Stuttgart Media University
Computer Science and Media

Gossip Insights

A Graph-based Keyword Extraction & Visualisation for Contextual Analyses of Social Media Trends

A thesis submitted to the faculty of 'Print and Media'
in partial fulfillment of the requirements for the
academic degree of 'Master of Science'.

Stuttgart, Summer 2018

Felix Heck, B.Sc.

Supervisors
Prof. Dr.-Ing. Johannes Maucher
Dr. Dan Chalmers
Yanick Nedderhoff, M.Sc.

To my beloved Bennetton Burger,
double with halloumi and mild chili sauce.

Thanks for your love, endless support
and encouragement along the way.

Abstract

When analysing social media trends with the help of Brandwatch Analytics, a Topic Word Cloud is available in addition to a large number of different visualisations, which represents the most prominent key words and sentences in the selected time interval.

Although the Topic Word Cloud provides a good overview of discussed topics and diverse insights into trends, further information, including contextual relationships between word groups, is lost in this visualisation. These can be useful for the rapid exploration of several discussed topics and their significance in the selected time interval. The aim of this thesis was to design and implement a prototypical solution to the problem, so that important topics can be extracted, visualised and explored in a contextual manner.

For this purpose, the problems outlined were analysed first, and on this basis, the requirements were specified and prioritised. In the scope of the conceptual design and implementation, a multi-stage process was developed which extracts significant keywords from a collection of tweets and ranks them by various heuristics and a graph; the constructed graph was also used for the visualisation of the keywords. Thereupon, the implementation resulting from the pilot experiment was qualitatively evaluated with stakeholders so that solutions could be worked out for identified weak points.

The result of this paper is a prototype that not only compensates for conceptual disadvantages of the Topic Word Cloud visualisation, but also optimises the underlying keyword extraction by considering various statistical heuristics.

Keywords: Natural Language Processing, Web Mining, Information Retrieval, Graph Theory, Data Visualisation, Twitter

Zusammenfassung

Bei der Analyse von Social Media Trends mit Hilfe von Brandwatch Analytics steht neben einer Vielzahl an unterschiedlichen Visualisierungen zudem eine Topic Word Cloud zur Verfügung, welche die prominentesten Schlüsselworte und -sätze im selektierten Zeitinterval darstellt.

Obwohl die Topic Word Cloud einen guten Überblick über diskutierte Themen und vielseitige Einsichten in die Trendentwicklung ermöglicht, gehen weitere Informationen in dieser Visualisierung unter; so auch kontextuelle Zusammenhänge zwischen Wortgruppen. Jene können der schnellen Exploration mehrerer Diskussionsthemen und deren Bedeutung im selektierten Zeitinterval zugutekommen. Diese Arbeit hat zum Ziel, hinsichtlich der Problemstellung eine prototypische Implementierung zu konzipieren und zu realisieren, sodass bedeutende Diskussionsthemen extrahiert sowie mit kontextuellen Relationen visualisiert und exploriert werden können.

Dazu wurde zunächst die beschriebene Problematik analysiert und darauf basierend entsprechende Anforderungen spezifiziert und priorisiert. Mit Hilfe dieser konnten bestehende Techniken und Ansätze identifiziert und evaluiert werden. Im Rahmen von Konzeption und Implementierung wurde ein mehrstufiger Prozess entwickelt, der aus einer Sammlung von Tweets signifikante Stichworte extrahiert und diese auf Basis verschiedener Heuristiken und eines Graphen bewertet. Der konstruierte Graph wurde darüber hinaus auch für die Visualisierung der Stichworte verwendet. Die aus dem Pilotversuch resultierende Implementierung wurde daraufhin mit Interessenvertretern qualitativ evaluiert, sodass für indentifizierte Schwachstellen Lösungsansätze erarbeitet werden konnten.

Das Ergebnis dieser Arbeit ist ein Prototyp, der nicht nur konzeptionelle Nachteile der Topic Word Cloud Visualisierung ausgleicht, sondern durch die Miteinbeziehung verschiedener statistischer Heuristiken auch die grundlegende Extraktion der Schlüsselworte optimiert.

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms

Nomenclature

Acronyms/Abbreviations

API	Application Programming Interface
BTM	Biterm Topic Models
IR	Information Retrieval
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
NER	Named-entity Recognition
NP	Noun Phrase
POS	Part of Speech
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
URL	Uniform Resource Locator
UX	User Experience
VSM	Vector Space Model

Glossary

Component	Modular visualisation tools which make up a dashboard and provide different data.
Dashboard	Multiple charts, summaries and other components which enable to analyse found mentions.
Mention	A content matching terms defined in a query like a webpage or a comment on social media.
Query	Search string including boolean and more advanced operators used to find online mentions.
Background Data	Data relating to the period before the actual peak or event. Other peaks are usually excluded.
Peak Data	Data relating to the period of the actual peak or event.

Chapter 1

Introduction

This chapter first introduces the problem that motivates the research in this thesis by, among other things, providing insights into the analysts' workflow and the issues that arise in the process. Subsequently, the requirements and objectives are defined and prioritised by the definition of the problem. Besides this, the thesis is delimited to clearly define the scope of the research. Finally, the structure of the theoretical work is outlined.

1.1 Motivation

Brandwatch Analytics is a software that enables real-time analyses of social media data from social media, blogs, forums as well as news and review sites by collecting mentions[1]. For this purpose the application continually gathers data using different methods like custom web crawlers, direct relationships to data providers or usage of an Application Programming Interface (API). Those methods use queries[2] defined by customers or internal research analysts as exemplified in listing 1.1. On top, provided dashboard[3] templates and components[4] allow to get quick insights into trends and further to customise the visualised data [1].

[1] A content matching terms defined in a query like a webpage or a comment on social media.
[2] Search string including boolean and more advanced operators used to find online mentions.
[3] Multiple charts, summaries and other components which enable to analyse found mentions.
[4] Modular visualisation tools which make up a dashboard and provide different data.

Listing 1.1: Exemplary Query

```
site:twitter.com AND (
  at_mentions:wagamama_*
  OR wagamama OR ("restaurant chain" NEAR/10 (asian OR japanese))
)
```

In case of significant peaks in line charts visualising the volume of mentions per time interval (figure 1.1), it might be of great use to quickly identify the most discussed topics to be able to react responsively to trends [10].

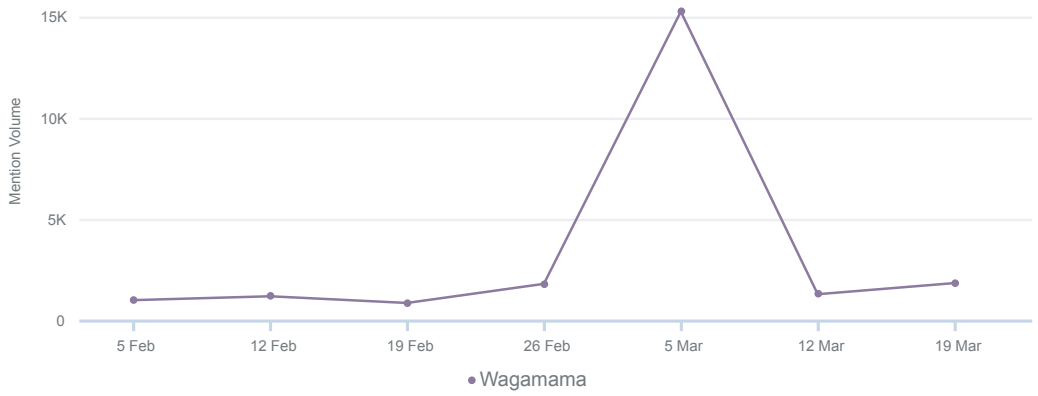

Figure 1.1: Exemplary Line Chart Component

A Line Chart Component related to the query shown in listing 1.1 including lots of background data and a significant peak.

The use of the Topic Word Cloud component enables the user to identify the most common terms in relation to the peak data[5]. More precisely, word clouds are usually an accumulation of n-grams, which occur most frequently in the corresponding corpus. Particularly frequently occurring n-grams are usually displayed in larger font sizes and are more likely to be placed in the centre (figure 1.2). Occasionally the visualisation is supplemented by the use of colours, which represents a specific segmentation; sentiment or named entities are examples of this.

[5] Data relating to the period of the actual peak or event.

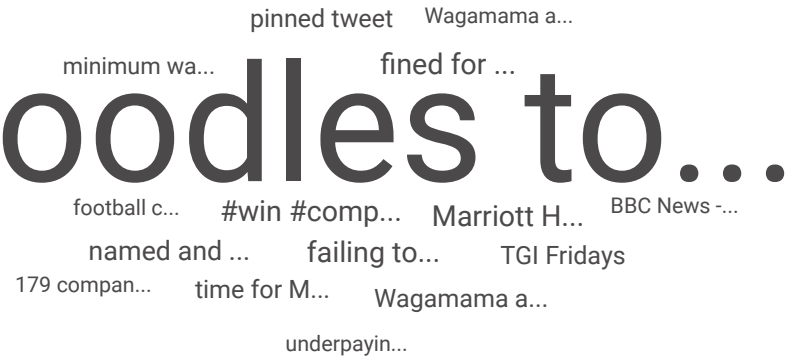

Figure 1.2: Exemplary Topic Word Cloud
The Topic Word Cloud component related to the peak shown in figure 1.1 covering a single discussed topic.

Although the Topic Word Cloud provides a good overview of discussed topics and a wide range of insights into trends, such as the visualisation of growth or sentiment based on a background analysis of time series, some disadvantages have been worked out in cooperation with data scientists and research analysts. In the following, these points of criticism, which are mainly due to the extraction and visualisation of contextual relationships or more generally to the design of the word clouds and how they are composed, are described in detail.

Since the axes of the word clouds often have no particular meaning, terms are arranged randomly and placed in such a way that only a minimum of free space remains. Due to this comparatively simple visualisation, a lot of mostly context-related information is lost. This leads to the first problem to be addressed in this thesis. The word cloud often contains several terms that belong to the same topic; however, this fact is not immediately apparent. To be able to recognise these relations, it is usually necessary to analyse the content and context of the respective terms by examining the mentions more closely.

Apart from the visualisation itself, the algorithm that extracts the terms from the corpus is also responsible for the loss of contextual information. To have a

better understanding of the problem, the individual steps of the current algorithm for selecting the n-grams are sketched (algorithm 1.1). It is worth noting that the sampling size was chosen to achieve an appropriate balance between accuracy and computing time, and that the algorithm contains further intermediate steps, which are, however, of no relevance to this problem.

Algorithm 1.1: Current Topic Word Cloud Algorithm

Input: corpus – A sample of up to 600 documents based on a query
blacklist – A list of blacklisted n-grams
threshold – A threshold for a n-gram's minimum frequency
Output: List of n-grams/score pairs

```
1  chunks ← split corpus by punctuation
2  ngrams ← extract n-grams from each chunk
3  terms ← empty list of future keyword candidates
4
5  def isRelevant(ngram) is
6      return ngram ∉ blacklist, terms and frequency(ngram) ≥ threshold
7
8  def getScore(ngram) is
9      return calculated score based on ngrams, its frequency and length
10
11 foreach ngram ∈ ngrams if isRelevant(ngram) do
12     score ← getScore(ngram)
13     append [ngram, score] to list of terms
14
15 return terms
```

With this background, it is now possible to describe the remaining problems which are mainly related to the n-gram's frequency and consequently to the lines 6 and 9 in the algorithm 1.1.

Interactions between users are ubiquitous in social media so that single posts are both referred to or commented on directly. This results in threads, whose individual posts might differ substantially in their choice of words. The reason for this is that, as in normal conversations, reference is usually made to the initial statement and the use of specific context-giving terms is not essential. As a result, terms are not seen as relevant because they do not occur frequently enough,

although a single topic is discussed in the narrower sense. Those relationships are also neglected in the Topic Word Cloud.

Furthermore, especially on Twitter, it is quite usual to share other posts. These shared tweets are also known as retweets and may also become viral in some cases. However, other pages enable the sharing of articles with prepared texts as well. This leads to the fact that there might be multiple mentions with the same choice of words. In the case of particularly popular retweets, their terms often emerge in the Topic Word Cloud, whereby other terms are entirely suppressed or partly disappear (figure 1.3), even though other topics were also discussed more frequently.

Figure 1.3: Topic Word Cloud with an Emerging Term

The problems outlined above result in analysts being deprived of contextual information that can be useful for analysing trends and identifying topics of interest to the company. As a result, those analysts must spend additional time extracting such information or to verifying that the visualisation is complete. Addressing these problems can lead to financial and time savings.

1.2 Objectives

This thesis aims to reduce the additional time-consuming tasks mentioned in Chapter 1.1 by conceptualising and realising a prototypical application. All identified problems were addressed as far as possible to provide analysts with intuitive and straightforward contextual insights into discussed topics and discussions in general in the future. The following questions are to be answered:

- Which individual steps of preprocessing might be necessary to handle different kinds of anomalies and specifics in social media content?
- What are common linguistic characteristics of keywords and key phrases in order to simplify their selection and extraction?
- How can proper keywords and key phrases be extracted and thereby ranked or rather weighted according to the requirements?
- How can threads and shared posts be taken into account, so that neither contextual links are lost, nor less prominent terms are suppressed?
- Which linguistic or statistical criteria or Natural Language Processing techniques can be used to shape contextual clusters algorithmically?
- How can the shaped clusters be visualised properly to quickly identify both discussed topics and their potential relations?
- Assuming there are multiple clusters of terms which are related to each other: how can a single topic be assigned to the individual clusters?

Thus, both the extraction of terms and their corresponding visualisation are part of the implementation. The detection of events is explicitly not part of the work, even if a more detailed delimitation is made in the further progress of the thesis in Chapter 1.4. All in all, event detection is excluded, as it would exceed the scope of this thesis and is already being researched internally. Which particular techniques and approaches are used remained open; these decisions were based on the analysis of related work and were made as part of the design process in Chapters 3 and 4.

1.3 Requirements

The requirements can be derived and prioritised based on the aforementioned research questions, the analysis of the existing approach and insights into its feedback.

Firstly, the number of mentions which are required for the algorithm to extract the keywords should be kept as low as possible to enable fast processing without losing accuracy and completeness. Secondly, the stored data is not annotated with

keywords, so the approach has to be unsupervised. Additionally, the mentions should be preprocessed and normalised adequately to ensure the highest possible quality of the extracted keywords; this is especially important for social snippets like tweets due to the high noise [97]. For the prototypical implementation, the focus will be on Twitter; more detailed reasons are described in Chapter 1.4.

Table 1.1: Prioritisation of the Requirements

Priority	**Requirement**
Premises	
obligatory	The algorithm is unsupervised
obligatory	The algorithm does require a relatively low number of mentions for meaningful results
obligatory	Document corpus is preprocessed and normalised for better generalisation and therefore better results
obligatory	Focus on Twitter content
Extraction	
obligatory	The algorithm should extract n-grams with $n \geq 1$
obligatory	Take contextual relationships into account to visualise multiple discussed topics
obligatory	Extract & visualise threads if available and relevant
optional	Take retweets and shared posts into account, especially regarding the visualisation
Clustering	
obligatory	Cluster the resulting n-grams to visualise contextual relationships as well as threads
obligatory	Enable an interactive visualisation
optional	Define a single n-gram as general topic per cluster

Furthermore, the keyword extraction should not only extract unigrams, but also n-grams ($n > 1$) with the help of linguistic characteristics, as additional surrounding words provide a more detailed context. Based on these requirements, which are mainly premises, the extraction itself as well as the weighting of the keywords must take into account the problems explained in Chapter 1.1, so that those do not have any negative influence on the future prototype. Finally, the extracted keywords must be clustered in such a way that contextual relationships are taken into

account. In addition to the algorithmic process, the resulting clusters must also be visualised and, if possible, assigned a single n-gram per cluster. An interactive visualisation is intended to enable the user to examine specific parts in more detail and to highlight these in presentations.

The requirements defined above are listed, categorised, prioritised by descending relevance for clarity in table 1.1. The temporally limited resources are the reason for the prioritisation. The practicability of the requirements is oriented to the respective functional scope and the associated efforts. Some optional requirements complement the obligatory ones and therefore presuppose them.

In general, the mandatory requirements are sorted in descending order of priority. Since the first requirements are premises and are the basis of further requirements for the implementation, they are mandatory. Extraction is the core of the prototype and is therefore the most important. Apart from the consideration of retweets, all requirements are obligatory. The focus is on contextual features in particular. Concerning clustering, clustering for visualising contextual information and interactions for a more straightforward examination are mandatory and thus corresponds to the prioritisation of extraction requirements. The assignment of individual topics to clusters is not essential for the prototype.

1.4 Delimitations

In the following, the partial aspects of the theoretical work of similar sub-areas are delimited in order to define the scope clearly; furthermore, the respective decisions are justified. In particular, reference is made to the choice of Twitter as the single source of data and the delimitation between event detection and description.

1.4.1 English Twitter Data as Single Source

First of all, the reasons for choosing Twitter as the single source of data are explained in more detail, since it is defined as a premise; both technical and content-related reasons are considered.

Twitter is one of the most important microblogging services that enables users to express their opinions or discuss topics interactively. This results in large amounts of information, which can be caused by personal as well as local or global events such as disasters and social movements [10, 46, 53, 86, 93]. For instance, 1 % of the public Twitter stream already covers about 95 % of all events deposited with news agencies [81,114]. Because the high amount of tweets covers a wide variety of event types, it is possible to identify and describe events using Twitter data only, even if the length of posts is limited [49, 59] and tweets contain a high linguistic noise [59, 97].

Twitter also offers a very high coverage in the Brandwatch ecosystem, as the provided API allows easy access to tweets and is further simplified by the Twitter Official Partner Program [53]. Thereby, using the Twitter GNIP service grants full coverage in real time [1]. Full coverage, in this case, means that all data related to queries is available. Due to the high coverage and the content characteristics, the prototype is mainly built on Twitter mentions.

Currently, Instagram provides the second best coverage. Due to the depreciation of parts of the API during 2018, Instagram will not be considered in this prototype [38].

Since most of Brandwatch's customers focus on English language content as well as the fact that English is the most widely used language on Twitter [75], this pilot experiment is initially designed for English language data.

1.4.2 Event Detection versus Description

The general topic will be narrowed down and delimited from event detection. As the objectives and requirements of the previous chapters indicate, this thesis focuses on the description of events and not on their detection. Nevertheless, both tasks are described to emphasise the differences. For a better understanding of an event, the characterisations of an event must be explained in more detail.

Primarily, a differentiation must be made between regular and relevant events. In general, an event can be defined as happening in the real-world that evolves over space and time [8, 89]. Thus, the difference to relevant events or rather anomalies

Figure 1.4: Data with Noise in the Form of a Single Peak [25]

is apparent in figure 1.5, which consists of a series of the single event shown in figure 1.4. In a broader view with more data, individual peaks converge into patterns so that peaks are typical and expected; there is not any relevance. To detect relevant events, the usual patterns must be known [25, 89].

> 'Anomalies are defined not by their own characteristics, but in contrast to what is normal.' – T. Dunning and E. Friedmann [25]

The detection of such anomalies is an independent field of research based on the application of various models. Since, on the one hand, the scope of the thesis is narrowly defined and, on the other hand, research is already being practised internally in this field, event detection is excluded. In contrast, the detected events and their data are used to describe those events in which an attempt is made to identify and visualise their cause.

Figure 1.5: Heartbeat Sequence of Regular Peaks with a Single Anomaly

1.5 Outline

The theoretical part of this thesis is structured as described below: an introduction to the topic as well as the definition of the problem and the objectives, the review of related work in the area of keyword extraction and community detection, the conceptual design and insights in implementation details, the evaluation, and the conclusion, including outlooks.

To provide an overview of existing research in the field of keyword extraction and community detection as well as to derive possible approaches and techniques, related research will be reviewed and analysed in Chapter 2.

Chapter 3 describes the iterative conceptual design and realisation of the prototypical algorithm as well as its steps based on the specified requirements and tasks. With the help of various extracted datasets, the algorithm is evaluated subsequently by each iteration. Moreover, the implementation gets qualitatively evaluated conclusively in cooperation with internal stakeholders of various roles. After evaluating the algorithm, for any identified weak points, potential approaches are worked out in Chapter 4.

Finally, Chapter 5 reflects on the algorithm, its overall performance and outlooks for further development.

Chapter 2

Background and Related Work

After narrowing down the thesis to the description of events, it remains open on the basis of which concepts this is supposed to be done. In general, topic modelling and keyword extraction can be mentioned as core concepts for information retrieval. There is also automatic document summarisation, which instead attempts to compress the content of the document to the essential lines [88]. Therefore, in the following, both concepts are introduced and related research is shown and evaluated in the context of the problem.

2.1 Topic Modelling

Topic modelling is the use of probabilistic statistical and mathematical techniques like as singular-value decomposition, which attempt to infer main topics [88]. Since there are different models for inferring topics, these are outlined below.

Deerwester *et al.* propose Latent Semantic Analysis (LSA), an improved variant of the TF-IDF scheme. Thus, the model utilises singular-value composition on a TF-IDF matrix with the aim to identify linguistic notions [91].

Hofmann proposes an enhancement called probabilistic Latent Semantic Analysis (pLSA) which models each word as a sample from a mixture model [42]. Sridhar outlines that pLSA does not perform well on documents which are not included in the training data and tends to be overfitted [96].

Models based on Latent Dirichlet Allocation (LDA) overcome the limitations of pLSA and generate clusters of distinguishable topics by identifying hidden and latent semantically related structures [15]. This technique is especially designed for the extraction of distinguishable topics from a wide variety of documents [40, 50, 88]. Also, LDA generalises well [96]. However, it may be seen as disadvantageous that the number of topics to be extracted must be determined before the process starts [47].

