardless of the source. Furthermore, no distinction is made between textual data from the two platforms, but they are collected in one dataset. In order to still be able to distinguish between the two platforms, a source identifier was added to the dataset to mark the origin of the documents. The scraping module works as follows. Once the target company has been identified, the unique profile name (account) must be passed to the scraper. The web scraper then searches for the company and creates *nameClasses*, a list of all the results for the search query on each platform. Once a direct match is found within the iteration of *nameClasses*, the company profile is accessed within the web browser. After that, the module scrolls as long as all the posts are collected within *URLs*, another list object that collects links for the respective posts, rather than directly accessing the network posts one by one. This allows the scraper to have information about how much of the posts are finished and which URLs are left to be scraped if the process gets terminated. While most possible data is collected, such as comments or dates, usernames are not collected for privacy reasons. An example of the data collected can be found in table 3.1. Overall, the result of the data generation process can be seen as a node-structured collection of documents for each company. Each company represents the beginning of a new spanning tree, followed by the nodes of the two platforms. Each platform node then consists of the underlying posts of the respective company, while each post in turn consists of all the comments of the respective posting. In order to control the quality of the data, some restrictions have been implemented: (i) Posts inviting the user to participate in a giveaway are not allowed, (ii) repeated comments on the same post are only stored once in the dataset and (iii) *URLs* are not allowed to be sorted by date in order to scrape data along the entire timeline of a company's social media history,

Company	Comment	Date
be quiet!	That sure is a lot of white	2019-05-13
meijer	The actions that spurred one of your stores in Michigan...	2013-12-16
Hessnatur	What washing and which cut is that in the picture?	2020-09-04
BrewDog	Is this at Cantillon or Kulminator?	2021-02-19
Hungryroot	delicious snack idea! especially for fall!	2022-03-04
meijer	Wish you were in Southern NV, I lived back east ...	2013-09-08
Birkenstock	can you check your dms :)	2020-08-28
meijer	I believe Wal-Mart is cheaper.	2016-08-28
everdrop	Great performance !!!	2020-12-04
Hungryroot	Please never take this away from us	2021-04-09
be quiet!	Looks great. Hope that is the color scheme they give away.....	2015-10-06
Fit for Free	I had written i around December last year, been ...	2021-03-23
meijer	love this place...will miss the one near me ...	2015-05-06

Table 3.1: *Sample of UGTD derived by the data collection process.*

rather than just the most recent or oldest. Since the focus of this work is to build a tool for analysing private companies with textual data, the restrictions follow this aim of bridging the gap between private investors and private companies. First, the by Customer2Vec analysed companies must not be listed on a stock exchange in any country. Furthermore, the textual data, social network comments, must be sufficient to reflect the customer behaviour of the respective company. Care has also been taken to include a variety of industries in the dataset in order to prove the concept in different domains. The UGTD set consists of 639,521 comments across 34 different companies from eight different countries. Despite the possibility to acquire this data it has to be mentioned that the data collection process is very time consuming, since the creation of the underlying dataset took approximately two months including the development of a functioning scraping tool and the scraping process itself. The selected companies can be broadly categorised into *Food*, *Fashion*, *Tech*, *Consumer Goods*, *Sports* and *Other* and sell their goods or services directly to customers. Most companies come from the fashion and tech industry and are based around shoe/footwear manufacturers such as *Birkenstock*, *Allbirds* or *Snocks* or subscription based businesses such as *Rent the Runway* or *Outfittery*. Hardware manufacturers were also included to get comments from companies whose business model is built around durable products. For example, the gaming equipment companies *be quiet!*, *Fanatec* and *Fractal*. Long-lasting products are also created by the bicycle manufacturers *CUBE* and *Simplon*. In addition, companies from the food and beverage segment have been incorporated as restaurant chains, such as *Ottos Burger*, *Hans im Glück* or *Coffee Fellows*. The supermarket chain *meijer* was also added to segment *Consumer Goods* together with *Bugaboo*, *mymuesli* and *Getaround*. Finally, an industry segment called *Other* is created, which contains the household cleaning company *everdrop*, the online bank *N26* and the skincare clinic *Face Reality Skincare*. A complete listing is provided in Appendix A. Figure 3.1 shows the timely distribution of the companies categorized by their industry. The illustration also provides information about the bad characteristics that are associated with social media and possible difficulties of modeling

so. The comment count for some companies is much higher as for others leading to a highly unbalanced dataset. Further not every company has observations in every year since the creation of the social network profile is of course decisive for the starting point.

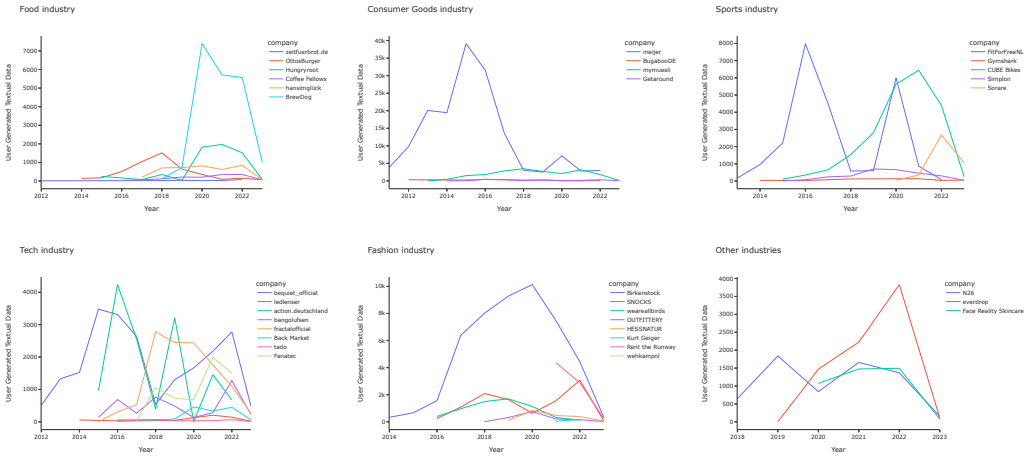


Figure 3.1: *Overview of UGTD per company and industry.*

The comment count for some companies is much higher as for others leading to a highly unbalanced dataset. Further not every company has observations in every year since the creation of the social network profile is of course decisive for the starting point. In the further proceedings of the work the impact of these irreglurarities will be discussed. This unbalance was deliberately maintained in order to assess its influence on later NLP concepts. This goes hand in hand with the requirement to develop a tool for practical application, since in reality it is very likely that a different number of UGTD will be found in the analysis of different companies. Additionally under practical considerations it would be inconsequential to remove information in favour of more balanced data since the aim of Customer2Vec is first and foremost the complete analysis of the social media profile of a given company.

4 Derivation of numerical representations of textual data behind Customer2Vec

The backbone of Customer2Vec are techniques from the area of NLP. NLP describes the process of gathering insights of e.g. unstructured textual data stored in a corpus C consisting of n documents. The textual data can be of any kind and of any context, such as annual reports, news articles or online reviews. In this thesis NLP is applied to social media comments from company profiles in order to structure the fuzzy data and create a distributed representation of such to make conclusions about discussed topics, products and brand context. Social media in general is a popular tool to which NLP is applied, such as in Jeong et al., 2019, Kolajo et al., 2022, Conway et al., 2019 or Bail, 2016 which are independent of the investment context in this thesis but are also about product planning, healthcare or advocacy. Regardless of the concrete application or intuition of applying NLP onto textual data the main prerequisite is to translate C into a numerical representation, more precise in a vector based representation of words w and documents d. The main idea thereby is to achieve that the numerical presentation of a given document is similar to documents which include the same words, or in more advanced algorithms even contexts. Since in general machines are not able to handle raw textual data it is necessary to transform it into a numerical representation. By doing so it is necessary to (i) store the document-word information and (ii) allow for drawings about the relatedness of two or more documents. Various techniques for this desired output are explained in the following and the used method within Customer2Vec is derived.

4.1 High dimensional naive encoding

From a human perspective it is quite comprehensible that the comparison between different documents is made by the words each document includes. The greater the intersection of shared words the more similar the vector representation should be. The result is then a document-term matrix $D \in \mathbb{N}^{n \times m}$. The probably most intuitive way to transform C into D is by creating the word per document distribution to quantify the existence of m words w in n documents d, such that:

$$b_{n,m} = \begin{cases} +1, & \text{if } w \text{ in } d \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Aggregating the vectors towards a matrix yields a document-term matrix D of the corpus such as:

$$\begin{pmatrix} & word_1 & word_2 & ... & word_m \\ document_1 & b_{1,1} & b_{1,2} & b_{1,...} & b_{1,m} \\ document_2 & b_{2,1} & b_{2,2} & b_{2,...} & b_{2,m} \\ ... & b_{...,1} & b_{...,2} & b_{...,...} & b_{...,m} \\ document_n & b_{n,1} & b_{n,2} & b_{n,...} & b_{n,m} \end{pmatrix}$$

As can be seen in Eq. (1) a convenient way to quantify the existence of a word in a document is the summation over a boolean representation of its existence. This approach leads to the fact that only individual words and their frequencies are considered during the vector creation (Kanakaraj and Guddeti, 2015). Consequently any sequential structure is eliminated and only the pure word frequency per document is of interest. Following this logic each document can be presented as a vector of length $\mathbb{N}^m$ (Akuma et al., 2022). This counting based algorithm and representation of textual data is generally known as Bag of Words (BOW). [1]

Since the BOW representation of C does not take into account the importance of a word given a document the *Term Frequency - Inverse Document Frequency* (TF-IDF) method introduces a measure of importance while creating the numerical representation of the corpus (Alzami et al., 2020). The TF-IDF represents the frequency calculated as weight of w in document d given all other words and is calculated by:

$$TF - IDF_{w,d} = TF_{w,d} * log\frac{n}{DF_w} \tag{2}$$

While TF describes the number of occurrences of term w in document d, n is the number of documents and DF is the number of documents containing word w. Each $TF - IDF_{w,d}$ is then replacing $b_{n,m}$ in D at the corresponding position.[2] Concluding naive encoding techniques

[1] An intuitive example can be drawn by letting the corpus C consist of the two documents 'Your shoes are comfy but they are expensive' and 'Discount for your comfy shoes?'. Then the vocabulary V consists of 9 words: Your, shoes, are, comfy, but, they, expensive, discount, for. By applying the BOW method in Eq. (1) C can be represented by D, such that:

$$\text{D} = \begin{pmatrix} & are & but & comfy & discount & expensive & for & shoes & they & your \\ document_1 & 2 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ document_2 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

[2] The example from above turns then into:

$$\text{D} = \begin{pmatrix} & are & but & comfy & discount & expensive & for & shoes & they & your \\ document_1 & 0.69 & 0.34 & 0.24 & 0.00 & 0.34 & 0.00 & 0.24 & 0.34 & 0.24 \\ document_2 & 0.00 & 0.00 & 0.38 & 0.53 & 0.00 & 0.53 & 0.38 & 0.00 & 0.38 \end{pmatrix}$$

aggregate the occurrences based on either Eq. (1) or Eq. (2) with the aim of creating D. Since $D \in \mathbb{N}^{n \times m}$ the document-term matrix typically yields a high dimensional numerical representation of textual data. The underlying UGTD set has 639,521 documents with 6,024 unique words leading to a matrix of 639,521 × 6,024. However BOW and TF-IDF are well known and popular word embeddings especially if a semantic meaning is not of interest.

4.2 Semantic word and document embeddings

The disadvantages of naive encoding models, such as BOW or TF-IDF, are that they do not capture contextual information based on word position in the text. Furthermore, the algorithms are not able to capture semantic meanings or word co-occurrences. Semantic word and document embeddings overcome these limitations by introducing a variety of numerical representations through the application of neural networks. The idea behind creating semantic word and document embeddings is that similar texts, either words or documents, should be represented by vectors that have a small distance to each other in an arbitrary space, even if the intersection of words is empty but the context is similar. Further, word vectors $\vec{w}$ should be close to document vectors $\vec{d}$ which they describe best (Angelov, 2020). Very popular implementations for this are called *Word to Vector* (Word2Vec) and *Document to Vector* (Doc2Vec) models which are discussed below.

Efficient Estimation of Word Representations in Vector Space by Mikolov et al., 2013

In order to achieve a numerical representation $\vec{V}$ consisting of word vectors $\vec{w}$ for a given Vocabulary V, the Word2Vec model allows different algorithms, e.g. the skip-gram implementation of the model proposed by Mikolov et al., 2013. The skip-gram model of Word2Vec embeddings uses each word and tries to predict words within a certain context window of size $\delta \in \mathbb{N}$, as outlined in the left figure of illustration 4.1 by applying the toy example from above with $\delta = 2$. By solving this supervised problem the model learns the context in which a particular word appears, while the context and the word have a similar vector. Since the underlying model is a neural network, each of its neuron helps to ensure that the numerical representation of w captures several contexts. This yields a distributed representation of each word and solves the restrictions of conventional word encodings (Angelov, 2020). The prediction model is a log-linear classifier, for faster training and creation of the word embeddings. The input layer consists of m words, which are encoded using 1-of-m coding, where m is the size of the vocabulary. The encoded input data is then processed through a feedforward neural network consisting of input, projection and output layers. While the input layer is a word distribution as explained above, the projection layer is constructed as a matrix with dimensionality $n \times$ *vector size*, while the *vector size* is a parameter to be tuned and describes the resulting dimensionality of each $\vec{w} \in \vec{V}$. Then the introduced skip-gram model tries to maximize classification of a word based on another word in the same sentence. So each word which is currently looked at, the current word, is used as an input to a log-linear classifier with continuous projection layer for predicting the context words, which are words next to the current word within a window of size δ. Additionally a

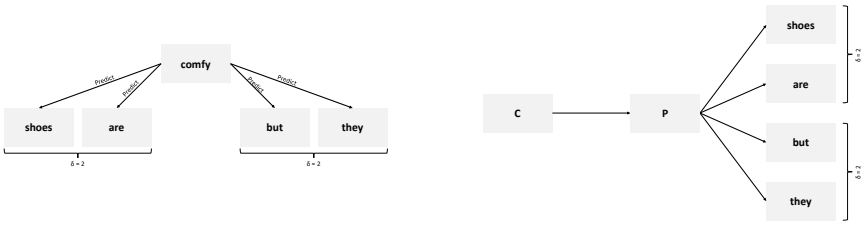


Figure 4.1: *Illustration of the skip-gram and DBOW implementations of Word2Vec and Doc2Vec.*

weight scheme is introduced resulting in higher weights to words closer to the context words and vice versa. While the output of the neural network is actually less of interest the weights within the hidden layer represent the word vectors $\vec{V}$ for the given vocabulary V.

Distributed Representations of Sentences and Documents by Le and Mikolov, 2014

Besides generating $\vec{V}$, it is also possible to transform a complete corpus C into semantic numerical representations $\vec{C}$ by extending proposed the Word2Vec model with document vectors $\vec{d} \; \forall \; d \in C$. An example is the Distributed Bag of Words (DBOW) version of Doc2Vec for generating document embeddings. The DBOW method uses $\vec{d}$ to predict words within a context window in the document. Angelov, 2020 suggested that the DBOW method produces better results than, for example, the Distributed Memory (DM) approach. The numerical representation of documents via DBOW, in this work social media comments, is achieved by constructing predictions of the next word given many contexts sampled from the document. In other words, while the prediction input for word embeddings $\vec{w}$ is represented by words, the Doc2Vec algorithm augments the word vectors with an additional numerical representation of the comment $\vec{d}$. While the distributed memory model uses a concatenation of the document vector with the word vectors, DBOW predicts randomly sampled words from the document. Thus, at each iteration of stochastic gradient descent within the neural network, a text window δ and a random word from the text window are sampled and formed into a classification task given the document vector. Next to δ and *vector size* the Doc2Vec model also requires a *sub-sampling threshold* and the related parameter *minimum count*. While the *sub-sampling threshold* determines the probability of removing high frequency words from the given context window with higher values indicating a lower probability of a high frequency word to be removed, the *minimum count* removes words with a frequency below the value at all (Angelov, 2020). The resulting vectors $\vec{V}$ and $\vec{C}$ can be used to draw conclusions about semantic meanings, relatedness between documents and/or words. The combination of Doc2Vec and Word2Vec results in a vector space that translates the given textual data into numerical representations in which similar words are close to each other and words are close to the documents that they best describe in vector space. To measure this distance, cosine similarity is a popular choice.

Cosine Similarity Cosine similarity S_C is a general concept to determine the distance between vectors, in theory the closer vectors are the higher the cosine similarity. This means that the closer a sample word vector $\vec{w}_1$ is to $\vec{w}_2$, the higher the cosine similarity score. Following the logic of the creation of $\vec{C}$ and $\vec{V}$, a higher cosine similarity means that the words and/or documents are either in the same context, have a similar meaning, or both. The cosine similarity S_C can be calculated as:

$$S_C(A, B) := cos = \frac{A \times B}{\|A\| \, \|B\|} = \frac{\sum_{i=1}^{v} A_i B_i}{\sqrt{\sum_{i=1}^{v} A_i^2} \sqrt{\sum_{i=1}^{v} B_i^2}} \tag{3}$$

Where v is the *vector size* and A and B are the representatives for the input which can be e. g. a word vector $\vec{w}$ or a document vector $\vec{d}$.

Finally, numerical representations, either by high-dimensional naive encoding or by semantic embeddings, of the corpus can then be used for further processing by inferring relatedness based on distance measures, such as cosine similarities. In addition, the resulting vector space can then be used for deeper analysis in more advanced NLP concepts, such as topic modeling or text classification tasks, see Jipeng et al., 2019 or Chen and Sokolova, 2018. Therefore, the desired word and document representation within Customer2Vec must be able to capture semantic relationships in order to take advantage of them.

4.3 Topic modeling

When analysing large corpora, such as the text of company profiles on social media, it is quite challenging to draw conclusions about the inherent topics T being discussed. This is where topic modeling comes in, revealing the underlying structure and latent factors in the textual data. In general, topic modeling allows elements to be described by common latent variables within an increasing size of the underlying dataset without compromising statistical relationships necessary for further processing with more or less power of meaningful dimension reduction (Vayansky and Kumar, 2020). By passing textual data into topic models, the algorithms reveal what concepts, events or categories are being discussed (Kherwa and Bansal, 2018). Considering the size of the underlying dataset topic modeling is a handy tool to gain insights into inherent topics and is therefore an important component within Customer2Vec. The advantage, besides the procedure being automatic, is that no prior human annotation, labelling or hand-coding is required. Only the number of topics k has to be defined in advance for some implementations. The found topics then represent the most repetitive words that are used whenever the found topic is discussed in the given data (Mohr and Bogdanov, 2013). The task of identifying the underlying topics of given textual data can be solved by different algorithms, which differ in complexity, parameters and numerical representation of the text, as discussed in the previous chapter. This chapter gives an overview of different topic modeling algorithms and derives the used model within Customer2Vec. Further it is argued why the chosen kind of topic modeling is best suited for social network data.

4.3.1 Documents as latent mixture of topics

Well-known and conventional models assume that each document is a mixture of latent topics. These types of topic models assume that observed variables interact with latent parameters in a specific probabilistic relationship, which then generates the textual data (Vayansky and Kumar, 2020). Once a numerical representation is obtained, e.g. via BOW (Eq. (1)) or TF-IDF (Eq. (2)), several models decompose D by approximation the matrix into a document-topic U and a topic-term matrix W such that

$$D \approx UW' \tag{4}$$

While U has the dimension of $n \times k$ and W has the dimension of $m \times k$. The parameter k has to be set in advance for conventional topic algorithms and determines the number of topics the model should end up with. Different algorithms share the input of a given document term matrix D, but their convergence to a solution of Eq. (4) differs. Two prominent examples are *Latent Semantic Analysis* and *Latent Dirichlet Allocation*.

Latent Semantic Analysis by Deerwester et al., 1990 *Latent Semantic Analysis* (LSA) overcomes the sparse and high-dimensional characteristic of D by creating a lower dimensional representation of it. This is done via the introduction of a third matrix Σ, such that:

$$D \approx Q\Sigma P' \tag{5}$$

Q is created by dividing the $r-th$ column of U by $||U_{\cdot}r||$, P is created by dividing the $r-th$ column of W by $||W_{\cdot}r||$ while $r \in 1, ..., k$. Σ is a $n \times m$ matrix whose main diagonal values are of interest which consist of the product $||U_{\cdot}r|| \times ||W_{\cdot}r||$. However, only the main diagonal of Σ is of interest, since it represents the singular values of D and is strictly positive by construction. Therefore the LSA algorithm selects the k largest singular values of D and discards the remaining columns of Q and P, so that the rank of the approximation of D is k. In general, the matrix factorisation in Eq. (5) is called *Singular Value Decomposition* (SVD). The matrix factorisation via SVD is used to find a linear subspace within the space of the original matrix D that describes the majority of the variation in the corpus (Vayansky and Kumar, 2020). After applying the SVD, Q, P and Σ have dimensions of $n \times k$, $k \times m$ and $k \times k$. Finally after the disposal of the remaining columns Q and P can be interpreted as U and W in Eq. (4). One major drawback is that LSA does not respect the semantic meaning of words and documents due to the matrix element wise representation of words and documents and therefore would require extensive pre-processing of the textual data. Further the number of topics, represented by k, has to be known ex ante. Additionally LSA emphasized weak performance related to short text applications, such as UGTD, as in Albalawi et al., 2020, due to the assumption that each document consists of several topics.

Latent Dirichlet Allocation by Jeong et al., 2003 In many related works the topic modeling method Latent Dirichlet Allocation (LDA) algorithm can be found, for example in Kolajo et al., 2022, Jeong et al., 2019 or Ramamonjisoa, 2014. LDA is a generative probabilistic model for collections of, for example, textual data. The underlying assumption of the model is that each document is constructed by a mixture of an underlying set of topic probabilities for each document and each topic is characterised by a probability distribution of words. This is done by assuming that the corpus is generated via the Dirichlet distribution and that each document is a mixture of topics with the presence of words according to a probability in each document. The use of Dirichlet priors is given by the famous plate notation in figure 4.2. m_d is the number

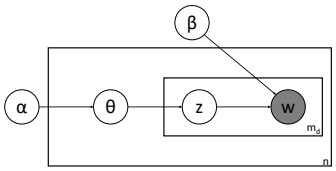


Figure 4.2: *LDA model represented in plate notation.*

of words in a document and n is the number of documents. From the left, α is the Dirichlet parameter that distributes k topics over $d \in C$. α has the dimension $n \times k$ and determines the probability that a document contains a given topic. From α the topic distribution θ for d is drawn so that $\theta \sim \text{Dirichlet}(\alpha)$. Finally, θ is used to select topic z, while $z \in 1, ..., k$. To obtain the topic-word matrix β, a sample is chosen that represents the word distribution given the topic z. From this distribution the word (marked in grey), that is the only observable variable in the model, is selected. In other words, LDA assumes that the creation of a document was generated by introducing a set of topics along with sets of words within each topic. The algorithm then reverses this process to find latent topics. Since θ and z are unknown parameters, the formulation of LDA can be expressed by:

$$P(\theta, z|d, \alpha, \beta) = \frac{\theta, z, d|\alpha, \beta}{P(d|\alpha, \beta)} \tag{6}$$

The equation is approximated by the Kullback-Leibler divergence between the approximation and $P(\theta, z|d, \alpha, \beta)$. By computing $P(z|d, \alpha, \beta)$, the result can be interpreted as the document-topic matrix U, while each entry of β corresponds to $P(w|z)$ and can be interpreted as the topic-term matrix W in Eq. (4). However, LDA has a disadvantage similar to LSA, at least in terms of brief social media comments, that it assumes that a document contains more than one topic. In addition, latent mixture models do not benefit from the semantic vector space created by Doc2Vec, as the inputs are based on BOW or TF-IDF even though there are attempts to solve for this behaviour as in Blair et al., 2019.

4.3.2 Creation of topics out of distributed representation $\vec{C}$

Traditional topic modeling methods, as outlined above, typically describe documents as a mixture of topics, which suffers from a potential bias in the determination of k. Furthermore, they do not allow for the comparison of cross-entity topics, such as similarities or other peculiarities. By using document embeddings $\vec{C}$ and word embeddings $\vec{V}$ for topic detection, it is possible to (i) create topics without any assumption about the set and (ii) create topic vectors $\vec{t}$ that can be used for cross-entity comparisons. An implementation of how topics can be created from distributed representations of C and V is the Top2Vec model of Angelov, 2020. Angelov proposes to create a distributed vector space consisting of $\vec{C}$ and $\vec{V}$ to compute $\vec{t}$, which is an aggregated form of a given set consisting of different $\vec{d} \in \vec{C}$ described by the closest words $\vec{w}$ that are a subset of $\vec{V}$, measured by the cosine similarity of $\vec{t}$ and $\vec{w} \in \vec{V}$. To achieve this goal, the Top2Vec model first reduces the dimensionality of $\vec{C}$ and then applies a clustering algorithm to group documents that share the same underlying latent topic t C_t. The centroid of $\vec{C_t}$ is then used to retrieve $\vec{t}$ a vector that represents the underlying topic of all documents in C_t. While the model uses *Uniform Manifold Approximation and Projection* for dimension reduction, the clusters are computed by *Hierarchical Density-based Spatial Clustering of Applications with Noise*.

Uniform Manifold Approximation and Projection by McInnes et al., 2018 The advantages of embeddings, which are able to capture the semantic meaning of a word and several linguistic regularities such as analogy relations, come with the disadvantage of having a very high dimensionality, the *vector size*, in order to reflect all the specific characteristics of a given word (Raunak et al., 2019). Because of this high dimensionality, a clustering algorithm would result in a large computational effort due to the curse of dimensionality (Ding, 2009). This is due to the fact that clustering algorithms depend on some kind of distance measure, e.g. the Euclidean distance as used in the Top2Vec model. However, distance measures show strange behaviour in high-dimensional space, as argued for example by Aggarwal et al., 2001. Therefore, dimension reduction is a popular preprocessing step before applying clustering algorithms. Angelov's Top2Vec implementation uses the *Uniform Manifold Approximation and Projection* (UMAP) algorithm for this downstreaming task. UMAP works by constructing an intermediate topological representation of the approximate manifold $\mathcal{M}$, which is simplified to a weighted graph. An advantage of the UMAP dimension reduction technique is its ability to preserve the local and global structure of high-dimensional data, unlike *Principal Component Analysis* which can only store linear information across dimensions. The UMAP algorithm constructs a high-dimensional graph from the given data, the manifold $\mathcal{M}$, and optimises a lower-dimensional graph that aims to be as similar as possible to the original, using cross-entropy as a loss function. The high-dimensional graph is constructed from a weighted graph where edges represent the probability that two points are connected in any space. To handle these connections, UMAP allows several parameters. The simplest input parameter is the dimensionality γ of the lower dimensional representation of $\mathcal{M}$, which is quite similar to many other dimension reduction techniques. Another parameter determines how $\mathcal{M}$ is approximated. The number of neighbours ϕ balances the focus of the algorithm between local and global structures. The smaller ϕ,

the more constrained the neighbourhood UMAP uses by approximating the manifold of the underlying data.

Hierarchical Density Clustering by Campello et al., 2013 For clustering, Top2Vec uses the *Hierarchical Density-based Spatial Clustering of Applications with Noise* (HDBSCAN) algorithm, more precisely the accelerated version of HDBSCAN by McInnes and Healy, 2017. HDBSCAN has an important and favourable behaviour towards the underlying data. The algorithm aims to build clusters of higher density by increasing the quality of such clusters by assigning noise to a single cluster. As social media texts are by nature very unstructured, short and noisy, this clustering behaviour lead to more interpretable results later in the process of inherent topic detection, so the assumption. However, for clustering, HDBSCAN relies on a single linkage and creates the distance matrix. Then a density estimate is made from the distance matrix by defining the core distance $core_e(x)$ for parameter e for a given point x, where e determines the number of nearest neighbours to point x.[3] In other words, $core_e(x)$ measures the distance from x to the $e-th$ nearest point, including x, leading to the radius with point x as the centre, including $e-1$ points. Then $core_e(x)$ is used to calculate the *mutual reachability distance* $d_{mreach-e}$, which is defined as

$$d_{mreach-e}(a,b) = max[core_e(a), core_e(b), d(a,b)], \tag{7}$$

where $d(a,b)$ is the original distance between points a and b from the distance matrix. By applying $d_{mreach-e}$ to the embeddings, low core distances (dense points) remain at the same distance from each other as in the distance matrix, but sparser points, i.e. high core distances, are pushed away to be at least their core distance from any other point. Once the mutual reachability metric is defined, a graph is drawn with the data points as vertices and the edges weighted according to $d_{mreach-e}$. Then a threshold is applied, disconnecting vertices whose edges are below the threshold. The resulting spanning tree is constructed so that the edge with the lowest weight is always used to connect the vertices. Once the spanning tree is created, the hierarchical clustering is done by sorting the edges of the tree by their distances in increasing order. Then the clusters are created by merging for each edge. However this results in a cluster hierarchy. To create flat clusters HDBSCAN condenses the created cluster hierarchy into a smaller tree with more data attached to each node instead of using single-linkage. At this step the parameter *minimum cluster size* becomes important. Whenever a group is split, it is checked whether any of the resulting groups have fewer points than the *minimum cluster size*. The answer to the question if *points in the cluster* > *minimum cluster size* reveals if the cluster is really considered as a cluster within HDBSCAN or if they just fall out of the cluster building. The result is a smaller number of nodes. From the reduced set of nodes finally the clusters are extracted by measuring the $stability$ of a cluster which can be seen as the $lifetime$ along the reduction of nodes via the *minimum cluster size*. The higher the $stability$ of a cluster is the less it is splitted via the node reduction according to the *minimum cluster size* and the more likely the cluster is extracted as a cluster under HDBSCAN. The data not included in the resulting

[3] Note that in the original paper e is actually k which is already used for the number of topics.

clusters are then seen as noise (McInnes et al., 2017). After the application of HDBSCAN the cluster labels are used to sort each d to the corresponding C_t while t is the label of the cluster reaching from 0 to the number of extracted clusters. The detected noise is captured in the cluster label -1 and is not used for topic modeling within Top2Vec. $\vec{t}$ is then created by calculating the centroid of the respective $\vec{C}_t$.

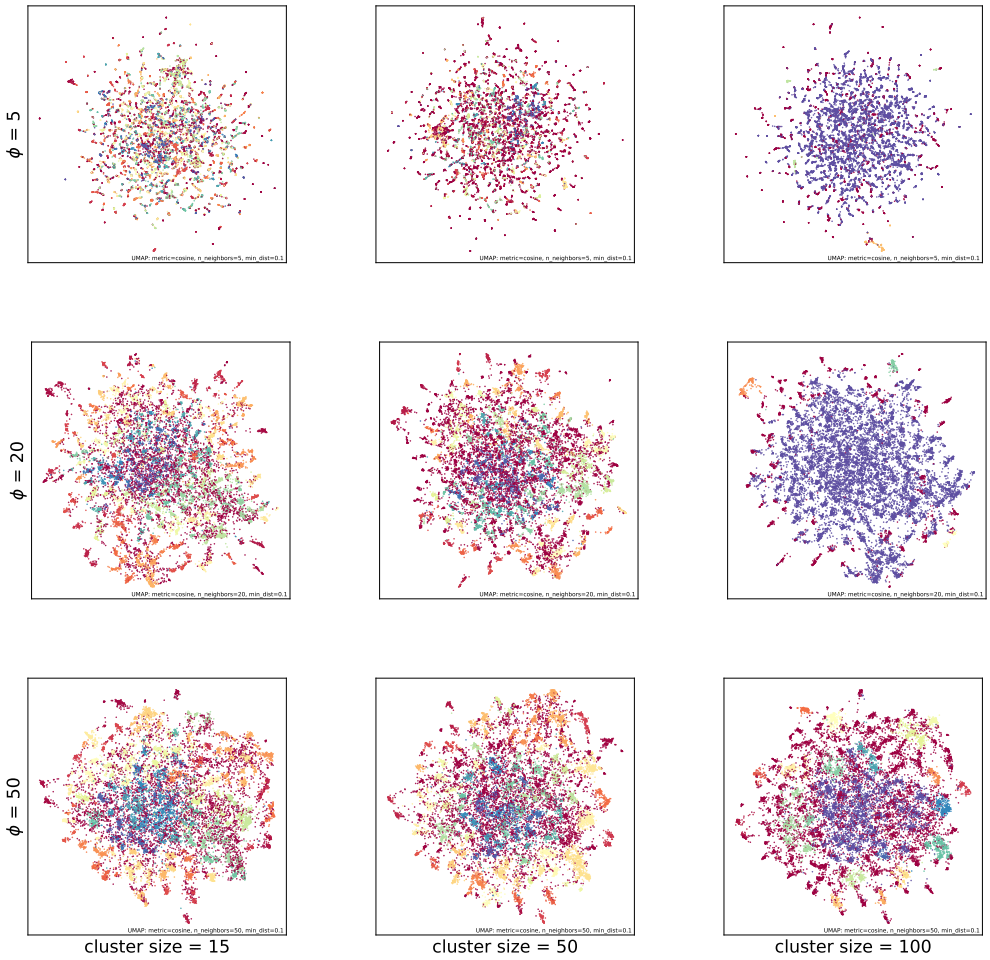


Figure 4.3: *Illustration of the relationship between UMAP and HDBSCAN, using the example of the company BrewDog, with a fixed value for gamma and varying values for ϕ and minimum cluster size.*

UMAP and HDBSCAN form the downstreaming task towards the topic creation for the given textual data within Top2Vec. What is crucial here is the interaction of the parameters in each model (Angelov, 2020). While the impact of γ is quiet obvious, meaning that the lower the value the lower is the dimension, ϕ and the *minimum cluster size* and its relationship are less trivial. In order to demonstrate the interplay of both parameters the comment observations of

the company *BrewDog*, a UK based brewery, were translated into a numerical representation via Doc2Vec with a vector size of 300 and 400 epochs within the neural network. Then the matrix was reduced to a two dimensional dataframe, e. g. $\gamma = 2$, with UMAP and a varying set for ϕ of [5,20,50] while the *minimum cluster size* takes the values of either 15, 50 or 100. The example in figure 4.3 demonstrates that small values for the *minimum cluster size* together with a small value for ϕ result in the highest amount of clusters and would therefore also mean that the number of topics would also be highest and vice versa, indicated by the colors. The trade-off between a reasonable and interpretable number of topics that are meaningful to the user lies in the parameter choice between *min cluster size* and ϕ. Figure 4.3 also shows how the noise reduction of HDBSCAN changes when the parameter selection is changed. The detected noise is marked dark red in the illustration and varies across the different parameter settings. The detected noise and therefore the data that is not used for clustering seems to increase, at least for the example of *BrewDog* with a higher value of the *minimum cluster size* and a more global view, e. g. with a high value for ϕ.