Even though pLSA and LDA have proven their value in long documents, they are not very suitable for short documents such as tweets [44,96,109]. Short texts lead to sparse contexts which complicate to identify senses of ambiguous words due to the lack of long-distance syntactic or semantic dependencies [96, 109].

Hong and Davison, as well as Steinskog *et al.* have addressed this issue and proposed pooling techniques to merge tweets into pseudo-documents according to certain characteristics, such as hashtags or authors [44, 97].

Zuo *et al.* propose the Word Network Topic Model to handle short documents with LDA. In a first step, a document for each term is created by merging contexts of the term in the corpus. Those documents and LDA are used to infer topics [115].

The mentioned weaknesses of conventional models in the context of short documents are addressed by Biterm Topic Models (BTM), as research by Yan *et al.* shows. In this case, the generation of unsorted word pairs that occur together in the document is modelled directly. As a result, BTM performs better than LDA and pLSA not only for short documents but also for longer documents [109]. Jonsson and Stolee verified these results [50].

Sridhar proposes to model a '[...] low-dimensional semantic vector space represented by the dense word vectors using Gaussian mixture models [...]'. The approach outperforms LDA on short documents since it does not rely on long distance syntactic or semantic dependencies [96].

Schneider proposes to use an inference algorithm additionally, which automatically identifies keywords for topics based on the assumption that keywords influence the topic assignments of nearby words. Aside from determining keyword-topic values, the number of topics is also regulated [90].

2.2 Keyword Extraction

The process of keyword extraction is a simplistic topic model and attempts to automatically identify the terms in an unstructured text that best describe the topic of the document [12, 65, 88]. The terminology for defining those terms includes keywords as well as keyphrases and key terms [12]. Keyphrases usually consist of several terms and are therefore n-grams with $n > 1$. Nevertheless, all terminological variants serve to characterise topics discussed in the document. More precisely, keywords can be used to index, classify and summarise a document or collection of documents [77].

Keyword extraction can be structured according to Ping-I, Shi-Jen and Zhang into statistical, linguistic, machine learning-based and other approaches [22, 113]. Machine-learning based approaches are mostly disregarded in the subsequent consideration. The reason for this is that supervised approaches are particularly common in this area, but cannot be used due to the absence of annotations. Therefore, the focus is on unsupervised approaches that do not contain any learning components [12, 14]. Below, these approaches are briefly characterised and research in the respective field is outlined.

2.2.1 General Unsupervised Approaches

HaCohen-Kerner presents a model based on research that shows that extracting keywords from titles and abstracts is successful. The extracted n-grams, where $1 \leq n \leq 3$, are weighted based on their full and partial occurrences [39].

Pasquier outlines a single document approach based on sentence clustering and LDA. Sentences are clustered using different algorithms to represent semantic relations. Each cluster represents a topic and keywords can be extracted [78].

Another approach for individual documents is proposed by Pudota, which uses linguistic and statistical features of n-grams, among others. The model is domain-independent [84].

Yang *et al.* propose an approach based on Shannon's entropy difference to define a new metric for ranking words' relevance. The entropy difference between the

intrinsic and extrinsic mode is used because it is assumed that in comparison to irrelevant words, the important words are intentionally placed [111].

Li *et al.* implement an approach which clusters keyword candidates by a semantic relatedness factor based on co-occurrences and an external knowledge source. Finally, KeyCluster extracts keywords per cluster based on POS patterns and rules [61].

Alrehamy and Walker propose another clustering-based model called SemCluster which uses internal ontologies and external knowledge sources to identify semantically relevant keyword candidates, to cluster these candidates and to extract the most relevant ones [9].

2.2.2 Statistical Approaches

Statistical approaches use simple methods that require no training and are also independent of language and domain. Statistics of words or n-grams are used to identify keywords in documents [22, 113]. The term frequency (TF) is one of the most important factors and is the basis for further statistics such as TF-IDF [60, 93]. Another metric is the standardised variable, also known as z-score, which indicates the deviation from the mean in units of standard deviation, which normalises the frequency [36, 60, 94]. Other examples include word co-occurrences and the PAT tree. The former statistic expresses n-grams co-occurring within a defined window, a sentence, paragraph or document [12]. These methods perform inadequately, especially in scientific papers, since essential words are only rarely mentioned and may therefore not be taken into account by the statistically empowered model [22, 113].

Sayyadi *et al.* and Li *et al.* argue that longer words or n-grams with a higher n contain more concrete information than short n-grams [58, 89], and a longer n-gram 'is likely to construct a more semantically specific phrase' [24].

Based on this assumption, the actual TF-IDF algorithm can also be extended, as Danesh *et al.* illustrate. In this case, the TF is reduced by the number of n-grams representing a superset of the term to be evaluated [24].

Li *et al.* propose to take the position of an n-gram in the underlying document into account based on the assumption that keywords are likely located within the title or first paragraph [58]. So Medelyan and Witten use a first occurrence heuristic with a linear decay function, too [63].

Lynn *et al.* propose a model called SwiftRank which utilises the position of sentences and avoids both language dependence and linguistic pre-processing such as named-entity recognition (NER) or POS tagging [62].

Also there are several approaches for mapping the co-occurrence, like Danesh *et al.* propose. In this case, not only the number is taken into account, but also the distance between the respective terms. A decay function is used here as well, which weights the pair of n-grams less as the distance increases [24].

2.2.3 Linguistic Approaches

Linguistic approaches that use linguistic properties of text parts include lexical, syntactic and discourse analysis [113]. For this purpose, the results of POS tagging and NER are used as well.

Hulth and Li Z. *et al.* propose the extraction of noun phrase (NP) chunks or n-grams using POS patterns, which are similar to NP chunks, to improve the results of a term selection approach. This generalises linguistic properties of common keywords [45, 59].

Li *et al.* propose to use NER and the resulting entities to extract keywords based on an idea similar to the proposal above. It is assumed that events can be described by extracting temporal, spatial and personal entities which are mainly part of NP chunks. [58].

Ritter *et al.* figured out that capital letters are used to emphasise terms in social snippets [86]. This intentional emphasisation by the author can be taken into account when extracting or ranking keywords.

Ercan and Cicekli propose extracting keywords with lexical chains since it is assumed that keywords are semantically related to the underlying text [28].

2.2.4 Graph-based Approaches

A graph is a mathematical model, which enables to explore relations and structures efficiently [18]. Graphs address the problem of missing contextual information in Vector Space Models (VSMs) and have in common that a text source is modelled as a graph by representing terms by nodes and connections by edges. The edges can represent various metrics and relations like co-occurrence, syntax and semantics [37, 41, 64, 89, 95]. The basic idea is to evaluate the graph by ranking the importance of individual nodes [21, 41]. Therefore, the graph-based approaches tend to combine several of the approaches already mentioned [12].

Ohsawa provided the initial approach with KeyGraph to cluster graphs with the help of co-occurrences and thus assign several topics to a document. In addition, statistical features are used to rank terms [73].

Erkan and Radev implement a stochastic approach called LexRank which calculates the importance of sentences based on the eigenvector centrality [29].

Mihalcea and Tarau propose a keyword and sentence extraction approach derived from PageRank which models weighted co-occurrence networks using a variable window and POS filters [65].

Palshikar proposes a single document approach which combines structural and statistical features to build an undirected graph. The edges are annotated with a dissimilarity measure between the connected words. Central nodes within the graph are keyword candidates [77].

Grineva *et al.* utilise community detection in a weighted and directed graph of semantic relationships between terms to extract keywords. The results indicate that the terms of the most relevant topics in the document tend to form a thematically coherent group [37].

Tsatsaronis *et al.* propose SemanticRank, which uses a knowledge-based measure of semantic relatedness between keywords to indicate semantic relationships [102].

TopicRank by Bougouin *et al.* is a combination of clustering-based and graph-based techniques. Extracted noun chunks are represented as nodes in a graph

and clustered into topics which are ranked using TextRank. Finally, a keyword for each relevant topic is selected [20].

Twitter Keyword Graph by Abilhoa and de Castro is an approach which represents tweets as graphs. The approach uses several pre-processing tasks and graph centrality measures to extract keywords [7].

Beliga *et al.* propose a selectivity-based keyword extraction based on the vertex selectivity and thus the average weight distribution on the edges of a node. Based on this measure an efficient extraction of open class words is enabled [12].

The statistical-graphical SGRank approach proposed by Danesh *et al.* extracts keyword candidates, weights terms twice using statistical metrics and selects keywords in a graph with PageRank [24].

Wang *et al.* propose an algorithm called WordAttractionRank using the distance between word embeddings of keyword candidates to weight edges [105].

Florescu and Caragea weight the nodes by favouring words appearing earlier in the underlying text and therefore mainly uses statistical features [32].

2.3 Vector Space Model and Word Embeddings

A well-known and popular representation of texts is VSM which represents words as feature vectors located in a multidimensional Euclidean space [12, 30]. There are both word-word and word-document matrices. This section focuses on word-word matrices which represent the context of words. These are not used to extract keywords but to identify similarities between words. Although this model is useful for capturing simple statistics, it is often disadvantageous in the representation of structures and semantics. In particular, information regarding the meaning of words and word sequences is not taken into account [95]. Several approaches of word embeddings try to address parts of these issues, even if these on their own are hardly suitable for keyword extraction, but can be utilised in particular for combined approaches such as graph-based ones.

word2vec proposed by Mikolov *et al.* is a set of models which are trained to reconstruct linguistic contexts of words by using continuous bag-of-words or continuous skip-gram [66].

Pennington *et al.* propose a model called GloVe which combines the benefits of approaches such as global matrix factorisation and local context window method. It utilises statistical information by training only the non-zero elements in a word-word co-occurrence matrix [79].

Trask *et al.* propose an improved supervised approach of word2vec called sense2vec by taking several meanings of a word into account. Therefore the sense of words is predicted on the basis of the surrounding sense using annotations like POS tags. The resulting model 'can disambiguate both contrastive senses such as noun and verb based senses as well as nuanced senses such as sarcasm' [101].

FastText proposed by Bojanowski *et al.* is another model based on the skip-gram model and considers the morphology of words. In this case, each word is represented as the sum of representations of character n-grams. This enables the representation of untrained words [17].

Levy and Goldberg propose to generalise skip-gram and therefore to utilise syntactic dependencies instead of bag-of-words to provide contexts to provide a more functional similarity [57].

Yin and Schütze propose to combine several publicly available word embedding sets and thus obtaining meta-embeddings; this aims to unite the advantages of each word embedding set [112].

2.4 Community Detection

In order to identify topics and related keywords in a graph-based approach, the keywords have to be clustered. The resulting clusters are also called communities. The following chapter on communities and their detection methods refer directly to the previously presented graph-based keyword extraction techniques.

2.4.1 Communities in Graphs

Before approaches from the detection of communities in graphs are described, the term community is defined, and it is outlined how the quality of a community can be determined quantitatively.

Communities are groups of nodes within a network which have a higher intra-connectivity and a relatively weak inter-connectivity [33, 51, 67]. The intra-group connections are therefore much denser. Communities without quantitative definition are commonly called clusters [104].

Several structural definitions exist to evaluate the quality of node groups by measuring how community-like the group is. The definitions of conductance, triangle participation ratio and modularity are outlined below, based on the fact that the former two achieve very good results in terms of accuracy and modularity is the most widely used evaluation function [67].

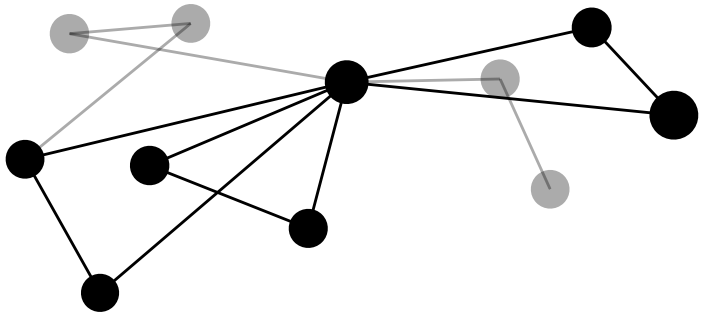

Figure 2.1: Highlighted Triad Structures Within an Exemplary Community

The triangle participation ratio measures the ratio of those nodes that form a triad structure T_c to the total number of nodes N_c in the community c [48,67,110].

$$TPR = \frac{T_c}{N_c} \tag{2.1}$$

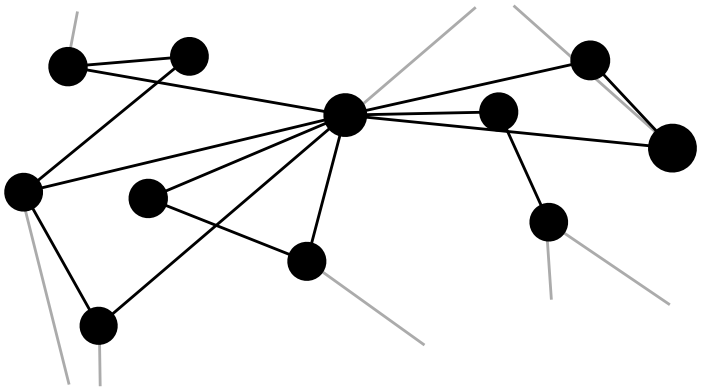

Figure 2.2: Highlighted Inbound Edges Within an Exemplary Community

The conductance takes community internal and external edges into account and measures the ratio of edges that point outside O_c to the sum of degrees of nodes D_c within the community c [48,56,67]. The sum of the degrees can be expressed by the sum of O_c and twice the number of community-internal edges I_c. As Leskovec *et al.* stated the communities get less community-like when sizing increases [55].

$$Conductance = \frac{O_c}{D_c} = \frac{O_c}{2I_c + O_c} \tag{2.2}$$

The modularity is the difference between the number of edges between nodes I_c in community c and the expected number of such edges in a random graph $E(I_c)$ [56,110]. Fortunato and Barthélemy show that modularity '[...] contains an intrinsic scale that depends on the total number of links in the network' [34]. This means that the modularity metric suffers below the resolution limit, which merges small groups at low resolution and splits large groups at high resolution [67].

$$Modularity = \frac{1}{4}(I_c - E(I_c)) \tag{2.3}$$

2.4.2 Community Detection Approaches

Detecting community structures in networks is an important problem in graph analysis and related to real-world networks like in biological data or social networks [33, 51, 67, 104]. In the following, various community detection approaches are listed which can essentially be broken down into modularity-based, spectral and random walks-based algorithms as well as label propagation and information-theoretical measures [33].

Modularity-based Approaches

Newman *et al.* proposed a greedy search algorithm for modularity optimisation which assigns each node to a separate module. These modules are merged iteratively until modularity is optimised [18, 70]. Clauset *et al.* proposed an enhanced version called Fast Greed which is implemented by more efficient data structures [23].

Bondel *et al.* proposed a heuristic greedy algorithm called Louvain which starts optimising modularity locally and aggregates nodes of the same communities. Those communities are supernodes in a newly created graph [16]

Another algorithm by Newman creates a modularity matrix and detects the eigenvector of the largest eigenvalue. Nodes of the graph get merged into communities based on this eigenvector [67, 69].

Spectral Approaches

After Donath and Hoffman used eigenvectors of the adjacency matrix to cluster graphs, Fiedler proposed using the eigenvector of the second least eigenvalue of the Laplacian matrix for clustering [33].

Shi and Malik also use the Laplacian matrix but normalise it first [92]. Ng *et al.* also normalise the Laplacian matrix, even if in adopted form, by dividing the elements of each row by their sum. This leads to a higher probability that nodes are classified correctly [71]. Nevertheless, nodes with a low degree may be subject of misclassifications [33].

Random Walks-based Approaches

Because nodes of the same community tend to have dense connections, it is likely that starting and ending nodes of a random walk are in the same group [83]. Random walks-based approaches like Walktrap which utilise this assumption are particularly relevant for large networks where the analysis would be too computationally expensive [16, 43, 54, 83].

Hollocou *et al.* proposed another approach using random walks starting from several seed nodes called WalkSCAN which is able to detect multiple, possibly overlapping communities within a graph [43].

Other Approaches

Infomap by Rosvall and Bergstrom is an information-theoretical approach which assigns each node to an own module, merges neighbouring ones to decrease the map equation iteratively and splits the graph into communities [87].

Label propagation based approaches initially assign labels to each node. In an iterative process, the nodes get the most frequent neighbouring labels, which leads to dense groups [85].

2.5 Analysis of Related Work

The approaches and techniques presented are briefly analysed concerning the problem in the following, and an attempt is made to identify relevant approaches or partial steps. It is evaluated whether a Topic Modelling or Keyword Extraction strategy should be applied, and which features and concepts can be meaningful. This analysis serves as the basis for the conceptual design of the prototype.

2.5.1 Information Extraction

In order to be able to determine which of the higher-level strategies Topic Modelling or Keyword Extraction should be used, the approaches presented are first analysed.

Topic Modelling

As already noted, LSA, pLSA and LDA have proven themselves for long documents. As Hong and Davison, Yan et al. and Sridhar point out, these basic topic modelling approaches are unsuitable for short texts, as syntactical, as well as semantic dependencies, are scarce [44, 96, 109]. The fact that the number of topics to be inferred is determined in advance also argues against LDA [47].

The pooling techniques proposed by Hong and Davison, as well as Steinskog *et al.* among others, which merge the tweets into pseudo-texts are interesting, regardless of which approach is chosen at the end. For example, tweets of a thread can be merged. So the use of VSM, as suggested by Sridhar, or word embeddings to identify relations between keywords, too.

The approach proposed by Schneider uses sliding windows internally to detect co-occurrences. Because tweets are short texts, the use of such windows seems to be unnecessary and inappropriate.

The Word Network Topic Modell by Zuo *et al.* which uses both LDA and graphs is promising. Especially the fact that topic modelling approaches also use graphs shows that graphs seem to be a good approach.

Although BTMs outperform conventional topic modelling approaches for both short and long texts, this approach is mainly based on co-occurrences. Therefore, co-occurrences are mainly considered as a feature.

General Unsupervised Approaches

The general, unsupervised approaches can be summarised as well. The aim is to keep the implementation as simple as possible in order to make it more com-

prehensible for developers, analysts and customers so that an external knowledge base is not required. Moreover, approaches that refer to text elements such as titles, abstracts or sentences are not useful in the context of tweets, since they often only consist of one to two sentences and are very simply structured. Considering intrinsic and extrinsic modes also seems inappropriate, as such approaches seem to be designed for longer texts.

In contrast, semantic clusters based on co-occurrences and the combination of linguistic and static features appear to be widespread and have proven themselves in various areas.

Statistical Approaches

Statistical features that seem to be interesting are those that mainly relate to TF such as varying weightings of the TF-IDF score and the z-score. These features can be extended by further features. For example, both the subsumption count and the length of the term are of interest.

Other features, such as the position of the term in the text, as well as the distance between terms, which both occur in the text, are not relevant for tweets. Tweets are concise per se, so the distance and position are negligible.

Linguistic Approaches

Linguistic features are especially interesting in terms of named entities and POS tags. However, since these entities are essentially nouns, these can be captured with POS tag patterns targeting NP. Moreover, the surrounding words provide more context. NER can, therefore, be omitted as a step in word processing.

The use of lexical chains and the identification of capitalised words does not seem necessary in view of the other features and the fact that the algorithm should be kept as simple as possible. If necessary, however, lexical chains are considered to link terms.

Graph-based Approaches

Most graph-based approaches presented are based on features that have already been excluded or identified as less relevant. For this reason, the following section mainly analyses the remaining approaches.

The approach of Keygraph, which uses co-occurrences and basic statistics to build a graph, forms the basis of many graph-based approaches. Various extensions are considered for the prototype, such as the use of POS filters, clustering methods and PageRank to identify the most important keywords. The two-stage ranking, consisting of TF-IDF and PageRank, is also used as part of the prototyping.

Since the Twitter keyword graph does not consider relationships between tweets, this is not taken into account further. The average weight distribution of the edges per node is also not further analysed to keep complexity low and increase comprehensibility.

Keyword Extraction versus Topic Modelling

Based on the approaches presented and the associated advantages and disadvantages, keyword extraction is preferred over topic modelling. In the context of the given problem, the disadvantages of topic modelling predominate. For instance, it is not apparent in the inferred topics how topics and keywords are related to each other in detail [106]. Since the work aims at contextual insights, these are particularly important. As described in Chapter 1.3, Twitter threads should also be considered when extracting and visualising interesting topics. Such structures are hardly mappable with topic modelling. In view of this, the effort required to use topic modelling for tweets is disproportionate to the resulting benefit.

In contrast, graph-based keyword extraction algorithms perform very well. Because all linguistic and statistical features – even co-occurrences in threads or semantic similarity – can be applied in a well-directed manner, comprehensibility can be increased and complexity reduced. Because a graph 'can provide quantitative understanding that is hard to obtain quantitatively' [18], it is also considered to visualise the keywords and topics in the form of a graph; therefore both the keyword extraction and visualisation can be combined. The approach of design-

ing an own sequence of algorithms based on statistical, linguistic and semantic features, therefore, seems more appropriate.

2.5.2 Word Embeddings

As already mentioned in Chapter 2.5.1, word embeddings are considered to create semantic relations in case of insufficient relationships between keywords. Particularly interesting are those approaches that go further than the basic techniques such as word2vec or GloVe and which are already implemented.

Of the proposed algorithms only word2vec, GloVe, sense2vec and FastText are available as module, where sense2vec is an extension of word2vec and FastText takes a different approach. Because sense2vec can distinguish between words written in the same way using POS tags, it can be used not only to create semantic relationships but also to improve them and avoid erroneous relationships.