Detection of topic words In order to interpret the topics and their quality for a human being, it is quiet convenient to look at the topic words $w*$, meaning those words which are most representative for the topic. Again the fact that the corpus is represented as distributed vectors representing topics, documents and words the identification of topic words is quiet simple. The approach to identify the most important words per topic, topic word sets $V*$ consisting of individual topic words $w*$, follows directly the construction of the embedding space of documents and words. Since words are closest to the documents in which they occur most and topics are the centroid of each document cluster, it follows that words with the lowest distance to the topic vector are the most relevant topic words. This gives rise to apply the concept of cosine similarity demonstrated in Eq. (3) and since the vectors are normalized in the process of the embedding creation it follows:

$$S_C(\vec{w}, \vec{t}) := cos = \frac{\vec{w} \times \vec{t}}{\|\vec{w}\| \left\|\vec{t}\right\|} = \frac{\vec{w} \times \vec{t}}{\sqrt{1}\sqrt{1}}, \tag{8}$$

which leaves the inner product of $\vec{w}$ and $\vec{t}$:

$$S_C(\vec{w}, \vec{t}) := cos = \frac{\vec{w} \times \vec{t}}{\|\vec{w}\| \left\|\vec{t}\right\|} = \vec{w} \times \vec{t}. \tag{9}$$

Since now $V*$ per topic t are identified it is necessary to assign each document the corresponding topic. This again can be done via changing Eq. (3):

$$S_C(d, t) := cos = \frac{\vec{d} \times \vec{t}}{\|d\| \, \|t\|} = \vec{d} \times \vec{t}. \tag{10}$$

Angelov uses Eq. (10) rather then resorting the documents by their label classes in order to assign topics also to the by HDBSCAN as noise detected document vectors.

Overall distributed representations of topics have several advantages, as outlined above. Firstly, no ex ante determination of topic sizes is necessary, which allows for a higher usability and faster application of the framework, since no prior information is required. Further topics are created as centroids of the document clusters created by HDBSCAN. This has the advantage that topics are directly linked to the documents around them, rather than being created by a probability distribution as in LSA. Furthermore, the number of topics can be directly derived from the creation of centroids (Angelov, 2020). In addition, the fact that documents, their topics and words are created within a consistent model, i.e. having the same dimension, makes it possible to draw conclusions about the relatedness between the vectors, but also among each other. Additionally numerical semantic representations of d and w by $\vec{d}$ and $\vec{w}$ show also a higher performance according to literature, such as in Baroni et al., 2014 and Top2Vec overall outperforms other topic models in Angelov, 2020. Therefore the used topic model inside Customer2Vec is Angelovs Top2Vec approach.

Topic Information Gain Since the creation of topics from a given corpus is an unsupervised task, performance measures based, on matching the found topics with the true inherent topics, cannot be directly applied. It is also conceivable to use, for example, document or word embeddings for classification tasks. However, in most cases the result has no underlying truth to which it could be compared, unless the user performs a manual data labelling of each document. However, this would make the automated topic creation approach redundant, as the document and word embeddings could then be used again for classification, with the manually labelled topics as the dependent variable. The reason why conventional topic modeling metrics, such as topic coherence, cannot be used for evaluation lies in the chosen continuous representation of topics in which documents are placed in a space to their corresponding topics rather than describing the documents as a mixture of topics. A solution to this problem, or a compromise of sorts, might be a metric that describes how informative the topics are to a human user, as in Angelov, 2020. The author suggests using the concept of *mutual information*, explained in Cover and Thomas, 2005 which is also the basis for the following derivation. Applied to topic modeling, mutual information describes a scoring of how well the detected topics T describe the documents C. For the scoring function, C is divided into the different sets C_t with $t \in T$, where each set corresponds to the document vectors with the same nearest topic vector $\vec{t}$. Then each topic is scored by the corresponding subset of C. Angelov argues that a topic vector represents the average topic of the individual topics of a group of documents close to each other in an arbitrary space. Thus, to evaluate the topic generated by the Top2Vec model, it is necessary to evaluate the unique documents whose average numerical embedding is used to generate the corresponding topic vector. This measurement is based on the concept of information theory. Applied to the inherent information of topics, words and documents, a given word w and a given document d can be seen as two distinct events from finite event spaces V and C with an assumed joint probability distribution $P(d, w)$. From the assumption that $P(d, w)$ exists, it follows:

$$P(d) = \sum_{w \in V} P(d, w) \tag{11}$$

and

$$P(w) = \sum_{d \in C} P(d, w). \tag{12}$$

Eq. (11) and Eq. (12) imply that when w/d is observed, it is always observed together with $d \in C/w \in V$. This is quite intuitive, since in this context words without documents or documents without words cannot be observed. Then the individual and joint probabilities can be used to calculate the *pairwise mutual information* (PMI) between w and d, which measures the difference between the amounts of information based on the actual observed probability $P(d, w)$ and the expected probability, assuming the independence of w and d measured by Eq. (11) $\times$ Eq. (12), and is given by

$$PMI(d, w) = log \frac{P(d, w)}{P(d)P(w)}. \tag{13}$$

Given the construction of the topic space, i.e. that the continuous representation of topics is given by subsets of documents belonging to each topic, any total information gain for each subset must be measured when described by the words closest to the corresponding topic vector. Measuring the difference in information gain between words and documents is therefore less important than their interaction over all possible events. This is done by weighting the PMI by the joint probability $P(d, w)$, resulting in *probability-weighted amount of information* (PWI), given by

$$PWI(d, w) = P(d, w) log \frac{P(d, w)}{P(d)P(w)} \tag{14}$$

As explained above, topics are represented by their corresponding topic words, which reflect the nearest word vectors $\vec{w}$ to the corresponding topic vector $\vec{t}$. Therefore, PWI has to be applied to the topic word sets $V*_t \in V$ and the document subsets $C_t \in C$, resulting in PWI(T). This gives

$$PWI(T) = \sum_{t \in T} \sum_{d \in C_t} \sum_{w* \in V*_t} P(d, w) log \frac{P(d, w)}{P(d)P(w))}. \tag{15}$$

Note that the term $\sum_{d \in C_t} \sum_{w* \in V*_t} P(d, w) log \frac{P(d,w)}{P(d)P(w))}$ can be expressed as the PMI of every possible document-word combination weighted by its probability $P(d, w)$. Now, to reflect that not every word is used within each topic document, but rather the corresponding topic words, the probability of topic word $w*$ is denoted as $P(w*)$ and is used to calculate the information gained about document d given topic word $w*$, such that:

$$P(d, w) = P(d|w) \times P(w*) \tag{16}$$

Finally the measure calculates the information gained about each document d given the corresponding topic words $V*_t$ as a prior. Since $w* \in V*_t \in V*$ is given it follows that $P(w*) = 1$, which leaves $P(d, w) = P(d|w)$. This leads to

$$PWI(T) = \sum_{t \in T} \sum_{d \in D_t} \sum_{w \in W_t} P(d|w) log \frac{P(d, w)}{P(d)P(w))}. \tag{17}$$

Angelov further argues that it is beneficial for the usefulness of the model not to evaluate all words related to a topic given the distribution, but to use only use a set of the first top words. Each top word-document co-occurrence per topic, weighted by probability, is then summed up and the desired PWI of the entire topic space PWI(T) is obtained.

5 Creation of distributed vector space from social network data for topic detection

The previous chapter explained the theory behind the distributed representation of documents and the creation of the topic vectors $\vec{t}$ used in Customer2Vec, arguing that a distributed representation of social media data that allows for semantic relationships is most appropriate for its development. Further a metric, the PWI, was introduced in order to evaluate the topic model. This chapter focuses on the concrete translation of the UGTD set into the Top2Vec framework from Angelov, 2020. The result is ultimately the desired distributed vector space of the underlying dataset from which latent topics are derived. It is argued that from an investor's point of view it could be very valuable to get an aggregated overview of what is most frequently discussed on the social media site of a potential target company or group of companies. The used approach is very unbiased towards the analyst's perception of the company and shows what is really being said in the network. Therefore, a good topic model should make the use of topic classification redundant in terms of easily interpretable topics. At this point, it is important to distinguish between the complete and the company-specific corpora. In the following, whenever vectors are related to the whole corpus, it is referred to *global embeddings*, while company-specific corpora and their vectors belong to the category of *local embeddings*. Due to the use of Doc2Vec for embedding, UMAP for dimension reduction and HDBSCAN for dense clustering, many parameters have to be set along the topic detection. The following section considers the most important and sensitive aspects and derives recommendations for the application by calculating corpus statistics, performing simulations and referring to the literature. Furthermore, the applicability of Angelov's original parameter setting to social media data is discussed. The remaining part of the thesis emphasises the implementation of Customer2Vec and comments on its performance in terms of usability, interpretable results and finally its contribution to the information asymmetry between companies and investors. It should also be noted that the underlying data is not exclusive to the approaches and results used, but rather serves as a large-scale proof of concept. The final tool and its applications should be applicable to any type of company that has a public social network profile of the sources used. Further implications on the parameter settings of each technique are discussed and for each application exemplary companies are used and their results are presented.

5.1 Pre-processing strategy for social media data

For most NLP applications, text pre-processing is a necessary step to improve the performance of the underlying tasks. Pre-processing is basically *denoising* the textual data by, for example, removing stop words or common words, such as *you* or *the*. However, the proposed framework of distributed representation of textual data is able to perform the analysis later in this thesis without any pre-processing of the traditional kind. This is due to the nature of the creation of word, document and topic vectors. Since common words appear in most documents, their word vectors $\vec{V}$ tend to be in a region of semantic space that is equally distant from all documents, as opposed by Angelov, 2020. Furthermore, since topics are interpreted by their topic words, words that are actually more specific to the topic should be closer to the corresponding $\vec{t}$ than common words. Therefore, Angelov argues that removing stop words is redundant. This desired behaviour was also investigated in the dataset for its nine most frequent common words. The cosine similarity between the corresponding word vectors and all document vectors was calculated, resulting in the distribution plots in Appendix B. The results show that most of the cosine similarities have a range around zero with no tendency towards tails, indicating a normal distribution of the distance of the common words to the documents. This is in line with Angelov's observations, making the removal of common words no longer necessary for the UGTD set. However, social media data is very unstructured, including slang or names of users that may be worth removing. The mere existence of such is not a problem for a functioning modelling of the data, but it disturbs the human interpretability, e. g. when one $w*$ is a name or unknown slang. Therefore, a pre-processing layer is introduced to handle this kind of texts. Basically, three types of social media specific noise are considered: (i) Linkages to other users (such as @username), (ii) slangs (such as hahahaha) and (iii) names. Because both networks offer the possibility to mention other sites or profiles, e.g. @jondoe or Jon Doe, this information needs to be excluded without reducing the information content. Since the proposed framework in this thesis aims at providing an automated analysis, a data-driven approach is preferred over, for example, lexicon-based methods, which may need to be extended with words with every additional company whished to be presented in the vector space by the user of Customer2Vec. Links in social networks are introduced by '@', which is quite convenient to remove the symbol together with the subsequent username linked to it. For this purpose, a function has been introduced that iterates over the documents and removes the observation if the link is the only information, but keeps the remaining words if text follows the link. This pre-processing step ensures that as much information as possible is retained in the data set. Unfortunately, names and slangs do not come with a handy marker such as '@' or similar, so a clustering based mechanism is implemented to first detect the noise and then remove the corresponding words from the data. The removal of slangs and names follows the theory of dense clustering of the distributed representation of textual data, in this case word vectors $\vec{V}$. Since names and slang should have a very irrelevant amount of information to explain the underlying data, most of it should be collected in the same cluster of *Irrelevance*. To create this cluster, $\vec{V}$ is processed in the HDBSCAN algorithm after dimension reduction by UMAP, similar to that applied to $\vec{C}$ for topic detection. Experimental analysis showed that a UMAP setting of ϕ = 15 and γ = 5 with at least 40 words within the same cluster leads to generally good results. Figure 5.1 shows the

detected clusters and their first five words for an exemplary vector space generated by Doc2Vec with 50 *epochs* and a *vector size* of 300. The cluster to be removed is marked by the words *van, de, hahaha, der, hahaha, den, lisa, marie, laura, sarah* and shows exactly the precision for the desired result. One might think of slangs and names as outliers and therefore it should be clustered in the *noise cluster* of HDBSCAN. However this is not how HDBSCAN works. Since slangs and names have similar representations they are not detected as noise but rather as one cluster. This cluster can now be used as an automatically generated lexicon for words to be removed from the vocabulary V. In addition, all documents containing fewer than 3 unique words from the topic modeling are removed, as this type of textual information is assumed to contain only emoticons or irrelevant content. This leaves a total count of 469,076 UGTD. However, slang and outlier clustering is not allowed to remove slang words that are very similar to *correct* words, such as comfy → comfortable. Fortunately, due to the semantic nature of $\vec{C}$ and $\vec{V}$, this is not observed. After applying the pre-processing steps, the data can be translated into a distributed representation for topic detection and other semantic applications.

Figure 5.1: *Results of passing $\vec{V}$ into the topic modeling framework with ϕ = 15, γ = 2 and minimum cluster size = 40.*

5.2 Distributed representation of social media comments

The backbone of all further applications of the distributed vector space, e.g. topic modeling, is the creation of $\vec{V}$ and $\vec{C}$ by the Doc2Vec model, in order to transform the corpus into a numerical representation such that $C \rightarrow \vec{C}$. While the desired outputs of Customer2Vec are company specific applications, the generation of each individual vector space is based on the complete UGTD set. The reasoning behind this procedure is the assumption that the embeddings $\vec{V}$ and $\vec{C}$ benefit from a larger input data set. In other words, it is assumed that the more often a word occurs, the better its numerical representation. Using the full documents should maximise this effect. Since the mapping of $C \rightarrow \vec{C}$ uses the complete data, the decision of the hyperparameters for the Doc2Vec model must also be based on the complete data set. In order to derive an appropriate application of Doc2Vec, special attention must be paid to the characteristics of the underlying textual data type. First of all, social media comments are short, noisy and not always informative, despite their informative value mentioned in chapter 2. Additionally the dataset is highly unbalanced towards each entities exposure of UGTD. Therefore, a targeted parameterisation towards an embedding space that is able to structure the data in a vector representation and from which informative comments can be extracted is absolutely necessary for the successful implementation of a social media mining tool as built in this thesis. For the creation of $\vec{w}$ and $\vec{d}$, perhaps one of the most important parameters is the introduced window size δ. Angelov uses in his implementation a δ of 15, which probably stems from the fact that he applied the Top2Vec model onto the famous 20 News Group dataset which tends to have longer documents as social media comments.[1] Since δ determines which words are selected and weighted more heavily for vectorisation, a value that is too large may result in poorer embeddings, as it hinders the model to look at a denser part of the document. To derive the correct value for δ, the total length of the documents passed is probably the best starting point. This is done by using the distribution of words for the whole corpus. Figure 5.2 shows the distribution of word counts for each company, and implicates that using a window size of 7 is in most cases very close to the median of all companies individual word counts of the respective documents. Regarding the parameter *vector size*, which defines the dimensionality in which the embeddings are represented, it is often argued in the literature that a value between 100 and 300 is sufficient to represent d and w, but approaches for concrete values can also be found. Patel and Bhattacharyya suggest using a vector size that is optimised based on corpus statistics. They argue that the number of pairwise equidistant words in the vocabulary gives a lower bound for vector sizes (Patel and Bhattacharyya, 2017). Angelov relies on the results of extensive testing of the Doc2Vec module by Lau and Baldwin, 2016, which suggests a general embedding size of 300. However, it may be argued that the embedding size depends not only on vocabulary statistics, such as pairwise equidistant words, but also on the nature of the textual data. While Angelov used the Top2Vec approach on the famous 20 News Groups dataset, Patel and Bhattacharyya used the Brown corpus from the NLTK package.[2] Therefore, the vector size is part of the following simulation, which derives the remaining parameters based on the topic information gain. Other parameters are derived directly from the original Top2Vec

[1] For the original parameter setting see github.com/ddangelov/Top2Vec/blob/master/top2vec/Top2Vec.py
[2] For the interested reader, the datasets can be obtained via sklearn and NLTK toolkits

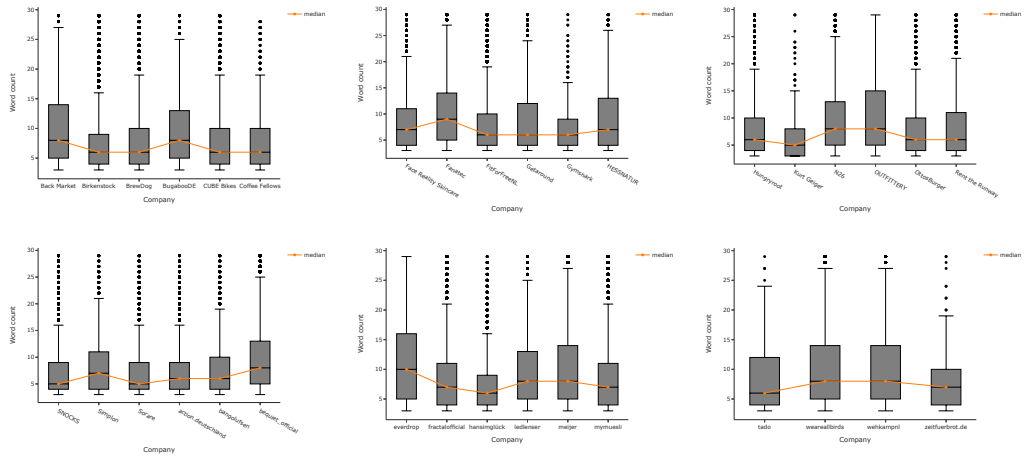


Figure 5.2: *Word frequencies of underlying UGTD. The orange line connects the median of each company. In order to create the box plots first the word count per document over all companies is computeted. Then the data is passed into the boxplot with a reduced view onto the range of [0:30] to conquer the impacts of outliers. While outliers are only removed for visualisation they are left in the data for the calculation of the median.*

implementation, to take advantage of the results mentioned in Angelovs paper. Specifically, this means that a sub-sampling threshold of 10^5, *hierarchical softmax* and a *minimum count* of 50 are used.

5.3 Sensitivity analysis of pointwise mutual information

While the derivation of an appropriate parameter set for the Doc2Vec model can to some extent be derived from corpus statistics, the majority of the parameters *vector size* and number of *epochs* and the UMAP and HDBSCAN parameters ϕ, γ and *minimum cluster count* are less trivial to select. For this reason, the concept of PWI introduced in chapter 4 is used to optimise the choice from a given set of parameters. The PWI is not only calculated per parameter set, but also per company, since the goal is a cross-company vector space that allows company-specific topic detection. This is also to address the unbalance of the UGTD set in terms of the number of comments per company, to avoid that the choice of parameters is derived solely in favour of companies with a higher number of UGTDs. For this purpose, the original use of all documents to create $\vec{C}$ must now be translated into company-specific applications for company-specific topics. In order to obtain company-specific topics, a horizontal reduction of document embeddings is performed via the simple condition that the embeddings are ordered 1:1 as the original documents. Thus, it is possible to derive the company-specific documents from the indices of the company-sorted data. While the document embeddings are reduced according to the company-specific documents, the word embeddings in theory can be left untouched. The reasoning behind this follows the idea of word embeddings. Since similar words

should have similar embeddings, it would not make sense to remove words and create entity specific word embeddings. However in the practical application it was observed that words from different companies in the same vector space got intermixed due to some similar business models of companies leading to similar linguistic expressions. One example are the footwear manufacturers *Allbirds*, *Birkenstock* and *Snocks*. As the business of all three companies revolves around related products, it happened that specific words from one company appeared in another collection of thematic words. Therefore only words w and their corresponding vectors $\vec{w}$ are used when they appear at least once in the company specific documents. Nevertheless, the document leverage approach is advantageous. It allows the creation of unique topics per entity while generating the vector space across all entities. To do this, the horizontal document matrix reduction mentioned above is used and fed into the topic creation algorithm, which allows for company-specific topics. While only the document matrix dimension is changed, the remaining steps follow the framework of dense clustering onto reduced dimensionality based on distributed document representations:

(i) Reduce document embeddings with UMAP by selecting ϕ and γ,
(ii) create dense clusters with HDBSCAN out of UMAP output and
(iii) derive topic vectors by calculating the centroid of each label cluster.

In order to illustrate the expressive power of PWI a exemplary derivation of topics without any hyperparameter tuning was performed. To create local topics following Angelovs approach the document reduction technique is applied and the resulting company vectors are passed into UMAP and HDBSCAN. For exemplary illustration a *minimum cluster count* of 15 reduced to five components with 15 neighbours was used for deriving topics of the companies *Meijer*, *Birkenstock* and *mymuesli*. Table 5.1 shows the logic of the PWI. The higher the score, the more differentiated the theme words are. The results for Meijer, for example, show comprehensive results with a high PWI, such as topics about snacks and other food (topics #31, #2 and #1). On the other hand, themes with a lower PWI score seem to be less meaningful. The same goes for Birkenstock and mymuesli. The first topics are quite intuitive and are about the look of Birkenstock shoes and the taste of mymuesli cereals, while topics with a lower PWI score are more like a collection of unmeaningful *outliers*. It can be concluded that model optimisation towards a maximum PWI score is desirable. Therefore, a simulation was conducted with the parameter values in table 5.2.

However, as can be seen from the topic number column t in table 5.1, the same value of *minimum cluster size* results in a different number of topics per company. In fact, the parameter is very sensitive. With a constant UMAP parameter setting and a varying value for *minimum cluster size*, a second simulation was performed to demonstrate the sensitivity. The results are shown in Appendix C and clearly show that as the minimum cluster size increases, the HDBSCAN algorithm disproportionately reduces the topic size. Also companies with very large sample sizes of UGTD are more robust to topic creation with a higher minimum cluster size, i.e. high minimum cluster sizes lead still to a moderate number of topics, as is the case for *FitForFree* or *Birkenstock*. For an overall optimisation towards a high PWI over the whole

Company	t	Topic Words	PWI(t)
Meijer	31	reeses, reese, twix, kat, snickers, butterfinger, kitkat, kats	9,120.55
	2	yum, royal, butter, boy, oven, peanut, reese, yummy	7,757.98
	1	yummy, delish, ya, tomato, looks, bite, curbside, peeps	6,037.08
	...	...	...
	8	brach, gmo, kats, mold, ripe, collab, player, shoot	9.76
	148	owensboro, nc, haute, xxx, bremen, bowling, tis, whoop	2.59
	144	ooooh, ooo, soooooo, ohhh, xxx, sooooo, yummm, ohh	0.00
Birkenstock	1	fancy, gotta, wipes, towels, omg, dang, wipe, lemon	5,618.11
	13	fancy, dang, beauties, omg, cute, ohhh, alexander, lemon	4,995.91
	15	ooooh, ooo, ohhh, love, addicted, omg, yummy, favorites	4,302.21
	...	...	...
	19	kyle, austria, ooooh, iii, munich, dis, ohhhh, nc,	4.58
	3	citizens, bananas, transformation, yay, expired, colorful, citizen	13.05
	10	cedar, trick, saginaw, south, north, smokey, opens, lines	166.44
mymuesli	9	delicious, looks, soooo, sooo, sounds, yummy, tasty, blueberry	3,144.96
	5	tasty, strawberry, mustard, tastes, delicious, hmm, wine, looks	1,925.59
	24	looks, tasty, delicious, tastes, sounds, sooo, wow, yummy	1,664.48
	...	...	...
	26	thrifty, acres, blueberry, dyson, vie, beans, soup, muffins	8.03
	11	blackberry, muller, claudia, cannes, eagle, schmidt, epil, function	37.02
	25	maria, vienna, daniela, martina, exists, austria, vendors, janssen	39.17

Table 5.1: *Local topics derived from Meijer, Birkenstock and mymuesli with the standard UMAP setting of ϕ = 15 reduced to five dimensions. After dimension reduction the vectors are clustered with a minimum cluster size of 15. The table shows per company three topics with the highest PWI scores and those three topics with the lowest PWI scores.*

Model layer	Parameter	Values
Doc2Vec	Epochs	50, 200, 400
Doc2Vec	Vector size	200, 300, 400
UMAP	Number of neighbors ϕ	5, 7, 10, 15
UMAP	Number of dimensions γ	5, 10, 15
HDBSCAN	Minimum cluster size	15

Table 5.2: *Parameter setting for PWI optimization.*

vector space $\vec{C}$, different numbers of topics per company can lead to the following problem. The less topics are used for the calculation of the PWI according to Eq. (17), the lower the value will be since the variation of $V*_t$ will be smaller, and vice versa. In particular, companies with less comment emergence would have a small PWI if the *minimum cluster size* is high. Therefore, local topics are detected for each company with a given UMAP setting and a *minimum cluster size* of 15. The topics are then reduced by iteratively merging each smallest topic with the most similar topic until <= k topics are reached.[3] This ensures that the gap between companies that tend to have a larger number of topics does not overshadow the results of companies with a smaller number of topics. Then, after merging the topics to <= k topics with the corresponding values for ϕ and γ taken from table 5.2, the PWI per topic and per company is calculated. To get a general overview, the results per parameter combination are aggregated by taking the median. Analysing the results via the median follows the concept of the PWI and the reduction of the impact of companies with a higher UGTD which have larger information scores. This addresses the unbalanced dataset since a larger amount of UGTD again leads to a higher PWI score by its definition in Eq. (17). Figure 5.3 shows the corresponding results for k = 20 while the PWI scores are plotted on the y-axis. Reducing the dimensionality clearly benefits the PWI and thus the interpretability of the topics by assumption. This can be seen from the fact that the maximum median values of each *vector size* - *epoch* combination has either $\gamma = 5$ or $\gamma = 10$ in table 5.3, except for the cases with *vector size* = 200. The PWI is almost always higher than for $\gamma = 15$. Nevertheless, a value of $\gamma = 10$ seems to be better than $\gamma = 5$ in some cases. In the case of the parameter ϕ, shown on the x-axis, the picture is less clear and generally mixed. While for some cases the variation of *phi* seems to be less important, the case of *epochs* = 200 shows the most variance in PWI across different values for this parameter, especially with a smaller *vector size*. The picture of this parameter setting is generally interesting, as no other parameter constellation can be found to have such a variance of the median PWI. The highest PWI for the given parameters can be obtained by using 50 *epochs*, a *vector size* of 300 and corresponding values for ϕ and γ of 5 and 10. However this parameter setting is not chosen to process since γ = 10 seems to be a slightly worse choice in the remaining scenarios and the differences compared to ϕ = 5 are only marginal. Nevertheless, the reason why a parameter setting with γ = 5 or 10 and *phi* = 5 works best in the median case may be a collection of different settings that favour the underlying algorithms. First, as argued by McInnes and Healy, 2017, smaller dimensions are better for clustering and additionally small neighbourhoods are better for reflecting local structures Angelov, 2020. Regarding ϕ, it can be concluded that lower values perform better, since the topics found in the UGTD across all companies may be very local and have few global relationships. Therefore, lower values allow UMAP to focus on this local information concentration and correctly use the right topic words $V*$. This is also due to the nature of UGTD. UGTD is more a collection of unstructured, noisy sentences, closely spaced in time and specificity under each post, than a temporally coherent conversation. A more local view, i.e. a smaller value for ϕ, therefore seems more appropriate. Given that the topic embeddings are created from the whole corpus, one might think that smaller values for ϕ are better because they can better capture the unique space of companies. However, this is not the case. The Doc2Vec algorithm does not discriminate between entities and assign them places in an arbitrary space,

[3] This process is taken from the original implementation in Angelov, 2020.

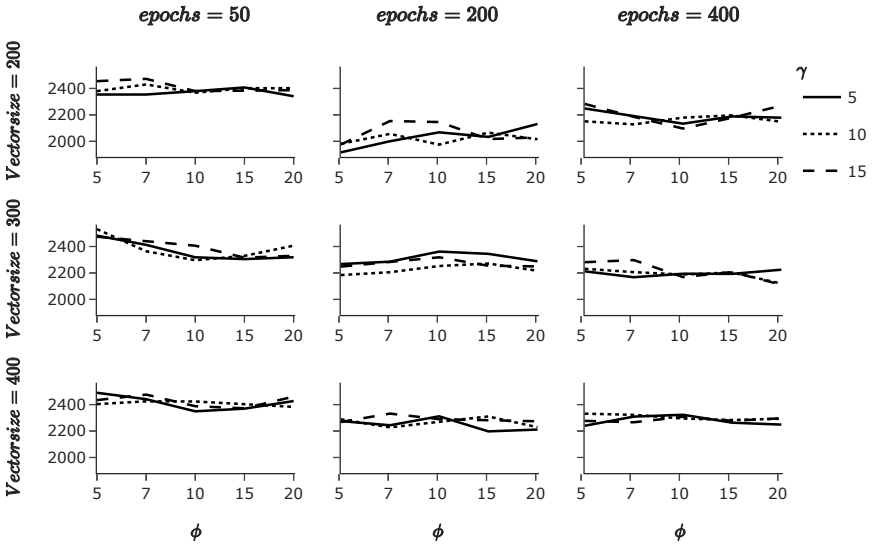


Figure 5.3: *Variation of epochs and vector size within the Doc2Vec plotted on the horizontal and vertical line. The UMAP parameter ϕ is plotted on the x-axis while γ determines the dashed lines and the corresponding PWI score is shown on the y-axis.*

but looks at the corpus as a whole and correctly assigns semantic meanings across companies. Therefore, the reason why a smaller neighbourhood benefits the PWI must be based on the fact that local issues can actually be better represented. Despite the argument for different results, the differences are small, as can be seen from the y-axes. Therefore it can be concluded that the parameter choice of the Doc2Vec model seems to be more import as for UMAP. Further the experiment confirms Angelovs implementation of choosing a smaller component value for γ since the values of 5 and 10 outperform 15 in most of the cases as can be seen in table 5.3. However the results for *vector size* = 300 seems to be most stable compared to the case of a *vector size* of 200 with varying number of *epochs*. For the following topic modeling results a *vector size* of 300 is chosen trained with 50 epochs and a UMAP parameter setting of *phi* = 5 and γ = 5.