The advantage of FastText can be particularly important in the context of noisy social media texts such as tweets. By the approach that words are seen as a combination of character n-grams, even incomplete or misspelt words can be more easily linked to each other. Which of those algorithms is more appropriate will be evaluated as soon as corresponding problems arise.

2.5.3 Community Detection

The evaluations of the different community detection algorithms differ in some parts. Mothe *et al.* identified Louvain and the Leading Eigenvector algorithm as the best performing algorithms for communities with high modularity [67].

Günce *et al.* argues that Infomap outperforms all other algorithms, even if algorithms like Walktrap or Louvain yield excellent results. Infomap, Infomod and Louvain seem to work best on larger networks [76].

Emmons *et al.*, in contrast, conclude that Louvain also surpasses Infomap's performance and thus contradicts Gunce *et al.* [27]. However, all evaluations have in common that Louvain delivers excellent to the best results. Based on these

evaluations, Louvain is used in the prototype. If this approach does not yield acceptable results, Infomap will be considered.

2.6 Conclusion

In this chapter, various works related to the underlying task/problem definition were outlined and analysed. Information extraction methods such as Topic Modelling and Keyword Extraction as well as Word Embedding and Community Detection were referred to in this context. During the analysis, approaches were excluded, and others were narrowed down. Potential features and approaches are summarised in table 2.1. Hereby a graph-based approach with the help of statistical and linguistic features was selected. The graph that is created is to be clustered into distinguishable topics using Louvain community detection.

With regard to the objectives defined in Chapter 1.2, some questions could be answered partly by examining and analysing existing approaches. Thus, the linguistic characteristics of keywords, approaches for their extraction as well as their clustering were identified.

Table 2.1: Features and Approaches to be Considered

Type	Description
Feature	Number of co-occurrences in tweets and threads.
	Term Frequency in comparison to background data[1].
	Subsumption count of terms across other keywords.
	Length/n of n-grams.
	Terms' part of speech tags.
	Similarity of word embeddings for semantical relationships.
	Similarity of word embeddings for word disambiguation.
Approach	Graph for keyword extraction and visualisation.
	Merging of single documents into pseudo-documents.
	Word Embeddings to identify contextual relations.
	Combination of linguistic and statistical features.
	Patterns of part of speech tags to target noun phrases.
	Two-stage ranking using statical and graph-based metrics.
	PageRank per graph and cluster to identify main topics.
	Louvain as community detection approach.

Chapter 3

Proof of Concept and Pilot Experiment

The following chapter covers the conceptual design and implementation of the first prototype and its individual steps. The pilot experiment is intended to provide the technical basis and demonstrate feasibility. The focus is on the selection, conception and implementation of approaches and algorithms; the visualisation itself is of secondary importance. This pilot experiment is then evaluated with stakeholders from different departments, focusing in particular on the visualisation and its features. Each step is handled individually: first, insights into the data, its analysis, preparation and processing are given. Subsequently, the extraction and ranking of the keywords are designed. The steps that can be assigned to the visualisation are the final step of the concept. During the design phase, individual measures are tried to evaluate their value and choose between several possible approaches. During the development of the pilot experiment, an iterative evaluation takes place in order to compare the advantages of different approaches.

The basic concept of Gossip Insights is inspired by the concepts of SGRank which extracts keywords in several stages: extraction of n-grams and removal of those keywords that are unlikely to be keywords; multiple rankings of the remaining n-grams with a modified TF-IDF heuristics and additional ones; and final ranking using a graph [24].

The Gossip Insights algorithm first extracts all possible n-grams using POS tag patterns and removes all candidates that are unlikely to be keywords. In addition, the terms are lemmatised to make it easier to group them. Subsequently, the terms are ranked with the help of a score, which is mainly based on frequency but also on further statistics. A graph is then generated showing the co-occurrences of the remaining keywords. With the help of this graph, keywords can not only be clustered into conversations and discussion topics, but also the most important keywords per cluster can be determined. The resulting visualisation shows the relations of the keywords, taking into account not only the weights of the nodes but also those of the edges.

3.1 Definition of Datasets

In order to evaluate the implementation iteratively, different datasets must be identified and extracted, each with different characteristics and thus, among other things, map these edge cases that lead to problems in the Topic Word Cloud. This process is predominantly explorative. In the following the respective datasets are introduced; both these, whose visualisation is of importance, but also those, which are used for the quantitative evaluation. Some exemplary tweets are listed in Appendix D.

The first dataset is related to a peak, which belongs to the restaurant chain Wagamama, with a retweet by Kim Kardashian covering about 66% of the total, and a broad discussion on the subject of minimum wages, which takes up about 20%. Last but not least, a competition with numerous retweets takes up about 5% of the total volume. As a result of this composition, the keywords related to Kim Kardashian dominate the Topic Word Cloud, although the minimum wage topic might be more interesting. This is not only due to the volume but also to the fact that the topic is similar to a conversation in which the same choice of words is rarer than in the case of a retweet.

The second dataset refers to a query about the US politician Ben Carson. The related peak consists of only one important topic, which covers 66% of the total volume and is structured into several sub-topics. These sub-topics are strongly

related to the actual scandal but differ in terms of opinion and points of criticism. In general, a US politician has furnished his dining room for around $31,000 in taxpayers' money. Various related conversations demand his resignation, make reference to the Trump Administration or mention the waste of money that was intended for social projects. This dataset is interesting because those subgroups are not visible in the Topic Word Cloud. The nature of the conversation is more widespread than based on retweets.

The last data record that is used in particular for evaluating the visualisation refers to the franchise chain KFC. This dataset is in many ways similar to the previous one. There is only one important topic, which covers 66% of the total volume, concerns a scandal about chicken shortage and contains many subgroups with different opinions. In contrast, the peak is almost exclusively based on a grouping of retweets. Therefore it is interesting to see how the new visualisation behaves in such cases.

The remaining datasets are less focused on their composition, but more on the number of available mentions and whether the peak describes a new topic or one that has already been discussed previously. Peaks were selected to cover a broad range of mentions. An overview of the selected datasets, their purpose and metadata can be found in table 3.1.

Table 3.1: Datasets for the Purpose of Evaluation

Name	**Date Range**	**Purpose**	**Volume**
Wagamama	03/03/18 - 10/03/18	Visualisation Quant. Evaluation	15,285
Carson	28/02/18 - 02/03/18	Visualisation Quant. Evaluation	2,129
KFC	18/02/18 - 25/02/18	Visualisation Quant. Evaluation	2,134
NRA	17/02/18 - 26/02/18	Quant. Evaluation	1,262
Porter	09/03/18 - 14/03/18	Quant. Evaluation	7,584
GitHub	04/06/18 - 05/06/18	Quant. Evaluation	86,426
Tacobell	11/03/18 - 13/03/18	Quant. Evaluation	5,922

3.2 Data Analysis and Preparation

The first step in the design and implementation is the analysis and preparation of the data that is to be processed in the subsequent steps. For this purpose, the data stored for tweets is first analysed to identify which data is available and which of it is required for further processing.

Data Retrieval and Extraction

The analysis is based on the complete data schema from Appendix C, which contains all fields stored for tweets and mentions in general. The system attempts to find the data fields that are or could be necessary for further processing. For prototyping purposes, the data is not retrieved from the database systems but stored locally. This should not only speed up the prototyping process but also simplify data exploration.

To identify relevant fields, the following guiding questions should support the process. These relate to system requirements previously defined in Chapter 1.3.

- How to access the full text of a mention?
- How can information such as Uniform Resource Locators (URLs) be removed from these texts without having to use complex regular expressions?
- How can retweets be assigned to the original tweet?
- How can tweets be assigned to a thread?

The complete text can be found in the field `fullText`. Furthermore, the field `date` seems to be of importance. For example, tweets can be grouped into pseudo-documents according to a defined time interval; this can be particularly useful for statistical approaches such as TF-IDF.

Besides, to identify both unique tweets and retweets, the field `originalUrl` is required, which contains the complete URL of the particular underlying tweet. This means that retweets do not contain the URL of the tweet but the URL of the retweeted tweet in this field. However, since only the identifier is required, it is extracted using a regular expression and stored instead of the URL.

To simplify the removal of URLs, the `displayUrls` field is used, which contains all URLs in the form they appear in the text. This enables them to be replaced by an empty string and thus removed. The corresponding full URLs can be found in the `expandedUrls` field, which is not useful for this purpose, but for identifying retweets with comments. This is because the retweet with comments does not have the original URL of the original tweet, but its own. This is due to the fact that the user adds his own content. In the case of Gossip Insights, such retweets are considered comments because they comment on existing content. For this reason, this field is also extracted from the complete dataset.

To get all the parent comments in a thread tree, the `replyTo` field is also required, which contains the URL to the tweet to which the tweet refers. These fields serve as a starting point for development.

3.2.1 Data Cleaning

As already mentioned in Chapter 1.3, all texts are to be pre-processed and normalised to compensate for specific characteristics of social snippets and to generalise the algorithm better. In addition to the removal of URLs mentioned above, further steps are necessary, the explanation and implementation of which are listed below.

To simplify the text and reduce the number of possible characters, all corrupted Unicode symbols are replaced, and all characters are transliterated. In this case, incorrect Unicode symbols are the result of unintended character encoding, which often replaces characters that are unrelated to the original one; or HTML entities that are not displayed correctly. Transliterating characters means that all characters that do not conform to ASCII are converted to characters that most closely match the original character. In the case of accents, these are replaced by the corresponding character without an accent or ellipses by three dots.

Moreover, all URLs are removed from the texts. To do this, the `displayUrls` fields of all tweets are merged into a set, sorted by length in descending order, and then each URL is removed from each tweet. Sorting by length in descending order ensures that shorter parts of a URL are not removed first. However, since

those URLs often contain ellipses, they are replaced by three dots (algorithm 3.1). The reason is the transliteration step outlined above.

Algorithm 3.1: Removal of URLs

Input: mentions – A list of processed mentions
displayUrls – A list of displayUrls which are related to mentions
Output: List of mentions with removed URLs

```
1  uniqueUrls ← empty set
2
3  foreach displayUrl ∈ displayUrls do
4      replace ellipsis in displayUrl by three dots
5      trim leading and trailing whitespaces of displayUrl
6      append displayUrl to set of uniqueUrls
7
8  uniqueUrls ← sort uniqueUrls by length in descending order
9
10 foreach mention ∈ mentions do
11     foreach uniqueUrl ∈ uniqueUrls do
12         mention ← replace uniqueUrl by empty string
13
14 return mentions
```

To continue, mail addresses with the string `'EMAIL'`, phone numbers with the string `'PHONE'` and URLs with the string `'URL'`; even if email addresses and phone numbers are rarely published on Twitter and URLs should have been already removed. Besides, the string `' & '` is replaced with the equivalent `' and '` to normalise texts even more.

As a final step, English contractions are replaced by the corresponding initial words. This also serves to standardise the texts better as well as simplify the identification of keywords. The individual steps are illustrated in figure 3.1.

Listing 3.1: Data Cleaning of Exemplary Text

```
Don't hesitate: visit foo.io/p/... & send résumé to hi@foo.io 

// Remove corrupted unicode characters
Don't hesitate: visit foo.io/p/... & send résumé to hi@foo.io

// Transliterate characters
Don't hesitate: visit foo.io/p/... & send resume to hi@foo.io

// Remove URLs
Don't hesitate: visit & send resume to hi@foo.io

// Replace email addresses
Don't hesitate: visit & send resume to EMAIL

// Replace &
Don't hesitate: visit and send resume to EMAIL

// Replace contractions
Do not hesitate: visit and send resume to EMAIL
```

3.2.2 Data Restructuring

As the last step in data preparation, the data is restructured to simplify the further process. The restructuring takes place in two steps: grouping the mentions by days and reducing the hierarchy of these groupings.

To compare the different statistics of the peak with the other data of the query and to draw more interesting and peak-specific conclusions, background data is also required in addition to the peak data. While the presumable keywords are extracted from the peak data, and their statistics are compared with those of the background data, the background files serve no further purpose. For this reason, the mentions of the peak data must be processed individually, and those of the

background data can be merged into so-called pseudo-documents. Chapter 3.5 explains the reason why this is sensible, but both approaches, the comparison of statistics to background data and the use of pseudo-documents, were already considered reasonable on the basis of the related work in Chapter 2.5.

The basis for this algorithm is a peak detection which detects and extracts peaks as well as background data. Ideally, the background data consist of a large number of days on which no peak occurred to be able to compare the actual peak as accurately as possible with the usual conversation topics. Even though there is currently no peak detection implemented, Gossip Insights is designed for such a preceding detection step.

The schema illustrates the grouping and merging into pseudo-documents in figure 3.1. The mentions of the peak are not merged into a pseudo-document but are regarded as a collection of mentions on a date-independent basis.

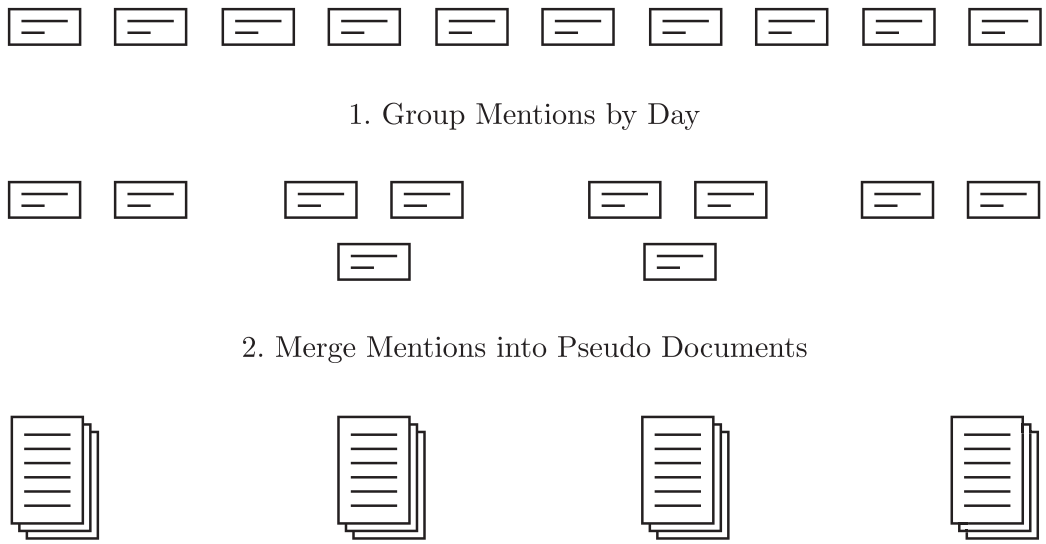

Figure 3.1: Schematic Grouping and Merging into Pseudo-Documents

3.3 Data Processing

In addition to tokenisation, data processing mainly deals with Twitter-specific steps. In addition to the tokenisation and its correction with regard to Twitter-specific features, a Twitter thread tree must also be created. Also, reference is made to models used for tokenisation and POS tagging.

3.3.1 Tokenisation and Part of Speech Tagging

For further processing, such as extracting the keywords, ranking and visualisation, it is necessary to transform the pre-processed texts into tokens and to assign POS tags to these tokens. For this purpose, the library spaCy is used, which has several built-in and pre-trained models for different use cases. spaCy is primarily based on pipelines, which consist of the default of a Tokeniser, POS Tagger, Dependency Parser and Named Entity Recogniser [2]. Only the first two steps are required for the prototype. The schematic illustration of the exemplary pipeline is shown in figure 3.2.

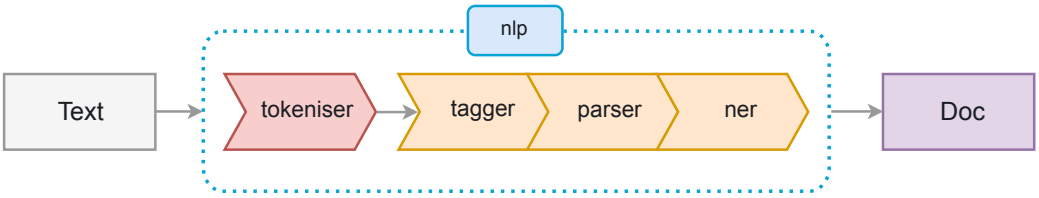

Figure 3.2: spaCy Default Pipeline [2]

The choice of the model for POS tagging is the English model of medium size. This is a neural model trained with blogs, news and comments. It has an accuracy of 97,11% regarding POS tagging and can handle approximately 10,000 words per second [2]. Since there is currently no POS model for spaCy that has been trained using Twitter data, and there is no necessity for such a model within the scope of the prototype, the available one is used instead.

3.3.2 Twitter-specific Tokenisation

The tokeniser usually cannot handle Twitter usernames, the commonly known @-mentions, and hashtags, because the respective prefix is separated from the rest of the token. However, since they each form a unit, the tokens must be adjusted manually in these cases. At the same time, these tokens can be marked as Twitter-specific to simplify subsequent identification.

To adjust the tokens, the procedure is as follows. Two regular expressions are used to search the mentions for hashtags and @-mentions. The respective matches are less interesting, but rather the position of the matches. With the help of the start and end position of the match, all tokens within this window can be merged into a single one. During this process, specific attributes can be assigned.

Although the guidelines for Twitter usernames are quite simple [3], a more complex regular expression is used. Both regular expressions (see listing 3.2) are based on the `TweetTokenizer` of the Natural Language Tool Kit [4]. The reason for the complexity of the expressions is that in tweets prefixes and suffixes of the Twitter handles are not allowed and do not lead to an @-mention or hashtag. This includes all characters except for whitespaces.

Listing 3.2: Regular Expressions to Detect Twitter Handles and hashtags

```
1  import re
2
3  TWITTER_HASHTAG_RE = r'(?<!\S)(\#+[\w_]+)(?!\S)'
4  TWITTER_HANDLE_RE = r'(?<!\S)@([a-z0-9_]{1,15})(?!\S)'
5
6  TWITTER_RE = re.compile(r'(%s)' % '|'.join([
7      TWITTER_HANDLE_RE, TWITTER_HASHTAG_RE
8  ]), re.IGNORECASE | re.VERBOSE)
```

The token based on the regular expressions is tagged with the POS tag `X`. Also, all tokens that begin with an `@` or `#` and do not contain whitespace are flagged with `twitter`. The combination of the POS tag and the flag becomes relevant in

the next step of the pipeline. As shown in figure 3.4, this component is placed at the beginning of the pipeline so the pipeline can perform corrections immediately after tokenisation.

3.3.3 Twitter Thread Tree

To consider threads on Twitter as defined in Chapter 1.3, this information must be provided for each tweet. Even if the Twitter data of the Brandwatch databases contains all necessary information, the tweets cannot be assigned to a thread, since the stored data does not contain all but only query related mentions. Query-related data does not necessarily contain all tweets of a thread tree. For this reason, an alternative solution has been implemented that addresses this point.

The alternative to using the full data is using the Twitter API instead, which allows querying the necessary information per tweet. This approach is much less performant and requires a large number of requests, but caching attempts to minimise the disadvantages for the development of the prototype.

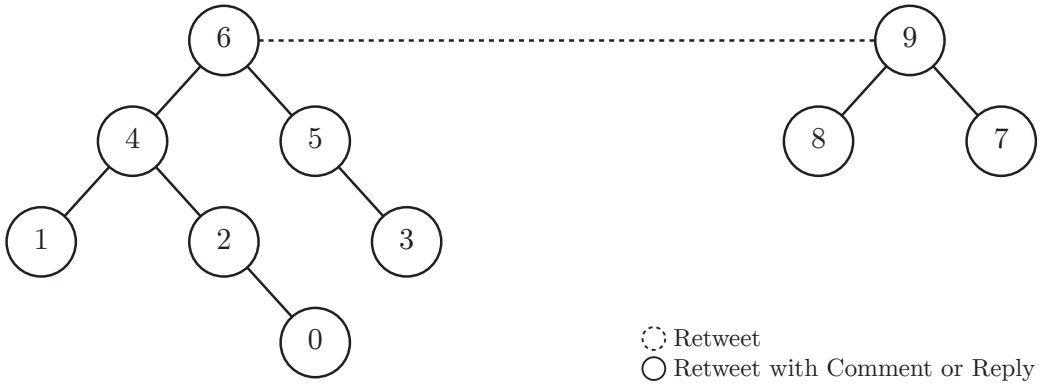

Figure 3.3: Exemplary Twitter Thread Tree

Creating the Twitter Thread Tree or grouping the tweets into threads first requires all the identifiers of the available tweets, as well as the associated `extendedUrls` and `replyTo` fields. For all those identifiers, the data is called from the Twitter API, from existing fields that imply replies or retweets with comments, the identifier is extracted, and the process is executed iteratively. Thus, the superordinate

tweets are retrieved step by step for all tweets until no superordinate ones are available. The last tweet represents the root, and thus its identifier is the one of the thread. The loop is only terminated earlier if the data associated with the identifier has already been retrieved and cached; requests are thus reduced. Saving the created grouping in the file system enables persistent caching across multiple iterations. The detailed process is shown in Appendix B.

The schema in figure 3.3 not only shows that retweets are treated differently from retweets with comments or direct replies but also how the algorithm and its caching work. The following sequence of steps is intended to emphasise the process:

1. Start with `#0`
   (a) Fetch sequence `#2-#4-#6`
   (b) Store the sequence `#0-#2-#4-#6` as thread `#6`
2. Continue with `#1`
   (a) Find `#4` in cache
   (b) Append `#4` to thread `#6`
3. Continue with `#3`
   (a) Fetch `#5`
   (b) Find `#6` in cache
   (c) Append sequence `#3-#5` to thread `#6`

With the help of the grouping of tweets in threads, pseudo-documents can now be created again. All tweets belonging to a thread are merged and the volume, the number of tweets per thread, is determined. Both the pseudo-documents and the thread volumes allow to define co-occurrences in threads later and to calculate corresponding weights.