The behaviour of HDBSCAN of not using the complete data for clustering and therefore for the topic creation within the Top2Vec framework but rather declaring part of the data as noise might be critical when the underyling data is text and especially short social media comments that tend to be noisy in general. The parameter setting of the highest median PWI lead to the disclosure of approximately 20% on average from every company as can be seen in appendix D. It seems that HDBSCAN mainly clusters comments as noise from companies that generally have less UGTD, which may be seen as an influence of the unbalanced dataset. However one might argue that the results under the consideration of removing documents from the topic detection are not wrong but rather in the worst case incomplete. Meaning that if the algorithm

Doc2Vec		UMAP		
Vector Size	**Number of epochs**	**n_neighbors**	**n_components**	**PWI**
200	50	7	15	2,472.39
200	200	7	15	2,153.65
200	400	5	15	2,283.10
300	50	5	10	2,531.42
300	200	10	5	2,361.78
300	400	7	15	2,297.21
400	50	5	5	2,489.45
400	200	7	15	2,270.04
400	400	5	10	2,331.40

Table 5.3: *Highest median PWI values of each vector size and epoch combination across companies and parameter settings.*

is able to cluster a specific set of documents C_t which share a latent topic then the users in reality frequently talked about it.

5.4 Local topic modeling results

After deriving a simulation-based parametrisation of the model layers Doc2Vec and UMAP and handling *minimum cluster size* by a handy hierarchical reduction method, local topics can now finally be derived by simply following the procedure within the simulation and setting the parameters to fixed values. While for a local comparison of different PWI values the *minimum cluster size* was set to 15, for a local analysis of a given topic the user might want to change the value. Translating the parameter into the usage of Customer2Vec, that means the lower the value is the more specific topics of the company specific comments will arise. However for exemplary illustrations the same hierarchical topic reduction as in the previous simulation is used in order to benefit from the PWI results. Local topics are shown in tables 5.4 and 5.5.

For each illustrative company, the top four issues with the highest PWI(t) were selected. The themes identified show clear company specificities and relationships to the underlying business model. For example, *Back Market* has the inherent theme of sustainability related comments (theme #7), which is quite intuitive as the company sells refurbished equipment. Another very differentiating theme seems to be topic #19, which indicates the discussion around the company's products. The remaining themes are more about customer service, which is consistent with other companies. Similar observations can be made in the case of *Face Reality Skincare*. Topic #17 revolves around the skincare company's products, while topic #11 suggests that social media users talk about results and skin conditions. The recognition of products through the topic modelling approach can also be seen in the case of *Bang & Olufsen*. Topic #13 aggregates document vectors which latent topic evolves around the products speakers and headphones,

Company	Industry	t	Topic Words	PWI(t)
Back Market	Tech	7	reusable, bags, plastic, straws, recycle	309.62
		18	dm, check, dms, pls, inbox	123.59
		17	email, sent, contact, response, message	69.57
		19	phone, cell, phones, text, activate	50.55
everdrop	Other	13	ripe, pamela, everdrop, ambitious, mold	906.94
		19	unpacked, plastic, regionally, reusable, regional	697.36
		17	email, delivery, ordered, answer, received	409.50
		18	tabs, cleaner, detergent, dishwasher, powder	376.35
Birkenstock	Fashion	15	looks, comfy, kinda, birkensocks, cute	2,877.27
		12	media, social, birkenstock, availability, styles	2,789.34
		18	birkensocks, pair, crocs, laces, gorgeous	1,740.77
		11	narrow, omg, width, bah, cutest	1,573.19
CUBE Bikes	Sports	16	enduro, ride, bike, dream, electrical	3,530.96
		8	bike, bikes, tm, dealer, stereo	1,800.16
		4	dream, bike, enduro, hardtail, master	1,456.07
		5	gravel, clockwork, hammer, horny, deposit	1,125.83
Coffee Fellows	Food	2	delicious, wooow, tasty, mega, looks	156.32
		5	coffee, iced, drink, cup, tea	125.47
		13	pic, send, upload, ig, page	97.31
		9	mega, ohhhhh, wooow, hammer, imb	75.78
Face Reality Skincare	Other	17	mandelic, toner, hydrabalance, serum, moisturizer	573.88
		11	results, acne, skin, skincare, client	437.14
		13	face, reality, acne, skincare, skin	244.12
		18	skin, transformation, acne, skincare, toner	210.10
Fanatec	Tech	17	csl, dd, ps, pedals, clubsport	503.76
		15	ps, dd, podium, xbox, wrc	387.77
		18	sim, ps, playstation, console, game	237.24
		0	steering, wheel, wrc, mclaren, bmw	232.06
action	Tech	3	action, shorts, erson, balcony, decoration	421.09
		18	decoration, hop, luna, pillow, cocktail	273.01
		14	goat, ati, pea, nearby, erson	112.32
		1	dortmund, ams, decoration, branches, apartment	94.11
Bang & Olufsen	Tech	16	beoremote, anniversary, rose, edition, gold	681.77
		13	speaker, music, headphones, lifestyle, sound	496.92
		19	zhang, lay, handsome, music, ambassador	459.67
		17	headphones, craft, speaker, bronze, technology	160.51
be quiet!	Tech	12	window, base, orange, silent, windowed	3,781.22
		19	psu, silentwings, silent, cooler, airflow	2,968.12
		0	orange, window, base, silver, windowed	2,579.40
		1	geforce, gb, gtx, radeon, voodoo	1,448.37

Table 5.4: *Top four local topic modeling results per company with optimized ϕ and γ towards the highest PWI(t) score I/II.*

Company	Industry	t	Topic Words	PWI(t)
Fit For Free	Sports	9	lessons, exercise, gyms, lesson, group	842.81
		19	cancel, subscription, lessons, exercise, lesson	633.14
		6	workout, gym, bodypump, exercising, workouts	451.64
		12	exercising, weights, les, zumba, lessons	398.59
Fractal	Tech	16	gtx, geforce, ti, gb, evga	2961.89
		18	radeon, geforce, gtx, strix, gb	1706.87
		9	meshify, tempered, define, tg, nano	1148.82
		13	itx, case, fractal, nano, matx	935.70
Hungryroot	Food	18	smoothie, pale, toast, amazing, blueberry	480.22
		8	delicious, bowl, sounds, tasty, yumm	371.67
		10	cookie, dough, batter, brownie, yum	368.62
		17	hungryroot, meals, sauce, meal, salad	248.30
Rent the Runway	Fashion	16	ambitious, caring, leader, strong, magical	741.01
		18	dress, wedding, rtr, dot, vest	455.62
		3	service, customer, chat, emails, response	329.13
		7	rtr, unlimited, member, rental, membership	274.95
Snocks	Fashion	13	mega, horny, underpants, cool, nails	595.31
		14	wanderbuddy, mega, pa, cool, wooow	576.05
		10	yeezy, boost, sneaker, adidas, snocks	574.99
		7	socks, shorts, snocks, sock, underpants	537.77
Simplon	Sports	15	pmax, simplon, rapcon, enduro, trails	220.09
		19	pmax, rapcon, simplon, anniversary, xt	185.64
		12	pic, moment, ig, hug, dope	153.21
		1	ambassador, rider, trails, rapcon, horny	137.59
Sorare	Sports	5	eth, sorare, releases, commenting, allows	338.54
		18	eth, league, football, goat, invitation	311.99
		3	dm, european, ig, giveaway, collaboration	186.58
		16	neymar, epil, blessing, ronaldo, ron	131.46
mymuesli	Consumer Goods	7	tasty, delicious, sounds, incredibly, muesli	2722.14
		11	delicious, tasty, looks, smoothie, mega	2594.92
		16	berry, mega, porridge, espresso, minis	1237.89
		1	muesli, mueslis, mymuesli, dresden, bircher	1111.43
Allbirds	Fashion	15	sheep, cruelty, animals, wool, cruel	579.83
		1	allbirds, comfortable, nz, mizzles, runners	403.56
		8	sizes, size, women, men, width	294.71
		14	comfortable, worn, toe, pair, feet	264.47
Wehkamp	Fashion	17	birth, zwitsal, sister, pregnant, child	2565.02
		14	trizone, oral, toothbrush, brushing, wolters	1094.79
		5	email, sent, mail, receive, received	1031.98
		7	wehkamp, wolters, msi, piet, nl	983.86

Table 5.5: *Top four local topic modeling results per company with optimized ϕ and γ towards the highest PWI(t) score II/II.*

while topic #16 even points to the discussion around specific editions of the company's products. In the case of *Allbirds*, the topic with the highest detected PWI seems to be about customers who are unhappy with the materials used in the company's shoes (topic #15). This seems to be a very specific topic in the company documents, as the PWI of this topic is higher compared to the other PWI(t) of *Allbirds*. However, not every topic is so intuitive and useful. The case of everdrop, for example, shows that the most distinguishable topic develops around brand ambassadors that the company uses for promotion and are frequently mentioned in the comments. One might think that the removal of the names mentioned in the preprocessing step was not successful, but it is more likely that the names of the brand ambassadors are in other clusters because they appear more often in the context of relevant topic words $w*$. Another drawback of the topics is the amount of spoken language such as 'ohhhhhh' or 'wooow'. Although it can be concluded that documents falling into this cluster are about positive moods, it is not possible to say what specific products or services these comments were made about. Still, the way an investor can benefit from the results is to get an overview of which products seem to be important to customers, at least to those who comment on social media, what problems they associate with the company and what problems customers have with, for example, its services. The PWI score is a great help in distinguishing relevant from irrelevant topics. It is argued that by generating the document and word vectors on the basis of the complete data set, i.e. across all companies, a vector space is created from which meaningful aggregations in the form of topics can be identified with the help of the PWI score. This vector space is used in Customer2Vec to create meaningful topics.

Local topic similarities In addition to company-specific topics, the distributed representation of C also allows semantic meanings to be drawn across companies. This is possible because $\vec{C}$ and $\vec{V}$ are created over the whole corpus and therefore have the same dimensionality across all companies. By comparing $\vec{t}$ across all companies, it can be investigated whether some topics are company specific, while other topics may be more similar. This procedure can be seen as an alternative to directly deriving global themes by building bridges between local companies via S_C. Once again, the power of a semantic vector space involving all companies is evident. To do this, a matrix is constructed that takes as values the cosine similarities between all topics across all 34 companies, e.g. :

$$\begin{pmatrix} & t_1 & t_2 & \dots & t_{k*34} \\ t_1 & S_{C1,1} & S_{C1,2} & S_{C\dots,\dots} & S_{C1,k*34} \\ t_2 & S_{C2,1} & S_{C2,2} & S_{C\dots,\dots} & S_{C2,k*34} \\ \dots & S_{C\dots,1} & S_{C\dots,2} & S_{C\dots,\dots} & S_{C\dots,k*34)} \\ t_{k*34} & S_{Ck*34,1} & S_{Ck*34,2} & S_{Ck*34,\dots} & S_{Ck*34,k*34} \end{pmatrix}$$

Since the parameter of *minimum cluster size* has to be defined for each company, the resulting topics are of different sizes. This leads to an obvious problem. While companies with less UGTD are not able to create topics with a large number of *minimum cluster size*, companies with more UGTD are not able to draw meaningful topics without larger values of *minimum cluster size*

since the resulting topics may be too specific. To balance this trade-off, a general minimum cluster size of 15 is used. The topics are then reduced to 20 using the same topic reduction method as in the PWI simulation. Then <= 20 topics per company are appended, resulting in a matrix that contains all topic vectors across all companies, while the number of topics for each company is stored in a separate data frame in order to be able to infer which vector belongs to which company. Conclusions about local topic similarities between companies can now be drawn by extracting the highest similarity for each company, with the condition that the company must not be the same as the original one. Example results are shown in table 5.6. The results show for some examples that the distributed representation of themes is able to draw relationships between companies for similar topic words. Since the company *action* sells various goods including sports and also technology, similarities can be observed towards the companies *Sorare* (Sports) and *Bang & Olufsen* (Tech). An example that the vector space is also able to draw similarites not only across the same type of product but in a more global perspective is the topic pair of *CUBE* and *Bugaboo*. Since both companies produce products that involve wheels, rims etc. there numerical representation seems to be comparable. Further for some companies, such as *Back Market* and *everdrop* the topic of sustainability seems to be inherent. This is in line with their business model, as they are active in the field of recycled electronics and sustainable household appliances. The distributed representation of the underlying textual data is clearly able to draw this relationship and assigns a high similarity score. In addition, the model is able to assign industry-specific phrases, such as *Birkenstock* and *Allbirds* in the footwear industry. The model is also able to draw similarities between companies that do not share a business model, but may share an important part of their operations. For example, *N*26 and *Sorare* do not have the same business model, but since *Sorare* is a provider of an online football app based on NFTs and *N*26 is a bank, the common theme between the companies is the general theme of finance. In addition to common topics with a high similarity score, company-specific topics can also be derived.

The last four topics have a lower maximum value of S_C with all other topics and reveal company specific complaints about the health policy of *meijer*, the production of *Allbirds* shoes, energy discussions about *tados* thermostat products and specific words about the business model of *Back Market*. However, not all common topics with a high degree of similarity are of such significance. This is entirely due to the highly fuzzy structure of social media data. However, the representation of $\vec{C}$ by $\vec{t}$ is still able to distinguish between specific and shared topic spaces across companies.

Company A	Topic Words A	Company B	Topic Words B	$S_C(\vec{t_A}, \vec{t_B})$
Sorare	football, boo, neymar, goat, father, watching, play, alice	action	rocky, tennis, boo, ceiling, hallway, sis, mae, pillows	0.94
Bang & Olufsen	bronze, headphones, craft, technology, speaker, design, optimized, sound	action	boxer, craft, fingers, headphones, lips, speaker, pup, music pot,	0.92
CUBE Bikes	rims, ams, sram, aim, analog, tt, tires, aerium	Bugaboo	tires, rims, cameleon, donkey, aim, stem, bugaboo, ams,	0.92
be quiet!	purebase, quiet, psu, case, english, pc, itx, nzxt	Bang & Olufsen	quiet, psu, fins, giveaway, noise, performance, silence, pc	0.89
CUBE Bikes	hpc, tm, trails, hybrid, cubes, gravel, mt, brakes	Simplon	bike, gravel, tm, mtb, ebike, enduro, brakes, trails	0.88
Hans im Glück	fries, potato, burger, burgers, sweet, ketchup, steam, cucumber	Ottos Burger	fries, potato, hype, chili, bagel, burger, camp	0.74
Allbirds	runners, comfy, allbirds, mens, shoes, combat, outdoors	Birkenstock	clog, restock, pairs, choc, comfy, cute, salted, caramel	0.69
Back Market	reusable, bags, recycle, plastic, recycling, containers, vegetables, disposable	everdrop	washable, reusable, fabric, bags, pads, plastic, unpacked, diapers	0.67
Back Market	refund, nor, refunded, thieves, terrible, response, returned, service	Getaround	service, cancel, customer, reviews, thieves, refund, charge, rent,	0.61
Sorare	sorare, eth, league, releases, commenting, allows, function, players	N26	deposit, bank, money, cash, transfer, dm, account, blocked	0.60
meijer	employees, mask, masks, customers, enforce, employee, people, medical	Fit For Free	customer, service, cancel, membership, subscription, canceled, email, cancel	0.57
Back Market	iphone, android, app, mobile, macbook, ipad, computer, gb,	Birkenstock	splendid, bronze, bells, rockin, awwww, copper, birkenstocks, metallic	0.46
Allbirds	cruelty, animals, wool, sheep, cruel, industry, animal, humane	Hessnatur	sustainable, regional, materials alternatives, fashion, produced	0.39
tado	heating, energy, devices, saving, systems, mode, services, detergent	meijer	clip, clipping, slot, printer, curbside computer, clipped, redeem	0.37

Table 5.6: *Local topic similarities derived by calculating the cosine similarity of each local topic to all other local topics. Then the topics with $max(S_C)$ are stored with the corresponding topic words.*

6 Customer2Vec - Module implementation

The previous chapter demonstrated the theory, application and interpretation of the created vector space based on $\vec{C}$ and $\vec{V}$ of which topics $\vec{t}$ are generated. Further the PWI scores were used to detect differential topics out of noisy social media comments. However the requirement of Customer2Vec is an analysis that is both interpretable and in-depth in order to map the topics and mood of the customer base. Topics are meaningful if not profound enough, especially considering the complete process including the dimension reduction with UMAP and the clustering of HDBSCAN. Additionally the distributed vector space also allows for further applications which use the semantic embeddings of $\vec{d}$ and $\vec{w}$. While topic modeling is an unbiased application of vector space, it also allows for the targeted analysis of, for example, specific keywords, extending Customer2Vec's portfolio. To connect the vector space and the resulting topics and similarities of words and comments with the mood of the customer, the concept of sentiment is used. More precise, the sentiment analysis algorithm used within Customer2Vec follows the Flair framework. The selection of the sentiment model is based on a simulation shown in Appendix E, which also includes a explanation of the general concept of sentiment analysis. These modules in combination with the proposed vector space form Customer2Vec. The aim in constructing the modules was to fully automate the analysis in order to make the usability and interpretability of the output as user-friendly as possible.

6.1 Topic evolvement and sentiment shifts

For a deeper analysis of the inherent topics per company, each document is linked to its corresponding theme. This is done by re-engineering the creation of topic vectors. Since $\vec{t}$ is the centroid of the surrounding document vectors $\vec{C}$, $\vec{d}$ must belong to the topic with the smallest distance, measured by cosine similarity, as in Eq. (10). In addition to the per-topic sentiment analysis, it is also instructive to look at sentiment shifts over time. Since the temporal distribution of comments is stored in the dataset, each document can be linked to its date of creation. Sentiment analysis is then performed for each document-year combination and then aggregated by topic by simply taking the average. In addition, information is stored on how many documents the topic contains for a given year. This makes it possible to look at emerging topics locally over time. The results are demonstrated for the companies *N*26 and *Allbirds*. Figure 6.1 shows that over the years there has been a recurring theme of customer service (topic #1 from the top) regarding *N*26, with sentiment always in the negative area. In addition, topic #2 developed positively with an increasing number of comments about the customer account

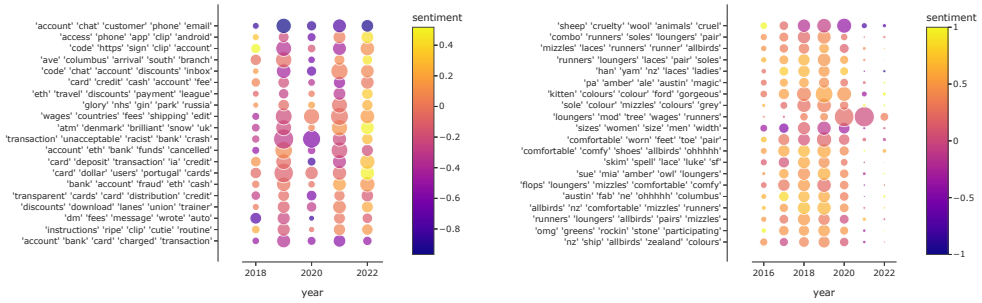


Figure 6.1: *Topic evolvement results for N26 (left) and Allbirds (right).*

and the online bank's app. Topic #11 was very strong in 2019 and 2020, with negative sentiment. During this period, people seem to have felt increasingly discriminated against and there was a general feeling of discomfort towards the bank. In the case of the shoe brand *Allbirds*, topic #1 is particularly striking and has developed more strongly from 2016 to 2020. Topic words show that during this time, customers have increasingly complained about how the brand produces its shoes and the raw materials it uses. This topic was already shown in table 5.5 but now it is possible to see its timely and *emotional* evolvement. Also noticeable is the increased mention of the shoe types Loungers, Tree and Runners in topic #9 with a moderate sentiment from 2018 to 2021. It can be concluded that a time and mood weighted illustration of topics is more illustrative as the pure analysis of topic words. For further research and extensions of Customer2Vec it might also be valuable not to only look at the movement of topics but also link it to the financial performance of the company.

6.2 Competitor benchmarking analysis

Topic models provide a good overview of the topics discussed for each company and by combining the results with sentiment analysis intuitive representations can be obtained. However, the distributed representation of C also allows for further applications by using the logic of the underlying classification task of the skip-gram model to create word vectors. Word vectors $\vec{V}$ are constructed to have a similar representation for words with a similar context. Theoretically, it should then be possible to infer a degree of relatedness of a given word w', the target word, to all other words $w \in V$. To examine the relationship, again the concept of cosine similarity S_C is used, e.g. $S_C(\vec{w'}, \vec{w})$. It is argued that a strategic choice of w' can provide additional insight into certain latent categories within the UGTD. To implement this, a filter mask is created such that:

$$I_{w'} = \begin{cases} 1, & \text{if } S_C(\vec{w'}, \vec{w}) \geq \lambda \\ 0, & \text{otherwise} \end{cases}$$

Target Word w'	Word w	$S_C(\vec{w}', \vec{w})$
Computer	PC	0.59
	System	0.56
	Phone	0.52
	Laptop	0.50
	Gaming	0.48
Fruit	Vegetables	0.64
	Veggies	0.57
	Fresh	0.57
	Berries	0.52
	Produce	0.51
Bank	Account	0.75
	Money	0.57
	Card	0.52
	Credit	0.50
	Cash	0.49

Table 6.1: *Word similarities based on cosine similarity over the complete Vocabulary V across all companies.*

Where $I_{w'}$ is a mapping vector consisting of zero or one to filter C for a given word w' and words whose cosine similarity is higher than a threshold λ. [1]. Applying $I_{w'}$ to the corpus then yields all documents in which at least one word has the property $S_C(\vec{w}', \vec{w}) \geq \lambda$. While in theory it is possible to compute the similarity of any word to any other word, in practice it is obvious that it is only possible to compute similarity measures between words that are present in the vocabulary, and therefore in the document, and whose frequency is above a certain number. However, it is still possible to get an overview of similar words to any given word in the dataset. For this reason, a mapping has to be introduced that identifies the word given by the vocabulary V in the word-vector matrix $\vec{V}$, so that $w \rightarrow \vec{w} : V \rightarrow \vec{V}$. The structure of V and $\vec{V}$ is useful for this, since the position of w in V is the same as the corresponding vector $\vec{w}$ in $\vec{V}$, the row index i can simply be used to extract the vector from the embedding matrix, resulting in the word vector of interest $\vec{w}'$, similar as the reduction technique used for C to only use company specific document vectors. Then $\vec{w}'$ is used to iterate over $\vec{V}$ and compute the cosine similarity using Eq. (3), such that $S_C(\vec{w}', \vec{w}) = \frac{\sum_{i=1}^{v} \vec{w}'_i \vec{w}_i}{\sqrt{\sum_{i=1}^{v} \vec{w}'^2_i} \sqrt{\sum_{i=1}^{v} \vec{w}_i^2}} \forall \vec{w} \in \vec{V}$. According to the scaling of cosine similarity, the higher $S_C(\vec{w}', \vec{w})$, the more similar w' and w must be. Since this task is unsupervised, it is not obvious to introduce a measure that reflects the correctness of the assumed similarity of words. However, qualitative measures such as human interpretability can be applied. For this reason, table 6.1 shows some examples of given target words and their top 5 comparison words. The results are quite impressive and follow the assumption that also moderate cosine similarities, e.g. less than 0.5, lead to intuitive results. It is important to note

[1] Please note that w' is also part of V, ensuring that documents containing w' are also marked with 1 in $I_{w'}$. Since $S_C(\vec{w}', \vec{w}') = 1$ and $\lambda \in [0, 1]$.

w'	Comment	Company	$S_C(\vec{w'}, \vec{w})$
Quality	i think it is pretty rotten to sell sandals an...	Birkenstock	0.26
	because all of be quiet ! products provide hig...	be quiet !	0.25
	great products with great performance but very...	be quiet!	0.25
	i love how comfortable these are. overall they...	Allbirds	0.24
	product tester	everdrop	0.24
Delivery	my order has not yet arrived	mymuesli	0.38
	no thanks. will definitely not order something...	mymuesli	0.38
	no thanks. will definitely not order something...	mymuesli	0.38
	unfortunately my order was not delivered and t...	Wehkamp	0.37
	order i order it	Wehkamp	0.37
Price	wow. and that's a sale.	meijer	0.31
	when is this sale over	meijer	0.29
	the product is good. but could be a little che...	meijer	0.28
	1 product but: d ...	Wehkamp	0.27
	i love this sale!	meijer	0.27

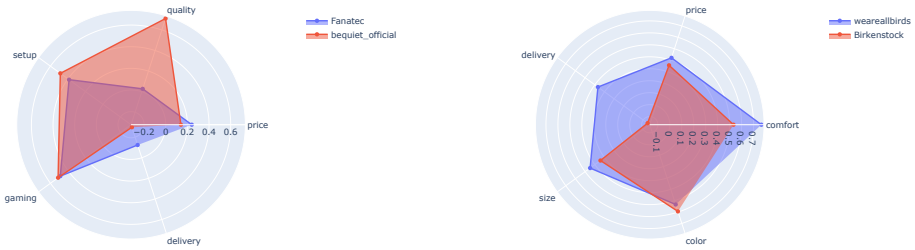
Table 6.2: *Semantic filtering results for w' = [quality, delivery, employees].*

that the results for table 6.1 are strongly influenced by the companies behind the selected words for w'. An example is the case of $w' = Bank$. Since most of the words that have a high cosine similarity to *Bank* come from the social network profile of the online bank *N*26, the fact that words like *Credit* are close to the word *Bank* is more likely due to the creation of the corpus, as the model really knows that in real life *Credit* and *Bank* belong together to some extent. Next to the possibility to introduce a semantic keyword search based on $S_C(\vec{w'}, \vec{w})$, due to the unified dimension of $\vec{w}$ and $\vec{d}$ it is also possible to filter complete documents for given w' by $S_C(\vec{w'}, \vec{d})$. This leads to $C^{\lambda}_{w'}$ and includes all documents which have the property of $S_C(\vec{w'}, \vec{d}) \geq \lambda \; \forall \; w' \in V'$ and $d \in C$. Following the semantic filtering logic introduced it allows to extract documents not only based on one word but also for words with, in the best case, the same meaning. This allows a semantic filtering of the document corpus of each company or over all companies according to a given word w' or a given set of words V', while w' and $V' \in V$.[2] For the example of $V' = [quality, delivery, price]$, the first 5 entries which do not include respective w' of the sub dataset $C^{0.2}_{quality,delivery,price}$ can be seen in table 6.2.

The results show that the word vector based search can enrich the filtering for a given keyword. The created distributed representation of words and documents is able to draw the relationship between e.g. the target word *quality* and properties of devices of the hardware manufacturer *be quiet!*. Furthermore, the keyword search is able to draw the relationship between e.g. the target words *delivery* and *price* to documents containing information about received orders and sale campaigns. Nevertheless the results also show that documents from companies with a high share of UGTD are at the top cosine similarities which might result to the case that other entities are under-represented for a given keyword. However since Customer2Vec is applied

[2]This method is implemented in the original Doc2Vec model within the gensim library.

per company this should not be an issue. Concluding the semantic creation of $\vec{d}$ and $\vec{w}$ allow filtering of documents by keywords. This approach can therefore be seen as an ex ante or biased *topic modeling* according to the given keyword or set of keywords.[3] The advantage is that if even the underlying topic model does not allow to reduce the topics to a given number in the first initialisation via the word vectors, it is possible to filter the documents for a given *topic* represented by the word vector of the category of interest or even a set. The logic is quite intuitive. The implementation for this approach is based on the Doc2Vec model and uses the document vectors for which generation word vectors are constructed. Based on the Doc2Vec model, the search algorithm applies the concept of cosine similarity between a simple mean of the projection weight vectors of w and the vectors for each word in the model, and then traces the document based on the document index. The method is similar to the word analogy and distance scripts in the original Word2Vec implementation. These two approaches, $S_C(\vec{w'}, \vec{w})$ and $S_C(\vec{w'}, \vec{d})$ are now used to develop two different dataset filtering algorithms. Firstly an approach to filter company specific comments for a set of categories that can be selected by the user and secondly an approach to define *product proxies* which are representatives for the products of a particular company mentioned in the comments.



Semantic Benchmarking Results					
Tech Industry			**Fashion Industry**		
	Companies			Companies	
Keyword	Fanatec	be quiet!	Keyword	Allbirds	Birkenstock
Quality	0.03	0.71	Delivery	0.37	-0.15
Price	0.24	0.14	Price	0.42	0.35
Delivery	-0.12	-0.30	Comfort	0.77	0.53
Gaming	0.49	0.51	Size	0.45	0.34
Setup	0.39	0.49	Color	0.53	0.59

Figure 6.2: *Company benchmarking results for companies in the fashion and tech industry.*

[3]The wording *topic modeling* here may be formally inadequate, but it reveals the underlying aim of clustering documents based on their inherent word vectors compared to the target word vector. Of course, since the target word has to be selected in advance, the results are biased to the extent that the topics are not created as a mean of distributed document representations.

The outlined strategy of document filtering based on semantic keyword search can be used to compare companies for a given set of categories. To aggregate the documents sentiment scores are used per category. In order to allow for a multidimensional perspective so called radar graphs are used to show the results, such as in figure 6.2. The results reveal clear differences of sentiment scores for e. g. gaming hardware producers. While *be quiet!* is able to outperform *Fanatec* in almost every category, documents belonging to *gaming* and *price* show less different sentiment scores between both companies. When comparing the shoe brands *Allbirds* and *Birkenstock*, the picture is similar. *Allbirds* seems to be more popular regarding the comfort of the shoes and the delivery process while prices, size and color variety only show small differences. Despite the illustrative outcome it has to be mentioned that the results are highly influenced by the number of comments that are scraped from each company profile.

6.3 Product naming approximation

The above application of word vectors can only be used if a more general set of words V' is used and is known in advance. Company- or industry-specific words, and thus documents, can also be extracted using word vectors without the need to know or make assumptions about V'. While the document filtering algorithm outlined above allows the creation of a filter mask for more general categories, in the sense that non-unique company-specific sub-vocabulary is required, a more company-specific vocabulary V_X can be extracted. Empirical experiments during the development of Customer2Vec have shown that applying cosine similarity, where w' is the brand name of a particular company of interest, reveals a very company-specific vocabulary V_X, while $V_X \in V$. This is due to the construction of $\vec{V}$, which again follows the logic of the skip-gram implementation of Word2Vec: Since words with similar context are represented by vectors close to each other in the vector space brand names and words associated with it are very close to each other because they are very different from the rest of the words in the vocabulary. The results of creating V_X can be separated into the products, called *product proxies* (px) and very unique words, called *brand context* (bx). An example of an embedding space including the company's brand names is shown in figure 6.3. From this point of view, two further analyses can be derived to analyse these two similarity consequences. Namely *Product naming approximation* and *Brand context recognition*. For the *Product naming approximation* the logic of the competitor benchmarking algorithm cannot be directly applied because it is not the documents containing similar words that are of interest, but the product proxies themselves. In order to extract px $S_C(\overrightarrow{brand}, \vec{w})\ \forall\ \vec{w}, \overrightarrow{brand} \in \vec{V}$ has to be calculated first. The result is a vector that gives knowledge about px for the respective company by looking at the cosine similarity scores.[4] Examples can be seen in table 6.3. Similar to the radar algorithm, a matrix $C^{\lambda}_{w'}$ is created using the identifier function $I_{w'}$, with the difference that w' is set to px:

[4] Another approach for px is to look at the respective website of the company in order to get the product names, but the word vector space reveals the product and also the name of the product the social media community uses. A third method would be to use the results of the company specific topics but then again a more manual search for the right topic would be necessary, therefore px are used for further analysis.