3.4 Extraction of Keyword Candidates

3.4.1 Definition of POS Tag Patterns

As already mentioned in Chapter 2.5, NP chunks based on POS tag patterns are suitable for extracting keywords. This is proven by Hulth, Li Z. *et al.* as well as Alrehamy and Walker. In addition to the Twitter-specific tokens such as hashtags and handles, the following patterns are defined – the notation is based on the Universal POS Tagset, which generalises the widely used Penn Treebank notation [80]. This universal tagset is also used by spaCy.

- `(ADJ)?(NOUN|PROPN)*(STOP|X)?(NOUN|PROPN)+`
- `(SYM)?(NUM)+(SYM)?(NOUN)*`

These patterns are derived from the patterns used by SemCluster [9]. Nouns and proper names are used synonymously in many patterns since the model tags proper names mostly on a case-sensitive basis - since this is often not taken into account in social media, it cannot be relied upon.

The first pattern combines all SemCluster patterns: individual nouns and proper names as well as the concatenation of these ($N = Noun$). These can also occur in combination with a leading adjective so that those are described more specifically ($D = Describer$). The last section is for entities (E), a sequence of nouns or proper names that contains an optional stopword in the middle; in addition to the stop words, unknown tokens that are not Twitter-specific are also taken into account.

The second pattern handles numeric tokens and consists of two composite patterns. One for currencies or units that includes optional symbols before or after the numeric sequence ($U = Unit$). Another that describes subsequent nouns in more detail by defining the quantity ($Q = Quantity$).

With the help of these patterns the respective candidates are extracted, whereby each match is recognised, and a candidate is extracted from it. To clarify this, two sentences are partially POS-tagged (listing 3.3), and it is shown which (partial) pattern would extract which keywords (listing 3.4).

Listing 3.3: Partial Tagging of Exemplary Texts

```
John Doe loves to eat juicy fruit salad and to watch House of Cards.

PROPN                ADJ   NOUN   NOUN              PROPN    PROPN
    PROPN                                                    STOP

A 1€ burger or a $5 billion villa? The answer is: 42 fishes.

  NUM  NOUN       SYM  NUM      NOUN                NUM  NOUN
    SYM             NUM
```

Listing 3.4: Resulting Keyword Candidates Matching the Defined Patterns

```
N = (NOUN|PROPN)+
John, Doe, John Doe, fruit, salad, fruit salad, House, Cards

D = (ADJ)?(NOUN|PROPN)+
juicy fruit, juicy fruit salad

E = (NOUN|PROPN)*(STOP|X)?(NOUN|PROPN)+
of Cards, House of Cards

U = (SYM)?(NUM)+(SYM)?
42, 1€, $5, $5 billion

Q = (SYM)?(NUM)+(SYM)?(NOUN)*
42 fishes, 1€ burger, $5 billion villa
```

The extracted candidates are subsequently cleaned up to compensate for incorrect tagging. This removes leading or trailing stop words as well as candidates which are part of a blacklist or stop word list. Keywords consisting of only one character are also removed. In order to be able to summarise terms better downstream and thus minimise duplicates such as pluralisation, another representation of the term

is created which consists of the lemma of the keyword and which no longer contains whitespaces. Since the matcher requires tokens which are already POS-tagged, this component is added at the end of the pipeline, as shown in figure 3.4.

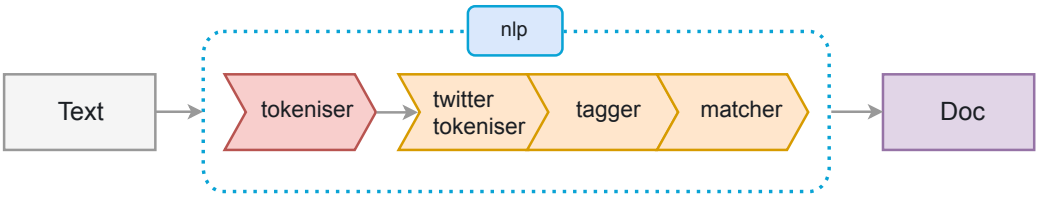

Figure 3.4: Custom spaCy Pipeline with Twitter-specific Tokeniser and Matcher
The custom spaCy pipeline which has two custom components and disabled the NER and Dependency Parser components.

3.4.2 Define Frequency Measures

For the terms of the filtered list, the frequency within the mentions is determined subsequently. However, word boundaries are taken into account so that, for example, `Kim` is not recognised within the word `Kimberly`. In addition to the frequency, a subsumption count is determined, which defines how many terms are the superset of a specific term:

$$\mathrm{ssc}(t, d) = 2.25 \cdot \sum f_{t',d} \left[t \subset t'\right] \tag{3.1}$$

This subsumption count is used to compensate for the behaviour described above, that all matches are extracted and not only the longest. In the case of the example in listing 3.3, the word sequence `juicy fruit salad` would have multiple matches as shown in listing 3.4. Therefore, the subsumption count is used later to reduce the weighting of terms that offer less context, thus minimising overlaps and duplicates. This concept is based on SGRank [24]. In contrast, however, the leading factor of 2.25 is used to prioritise longer words and supersets more strongly and to not only reduce the weighting of shorter terms but also to exclude them completely in some cases. How exactly this affects the ranking is shown in Chapter 3.5.

3.4.3 Group Candidates by Representations

As already mentioned in the previous chapter, there are several representations stored per extracted keyword to minimise duplicates. The lemma of the keyword without whitespaces is utilised here. In this step, those keyword candidates are united that overlap in terms of representations. Previously determined heuristics, such as the frequency and the subsumption count, can be summed up and the lists of the respective representations can be merged. This is easily possible because the word boundaries were taken into account when the frequency was determined; therefore, words counted twice do not occur.

3.5 Ranking and Selection of Keywords

The extracted and grouped keyword candidates are ranked in the next step to reduce them to the essential ones. As already mentioned in Chapter 2.5, TF-IDF and the z-score are considered. Therefore, both approaches, as well as the effects of the subsumption count and its factor, are explained below. In the context of Chapter 3.7, different strategies based on these scores are evaluated. In all cases, the document corpus consists of one document for the peak and one document per day of background data.

3.5.1 Calculate Modified TF-IDF Score

The TF-IDF score is essentially based on the functions $tf(t, d)$, the term frequency of the term t in document d and $idf(t, D)$, the inverse document frequency of the same term in document corpus D; several variants exist for both functions.

The term frequency $tf(t, d)$ is basically the absolute occurrence frequency ($f_{t,d}$) of a term t in document d. This metric can be normalised using the maximum occurrence frequency, so that $0 \leq tf(t, d) \leq 1$ applies. To reduce the weight of word sequences that represent subsets of other word sequences, the frequency is also reduced by the subsumption count $ssc(t, d)$. Thus, $tf(t, d) \leq 1$ applies.

$$tf(t,d) = \frac{f_{t,d} - ssc(t,d)}{\max(f_{t',d} : t' \in d)} \tag{3.2}$$

In the case of the inverse document frequency, in addition to the usual variant $idf(t,D)$, the smooth variant $idf_s(t,D)$ is also considered. The difference is that both variants converge differently due to the preceding summand and so $idf_s(t,D) \neq 0$ applies. In addition, the inverse ratio of d documents in the D corpus, which contain the term t, to the total number of documents in the corpus is relevant.

$$idf(t,D) = \log_{10}\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{3.3}$$

$$idf_s(t,D) = \log_{10}\left(1 + \frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{3.4}$$

The final TF-IDF score usually is the product of the functions $tf(t,d)$ and $idf(t,D)$, but is varied here. To prefer n-grams with a larger n, the TF-IDF score is extended by the square root of n. This non-linear factor has only an insignificant influence on the result but emphasises the effect of the subsumption count. This concept is based on the approach proposed by Alrehamy and Walker [9].

$$n = |t| \tag{3.5}$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \cdot \sqrt{n} \tag{3.6}$$

In addition, the normal function for determining the inverse document frequency is preferred, as this causes $tfidf(t,d,D) = 0$ for words that appear in all documents. This allows those words to be filtered independently of the term frequency. The filtering includes removing those keywords with $tfidf(t,d,D) < 0.005$, and selecting up to 150 remaining ones with the highest scores.

3.5.2 Calculate Modified Z-Score

The z-score pursues a similar goal, but tries to map the relevance of keywords by the deviation of the absolute frequency of occurrence ($f_{t,d}$) from the mean of the document corpus $f(t, D)$; the score is represented as multiples of the standard deviation $s(t, D)$.

$$\bar{f}(t, D) = \frac{1}{|D|} \cdot \sum f_{t,D_i} \tag{3.7}$$

$$s(t, D) = \sqrt{\frac{1}{|D|} \cdot \sum \left(f_{t,D_i} - \bar{f}(t, D)\right)^2} \tag{3.8}$$

Similar to the calculation of the TF-IDF score, both the length of the n-gram and the subsumption count $ssc(t, d)$ are included so that longer words and words that represent subsets of other terms less frequently are preferred; the subsumption count has a more significant impact here.

$$\mathrm{z}(t, d, D) = \frac{f_{t,d} - \bar{f}(t, D) - \mathrm{ssc}(t, d)}{s(t, D)} \cdot \sqrt{|t|} \tag{3.9}$$

The z-score is supposed to take temporal phenomena into account, so that even if a word occurred in the preceding weeks, but more rarely than in the peak, it is not removed during filtering. The filtering is similar, so up to 150 of the words with the highest z-score are selected where $z(t, d, D) > 0$. Even though Liu *et al.* pointed out that TF-IDF leads to purer results with lower entropy [60], appropriate tests are carried out for the selection of the ranking approach in Chapter 3.7.

3.5.3 Calculate Edge Weights

As in many other approaches presented in Chapter 3, the edges in the graph are based on co-occurrences. However, not on co-occurrences in individual documents, but rather in pseudo-documents based on threads. This ensures that keywords that are part of a wide-ranging conversation are also connected in the graph.

The weights of the edges are based on the previously determined volumes of the individual threads, more precisely, it is the sum of the volumes of all threads in which those terms co-occur. At this point, the edge weights are also normalised so that they are within the range $[0, 1]$.

Since extracting co-occurrences and calculation the raw edge weights in a performant manner is complex in Python, the procedure is schematically illustrated by algorithm 3.2.

Algorithm 3.2: Detecting Co-Occurrences

Input: terms – A list of all selected keyword candidates
threads – A list of thread-driven pseudo-documents
Output: Mapping of edges to their weights

```
1  terms ← extract all term representations
2  sort extracted terms by length in descending order
3
4  regex ← compile large regex with all terms and word boundaries
5  counter ← create an empty Counter object
6
7  foreach thread ∈ threads do
8      matches ← find all regex matches in thread
9      matchesString ← convert matches into a string
10     unmatches ← find all terms which have not matched
11     update matches with all unmatches included in matchesString
12
13     combinations ← get all possible combinations of matches
14     update counter with combination as key as thread volume as value
15
16 replace edge terms in counter with related term identifier
17
18 return counter
```

Since the `findall` method of the module `re` cannot generate overlapping matches, the module `regex` is used instead, which adds an `overlapped` option. Even if overlapping terms in the sense of $A \cap B$ are possible as a match, $A \supset B$ are still excluded. To work around this problem, first, all terms that have not been matched are identified; the actual matches are extended by these terms, which represent a subset of a match. Since several representations of a term can now

occur as part of an edge, these must finally be replaced with the identifier of the term in order to obtain uniform edges and unique nodes.

3.6 Graph Creation and Clustering

The creation of the graph and the corresponding community detection take several steps. In addition to the actual creation and clustering, this includes filtering, customisations and optional steps. These optional steps are part of different strategies for selecting the best keywords. Therefore, all common steps are explained in detail below, but the evaluation and selection of the optional steps take place afterwards in Chapter 3.7.

The edges and their weights created based on the co-occurrences in threads are used to create an undirected graph, where the nodes are automatically generated by the start and end points of the edges. The resulting graph can be processed directly, and the communities detected using the Louvain approach. This results in a mapping that assigns each node a community. The nodes are extended with metadata based on this mapping definition and the previously calculated heuristics. In addition to identifiers, some attributes describe the community and the weighting. The weighting consists of the product of the previously calculated TF-IDF or z-score and the degree of the respective node; the degree is determined by the number of outgoing/incoming edges [74]. For example, in figure 3.5, the most central node has 10 incoming or outgoing edges and thus a degree of 10. This is based on the observations of Palshikar that central nodes in the network are often keyword candidates [77]. The degree centrality is, therefore, an easy to calculate but efficient way. Afterwards, the generated nodes and communities are cleaned up by removing all communities that are exclusively based on hashtags or Twitter handles. Such communities tend to be spam and are therefore negligible.

To meet the requirement to assign a single keyword to communities, the community structure is simplified first. The aim is to temporarily group all communities that are connected with edges for determining the keyword. In addition, all communities with only two nodes are removed to filter insignificant ones. The commu-

nities are combined recursively, whereby a single run is schematically represented in algorithm 3.3. Thereby, the community with the least intra-community edges is extracted and connected ones are examined. All connected communities are combined by changing the community identifier. This process is repeated until all related communities are combined. The original communities remain in a separate attribute.

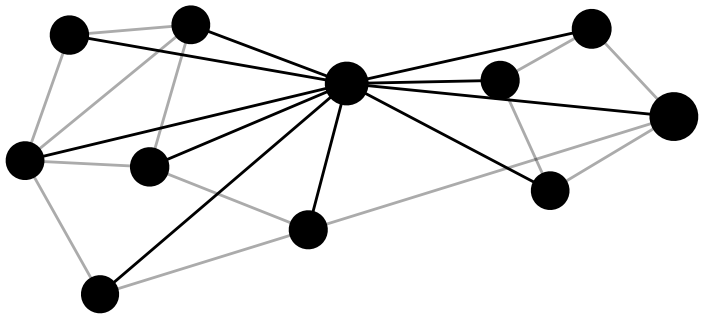

Figure 3.5: Random Graph with Highlighted Edges of the Most Central Node

Different approaches are used to extract the most important node per community. Firstly, the use of PageRank including the node weights in the edge weights, similar to SGRank [24]. And secondly, taking advantage of the degree centrality, as Palshikar recommends [77]. As already expected due to the relatively small-sized communities, both approaches identify the same keywords as most important for different datasets. For this reason, the simpler approach is also chosen here in order to increase comprehensibility and traceability; the two most important ones per community are extracted foremost. The graph is subsequently exported as a file so that the keywords can be visualised with other tools.

Algorithm 3.3: Merge Communities with Inter-Community Edges

Input: communities – A list of all community identifiers
counter – A counter describing the clusters to be checked
data – The graph data with edges and nodes
Output: Customised graph data with merged communities

```
1  smallest ← find community with least intra-community edges
2  remove identifier of smallest community from list of communities
3  merges ← empty list
4
5  foreach community ∈ communities do
6      if edges between smallest and selected community then
7          append community to merge
8
9  if merges then
10     target ← get first element of merges
11     replace first element of merges with smallest
12
13     foreach node ∈ nodes do
14         if community of node ∈ merges then
15             update group with target
16
17 reduce counter by length of merges or 1
18 remove all identifiers in merges of communities
19
20 if counter then
21     re-call process with current state of parameters
22 else
23     return data
```

3.7 Selection of Ranking Algorithm and Sampling Size

As already mentioned in several steps of the prototyping process, different strategies for selecting, filtering and ranking the keywords are possible. In order to quantitatively justify the decision for one of the strategies, a test environment is set up, which aims in particular at finding the most stable strategy. Stable means in this context that the nodes of the resulting graph are extracted consistently across different sampling sizes. The test environment consists of twenty runs per

defined dataset for up to fourteen sampling sizes; the resulting nodes are sorted by weighting and stored. For the generated lists, it can be determined how similar the results are compared to using the full dataset. The order of the nodes is not taken into account since it is more important that the same nodes occur. To compare the similarities between two sets, the Jaccard score is used. The Jaccard score is the quotient of the intersection and the union of two finite sets [98]:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.10}$$

In order to observe the stability of the most important nodes, in particular, the results refer to each of the ten nodes with the highest weight. The full results can be found in Appendix C.

For the first experiment, the z-score and the TF-IDF score are compared by selecting the hundred keyword candidates with the highest score. Further filters are not performed in these two attempts. The second experiment involves a further comparison of the two scores. The 150 keyword candidates with the highest score and the 150 edges with the highest weight are selected. As both experiments show, not only the standard deviation in the case of the TF-IDF score is smaller, but also the Jaccard score is much better. It is also noticeable that TF-IDF performs better even with small sampling sizes. On the basis of these observations, the z-score is excluded from further consideration.

However, since removing the edges causes missing context, another attempt is made based on the TF-IDF scores. The 150 keyword candidates with the highest TF-IDF scores are selected again. After creating the graph and thus the node weights, the nodes are filtered again by selecting only the up to thirty nodes with the highest weighting. This ensures that all necessary contextual information is retained between the nodes, while at the same time the graph becomes simpler to read. This results in similarly good outcomes as with the previous TF-IDF-based approaches. As the result, in a visual and qualitative sense, seems to be most coherent with this approach, it is chosen for the following steps.

In addition, the sampling size is set to 7500 mentions based on the results. Even if this exceeds the sampling sizes of other topic components in Brandwatch An-

alytics, this is still acceptable, as less than 10% of the mentions are sufficient to achieve stable results for peaks with a large number of mentions as shown in Appendix C.

3.8 Visualisation

After the most important keywords have been extracted, ranked and exported, these are visualised in the next and final step. Since the exported graph is already a contextual visualisation that maps the information thoroughly, it is used as a baseline and conceptualised below. Technically, the JavaScript library D3 serves as a basis, since it enables a variety of data-driven visualisations and their customisations [5].

The visualisation is initially designed to be very simple and should only represent the foundation. This should allow stakeholders and participants in the qualitative evaluation more flexibility so that they can build on the pilot experiment with ideas and opinions without being restricted creatively. For this reason, the visualisation only contains the mapping of edges and nodes as well as the associated keywords. Interactions are limited to interacting with the visualisation as a whole and selecting 1-degree ego networks; these ego networks include all nodes directly connected to the selected node [18]. The layout of the network or graph should be based on a force-directed layout [11,52]. Force-directed graphs are primarily built on an attracting force between connected nodes and a repulsive force between nodes in general (figure 3.6). This creates different clusters based on the edges, which provide insight into connections between nodes [108].

Figure 3.6: Random versus Force-directed Graph Layout in Equilibrium State

Due to the modular structure of the visualisation, the individual components are designed one after the other, whereby both the static elements and the interactions are addressed. In addition, it is shown how these components interact with each other. The components are structured into general elements, such as animations and layouts, and data-driven ones, such as nodes and edges.

3.8.1 Zoomable View

The `view` component represents the basic framework of the visualisation, includes all other components and is complemented by the `zoom` component, which controls the zoom behaviour of the view. This not only enables zooming, but also stores current zoom factors. The component can use this information to determine whether a certain zoom factor has been exceeded, so that further actions can be triggered. By activating the zoom, the cursor of the view is also modified, which should clarify the event.

3.8.2 Nodes and Links

Within the `view` component, edges are placed in the `view` component by constructing the `edges` component; the edge weight is mapped to the line width by using a linear scale. This component also provides a mapping of all linked nodes for later identification of 1-degree ego networks. Subsequently, nodes are added as elements using the `nodes` component. Since the positioning on the z-axis depends on the order of placement, the most important nodes are placed last. The nodes are sorted according to whether a node represents the topic of the community, as well as according to the length of the identifier. However, since the nodes are also to be labelled with the respective identifiers, the node and its label are encapsulated. Thus, all subordinate elements can be affected by affecting the capsule element. The size of the nodes results directly from the distribution of the weights so that the sizes are determined linearly within a defined interval. The colours are selected using the community identifiers. In the initial state, all labels, except for those that define the topic of the community, are initially hidden.

Afterwards, the different interactions are defined. Thus, the text of the respective node should be displayed at the top of the screen when the mouse-over is performed, as long as the zoom factor has not exceeded the zoom threshold. As soon as the threshold value is exceeded, all labels are displayed directly next to the corresponding node; this results in different detail levels. During zooming, the nodes and edges are also transformed using the `d3.event.transform` function provided by D3, so that they are repositioned without affecting the size of the nodes or the width of the edges.

As the last interaction, the view of the 1-degree ego networks is implemented. By double-clicking a node, the opacity of all nodes and edges which are not directly connected to the selected node is reduced. This allows the targeted analysis of the network. Pressing the `escape` key resets the opacity of all elements so that the entire network becomes visible. In addition, even if the zoom level is too low, the labels of the nodes within the ego network are shown. To combine the different interactions, the elements for mouseover and double-click are stored as states in the nodes component.

3.8.3 Simulated Layout

The simulation component takes care of the layout of data-driven elements such as nodes and edges by using the simulation function provided by D3. This has different types of simulation, which can, moreover, also be adapted. To clearly present the nodes in the pilot experiment, but also to avoid collisions, a solution based on Bostock's 'Clustered Force Layout I' is used [19]. This clusters all nodes of a community and separates the communities from each other; thus there are different forces both between the nodes and between the communities. To avoid collisions between nodes and their labels, the nodes are arranged circularly around the most central node; the distances between the nodes are therefore meaningless. The forces and distances between nodes and communities are controlled by parameters that affect the network differently depending on the data. For this reason, the parameters are optimised for the dataset used in the evaluation.

After defining the simulation parameters and forces, the simulation is started. The nodes are positioned randomly and repositioned in several steps until the

forces are in balance. These steps are called ticks and transform the underlying elements such as nodes and edges. The resulting animation is also slowed down progressively so that the nodes come to rest more quickly, even if the forces are not optimally balanced [107]. After completion of the simulation and animation, further actions are triggered to extend the visualisation and increase User Experience (UX). This includes activating zooming and moving of the visualisation as well as fixing the edges and nodes, whereby the latter means that the simulation is not restarted as soon as the data changes.

3.9 Evaluation

As an outcome of the conception and prototyping of Gossip Insights several examples using different datasets were created. To validate the design, interactions, structure and idea of Gossip Insights the user research is implemented and presented below. The conclusions of both the research sessions and evaluating the defined objectives and requirements (Chapter 1.2 and 1.3), are outlined below. Furthermore, it is derived which further steps are necessary and how this can improve the outcome of the pilot experiment. After prioritising, for some problems not only solution approaches are drafted but also implemented directly.

3.9.1 Research Methodology

A series of limited scope usability testing sessions are hold, with maximum 30 minutes per session. During the session, the participants are asked to work through a scenario with several tasks. The tasks are related to the main issues of the Topic Word Cloud which are tried to be addressed with the Gossip Insights prototype. The participants are requested to interact with both the Topic Word Cloud and the prototype. The interactions are observed, events are notated and the findings to derive further actions and improvements are summarised. The participants are split into two groups, the first is starting with the Topic Word Cloud and the second with the prototype to avoid biases. The sessions are screen recorded.

The questions and tasks listed below as well as their sequence are only a rough framework. Based on the progress of the interview, it can be decided whether certain questions are useful in the respective situation, other questions are more relevant, or the order must be changed. Reasons for this can be, among other things, that the participant answers the question on their initiative, that technical requirements are not met – especially in remote sessions – or that the participant lacks experience in specific areas.

3.9.2 Objects of Research

As already mentioned, two components will be the objects of research. These refer to the same query that is tailored to the restaurant chain Wagamama. This query and the peak associated with it are characterised by the fact that many problems of the classic Topic Word Cloud are covered. In the Topic Word Cloud, for example, a single term dominates that stands out from the crowd of other keywords. Also a topic that includes an authentic conversation with many unique contributions is covered. Retweets are also considered, since two of the three topics are mainly based on retweets of a single post. Last but not least, a range of topics with very different volume shares is covered: from about two-thirds to a few percent. Images of both components are shown in Appendix E.

3.9.3 Instructions and Scenario

The session is introduced with general instructions and the given scenario in order to provide the participant with the context that would be present in the normal working day:

We are going to be looking at both the Topic Word Cloud and the current prototype of the Gossip Insights component. While we are working through these tasks, if you would like to do something that is not included in the prototype, or if the prototype does not do what you expect it to do, just let me know. Please speak your thoughts aloud while interacting.

You are working on a report on the last month for Wagamama. You just identified a significant peak in the line chart and your task is to figure out the discussed topics causing this peak.

To better evaluate each participant's answers, the participant is subsequently asked to describe his or her role and associated responsibilities as well as familiarity with the Brandwatch Analytics and topic-specific components.

3.9.4 Research Questions

The research questions are grouped into general questions that are to be answered by the research sessions. These relate to different aspects of the prototype but are mainly focused on UX and comprehensibility. Contextual aspects from Chapter 1 are also considered.

- Do users find the navigation intuitive?
- Do users understand the meaning of colours, edges and nodes?
- Do users miss any interactions or hints?
- Do users find relationships between keywords (faster)?
- Do users get the required contextual information better than with existing components like the Topic Word Cloud?
- Which future use cases can internal stakeholders imagine

Based on these questions, more detailed questions and tasks are derived for the sessions, which can be divided into three sections: Topic Word Cloud, Gossip Insights and concluding questions, which help to compare both components directly. The full list of the research question guideline is shown in Appendix F.

3.9.5 Participants

In this case, the evaluation sessions are limited to internal stakeholders. In addition to the time limit and the stage of the prototype, this is mainly due to the fact that this year's product roadmap for Brandwatch Analytics has already been

decided. Involving external stakeholders such as customers could raise misguided expectations and hopes, which should be avoided.

The aim is to recruit participants with an analytical or product-oriented perspective. Employees from different departments, who differ in their relation to Brandwatch Analytics, were invited to participate. In order to evenly group the participants, an even number of participants is required per team/department. The participants and their assignment to team, department and group can be seen in the table 3.2.

Table 3.2: UX Research Participants

Name	**Role/Team/Department**	**Group**
Lydia Shaw	Social Media Data Analyst Professional Services Customer Success	1
Christopher Carnes	Social Media Data Analyst Professional Services Customer Success	1
Sarah Barber	Data Analyst Manager EMEA Professional Services Customer Success	2
Taya Reznichenko	Project Manager Professional Services Customer Success	2
Edward Crook	Director Strategy & Insights Revenue	1
Peter Fairfax	Senior Research Analyst Product, Analytics & Partners Revenue	2
Amy Barker	Product Manager Strategy & Insights Product	1
Emelie Swerre	Product Manager Product, Audiences Product	2

3.9.6 Research Findings

In the following, all feedback sessions are summarised and the most important conclusions are extracted in order to define necessary adjustments and to conceptualise suitable solutions. The two components are compared again to highlight the differences. The individual session notes are listed in Appendix G.

A big advantage of the Topic Word Cloud is that it seems to be very easy to identify retweet-based keywords, as they usually dominate the Topic Word Cloud. The significant font size also makes it obvious which keywords were generally mentioned most often. However, this leads to some disadvantages, as smaller keywords are often suppressed or ranked as less important even though it only visualises the volume. As a result, keywords in more widespread conversations are not considered further. Additionally, none of the participants was able to identify relationships between keywords without using the search and boolean operators, analysing the mentions behind or browsing different levels of the Topic Word Cloud. To clarify these points, the structure of the navigation is briefly explained below. The cloud consists of several levels, each with a different set of keywords. Clicking on a keyword leads to the next level, which displays new keywords that are related to the clicked one. The truncated keywords and ellipses are also a barrier. This revealed that due to the lack of context, users usually check the mentions to get clarity and the keywords are not trusted as stand-alone information. Last but not least, users are dissatisfied with the navigation concept, as they have to click through several levels to get the desired information. Besides, by each level the reference to the actual Topic Word Cloud is lost. The fact that individual analysts use external tools supports this statement; for this, a sample of mentions is exported and trends are analysed with external tools like other word cloud visualisations.

At first, the difficulties in understanding Gossip Insights its navigation concept were similar. On the one hand the required combination of mouse and keyboard usage was criticised and on the other hand other interactions were expected. A double click interaction without a single click interaction seems to be inconsistent. Since a graph as a visualisation was unusual for all participants, a legend of the basic interactions would simplify the first contact with the tool. According to the

participants, this legend could also be further expanded by other explanations, such as hints on segmentation. The underlying idea was to replace the cluster-based colour segmentation with a more fine-grained one, such as @-mentions, hashtags named entities or emojis. Since all participants were confused by the presentation of both @-mentions and authors, authors could be removed at the same time. In addition to segmentation, users also want to extend the visualisation with additional information and metrics. Besides using the edge length to visualise its strength, statistics and related mentions, would also enhance UX.

Despite the lack of mentions, which are usually used to gain more confidence in their conclusions, most participants identified all topics and important connections quickly. As a result, Gossip Insights was rated as more insightful and natural, but less helpful due to missing mentions. This experience seems to be due to the fact that all participants were able to quickly acquire a detailed understanding of the respective parts of Gossip Insights – such as nodes, edges, clusters and node size. Even the different types of clusters, such as widespread discussions or retweet-based clusters, were usually identified. Even though the navigation concept was unclear with regard to the different user inputs, the several levels of detail were a feature often used to reduce the visualisation its complexity.

3.10 Conclusion

This chapter covered the first stage of the prototyping process, which consists of a pilot experiment and its evaluation with internal stakeholders. During the evaluation, further requirements were identified in addition to those defined in the beginning. These were prioritised and described in detail to be able to carry out their conception and implementation in the next step.

Concerning the objectives defined in Chapter 1.2, further questions could be answered through experimental procedures and questions already answered could be supplemented with details. Thus, pre-processing steps were described with which anomalies and noise in texts of social media can be reduced, and sharings can be included. Besides, the Thread Tree was introduced as an approach that allows threads to be considered. Furthermore, a technique was selected to select the most

important keywords for each cluster as well as open points regarding the extraction and ranking could be answered by choosing suitable algorithms/heuristics. The feedback also revealed how the visualisation of the clustered keywords can be optimised. This means that all remaining targets have been met.

Chapter 4

Prototype

On the basis of the feedback sessions and the conclusions derived in Chapter 3.9.6, not only various steps are conceived in the following, but also insights into their implementation are given. This refers to both the visualisation and the basic steps of data processing. The aim is to improve the Gossip Insights prototype to further fulfil the demands of the users. Due to the prototype character, this is explicitly not about selecting the best model for tokenisation and POS tagging.

4.1 Embed Related Mentions

Similar to existing topic components, it should be possible to display the mentions for each keyword in which the respective keyword occurs. In previous components the twenty most current mentions were displayed and it is possible to apply various filters; a pagination is usually implemented to get further mentions. This should also be implemented in the graph, apart from the fact that neither pagination nor duplicates should be displayed; retweets are reduced to a single tweet to preserve the simplicity. Furthermore, filters are not necessary for the prototype.

To find the identifiers of the corresponding mentions, the pipeline is extended. In this step, the program searches in reverse chronological order for matching mentions for each node (algorithm 4.1); the identifiers are mapped to the nodes. Up to 250 unique mentions are assigned to the nodes, even if only a fraction of

them is displayed. Since the mentions should not only be embedded in nodes but also in edges – to show mentions in which both keywords occur – it is ensured that both nodes have been assigned sufficient mentions to obtain meaningful intersections. However, it is not the list of unique mentions assigned to the nodes, but the complete one. This can be used to determine at a later stage how much is made up of retweets (Chapter 4.4).

Algorithm 4.1: Algorithm to Fetch Node-related Tweets

Input: mentions – A list of processed mentions
ids – A list of identifiers matching the mentions
nodes – A list of all nodes
Output: List of nodes with associated mentions

```
1  mentions ← reverse order of mentions
2  ids ← reverse order of ids
3
4  foreach node ∈ nodes do
5      mentionIdsSet ← empty set
6      mentionIdsList ← empty list
7
8      foreach mention ∈ mentions do
9          if terms of node ∈ mention then
10             append mention to set of mentionIdsSet
11             append mention to list of mentionIdsList
12
13         if length of mentionIdsSet = 250 break
14
15     assign mentionIdsList to node
```

In contrast to the Topic Word Cloud, the mentions are not displayed in an overlapping window that covers the actual visualisation, but in a sidebar. This is based on the findings in chapter 3.9.6, which show that users have difficulties with this view because context and the reference to the visualisation are lost. The sidebar remains hidden at first and is displayed as soon as loading of the mentions – caused by the double-clicking a node – is finished; the mentions are embedded with the help of the Twitter gadget [6]. Besides, the sidebar clearly shows which keywords are assigned to these. Since the sidebar only occupies a small area of the screen, you can interact with the visualisation as usual. If the user double-clicks another node, the sidebar remains open, mentions and header

are exchanged, and the container is scrolled to the top. If the selected node is behind the potential position of the extended sidebar, the visualisation is shifted so that the node is still visible; closing the sidebar reverses this shift.

The pagination and filters are considered for future implementations, but is not important for the prototype. In addition, the used Twitter widget can be replaced by a custom implementation for further customisations.

4.2 Remove Authors

The reason why authors are in the graph besides @-mentions is that Twitter flags retweets with `RT <author_handle>:`. Thus the authors, whose tweets have been retweeted very often, get into the visualisation. To avoid having to look for extraction of the keywords, which of the Twitter handles originate from retweets and which are actual @-mentions, those flags are already removed during preprocessing using regular expressions.

Unlike when modifying the generated tokens, a simplified pattern can be used here. The reason for this is that by combining `RT` and `<author_handle>` at the beginning of the mention, the probability of removing other text fragments is very low. Optionally, a trailing string consisting of colon and whitespace is removed. Together with the `IGNORECASE` flag, the following regular expression is compiled in advance:

Listing 4.1: Regular Expression to Remove retweet-Flags from mentions

```
1  import re
2
3  TWITTER_RETWEET_RE = r'(rt\s@[a-z0-9_]{1,15}):?\s?'
4
5  re.compile(TWITTER_RETWEET_RE, re.IGNORECASE)
```

4.3 Change Colour-based Segmentation

The colour-based segmentation will no longer be used for clusters. During the feedback sessions, reference was made to various segmentations already existing in Brandwatch Analytics. These include @-mentions, hashtags, sentiments (positive, neutral and negative), named entities (organisations, people and locations) and remaining usual keywords. The colours are used according to the style guide to embedding such segmentation into nodes.

In the first step, particular focus is placed on hashtags, @-mentions and usual keywords are highlighted accordingly. Sentiment and named entities are therefore initially neglected and considered for future steps and extensions. This is mainly due to the fact that this would exceed the time and functional scope of the prototype. Since hashtags and @-mentions can be easily identified by their prefix, they can be included in the schedule.

4.4 Highlight Retweet-based Clusters

Even if most participants of the UX research sessions were able to independently identify the different nature of the clusters and thus in particular to find retweet-based clusters, these should be explicitly marked as such by embedding this information in the nodes. Since the colour is already used as an attribute for the segmentation of the keyword types, other shapes are used in this case; all retweet-based clusters or subclusters are therefore displayed as squares.

To identify these clusters, this step uses the embedded list of associated mentions. For this purpose, all mentions are unified per subcluster, not the aggregated ones, and the ratio of uniques to all mentions is determined. The threshold value is set at 90% so that those subclusters whose unique tweets represent only 10% are marked as retweet-based.

$$retweetRatio = 1 - \frac{|\{x_1, \ldots, x_n\}|}{|(x_1, \ldots, x_n)|} \tag{4.1}$$

Also, retweet-based clusters should be hideable to be able to focus on more wide-ranging clusters. Initially, however, all clusters are displayed so that no information is withheld.

4.5 Simplify Navigation Concept

The basic navigation concept should be based on the depth of the interaction and be reflected in the provided information. No inconsistencies should be created, i.e. the interactions start with the hovering, can be extended by a single click and end with a double click. When hovering a node, the respective label is displayed directly above the node; when clicking once, all nodes and edges that are not directly connected to the node are hidden; when double-clicking, the corresponding mentions are displayed, as described in Chapter 4.1. Equivalent to the interactions with the nodes, these are also introduced for edges, so that the edge and the two associated nodes are shown with one click, while the others are hidden. A double click, displays the mentions belonging to the edge. This edge-specific behaviour is almost equivalent to the previous solution, so that it can be easily extended.

Using the keyboard to return to the overview is replaced by a single click: if the whitespace is the target of a click and not a node, one step back is taken. So it is possible to get back from the view of the mentions and the subgraph.

To simplify the different states of nodes and edges - shown, hidden and greyed out - and their visualisation, various attributes are used for design: `opacity`, `fill-opacity` or `stroke-opacity`, as well as `display`. This allows the display of the respective elements to be influenced at several levels.

4.6 Legend for Navigation and Segmentation

To simplify the introduction and the working with the visualisation according to the feedback of the users and the resulting findings, Gossip Insights is extended by a legend in the form of a further sidebar. In addition to explaining the main

interactions, this should also show the segmentation and provide the opportunity to hide and show the retweet-based clusters. To illustrate the depth of the interactions, this is reflected in the legend by starting with the hovering and listing clicks and double-clicks afterwards. However, since the sidebar can be distracting and irritating, especially at a higher zoom level, its state is also linked to the zoom level. As soon as the zoom threshold is exceeded, the sidebar disappears, allowing the user a better overview.

4.7 Use Distance Between Nodes as Metric

The UX research sessions revealed that participants either misunderstood the rather simple circular layout or asked for more informational content. This coincides with the statements of Borgatti *et al.* that poorly laid out networks not only convey too little information but can also be misunderstood [18]. As a result, the layout is adapted, and the weighting is embedded in the edges. Besides the width, colour and style of the edges, the distance between nodes is available as a possible type of embedding [18].

To do this, the models and forces of Bostock's 'Clustered Force Layout I' are replaced by those provided by D3, making not only the layout more contextual but also easier to maintain. Instead of the usual force layouts based on spring forces and Coulomb's law, charge-based forces are used. While negative charges, i.e. low weightings or unconnected nodes repulse, positive charges, i.e. nodes that are connected with strongly weighted edges, attract [52, 107]. The distance is also influenced by a weak geometric constraint, where a function determines the optimal distance between the respective nodes [107]. This results in natural subclusters according to the data, which, among other things, visualise conversations on the topic of a cluster (figure 4.1).

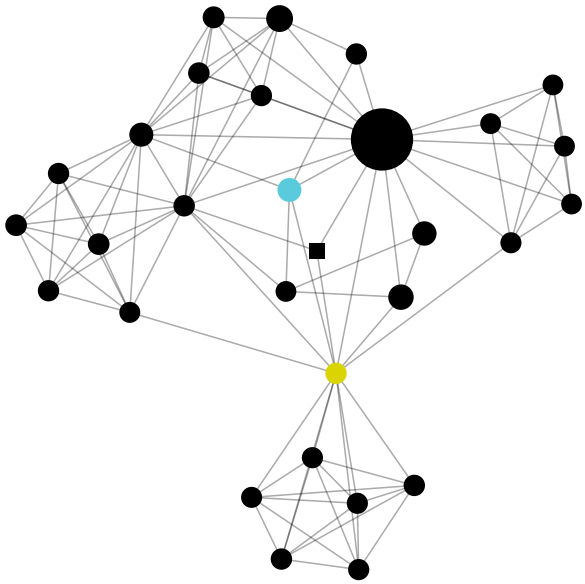

Figure 4.1: Subclusters within a Common Topic Using the query ‘kfc’

4.8 Evaluation

To evaluate the prototypical implementation finally, the requirements defined at the beginning (Chapter 1.3), the approaches derived from related work (Chapter 2.5) and the UX research findings (Chapter 3.9.6) are examined and evaluated. At this point it is explicitly emphasised that this is a prototypical implementation and especially the functionality and the concept are in focus; the quality of the implementation or the selection of the best models is of minor importance.

Concerning the requirements, as can be seen from table 4.1 - both the mandatory and the optional ones are fulfilled. The final version of Gossip Insights is an unsupervised approach that extracts n-grams of different lengths out of a sampled and normalised Twitter dataset using POS tag patterns and multi-stage rankings. By using graphs, contextual relationships are included and displayed, resulting in clustering as well as nodes and edges; both the edges and the clusters take threads into account. The visualisation in the form of a graph also enables new metrics for weighting the nodes, so that retweets are less significant. Gossip Insights is

completed by a variety of interactions with the visualisation and the mapping of the most important node per cluster.

Table 4.1: Implementation Status of the Requirements

Priority	Requirement	
Premises		
obligatory	The algorithm is unsupervised	✓
obligatory	The algorithm does require a relatively low number of mentions for meaningful results	✓
obligatory	Document corpus is preprocessed and normalised for better generalisation and therefore better results	✓
obligatory	Focus on Twitter content	✓
Extraction		
obligatory	Algorithm should extract n-grams with $n \geq 1$	✓
obligatory	Take contextual relationships into account to visualise multiple discussed topics	✓
obligatory	Extract & visualise threads if available and relevant	✓
optional	Take retweets and sharings into account, especially regarding the visualisation	✓
Clustering		
obligatory	Cluster the resulting n-grams to visualise contextual relationships as well as threads	✓
obligatory	Enable an interactive visualisation	✓
optional	Define a single n-gram as general topic per cluster	✓

Reviewing the approaches that were classified as reasonable and worth considering in the analysis of related work, it results that nearly all approaches and features were meaningfully implemented in the final version of Gossip Insights without unnecessarily increasing the complexity (table 4.2). Thus, only word embeddings and PageRank were omitted. Through co-occurrences, sufficient contextual relationships can be created to visualise the keywords properly. No clusters arise, for which the additional creation of a semantic context would be necessary. As already explained, PageRank was used temporarily but was replaced by the more straightforward approach with degree centrality, which achieves the same results without increasing complexity.

Table 4.2: Implementation Status of Considered Features and Approaches

Type	Description	
Feature	Number of co-occurrences in tweets and threads.	✓
	Term Frequency in comparison to background data.	✓
	Subsumption count of terms across other keywords.	✓
	Length/n of n-grams.	✓
	Terms' part of speech tags.	✓
	Similarity of word embeddings for semantical relationships.	
	Similarity of word embeddings for word disambiguation.	
Approach	Graph for keyword extraction and visualisation.	✓
	Merging of single documents into pseudo-documents.	✓
	Word Embeddings to identify contextual relations.	
	Combination of linguistic and statistical features.	✓
	Patterns of part of speech tags to target noun phrases.	✓
	Two-stage ranking using statical and graph-based metrics.	✓
	PageRank per graph and cluster to identify main topics.	
	Louvain as community detection approach.	✓

Regarding the findings summarised in Chapter 3.9.6, almost all weaknesses have been eliminated within the context of the prototype, so that it can be assumed that this adds further value; various brief practical tests with analysts confirm this impression. Due to the limited time and the narrow scope, two features are not implemented in the prototype. In addition to the statistics for nodes and clusters, this also includes the segmentation of named entities.

In summary, Gossip Insights meets the results of the requirements analysis and user feedback. Only two of the subsequently requested features are not implemented due to the limited time frame. In addition to the formal requirements, the prototype also proves itself in practice with various datasets.

4.9 Conclusion

This chapter covered the second stage of the prototyping process, which consists of the pilot experiments its further development into a prototype and a final evaluation. While the first evaluation focused on the basic concept and the UX, the final evaluation deals with fulfilling the requirements and user requests. The identified requirements were nearly completely implemented so that the resulting prototype not only meets all requirements but also meets the needs of the user. Screenshots of the final Gossip Insights visualisation of three different queries are provided in the Appendix H.