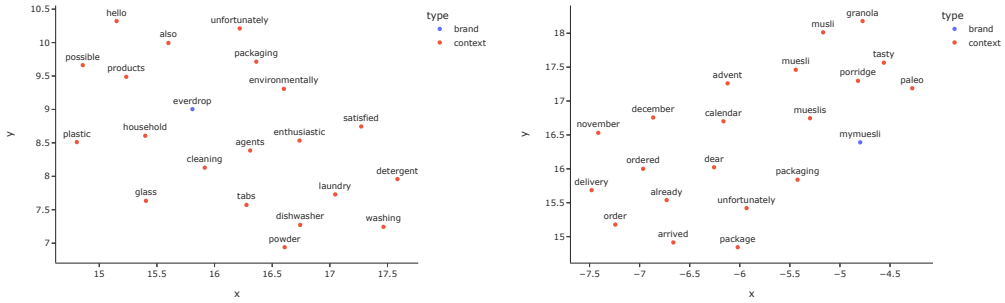


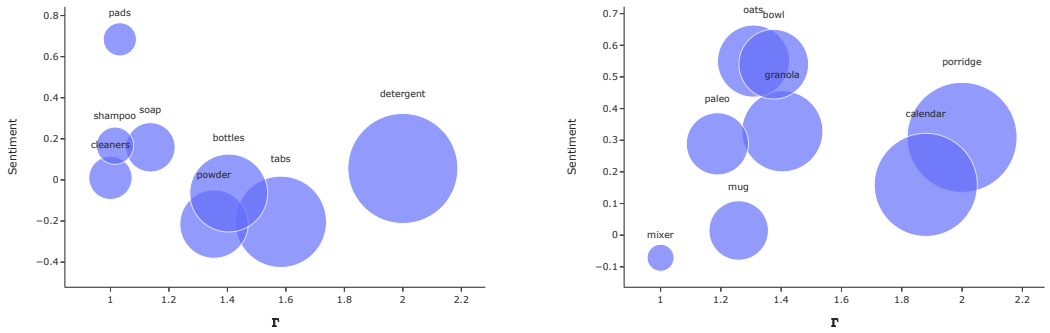
Figure 6.3: *Brand context words of everdrop and mymuesli brought to a two-dimensional representation via UMAP.*

Company	**Product Proxy**	$S_C(\overrightarrow{brand}, \vec{w})$
BrewDog	beer	0.67
	punk	0.53
	ipa	0.48
	elivs	0.41
	lager	0.37
Allbirds	runners	0.63
	loungers	0.40
	sandals	0.39
	animals	0.38
	laces	0.36
Fanatec	wheel	0.67
	dd	0.65
	pedals	0.61
	csl	0.60
	base	0.52

Table 6.3: *Exemplary product proxies from BrewDog, Allbirds and Fanatec.*

$$I_{px} = \begin{cases} 1, & \text{if } px \in d, \\ \text{otherwise} & 0, \end{cases} \tag{18}$$

Another difference is that the similarity between the product proxy and the words in each document does not need to be considered, as it is the product proxy and only the product proxy that is of interest. One might argue that similar product names refer to the same product, but then these words should also have a high similarity to the product and therefore to the brand name. Then I_{px} is used to create C_{px}. This allows to limit the corpus to only those documents that contain information about a certain product or group of products.



Product Proxies Results

everdrop				mymuesli			
px	**Score**	**Actuality**	**# Documents**	**px**	**Score**	**Actuality**	**# Documents**
detergent	0.06	2.00	627	porridge	0.31	2.00	466
tabs	-0.20	1.58	432	granola	0.33	1.40	253
powder	-0.21	1.35	242	calendar	0.16	1.88	409
cleaners	0.01	1.00	97	paleo	0.29	1.19	149
soap	0.16	1.14	126	mug	0.01	1.30	136
shampoo	0.17	1.02	72	oats	0.55	1.31	200
bottles	-0.07	1.40	314	bowl	0.54	1.38	187
pads	0.68	1.03	57	mixer	-0.07	1.00	28

Figure 6.4: *Product naming approximation results for everdrop and mymuesli.*

The further processing of the product proxies builds on the work of Jeong et al., 2019, who built a mining tool for product planning opportunities based on keyword-based topic modeling. They use the opportunity algorithm developed by Ulwick in 2005, which defines the opportunity as *Importance + (Importance - Satisfaction)*, where satisfaction was calculated by sentiment analysis and importance by calculating the frequency of mentioned product topics. However, this

paper uses an additional metric that reflects the degree of actuality referred to as Γ. In order to quantify the recent occurrence of words or topics associated with a particular company, the following logic is applied. Let Y be a set of unique years from a given set of documents C_{px}, sorted in ascending order. Then a second vector Θ is created, ranging between 1 for the lowest year value and the value of the length of Y for the highest year value. Then the years in Y are multiplied by the corresponding values in Θ. Finally the degree of actuality is calculated by:

$$\Gamma = Y' \times \Theta \tag{19}$$

This way it is ensured that the actuality measure has higher values for documents which have more current information about the word of interest and vice versa. Once *actuality* is defined product proxies are extracted using V_X. Then the sentiment of each individual document is calculated to reflect the mood around the named proxy.[5] Once the aggregated sentiment is calculated the respective actuality is determined following Eq. (19). The sentiment and actuality-based analysis is illustrated by scatter plots. Examples of the results for the companies *everdrop* and *mymuesli* can be seen in figure 6.4. The illustration shows the results of the procedure for the companies *mymuesli*, a producer of breakfast cereals, and *everdrop*, a producer of environmentally friendly cleaning products. The size of the bubbles reflects the number of documents in which the proxy appears. In the case of *everdrop*, consumers seem to have a positive sentiment towards the company's pads in the past, but are less satisfied with the tabs and detergents in more recent comments. The results for *mymuesli* show that the advent calendar and the product porridge is often discussed with moderate sentiment and that customers have been satisfied with the bowl and oats in the past.

6.4 Brand context recognition

In order to create a brand context specific vocabulary, the *product proxies* outlined above are removed from V_X. While the *product proxies* explain certain characteristics of the comments for the associated products, such as sentiment or timeliness, the assumption is that the remaining words in V_X reveal the context with which the company's brand name is associated. The company specific dataset C_X is then filtered for documents containing words listed in V_X, similar to I_{px}. Again, a threshold λ is used to define at what similarity the word is considered company specific. For the remaining words, the number of occurrences in the company-specific data is calculated, and all words with more than a certain number of occurrences are processed further.[6] Then C_X is filtered to determine the company documents in which the words in V_X occur and the word frequency of V_X is stored in C_X. While only word frequency information is processed up to this point, the context of the company specific words appart from the brand

[5] Please note that similar to the assumption that every document involves one topic, here the assumption is that if the product is mentioned the comment is all about it. This implies that if a user is commenting not only to one product but to several, the mood would be the same if the comment would only be about the proxy.

[6] A value of 100 showed the best performance in terms of the trade-off between showing enough context words and relatively high frequencies.

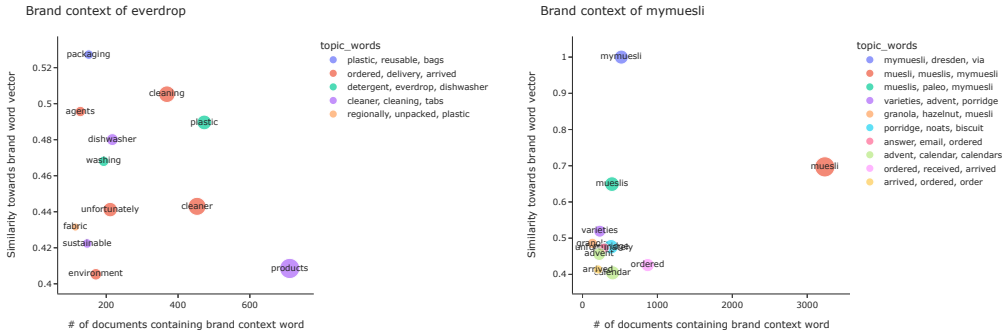


Figure 6.5: *Brand context recognition results of everdrop and mymuesli.*

name may be desirable. For this reason, the documents of the respective company, which contain information about the brand context, are processed through the topic modeling framework. This allows a closer look at the context of the underlying company words in their document environment, aggregated on a topic basis. The corresponding topic words are then used to gain further insight into the company words and mark the brand context, while the topic words are weighted by their occurrence in C_X. The results are summarised in a scatterplot showing the word frequency on the x-axis and the similarity towards $\overrightarrow{brand}$ on the y-axis, with the colour indicating the coherent topic in which the word occurs most frequently and the size indicating the frequency of the word within the topic. In this way, larger bubbles indicate the words that are more frequent in the clusters as a whole. Words with smaller bubbles are more reflective and specific to that theme. The brand context for *everdrop* in figure 6.5 shows that many users frequently talk about the companies washing and cleaning products and also about sustainability issues such as packaging or plastic. The results in the graph therefore suggest that the brand *everdrop* is perceived as environmentally friendly. In the *mymuesli* example, the picture is less clear. It seems that many documents around the brand name develop around the company's *advent calendar*, while also the ordering or delivery process is often discussed in the context of the brand. This leads to the conclusion that *mymuesli* is popular with its customers, especially due to campaigns such as the calendar, but that they communicate their negative experiences.

7 Discussion and outlook

This thesis proposes Customer2Vec, a social media mining tool for identifying and evaluating trends, topics and opinions' using distributed representations of customer generated data in vector space. To obtain data from the social web, scraping modules were implemented for the Instagram and Facebook platforms, collecting over 600,000 UGTD from 34 companies over an average period of 6 years. To structure the data, UGTD were first transformed into a numerical representation using Word2Vec and Doc2Vec, reduced to a lower dimension using UMAP and aggregated into clusters using HDBSCAN. The centroid of each C_t was then used to form the topic vector $\vec{t}$. While $\vec{d}$ and $\vec{w}$ were generated over the whole corpus, Customer2Vec allows a company specific application by reducing the document vectors to an entity unique matrix containing only documents that are inherent to the social media data of the respective company. This allowed the introduction of the distinction between global (full corpus) and local (company specific corpus) experiments. In order to parameterise the different algorithms appropriately, a simulation of the sensitivity of the PWI was performed. The results showed that the Top2Vec model is able to generate human interpretable topics from the distributed document representation without any ex ante assumptions about topic amount or content. However, the application of the Top2Vec algorithm revealed a high sensitivity of the parameters *minimum cluster size* to the topic size. In addition to topic modeling, the semantic nature of $\vec{w}$ and $\vec{d}$ together with sentiment analysis were used to develop algorithms for *Topic evolution and sentiment shifts*, *Competitor benchmarking analysis*, *Product naming approximation* and *Brand context recognition*. It is argued that topic modeling in combination with the additional applications offers the opportunity for an potential investor to gain an overview of discussed topics, reveal certain problems that customers have or have had with the company or the mentioned products and the associated mood. The initial suspicion of social media data being noisy and difficult to structure turned out to be true. The underlying UGTD included slangs, linkages and was also very unbalanced across the companies. However the individual companies still benefit from the application of the framework onto the whole sample. Additionally despite under-represented companies such as *tado* the framework was still able to capture company specific vocabulary and topics. For further research the results may be connected with the aim of predicting performance indicators of the underlying companies such as Revenues or Net Profit. However this research might be disturbed by some kind of survivor ship bias. Meaning that only successful companies, e. g. with a high Net Profit, are able to afford social media marketing and therefore receive a larger exposure of UGTD. Although the overall vector space for all companies is positive, it might be interesting to subdivide it according to the respective industries, which would eventually lead to industry-specific vector spaces. The advantage of this might be to capture domain specific topics. The overvall question if a distributed

vector space created from social media data is appropiate for a possible analytics tool can be answered positively. The work of Mikolov et. al. and the resulting Doc2Vec and Word2Vec models, as well as Angelov's framework, have contributed significantly to the implementation demonstrated in this work.

Bibliography

Angelov, D. (2020). Top2vec: Distributed representations of topics.

Choi, J., Oh, S., Yoon, J., Lee, J.-M., & Coh, B.-Y. (2020). Identification of time-evolving product opportunities via social media mining. *Technological Forecasting and Social Change*, *156*, 120045.

Retterath, A. (2020). *Essays on machine learning and the value of data in venture capital* [Doctoral dissertation, Technical University Munich].

Asur, S., & Huberman, B. A. (2010). Predicting the future with social media.

Scharfman, J. A. (2012). *Private equity operational due diligence: Tools to evaluate liquidity, valuation and documentation: Tools to evaluate liquidity, valuation, and documentation.* Wiley Finance Series.

Farzindar, A. A., & Inkpen, D. (2020). *Natural language processing for social media.* Springer International Publishing.

Stelzner, M. A. (2023). How marketers are using social media to grow their business. *Social Media Examiner.*

Kirtiş, A. K., & Karahan, F. (2011). To be or not to be in social media arena as the most cost-efficient marketing strategy after the global recession. *Procedia - Social and Behavioral Sciences*, *24*, 260–268.

Ballestar, M. T., Cuerdo-Mir, M., & Freire-Rubio, M. T. (2020). The concept of sustainability on social media: A social listening approach. *Sustainability*, *12*(5), 2122.

Kolajo, T., Daramola, O., & Adebiyi, A. A. (2022). Real-time event detection in social media streams through semantic analysis of noisy terms. *Journal of Big Data*, *9*(1).

Ramamonjisoa, D. (2014). Topic modeling on users's comments.

Usmani, U. A., Haron, N. S., & Jaafar, J. (2021). A natural language processing approach to mine online reviews using topic modelling, 82–98.

Jeong, B., Yoon, J., & Lee, J.-M. (2019). Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis. *International Journal of Information Management*, *48*, 280–290.

Conway, M., Hu, M., & Chapman, W. W. (2019). Recent advances in using natural language processing to address public health research questions using social media and Consumer-Generated data. *Yearbook of Medical Informatics*, *28*(01), 208–217.

Bail, C. A. (2016). Combining natural language processing and network analysis to examine how advocacy organizations stimulate conversation on social media. *Proceedings of the National Academy of Sciences*, *113*(42), 11823–11828.

Kanakaraj, M., & Guddeti, R. M. R. (2015). NLP based sentiment analysis on twitter data using ensemble classifiers.

Akuma, S., Lubem, T., & Adom, I. T. (2022). Comparing bag of words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology*, *14*(7), 3629–3635.

Alzami, F., Udayanti, E. D., Prabowo, D. P., & Megantara, R. A. (2020). Document preprocessing with TF-IDF to improve the polarity classification performance of unstructured sentiment analysis. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 235–242.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents.

Jipeng, Q., Zhenyu, Q., Yun, L., Yunhao, Y., & Xindong, W. (2019). Short text topic modeling techniques, applications, and performance: A survey.

Chen, Q., & Sokolova, M. (2018). Word2vec and doc2vec in unsupervised sentiment analysis of clinical discharge summaries.

Vayansky, I., & Kumar, S. A. (2020). A review of topic modeling methods. *Information Systems*, *94*, 101582.

Kherwa, P., & Bansal, P. (2018). Topic modeling: A comprehensive review. *ICST Transactions on Scalable Information Systems*, *0*(0), 159623.

Mohr, J. W., & Bogdanov, P. (2013). Introduction—topic models: What they are and why they matter. *Poetics*, *41*(6), 545–569.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence*, *3*.

Jeong, B., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*.

Blair, S. J., Bi, Y., & Mulvenna, M. D. (2019). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence*, *50*(1), 138–156.

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction.

Raunak, V., Gupta, V., & Metze, F. (2019). Effective dimensionality reduction for word embeddings.

Ding, C. (2009). Dimension reduction techniques for clustering, 846–846.

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space, 420–434.

Campello, R. J. G. B., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates, 160–172.

McInnes, L., & Healy, J. (2017). Accelerated hierarchical density based clustering.

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, *2*(11).

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors.

Cover, T. M., & Thomas, J. A. (2005). *Elements of information theory*. Wiley.

Patel, K., & Bhattacharyya, P. (2017). Towards lower bounds on number of dimensions for word embeddings. *International Joint Conference on Natural Language Processing*.

Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation.

Min, W. N. S. W., & Zulkarnain, N. Z. (2020). Comparative evaluation of lexicons in performing sentiment analysis. *Journal of Advanced Computing Technology and Application*, *2*(5), 8.

Hutto, C., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 216–225.

Agerri, R., Vicente, I. S., Campos, J. A., Barrena, A., Saralegi, X., Soroa, A., & Agirre, E. (2020). Give your text representation models some love: The case for basque.