Chapter 5

Conclusion and Future Work

The contents of this work are summarised in the abstract. This chapter reflects on the work and provides a perspective for the further development.

In the process of researching approaches, technologies and implementations that deal with the given problem, only individual aspects and not an entirely suitable solution were identified. On closer examination of these, approaches were extracted which seemed reasonable and promising in combination. The challenges became evident only in the conceptual design and implementation of the pilot experiment. First, the extraction of keywords based on POS tag patterns turned out to be difficult because models with noisy social media data partly encounter difficulties. Second, the balanced and meaningful selection of keywords through ranking and selection of keyword candidates, nodes and edges. Moreover, last but not least, contextual visualisation, which embeds various metrics while maintaining the balance between detail and overview. However, the use of the pilot experiment for several datasets and its evaluation in the form of a UX research session revealed, apart from proof of the concept, that the implementation has to be extended by further features, which increase in particular the UX, but also the confidence of the users in the visualisation. For this purpose, solutions could be realised immediately after that. Their implementation, the fulfilment of all requirements, objectives and almost all user requests as well as the convincing way the prototypical implementation works with different datasets ensure that analysts can already use it on a trial basis.

For the further development of the prototype, various modifications and extensions are possible; a summary of some of them is given below, which is intended to provide a perspective for the future. First of all, the features that were requested during the UX research sessions, such as embedding statistics on clusters, nodes and edges – like the volume, share or sentiment – as well as extending the segmentation by named entities or emojis, which would need a NER step. The mentions and the associated sidebar can also be adjusted. It is feasible to replace the Twitter widget with a custom implementation to highlight the corresponding keyword in the individual tweet. Also, interactions and features of existing components could be adapted to make mentions filterable and sortable as well as to introduce pagination. As Borgatti *et al.* demonstrate, graphs can also be extended to scatter plots by using axes to visualise attributes. Thus, the volume, the cluster size, impact, sentiment, trend related factors or others can be mapped [18]. Even if the prototype already allows to highlight 1-degree ego networks, it would be potential to extend this functionality. For example, with n-degree-highlights or the temporary removal of ego nodes to identify subclusters more easily. This also includes highlighting nodes which are connected to all other nodes within a cluster to simplify the identification of nodes worth removing. The last functional extension is the visualisation of time series to be able to observe the evolution of a graph within an interval. The remaining aspects are mainly focused on improving the suitability as a production system. In addition to the direct connection to the database systems and the already mentioned integration into the peak/event detection, the sampling size and thus the required time can be minimised. Moreover, the delimitations defined in chapter 1.4 can be lightened, and thus other data sources, as well as other languages, can be considered. This would mean that POS tagging models and stopword lists would have to be evaluated for other languages. Furthermore, it would be necessary to validate that Gossip Insights performs well even with longer texts.

It remains to be seen how the company will continue to develop or how others will adapt to Gossip Insights. However, the concept and the prototype provide a solid basis for adaptations and extensions that affect not only the performance and suitability for production systems but also the functionality.

Appendix A

Stored Mention Data

Listing A.1: Exemplary Mention Returned by Brandwatch API


```
{
  "accountType": "individual",
  "added": "2018-03-19T09:41:34.817+0000",
  "assignment": null,
  "author": "QueenThrift",
  "authorCity": "Redhill",
  "authorCityCode": "re20",
  "authorContinent": "Europe",
  "authorContinentCode": "eu",
  "authorCountry": "United Kingdom",
  "authorCountryCode": "uk",
  "authorCounty": "Surrey",
  "authorCountyCode": "sur7",
  "authorLocation": "eu,uk,engb,sur7,re20",
  "authorState": "England",
  "authorStateCode": "engb",
  "avatarUrl": "https://pbs.twimg.com/profile_images/851534850364493824/
     JYKLzDyS_normal.jpg",
  "averageDurationOfVisit": 20,
  "averageVisits": 6,
  "backlinks": 49850734,
```

```
  "blogComments": 0,
  "categories": [],
  "categoryDetails": [],
  "checked": false,
  "city": "Redhill",
  "cityCode": "re20",
  "continent": "Europe",
  "continentCode": "eu",
  "country": "United Kingdom",
  "countryCode": "uk",
  "county": "Surrey",
  "countyCode": "sur7",
  "date": "2018-02-17T01:36:30.000+0000",
  "displayUrls": [
     "pic.twitter.com/sAjPG229Zg"
  ],
  "domain": "twitter.com",
  "engagement": 0.0,
  "expandedUrls": [
     "https://twitter.com/BarnesyGillian/status/964272837384122374/
        photo/1"
  ],
  "facebookAuthorId": null,
  "facebookComments": 0,
  "facebookLikes": 0,
  "facebookRole": null,
  "facebookShares": 0,
  "facebookSubtype": null,
  "forumPosts": 0,
  "forumViews": 0,
  "fullText": "RT @BarnesyGillian: @wagamama_uk your window decorations
     are a bit saucy aren't they? pic.twitter.com/sAjPG229Zg",
  "fullname": "Rachel",
  "gender": "female",
  "id": 176259116247,
  "impact": 48,
  "importanceAmplification": 30,
  "importanceReach": 66,
  "impressions": 2104,
  "influence": 713,
  "insightsHashtag": [],
  "insightsMentioned": [
```

```
        "@barnesygillian",
        "@wagamama_uk"
    ],
    "instagramCommentCount": 0,
    "instagramFollowerCount": 0,
    "instagramFollowingCount": 0,
    "instagramInteractionsCount": 0,
    "instagramLikeCount": 0,
    "instagramPostCount": 0,
    "interest": [
        "Food & Drinks",
        "Animals & Pets",
        "Family & Parenting"
    ],
    "language": "en",
    "lastAssignmentDate": null,
    "latitude": 0.0,
    "locationName": null,
    "longitude": 0.0,
    "matchPositions": [
        {
            "start": 21,
            "text": "wagamama",
            "length": 8
        }
    ],
    "mediaFilter": null,
    "mediaUrls": [
        "http://pbs.twimg.com/media/DWHJA1iXOAYOBRl.jpg"
    ],
    "monthlyVisitors": 6000000000,
    "mozRank": 9.6,
    "noteIds": [],
    "originalUrl": "http://twitter.com/BarnesyGillian/statuses
        /964272837384122374",
    "outreach": 7,
    "pageType": "twitter",
    "pagesPerVisit": 22,
    "percentFemaleVisitors": 46,
    "percentMaleVisitors": 54,
    "priority": null,
```

```
  "professions": [],
  "queryId": 1999288959,
  "queryName": "wagamama",
  "reach": 713,
  "replyTo": null,
  "resourceId": 176259116247,
  "resourceType": "page",
  "retweetOf": "http://twitter.com/BarnesyGillian/statuses
     /964272837384122374",
  "sentiment": "neutral",
  "shortUrls": [
     "https://t.co/sAjPG229Zg"
  ],
  "snippet": "RT @BarnesyGillian: @wagamama_uk your window decorations
     are a bit saucy aren't they? pic.twitter.com/sAjPG229Zg",
  "starred": false,
  "state": "England",
  "stateCode": "engb",
  "status": null,
  "subtype": null,
  "tags": [],
  "threadAuthor": "BarnesyGillian",
  "threadCreated": null,
  "threadEntryType": "share",
  "threadId": "0",
  "threadURL": null,
  "title": "Rachel (@QueenThrift): RT @BarnesyGillian: @wagam ...",
  "trackedLinkClicks": 0,
  "trackedLinks": null,
  "twitterAuthorId": "393338307",
  "twitterFollowers": 2104,
  "twitterFollowing": 514,
  "twitterPostCount": 34263,
  "twitterReplyCount": 0,
  "twitterRetweets": 0,
  "twitterRole": null,
  "twitterVerified": false,
  "updated": "2018-03-19T09:41:34.817+0000",
  "url": "http://twitter.com/QueenThrift/statuses/964674887804743680",
  "wordCount": null
}
```

Appendix B

Twitter Thread Tree

Algorithm B.1: Utilities for Creating the Thread Tree

```
1  def getTweetIds(ids, replyTos, expandedUrls) is
2      merge elements of replyTos and expandedUrls
3      extract identifier for each element
4      remove falsy values and duplicates
5      update list with ids
6      return list of unique tweet identifiers
7
8  def getThread(threads, tweetId) is
9      check if any of the stored threads contains tweetId
10     return threadId or None
11
12 def getThreadId(cache, tempCache) is
13     return cache or first element in tempCache
14
15 def prepend(tweetId, tempCache) is
16     prepend tweetId to tempCache if truthy
17     return tempCache
18
19 def getReplyId(status) is
20     replyId ← in_reply_to_status_id_str of status or None
21     quotedId ← quoted_status_id_str of status or None
22     return replyId or quotedId
```

Algorithm B.2: Thread Tree Creation Process

Input: ids – A list of all extracted mention identifier
replyTos – A list of all extracted replyTo fields
expandedUrls – A list of all extracted expandedUrls fields
Output: Mapping of thread identifiers to subordinate mentions

```
1  tweetIds ← getTweetIds(ids, replyTos, expandedUrls)
2  threads ← empty dictionary
3  tempCache ← empty list
4
5  foreach tweetId ∈ tweetIds do
6      remove all elements in tempThread
7      replyId ← tweetId
8      status ← None
9      cache ← None
10
11     if getThread(threads, tweetIds) then
12         return threads;
13
14     while true do
15         prepend(replyId, tempCache)
16         status ← fetch tweet data by identifier
17         replyId ← getReplyId(status)
18         cache ← getThread(threads, replyId)
19
20         if cache or not replyId then
21             replyId = None
22             break
23
24     prepend(replyId, tempCache)
25     threadId ← getThreadId(cache, tempCache)
26     update threads with {threadId: tempCache}
27
28 return threads
```

Appendix C

Plots of Sampling-related Jaccard Scores

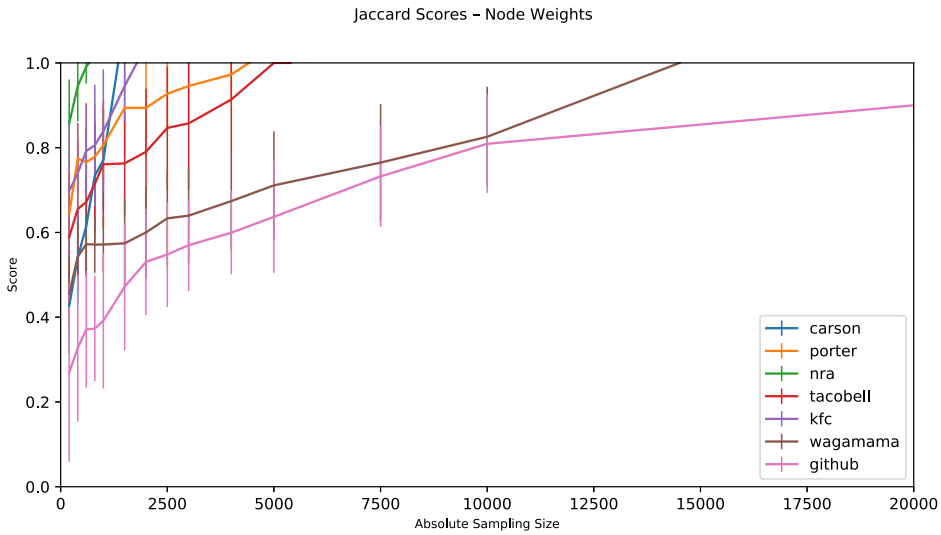

Figure C.1: Scores of the Nodes Extracted by Using the Top 150 TF-IDF Scores and the Top 30 Weighted Nodes

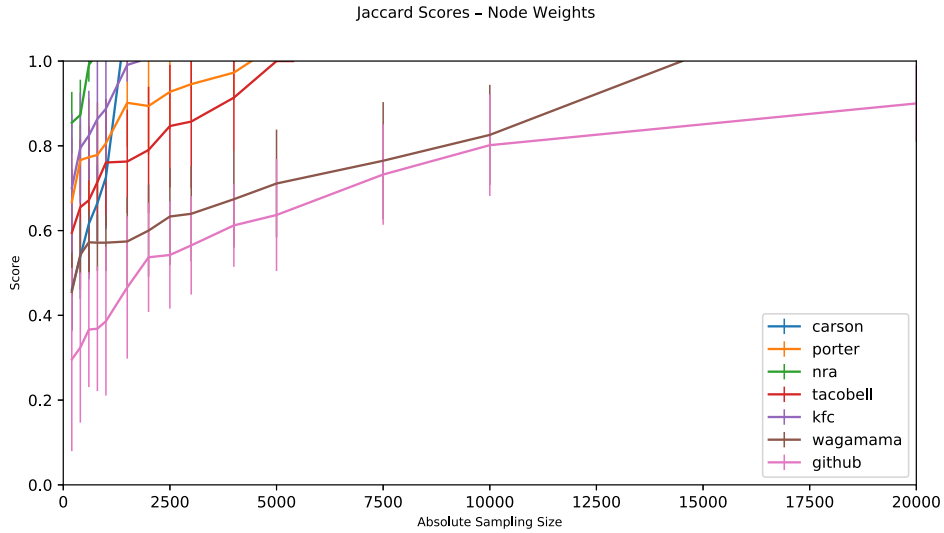

Figure C.2: Jaccard Scores of the Nodes Extracted by Using the Top 100 TF-IDF Scores

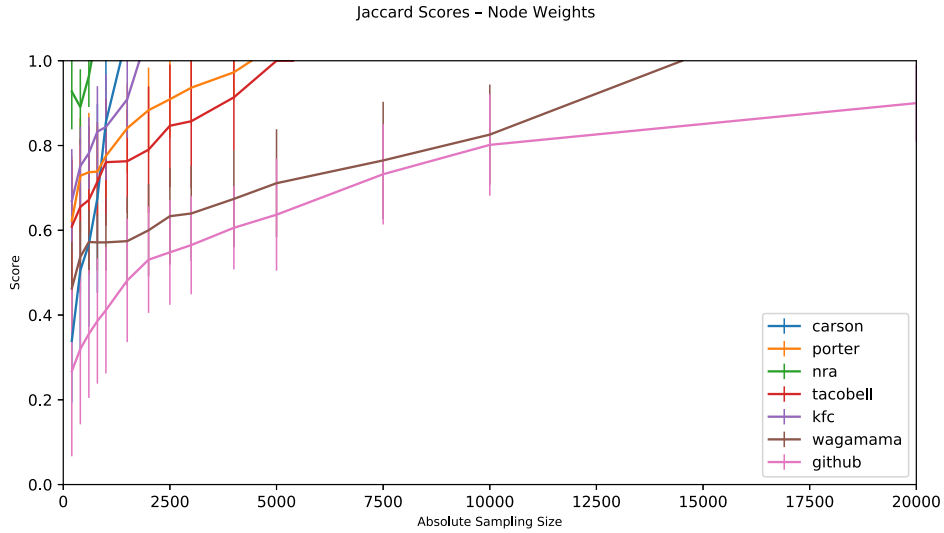

Figure C.3: Scores of the Nodes Extracted by Using the Top 150 TF-IDF Scores and the Top 150 Weighted Edges

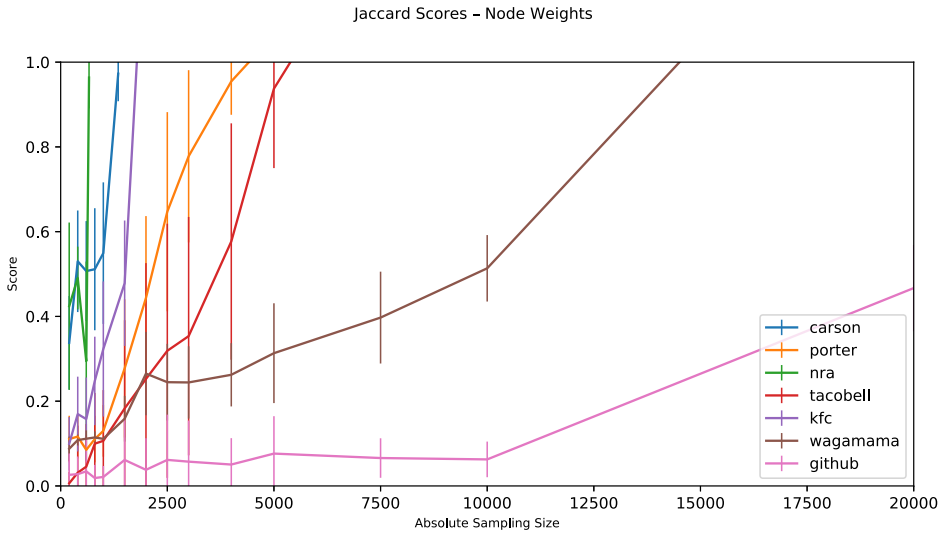

Figure C.4: Scores of the Nodes Extracted by Using the Top 100 Z-Scores

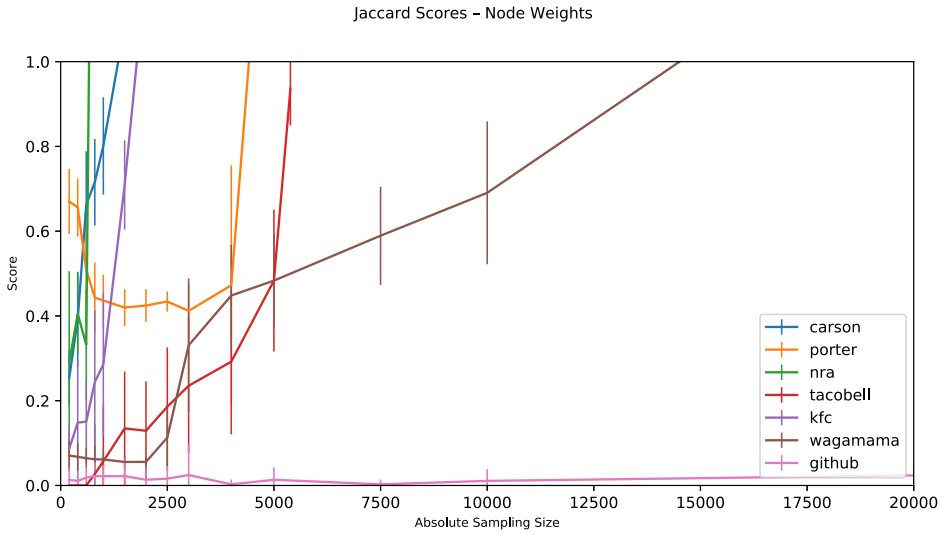

Figure C.5: Scores of the Nodes Extracted by Using the Top 150 Z-Scores and the Top 150 Weighted Edges

Appendix D

Exemplary Tweets of Datasets

Query: Wagamama

Figure D.1: Wagamama – Tweet about Kim Kardashian [103]

Figure D.2: Wagamama – Tweet about Minimum Wage [68]

Figure D.3: Wagamama – Tweet about a Competition on Mothers's Day [31]

Query: Carson

Figure D.4: Carson – Tweet about the General Dining Room Scandal [82]

Figure D.5: Carson – Tweet about Trump Administration [35]

Figure D.6: Carson – Tweet about Firing Ben Carson [26]

Figure D.7: Carson – Tweet about Spending Social Projects' Money [100]

Query: KFC

Figure D.8: KFC – Tweet about Journalists [72]

Figure D.9: KFC – Tweet about an Employee’s Statement [13]

Figure D.10: KFC – Tweet about the Police [99]

Appendix E

Screenshots of Objects of User Experience Research

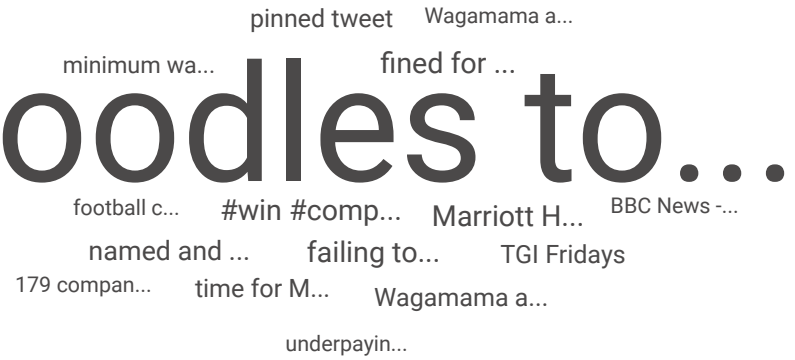

Figure E.1: Topic Word Cloud Related to Wagamama

The basic Topic Word Cloud related to the query about Wagamama. Hovering a specific term shows the complete keyword as well as information about the count of related mentions.

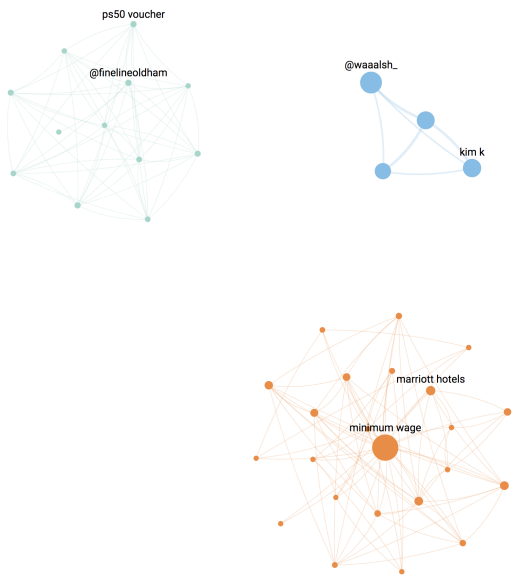

Figure E.2: Overview of the Gossip Insights Visualisation

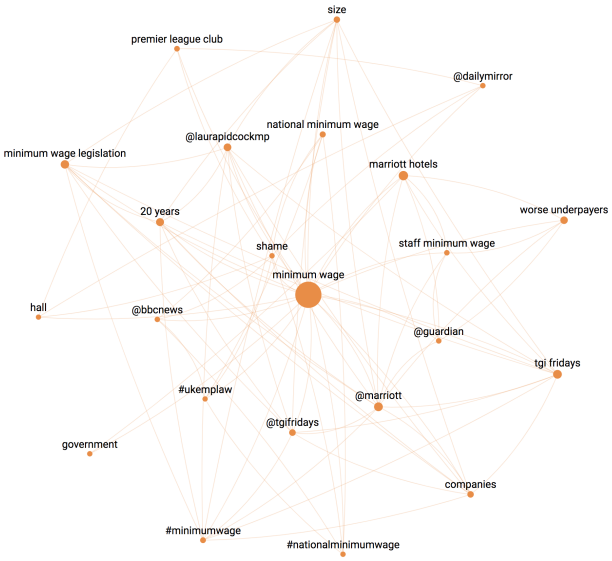

Figure E.3: Detail View of a Single Cluster in Gossip Insights

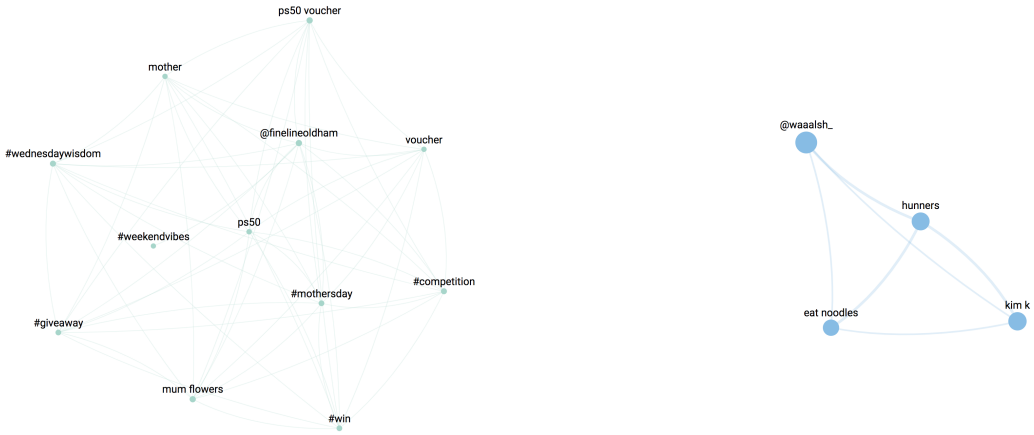

Figure E.4: Detail View of Two Clusters in Gossip Insights

Appendix F

Research Question Guideline

Topic Word Cloud

- What are the mainly discussed topics at a first look?
- What are the most important insights to you here?
- What is the relationship of Marriott Hotels to TGI Fridays?

Gossip Insights

- Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?
- What are the most important insights to you here?
- Interact with the prototype on your own.
  Figure out basic navigation.
- What are the mainly discussed topics at a first look?
- What is the competition about?
- Who is responsible for the 'eating topless noodles' news?
- Is there anything you think is not required?
- Is there anything missing in your opinion?

Conclusion

- Do you have any other thoughts or comments you would like to share?
- Which visualisation felt more natural, insightful or useful?

Appendix G

User Experience Research Session Notes

Christopher Carnes

Role: Social Media Data Analyst
Team: Professional Services
Department: Customer Success
Group: 1
Session: 25 June 2018 – 20 minutes – remotely

How would you describe your role and responsibilities?

- Helps and makes sure that clients get as much as possible out of Brandwatch.
- Creates dashboards/queries/reports on clients' behalf.

To what extent do you use Brandwatch Analytics?

- Every day as main part of his daily job.

How you ever tried to identify discussed topics? How often?

- Every day or rather every time he sets up a dashboard.
- Topics are the most important part of the dashboards.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- 'noodles topless' as most important topic according to the volume.
- Other topics around 'noodles topless' seem to be mentioned along with it.
- Detects TGI Fridays, Marriott Hotels and multiple minimum wage related keywords. Even identifies that minimum wage is an important topic but is not sure how this is related to 'noodles topless'.
- Assumes that the peak is about minimum wage.
- Does not detect the keywords about mother's day.

What are the most important insights to you here?

- A lot of people talked about Wagamama in the context of minimum wage.
- 'noodles topless' seems to be important as well.

What is the relationship of Marriott Hotels to TGI Fridays?

- Cannot figure out a real relationship at a first look.
- Needs to click through the component to get insights.
- Both Marriott Hotels andTGI Fridays are mentioned in articles about minimum wage.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- Couple of different networks.
- Minimum wage & Marriott Hotels are in one cluster.

- @-mentions and another person (Kim K) in another cluster.
  Not sure if the @-mention is an author or a mentioned account.
- Voucher and another Twitter handle.
- Not much description and so he is not sure what is the relation between those networks.

What are the most important insights to you here?

- He assumes that there is a relation between keywords, like minimum wage and Marriott Hotels.

Interact with the prototype on your own.
Figure out basic navigation.

- He is amused and interested about the mouseover effect of nodes, which reveals the underlying keyword.
- He identifies dragging as navigation tool quite fast.
- But he is not sure about how to interact further. Needs help.
- Zoom which and the escape key do not work remotely.
- He is not sure how to get back to the overview.

What are the mainly discussed topics at a first look?

- Marriott Hotels and minimum wage.
- Detects new keywords related to minimum wage:
  'premier league club' and '#ukemplaw'.
- He is not sure what is the role of '@dailymirror'.
- He really likes the way you can see the direct connections between keywords.
- He is able to identify important topics, authors and hashtags even without checking the underlying mentions.
- @waalsh_ and Kim K, not sure what is that topic about without mentions.
- He would have to take a look at the mentions for further details and insights.
- A competition with a voucher on mother's day.
- He assumes that the voucher is related to mother's day.
- He highlights that he likes the visualisation for more contextual information.

What is the competition about?

- Mother's day and a voucher.

Who is responsible for the 'eating topless noodles' news?

- @waalsh_

Is there anything you think is not required?

- n/a.

Is there anything missing in your opinion?

- Labels or better distinction between mentioned users and authors.
- Getting information about the volume per keyword.
- Getting access to mentions.

Conclusion

Do you have any other thoughts or comments you would like to share?

- It is quite hard to detect the edges. He suggests to increase the contrast.
- Labels for navigation controls.
  Like zooming, double clicking and going back to the overview.
- Clarify how those clusters are connected and what they have in common.
- Good to see different clusters for main topics.

Which visualisation felt more natural, insightful or useful?

- Gossip Insights felt more insightful.
- Topic Word Cloud just shows which topic got the most volume.
- Gossip Insights shows the individual connections between the topics.
- Get more context in form of people, organisations and hashtags.

Key Findings

Topic Word Cloud

- He gets the most topics but misses smaller ones.
- He does not the relationship between the several keywords without checking the underlying mentions.
- He needs to click through the component to get sub word clouds.

Gossip Insights

- He has a good basic understanding of the visualisation.
- Nice to get clustered, more fine grained topics for fast insights.
- The contrast of the edges is too low.
- He gets the basic navigation quite fast and intuitively.
- He struggles with deeper navigation concepts.
- Labels/hints for navigation and some kind of legend are missing.
- It is easy to get an overview without having a look at the mentions.
- He would like to get see cluster related mentions and statistics.
- He is confused of the mixture of @-mentionsand authors.
- He is not sure about relationships between clusters.

Amy Barker

Role: Product Manager
Team: Product, Analytics & Partners
Department: Product
Group: 1
Session: 26 June 2018 – 26 minutes

How would you describe your role and responsibilities?

- In between commercial, engineering, marketing & design.
- Decide which features will be included in the products to achieve business goals.

To what extent do you use Brandwatch Analytics?

- Used it especially in her previous role as analyst.
- Set up dashboards and queries in an advanced way.
- Gets in touch with Analytics 2-3x per day.

How you ever tried to identify discussed topics? How often?

- Every day or rather every time she set up a dashboard.
- Knows all the components as product manager.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- It is hard to get an overview because of ellipsis.
  So keywords are delimited until you hover them with the mouse.
- Identifies 'noodles topless' as keyword belonging to a retweets.
- Gets a topic about competition.
- Needs to see the mentions to get confident.
- The shown mentions are not perfect, because there is no good ranking metric.

What are the most important insights to you here?

- There are lots of retweets of the 'noodles topless' tweet.

What is the relationship of Marriott Hotels to TGI Fridays?

- Cannot figure out the relationship immediately.
- Needs to search with boolean operators to get relationship.
- Gets minimum wage as common topic.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- Identifies nodes as topics.
- Figured out tree clusters.
- Identifies edges as co-occurrences.
- Guesses that the node size is related to the frequency.
- Not sure if Twitter handles are @-mentions or authors.
- Not sure what it means when nodes are equally sized.

What are the most important insights to you here?

- The topic about 'noodles topless' has much less edges and nodes than the remaining ones.
- Some clusters have nodes with the same size and others have nodes with differing nodes sizes.
- Guesses that clusters with equally sized nodes are retweets.

Interact with the prototype on your own.
Figure out basic navigation.

- She identified the dragging and hover quite fast.
- Not sure how to interact further.
- Tries to single click and right click, without any reaction.
- Needs help with double click and going back to the overview.

- Assumed that clicking the whitespace would lead to go back to the overview.
- Happy to see all the keywords in the detail view.

What are the mainly discussed topics at a first look?

- Identifies that the shown conversation is the same as before.
- Does not get how '20 years' is related to minimum wage and why the 'minimum wage' node is much bigger. But was just a tiny mistake, she selected another node.

What is the competition about?

- Mother's day and a voucher.
- Has no idea what a 'ps50 voucher' is.
- Would like get the mention for further details.

Who is responsible for the 'eating topless noodles' news?

- @waalsh_

Is there anything you think is not required?

- Clusters for retweets because clusters imply that the topic is more wide spread. Maybe those clusters should be collapsed into a single node.
- Authors are rather confusing than helpful in this visualisation

Is there anything missing in your opinion?

- Getting access to mentions to get confidence and more detailed insights.
- A distinction between @-mentions and authors.
- Or even better: a segmentation like page type, gender, author, @-mention, hashtag,...
- Hints about the node size or insights what is the nature of the cluster, e.g. retweet, wide spread conversation,...

Conclusion

Do you have any other thoughts or comments you would like to share?

- In a similar project it is possible to extend the selected subnetwork iteratively. Would be nice to have this feature in this visualisation as well.
- The feature would enable a more exploratory style.
- Maybe use the same colour for each cluster to clarify that all the topics are related to one query and to enable colour-based segmentation.
- Suggests to keep the hover text near the node to keep the focus.
- Suggests to show subnetwork on hover.
- Suggests to replace double with single click and
  to show mentions or statistics on single click.
- Suggests to avoid mixing keyboard and mouse actions.
- Nice to see a separation of different discussions and relationships.

Which visualisation felt more natural, insightful or useful?

- Gossip Insights feels more insightful.
- Topic Word Cloud feels more helpful at the current stage because you get more confidence with mentions and statistics.

Key Findings

Topic Word Cloud

- Hard to get relations between keywords in Topic Word Cloud and other existing topic visualisations.
- Ellipses prevent to get a fast overview without hovering single keywords.
- It is easy to identify keywords based on a retweet because of the emerging size.
- She gets all topics but needs to take a look at mentions for confidence and search with boolean operators to get relationships.

Gossip Insights

- She gets really fast a detailed understanding of all the components in the visualisation.
- Nice to get clustered keywords which their relationship, it is easy to identify different topics.
- She is confused by the mixture of @-mentionsand authors.
- Identifies retweet-based clusters fast due to the nodes' size.
- Navigation concept is not intuitive enough.
- She gets the basic navigation concept but is confused of the mixed keyboard and mouse commands.
- She expects that clicking the whitespace is a 'go back' gesture.
- Colours should be used for other segmentation instead of highlighting the clustering.
- Even if Gossip Insights is good to get an overview it is necessary to get access to the related mentions.
- Statistics might be useful to understand the importance of keywords.

Peter Fairfax

Role: Senior Research Analyst
Team: Strategy & Insights
Department: Revenue
Group: 2
Session: 26 June 2018 – 25 minutes

How would you describe your role and responsibilities?

- Manage all the research projects for larger clients, especially pharmaceutical brands.
- Define and select methodologies and approaches for research.

To what extent do you use Brandwatch Analytics?

- Multiple times every day as part of his job.
- But he uses custom scripts and solutions as well.

How you ever tried to identify discussed topics? How often?

- He uses topic components like the Topic Word Cloud every day.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- Identifies nodes as Twitter handles and phrases.
- Assumes that node size is related to the term frequency or another kind of importance metric.
- Identifies edges as co-occurrences.
- Identifies that there are differences in the thickness of edges and assumes that is related to the edge weight.

What are the most important insights to you here?

- He doesn't get any specific informations except of there are multiple discussed topics.
- After he gets hints related to the navigation, he gets further insights.
- Assumes that Marriott Hotels are mentioned in the context of minimum wage.
- Identifies that minimum wage is more widely discussed than the other topics.

Interact with the prototype on your own.
Figure out basic navigation.

- He doesn't identify the different ways to navigate.
- He asks for help how to get more details.
- He identifies how to zoom in and drag the visualisation.
- He expects to get back to the overview which a click in the whitespace.

What are the mainly discussed topics at a first look?

- Marriott Hotels and minimum wage.
- Detects new keywords related to minimum wage:
  'worst underpayers' and 'national minimum wage'.
- He is able to identify important topics, authors and hashtags even without checking the underlying mentions.
- @waalsh_ and Kim K, not sure what is that topic about without mentions.
- A competition with flowers on mother's day.

What is the competition about?

- Mother's day and flowers for mums .

Who is responsible for the 'eating topless noodles' news?

- @waalsh_

Is there anything you think is not required?

- n.a.

Is there anything missing in your opinion?

- He would like to see some example mentions to get confirmation.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- He takes a look at the snippets to get more insights.
- Identifies the topic about Kim Kardashian and its volume.
- Does not identify other topics based on the Topic Word Cloud.

What are the most important insights to you here?

- Kim Kardashian and topless noodles is the main topic with about $^2/_3$ of the total volume.

What is the relationship of Marriott Hotels to TGI Fridays?

- He skips the Topic Word Cloud again and switches directly to the related mentions.
- He uses the search and boolean operators to identify the common topic.
- Identifies that both are not paying the minimum wage.

Conclusion

Do you have any other thoughts or comments you would like to share?

- It is quite hard with the Topic Word Cloud to identify relationships without using the search and boolean operators.
- It is much easier with Gossip Insights to get such relationships.
- He would like to get more information about a pair of connected keywords.
- Get more information about the nature of a conversation, like retweet-based or more widely discussed topics.
- Topic Word Cloud is driven by retweets or automatically generated tweets, so it less helpful in uncovering unique conversations.

Which visualisation felt more natural, insightful or useful?

- Gossip Insights feels more natural but unless related mentions are missing, more context is missing.
- It is good for discovering topics but not to get confirmation and confidence.
- Instead the Topic Word Cloud provides more details with related mentions.

Key Findings

Topic Word Cloud

- He does not get all discussed topics, just the emerging term.
- He needs to take a look at the mentionsto get confidence.
- He needs to use the boolean search to get relationships.
- He complains about that the Topic Word Cloud is driven by retweetsand automatically generated tweets.

Gossip Insights

- He is able to identify different natures of clusters, for example the one about minimum wage is more widely spread and the one about the competition is retweet-based.
- He has a really good and detailed understanding of the various components within the visualisation.
- It is easy to get relations between keywords, more context and even to identify all topics without checking the related mentions.
- He would like to get related mentionsanyways, just to get confidence.
- He is confused by authors in the visualisation.
- He struggles with the deeper navigation concepts and how to go back to the overview.

Sarah Barber

Role: Data Analyst Manager EMEA
Team: Professional Services
Department: Customer Success
Group: 2
Session: 26 June 2018 – 19 minutes

How would you describe your role and responsibilities?

- Leading position in Professional Services.
- Lead a team of analysts who take project on for clients.
- Projects should support the clients' understanding of Brandwatch Analytics.
- In the meantime her job is more about leading than executing tasks on customer behalf. But she used to work as analyst.

To what extent do you use Brandwatch Analytics?

- Uses Brandwatch Analytics quite less compared to analysts.

How you ever tried to identify discussed topics? How often?

- She used to work with topic components and Brandwatch Analytics, so she is familiar with the platform.
- She even used to identify topics.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- She identifies nodes as keywords, authors or @-mentions.
- She assumes that the node size is related to a metric how important the node is, e.g. term frequency.
- She identifies edges as co-occurrences.
- She assumes that clustered nodes are related to the same topic.

- She figures out that totally separated clusters do not have anything in common except of the query.
- She is not sure if the Twitter handles are @-mentions or authors.

What are the most important insights to you here?

- She gets which clusters are based on retweets.
- She would need mentions for further context and information.

Interact with the prototype on your own.
Figure out basic navigation.

- She identifies the hover effects but needs help for deeper navigation like clicks and zoom.
- She struggles with getting back to the overview.

What are the mainly discussed topics at a first look?

- She identifies all three topics: minimum wage, Kim Kardashian and the voucher.

What is the competition about?

- She relates the voucher to flowers and mother's day.
- She assumes that the topic is less important because of the small nodes.

Who is responsible for the 'eating topless noodles' news?

- She gets Kim Kardashian as responsible person without any struggles.

Is there anything you think is not required?

- She would like to see mentions.

Is there anything missing in your opinion?

- Clearer distinction between @-mentions and authors.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- She struggles to identify keywords because of ellipses.
- She is not able to identify topics or relating keywords at a first look.
- It is not intuitive to get to the mentions related to the keywords.

What are the most important insights to you here?

- That the most important and emerging keyword is quite unimportant because it is neither about the query nor about Kim Kardashian.

What is the relationship of Marriott Hotels to TGI Fridays?

- She is not able to identify a relationship between both terms at a first look.
- She needs to click through the component, to see the sub word clouds and the mentions.
- She needs lots of time to get the relationship.
- Her suspicion is that both are related to minimum wage and gets finally proof after long time of clicking through.

Conclusion

Do you have any other thoughts or comments you would like to share?

- She would like to see what the node size indicates, e.g. more statistics or a legend.
- She thinks that the Topic Word Cloud does not give any insights into topics with less volume and a more unique conversation.
- The navigation concept of Topic Word Cloud is even worse because you have to navigate more often forward and backwards. You do not have a split screen.
- Gossip Insights enables to answer faster to questions like what is the peak about.
-

Which visualisation felt more natural, insightful or useful?

- Gossip Insights was definitely more useful in her opinion.
- In her opinion Gossip Insights was quite intuitive and she did quite well for the first time.
- Gossip Insights is easier to work with because you can see the connections between terms.

Key Findings

Topic Word Cloud

- The Topic Word Cloud does not allow to get fast insights because of ellipses and missing relations.
- It is necessary to click through to get further information but the navigation concept is bad and not intuitive.
- Does not get the interesting topics because of the issues mentioned above.

Gossip Insights

- She gets fast a detailed understanding of all the components in the visualisation and even the importance metric in the form of the node size.
- She is able to identify mainly discussed topics without any struggles.
- She is confused by the nodes which represent authors.
- She identifies different natures of clusters, like more unique conversation or retweet-based ones.
- In her opinion Gossip Insights is more insightful and helpful than Topic Word Cloud.
- Topic Word Cloud is not the best component for more widely spread conversations.
- She struggles with the navigation except of the hovering.
- She would like to get more statistics and a legend for the navigation concept.

Taya Reznichenko

Role: Project Manager
Team: Professional Services
Department: Customer Success
Group: 2
Session: 25 June 2018 – 19 minutes – remotely

How would you describe your role and responsibilities?

- She is a project manager for research projects.

To what extent do you use Brandwatch Analytics?

- She needs to understand Brandwatch Analytics in order to get accurate estimations for her projects and to explain components correctly to clients.
- Sometimes she uses the platform instead of analysts when the scope is small enough.

How you ever tried to identify discussed topics? How often?

- She does not use components on a daily basis but she knows all of them.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- She identifies nodes as keywords and people.
- She seems to be confused why people and keywords are mixed.
- She assumes that there should be a difference concerning the clusters but does not find anything.
- She assumes that the node size visualise the volume or similar metrics.
- She does not get what the edges mean.

What are the most important insights to you here?

- n/a.

Interact with the prototype on your own.
Figure out basic navigation.

- She likes the multiple levels of detail.
- She likes the detail view with reduced count of edges.
- She assumes that the distance between nodes is important.

What are the mainly discussed topics at a first look?

- She gets really detailed insights into the cluster about minimum wage.
- She names all the related organisations, media and the government.
- She cannot share any insights related to the topic about Kim Kardashian because of missing context.
- She describes the topic about mother's day and vouchers and shares all possible insights.

What is the competition about?

- Mother's day and vouchers.

Who is responsible for the 'eating topless noodles' news?

- She cannot figure out who is responsible.

Is there anything you think is not required?

- She would keep the visualisation as simple as possible and remove unnecessary/unimportant edges.

Is there anything missing in your opinion?

- She would like to show how exactly keywords are related to each other to get more context, e.g. show mentions.
- She would like to add the distance between nodes as a metric.
- She would like to get a segmentation between keywords, people and hashtags. Maybe even a segmentation with named entities.
- Optionally toggle the segmented keywords.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- She identifies the topics about minimum wage and Wagamama without being asked for.
- Keywords about the voucher are ignored.

What are the most important insights to you here?