Alan Akbik, D. B., & Vollgraf., R. (2018). Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics, pages 1638–1649.*

A Company Overview

Company Name	Business description	Country
action	*International discount retailer, mainly non-food products*	Netherlands
Allbirds	*Footwear retailer*	United States
Back Market	*marketplace for refurbished electronic devices. The Company's business model revolves around matching certified refurbished electronics dealers with electronics buyers.*	France
Bang & Olufsen	*Bang & Olufsen ("B&O") is a manufacturer of high-end audio and multimedia equipment. The Company's business model revolves around designing, manufacturing and selling audio and multimedia equipment including music systems, speakers, headphones and televisions.*	Denmark
be quiet!	*Producer of computer components. The Company's business model mainly revolves around the development and production of high-performance computer hardware for applications in video gaming and e-sports.*	Germany
Birkenstock	*Producer of sandals and closed shoes for use in everyday, leisure and work settings. The Company's business model primarily revolves around the development and production of premium comfort footwear as well as subsequent distribution*	Germany
BrewDog	*BrewDog is a brewery and pub chain, operating through a multi-channel distribution network mainly focused on craft beers. The Company's business model revolves around selling beer directly to customers in ~60 countries through its online site and >100 bars as well as by supplying beer to supermarkets and partnering bars.*	United Kingdom
Bugaboo	*Manufacturer of mobility products. The Company's business model revolves around designing, manufacturing and distributing a range of mobility products, which are primarily used to travel with infants.*	Netherlands
Coffee Fellows	*Coffee Fellows is a coffee shop franchise. The Company's business model primarily revolves around (i) the management and monitoring of franchises along with (ii) the operation of directly-owned shops.*	Germany
CUBE	*Manufacturer of bicycles. The Company's business model mainly revolves around the design, manufacturing and sale of bicycles, related accessories and equipment.*	Germany
everdrop	*Developer of sustainable cleaning and dishwashing detergents. The Company's business model mainly revolves around the development of dissolvable cleaning tablets and sustainable detergents.*	Germany
Face Reality Skincare	*The company has a clinic in San Francisco with products and protocols personalized based on each client's acne type and severity, and skin type.*	United States

Table A.1: *Company Overview I/III. Business descriptions were taken from GainPro and Forbes.*

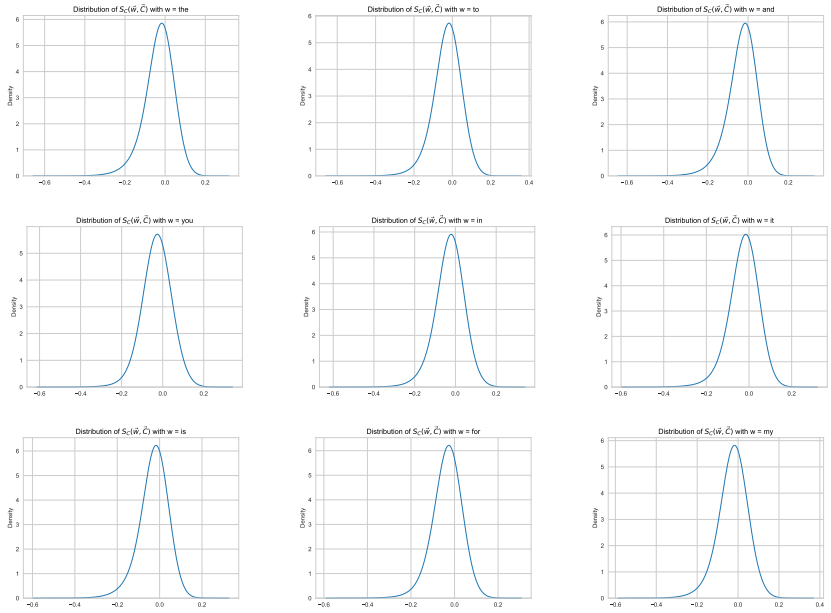
Company Name	Business description	Country
Fanatec	*Development of controllers, racing wheels, pedals, cockpits and connecting accessories for the main game consoles and the PC.*	Germany
Fit For Free	*Fitness chain operator. The Company's business model revolves around the operation of a chain of fitness centres in the Netherlands and Poland.*	Netherlands
Fractal	*Designer and wholesaler of premium gaming PC hardware. The Company's business model revolves around the design, marketing and wholesale of PC chassis, power Dsupplies, water coolers and fans for high-performance gaming computers.*	Sweden
Getaround	*Getaround is an online car sharing or peer-to-peer carsharing service that connects drivers who need to reserve cars with car owners who share their cars in exchange for payment.*	United States
Gymshark	*Online retailer of fitness clothing. The business model of the Company revolves around designing, manufacturing, marketing and selling sports apparel and associated accessories, for both women and men.*	United Kingdom
Hans im Glück	*Fast-casual burger restaurant franchise. The Company's business model revolves around the franchising of its eponymous brand and concept as well as the operation of its own establishments.*	Germany
Hessnatur	*Retailer of organic apparel and bed & bath goods. The Company's business model revolves around the design, marketing, distribution and sale of its non-chemically-processed and organic products.*	Germany
Hungryroot	*Offers online grocery service that delivers modern healthy food with recipe and meal planning support.*	United States
Kurt Geiger	*Luxury footwear and accessories retailer. The Company designs and distributes women's footwear, specialising in occasion-wear shoes, trainers, jewellery and related accessories.*	United Kingdom
Ledlenser	*Manufacturer of lighting equipment. The Company's business model mainly revolves around the manufacturing and distribution of LED flashlights and headlamps, both for professional and outdoor applications.*	Germany
meijer	*Meijer is a family-owned grocery chain operating in the Midwest.The supercenter stores offer groceries as well as departments such as fashion, automotive, home decor, health and beauty care, pharmacy, electronics and banking.*	United States
mymuesli	*mymuesli is a producer and distributor of customised cereals. The Company's business model mainly revolves around the procurement, development, production and distribution of food products as well as related non-food products and services in the food and nutrition sector.*	Germany

Table A.2: *Company Overview II/III. Business descriptions were taken from GainPro and Forbes.*

Company Name	Business description	Country
N26	*N26 is a digital bank offering a mobile banking platform and other related financial services. The Company's business model revolves around the offering of basic & premium bank accounts, savings accounts through partner banks distribution of proprietary and partnership credit products insurance services and cryptocurrency trading.*	Germany
OttosBurger	*Burger chain originated from Hamburg, Germany.*	Germany
Outfittery	*Curated online retailer of clothing and apparel. The Company's business model revolves around the online sale of >100 clothing brands, where the customer is supported by a personal stylist.*	Germany
Rent the Runway	*E-commerce platform that allows users to rent, subscribe, or buy designer apparel and accessories.*	United States
Simplon	*Manufacturer of bicycles. The Company's business model mainly revolves around the design and assembly of road, e-road, mountain, e-mountain, trekking bikes as well as e-bikes.*	Austria
Snocks	*Snocks is a socks and underwear D2C brand. The Company's business model revolves around the design, marketing and retail of high-quality functional undergarments for both sexes.*	Germany
Sorare	*Football platform based on non-fungible tokens ("NFTs"). The Company's business model revolves around creating and selling a limited supply of officially licensed digital football cards.*	France
tado	*Group of connected home thermostat developers and providers. tado's business model mainly revolves around the development and provision of smart home thermostats, which can be controlled via a self-developed smartphone app.*	Germany
Wehkamp	*Wehkamp is an online fashion retailer. As such, the Company's business model revolves around the operation of its e-commerce platform, through which it sells fashion products.*	Netherlands
Zeit für Brot	*Fresh-product organic bakery chain. The Company's business model mainly revolves around the production of bakery and pastry products.*	Germany

Table A.3: *Company Overview III/III. Business descriptions were taken from GainPro and Forbes.*

B Distribution of $S_C(\vec{w}_c, \vec{d}) \forall \vec{d} \in \vec{C}$



Word	the	to	and	you	in	it	is	for	my	of
25th Quartile	-0.31	-0.31	-0.31	-0.28	-0.29	-0.3	-0.29	-0.28	-0.28	-0.32
Mean	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03	-0.03
Median	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02	-0.02	-0.03	-0.02	-0.02
75th Quartile	-0.26	-0.24	-0.25	-0.22	-0.23	-0.24	-0.24	-0.23	-0.23	-0.26

Figure B.1: *Distribution of S_C between the vectors of the most frequent common words and each $\vec{d} \in \vec{C}$.*

C Sensitivity of minimum cluster size in HDBSCAN for topic detection

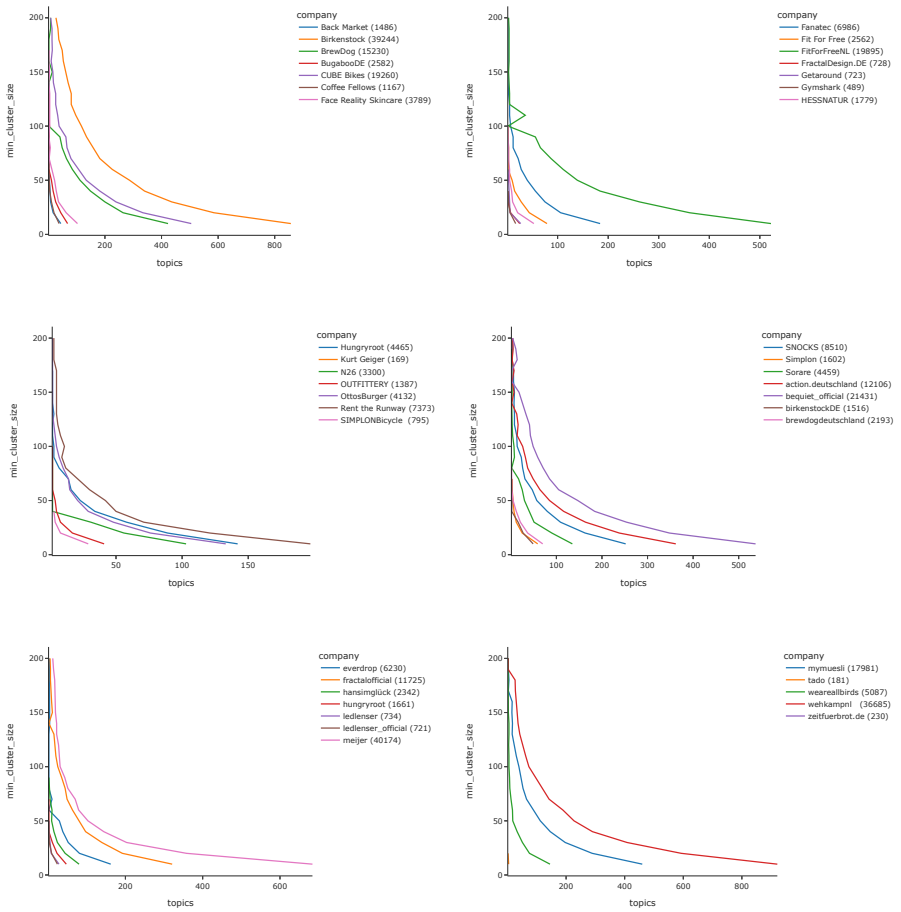


Figure C.1: *Simulation results of calculating topic vectors on company level with a $\gamma = 5$ and $\phi = 15$.*

D Document share labelled as noise by HDBSCAN

Company	Labelled as noise in %	# Clusters	UGTD count
Gymshark	0.47	9	415
zeitfuerbrot.de	0.38	3	248
OUTFITTERY	0.37	27	1,494
Back Market	0.32	33	1,633
ledlenser	0.31	28	1,483
Coffee Fellows	0.29	32	1,433
Hessnatur	0.27	39	1,878
Bugaboo	0.27	48	2,659
Getaround	0.26	37	1,831
everdrop	0.23	157	7,910
Simplon	0.22	70	2,773
Fanatec	0.21	140	6,030
Allbirds	0.19	131	6,159
Rent the Runway	0.19	157	7,619
Face Reality Skincare	0.18	94	4,164
N26	0.17	148	6,677
OttosBurger	0.16	117	4,724
Hans im Glück	0.16	97	4,025
Sorare	0.15	107	4,506
Kurt Geiger	0.14	6	221
BrewDog	0.14	379	20,653
Wehkamp	0.14	807	37,037
tado	0.14	6	328
be quiet!	0.13	475	21,850
mymuesli	0.13	432	20,162
Bang & Olufsen	0.13	113	4,796
CUBE Bikes	0.12	488	22,403
Birkenstock	0.12	931	49,092
Snocks	0.12	241	10,580
action	0.12	320	13,492
meijer	0.11	3,171	155,959
Hungryroot	0.11	144	6,321
Fractal	0.11	317	14,537
Fit For Free	0.11	550	23,984

Table D.1: *% - labelled as noise of HDBSCAN by company for chosen parameter setting.*

E Derivation of sentiment analysis framework

Sentiment analysis is a technique from the field of NLP and allows the quantification of the *mood* of a given document or token to be quantified in a numerical representation. From a more technical point of view, sentiment analysis is a binary or multi-class classification task to determine whether the text is positive, negative or neutral. While the underlying goal is the same for all approaches, they differ in their computational logic. A general distinction can be made between lexical-based and embedding-based sentiment classifiers. Lexicon-based emotion inference methods use predefined vocabularies in which each word is associated with a particular sentiment and polarity. While polarity is a float value within [-1,1], where 0 is neutral, -1 is very negative and 1 is very positive sentiment. As lexical-based techniques do not require pre-labelled data, they are highly dependent on the corpus on which they are built. Therefore, it may be worthwhile to investigate the degree of fit of the underlying data for the lexicon created before using it Min and Zulkarnain, 2020. In order to map the words from the text to the underlying lexicon, the BOW method can be used as a downstream task, and then simply aggregating the sentiment per word associated in the lexicon for an overall score. Very prominent methods of this approach are *Valence Aware Dictionary for Sentiment Reasoning* (VADER) and TextBlob. The heuristics in VADER capture not only the desired sentiment and polarity score, but also the *strength* of the underlying emotion, referred to as *intensity*. What *intensity* does is capture the fact that, for example, *'This product is great'* is less positive than *'This product is GREAT!!! <3'*. Shifts in polarity, e.g. detected by *'but'* within a document, are also reflected by weighting the rating of the sentence, e.g. *'The product is great, but it's way too expensive'*. Hutto and Gilbert, 2014 use these exemplary heuristics to mimic the way a human would interpret the text, thereby improving the results. Another approach is the famous TextBlob lexicon created by Steven Loria. While VADER covers 7,052 words, TextBlob has only 2,919, but is also very popular because of its convenient API. Flair belongs to the so-called embedding-based types of sentiment analysis and uses document and word vectors similar to those described above. Flair employs pre-trained language models that can be used to generate contextual embeddings (Agerri et al., 2020). The algorithm uses recurrent neural networks, which in the context of textual data means that the model predicts the next character based on previous characters. The framework is trained on the IMDB dataset, which contains various movie reviews. [1] A complete explanation of the technicals can be found in Alan Akbik and Vollgraf., 2018. In order to derive the most suitable sentiment analysis framework for Customer2Vec, a small data sample of 300 documents was

[1] The IMDB dataset can be downloaded from https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

randomly selected and pre-labelled with the convenient labels 0 (neutral), 1 (positive) and -1 (negative). The three proposed sentiment classifiers were then tested on this sample without text pre-processing. The results were analysed for each category of neutral, positive and negative, focusing on the correct decision of positive and negative sentiments with varying threshold.

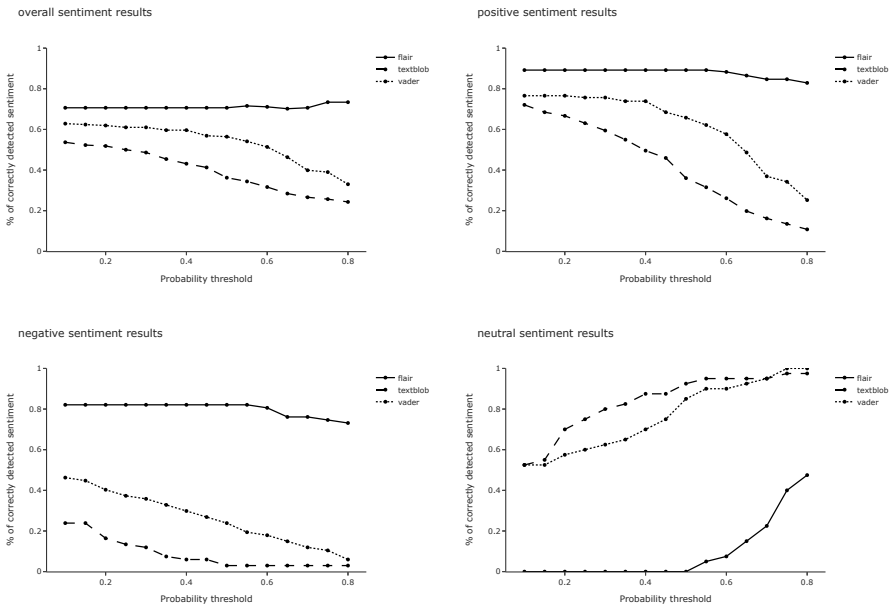


Figure E.1: *Sentiment analysis benchmarking.*

The results show that the Flair sentiment framework has a very high accuracy in detecting positive and negative sentiments, but suffers in correctly assigning a neutral score to the 0-labelled documents. TextBlob and VADER show similar behaviour with respect to the different label categories, with stronger performance in detecting neutral sentiments and weaker performance in detecting negative sentiments. Looking at the overall results and assuming an intuitive threshold of 0.5, Flair is able to outperform VADER and TextBlob and is therefore used in Customer2Vec whenever sentiment analysis is applied.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.
Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Passau, den 9th June 2023

.....
(Fabian Dick)