- She needs to hover all keywords to skip the ellipses.
- She is slightly confused because of emerging keyword which does not provide any further details.

What is the relationship of Marriott Hotels to TGI Fridays?

- She needs to click through and is a bit confused of the underlying navigation concept.
- She wants to take a look at the mentions.
- She assumes that both are related to minimum wage but she is not sure and does not get any proof.

Conclusion

Do you have any other thoughts or comments you would like to share?

- In her opinion it is quite hard to get specific relationships without delving into the details view.
- She would like to exclude specific clusters.
- She likes that topics are clustered, so it is more obvious.

Which visualisation felt more natural, insightful or useful?

- She says that she did not get that there is a relationship between the clusters in Gossip Insights.
- The Topic Word Cloud seems to be more confusing because it provides just keywords without any connection or further context.

Key Findings

Topic Word Cloud

- She misses lots of information because of missing relationships and clustering.
- Navigation concepts seems to be confusing and misleading.
- She does not feel confident with her assumptions because she does not find proofs.

Gossip Insights

- She likes the clustering into conversations/topics.
- She likes the multiple levels of detail.
- She says that there are too much edges to keep the overview.
- It seems to be obvious what are the roles of the nodes but not of the edges.
- She would like to add some kind of segmentation.
- She would like to change the layout and to take the distance between nodes as metric into account.

Emelie Swerre

Role: Product Manager
Team: Product, Audiences
Department: Product
Group: 2
Session: 27 June 2018 – 12 minutes

How would you describe your role and responsibilities?

- She manages Brandwatch Audiences.
- Works in cooperation with lots of different teams.
- Has lots of contact with customers.

To what extent do you use Brandwatch Analytics?

- She sees herself a 'pro' Brandwatch Analytics user.
- Lots of contact with other product managers.
- She used to be in the research team, so she is comfortable with the platform.

How you ever tried to identify discussed topics? How often?

- She is familiar with all available topic components in Brandwatch Analytics.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- She identifies nodes as topics and people.
- She guesses that the node size is related to the importance and occurrence frequency of those terms.
- She recognizes multiple communities in the network.
- She is confused by the authors.
- She assumes that the authors indicate how much a topic is related to those authors.

- She identifies the edges as semantic relationships between keywords.

What are the most important insights to you here?

- She recognizes that the peak is driven bei three conversations/topics.

Interact with the prototype on your own.
Figure out basic navigation.

- She gets the hovering and panning navigation concept quite fast.
- Needs help with deeper navigation concepts.
- She gets that the nodes are a combination of people, hashtags and keywords.

What are the mainly discussed topics at a first look?

- She would like to see the sentiment.
- She assumes that one cluster is about minimum wage but is not sure if Wagamama is paying minimum wage or not.
- She gets the second topic about mother's day and vouchers.
- She is not able to identify the third topic because of missing context in the sense of mentions.

What is the competition about?

- She identifies the reason for the competition as a voucher for mothers's day.

Who is responsible for the 'eating topless noodles' news?

- She is not able to identify who is responsible.

Is there anything you think is not required?

- n/a.

Is there anything missing in your opinion?

- She would like to see related mentions.
- She would like to see a better distinction between different keyword types.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- n/a.

What are the most important insights to you here?

- n/a.

What is the relationship of Marriott Hotels to TGI Fridays?

- She identifies the relationship quite fast with clicking through various levels of the Topic Word Cloud.
- She does not need the search and any boolean operators.
- She uses the underlying mentions to get confidence.

Conclusion

Do you have any other thoughts or comments you would like to share?

- She would use colours for segmentation.
- She does not know what the node size means exactly.
- She likes the segmentation in the Topic Word Cloud and misses such feature in Gossip Insights.
- She likes seeing relationships between terms.

Which visualisation felt more natural, insightful or useful?

- In her opinion Gossip Insights is more insightful regarding relationships between keywords.
- The Topic Word Cloud just shows keywords and does not give any hints what the topics are.
- It is easier to get an overview.

Key Findings

Topic Word Cloud

- Easy to get relations between keywords in Topic Word Cloud by clicking through.
- It is easy to identify keywords based on a retweet because of the emerging size.

Gossip Insights

- She gets a detailed understanding of all the components in the visualisation.
- Nice to get clustered keywords which their relationship, so it is easy to identify different topics.
- Navigation concept is not intuitive enough.
- She is confused by the mixture of @-mentionsand authors.
- Colours should be used for other segmentation instead of highlighting the clustering.
- Even if Gossip Insights is good to get an overview it is necessary to get access to the related mentions.
- Statistics and a legend might be useful to understand the visualisation better.

Edward Crook

Role: Strategy & Insights Director
Team: Strategy & Insights
Department: Revenue
Group: 1
Session: 2 July 2018 – 19 minutes – remotely

How would you describe your role and responsibilities?

- Head of the research team in North America.
- Lots of experience at Brandwatch.
- Helps the team to create reports on behalf of clients and be some kind of social data consultancy for clients.

To what extent do you use Brandwatch Analytics?

- He uses Brandwatch Analytics on a weekly basis and is really confident with the platform.

How you ever tried to identify discussed topics? How often?

- He knows all the available topic components.
- Even though the components change fast, he feels confident to be up to date.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- He struggles because all the terms are truncated, so it is neither obvious nor intuitive.
- He needs to click on the biggest keyword to get the mentions.
- With the help of mentions he identifies the topic as a retweet-based on.
- He explains that he usually excludes retweets after identifying them.

- He misses the remaining topics because he identified the main cause for the peak.

What are the most important insights to you here?

- He identifies the hugest keywords.
- He usually just exports the data with a random sample and use an external platform to get better insights, because he struggles with using this component in general.

What is the relationship of Marriott Hotels to TGI Fridays?

- He does not assume that there is a relationship, so he usually would miss this.
- Usually, after he identified a relationship, he uses the search with boolean operators.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- He identifies nodes as a mixtures of Twitter handles and keywords.
- He assumes that the edges are co-occurrences.
- He assumes that the colour is just caused by a clustering algorithm.
- He assumes that the node size is based on the term frequency.

What are the most important insights to you here?

- There a multiple clustered topics which are not related to each other.
- He likes the different levels of details.
- He is interested in the visualisation its limitations and how its created.
- In his opinion it makes sense how the graph is composed and he identifies the underlying reasons for on his own.

Interact with the prototype on your own.
Figure out basic navigation.

- Navigation is done by the facilitator.
- He asks for more interactions like hovering for detailed informations and clicking.
- He is interested what happens as soon as a keyword belongs to multiple clusters.
- He likes the solution that keywords are not duplicated in such cases and all information is visible.

What are the mainly discussed topics at a first look?

- n/a.

What is the competition about?

- n/a.

Who is responsible for the 'eating topless noodles' news?

- n/a.

Is there anything you think is not required?

- n/a.

Is there anything missing in your opinion?

- He would like to see more information related to the edges.
- He would like to get statistics for nodes, clusters and edges.

Conclusion

Do you have any other thoughts or comments you would like to share?

- He likes the exploratory nature of Gossip Insights.
- A new feature in his opinion would be some kind of time series, to see how the keywords and the network change over time.

Which visualisation felt more natural, insightful or useful?

- It is not clearly stated, but based on the feedback on both components, he will probably prefer Gossip Insights.

Key Findings

Topic Word Cloud

- He does not get to much information at a first look because the terms are truncated.
- He needs mentions to get further information about a specific keyword.
- He misses relationships and other topics because there is no clustering but an emering term.
- He usually does not the Topic Word Cloud because he does not like it. He uses external tools instead.

Gossip Insights

- He gets a really detailed and fast understanding of the the parts within Gossip Insights.
- He likes the different levels of details and how the visualisation is composed.
- He would like to get more information about cluster,s nodes and edges.
- He likes the exploratory nature of the visualisation/component.
- He would like to see the component showing the clusters in a time series.

Lydia Shaw

Role: Social Media Data Analyst
Team: Professional Services
Department: Customer Success
Group: 1
Session: July 16th 2018 – 12 minutes

Note: Unfortunately there were technical problems during this session, so that besides the screen recording only handwritten notes and no audio recording are available.

How would you describe your role and responsibilities?

- She is a Social Media Data Analyst.
- Creates dashboards, reports and queries for clients on a weekly basis.
- Explains Brandwatch Analytics and its features to clients.

To what extent do you use Brandwatch Analytics?

- She uses the platform on a daily basis.
- She uses most of the available features.

How you ever tried to identify discussed topics? How often?

- Every time she has to create dashboard or report, she tries to identify the mainly discussed topics.
- She is aware of all available topic components.

Topic Word Cloud

What are the mainly discussed topics at a first look?

- She clicks trough the Topic Word Cloud.
- Gets quite fast the three main topics.
- mentions that she does not like the navigation concept and the component itself. She usually tries to figure out the topics by browsing the mentions.

What are the most important insights to you here?

- That the topic about Kim Kardashian is mainly based on a retweet.
- She uses the search to gain confidence.

What is the relationship of Marriott Hotels to TGI Fridays?

- She clicks through both underlying topic clouds to find overlapping keywords.
- Was intuitive and fast but she mentions that she usually would not assume a relationship between those keywords.

Gossip Insights

Without interaction, could you walk me through what you are seeing here and your understanding of the visualisation?

- She identifies that there are three conversations without any explicit relationship to each other.
- Gets that the nodes are keywords, authors, @-mentions or hashtags.
- Is not confused by the mixture of authors and @-mentions.
- Gets that edges are co-occurrences.
- Assumes that the distance between nodes matters because the nodes are positioned in a circular way.

What are the most important insights to you here?

- There are three independently discussed topics.
- The topic about Kim Kardashian is the one with the highest volume.
- The topic about minimum wage is most widespread.

Interact with the prototype on your own.
Figure out basic navigation.

- Figures out the hovering and dragging interactions really fast.
- Struggles with other interactions like zooming, double-clicking and going back to the overview.
- mentions that she would like to get rid of keyboard interactions.

What are the mainly discussed topics at a first look?

- Gets all topics without struggles: Kim Kardashian, voucher on mother's day and minimum wage.

What is the competition about?

- A voucher and flowers on mother's day.

Who is responsible for the 'eating topless noodles' news?

- She identifies both the author and Kim Kardashian.

Is there anything you think is not required?

- n/a.

Is there anything missing in your opinion?

- n/a.

Conclusion

Do you have any other thoughts or comments you would like to share?

- She mentions again that she does not like the Topic Word Cloud and usually use other components.
- She likes the more natural navigation concept and the graph as visualisation.
- She likes the more colourful visualisation.

Which visualisation felt more natural, insightful or useful?

- She liked Gossip Insights more because it is easier to get the topics, to see relationships and to get context.
- In her opinion Gossip Insights felt more natural, insightful and helpful.

Key Findings

Topic Word Cloud

- Navigation concept is misguiding and confusing.
- It is easy to get informations if you know what are you looking for.
- She does not assume relationships based on the Topic Word Cloud.
- She usually does not use the Topic Word Cloud.

Gossip Insights

- She likes that it is easy to identify different topics because of the clustering and shown relationships.
- She likes the colourful visualisation.
- She does not miss any feature or thinks that specific features are not necessary.
- The navigation with keyboard and mouse is confusing.
- She would like to use more intuitive interactions like single-clicks.
- It is easy to get a detailed understanding of Gossip Insights.

Appendix H

Screenshots of the Final Prototype

Below are screenshots of Gossip Insights for various queries. The light theme is enabled - the dark theme's visualisation and its legend consist of a dark background with white writing.

Most screenshots are available for the query 'wagamama' to show all views, interactions and states. In addition to the initial view, the hovering of nodes and the hiding of retweet-based clusters, this also includes various detail views as well as selected 1-degree ego networks or edges as well as the shown mentions sidebar. For the queries, 'kfc' and 'carson', only one screenshot of the initial and the detail view are provided for each. In addition to the feature scope, this should also provide an understanding of extracted keywords and the formed subclusters.

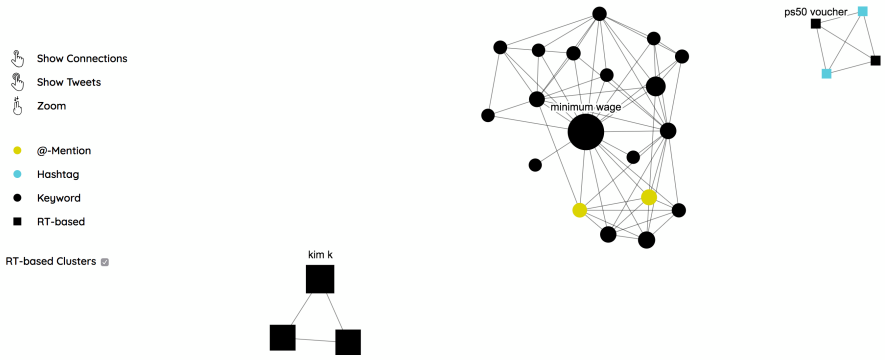

Figure H.1: Wagamama – Initial View

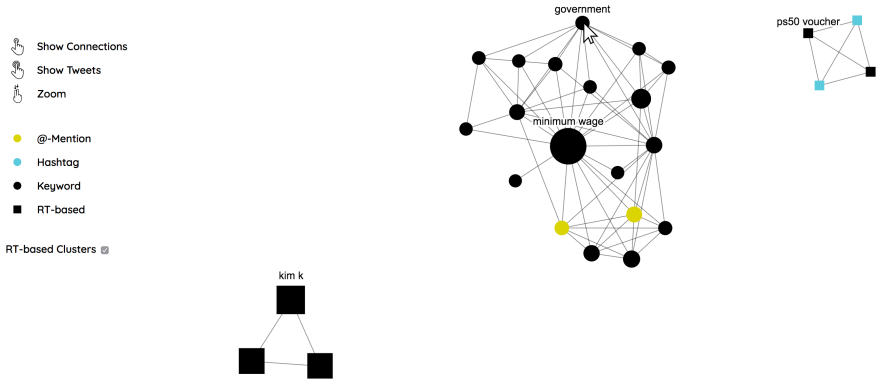

Figure H.2: Wagamama – Hovered Node in Initial View

Figure H.3: Wagamama – Hidden Retweet-based Clusters

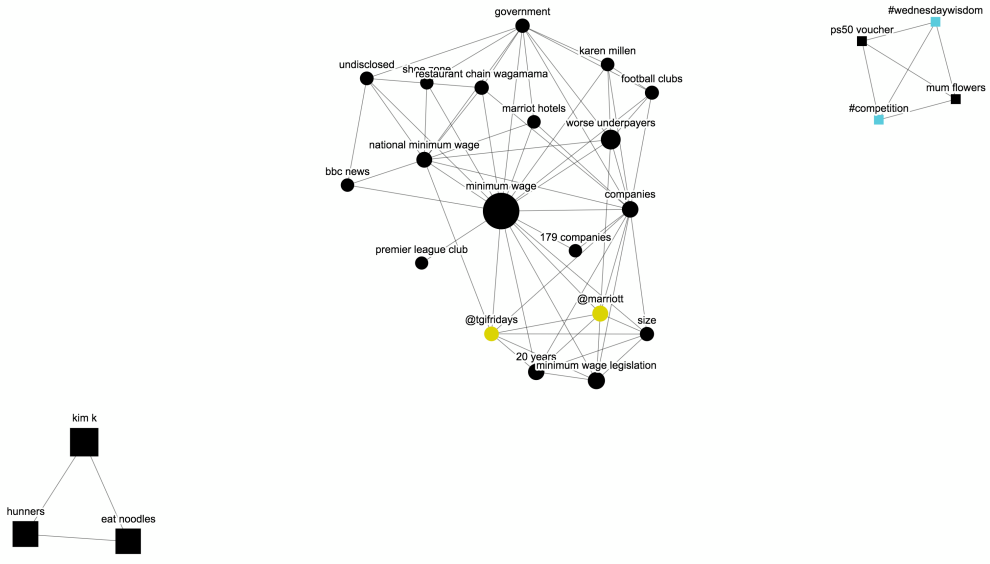

Figure H.4: Wagamama – Detail View

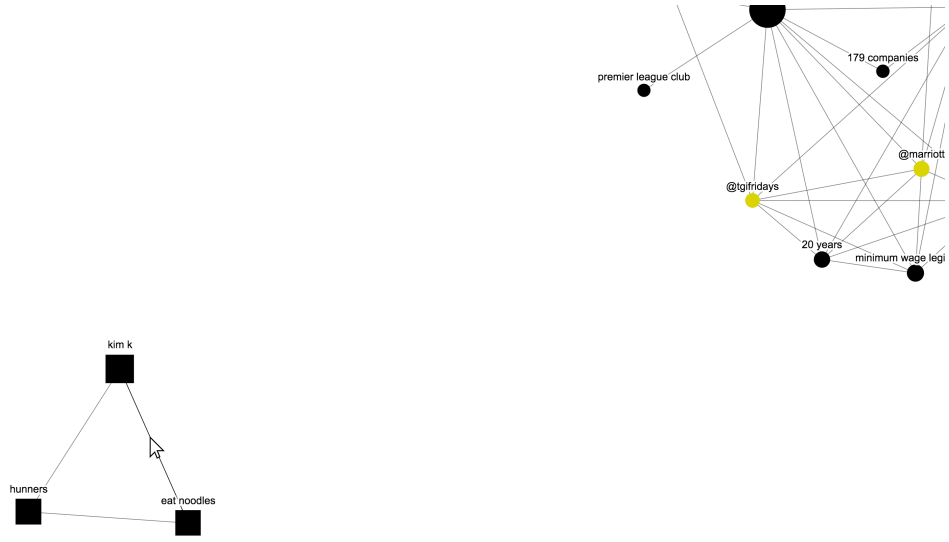

Figure H.5: Wagamama – Detail View of a Single Cluster

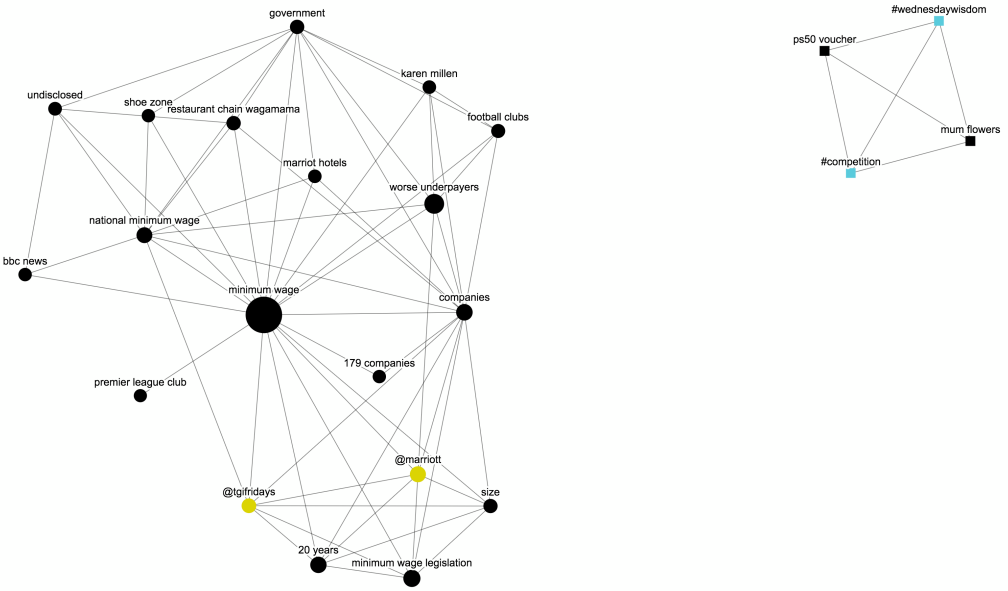

Figure H.6: Wagamama – Detail View of Remaining Clusters

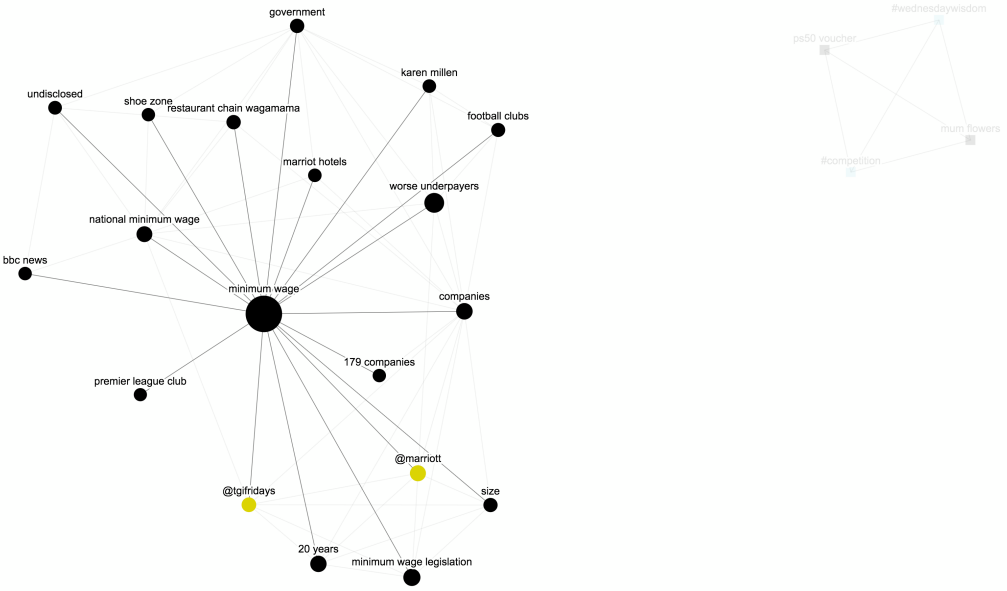

Figure H.7: Wagamama – Selected 1-Degree Ego Network

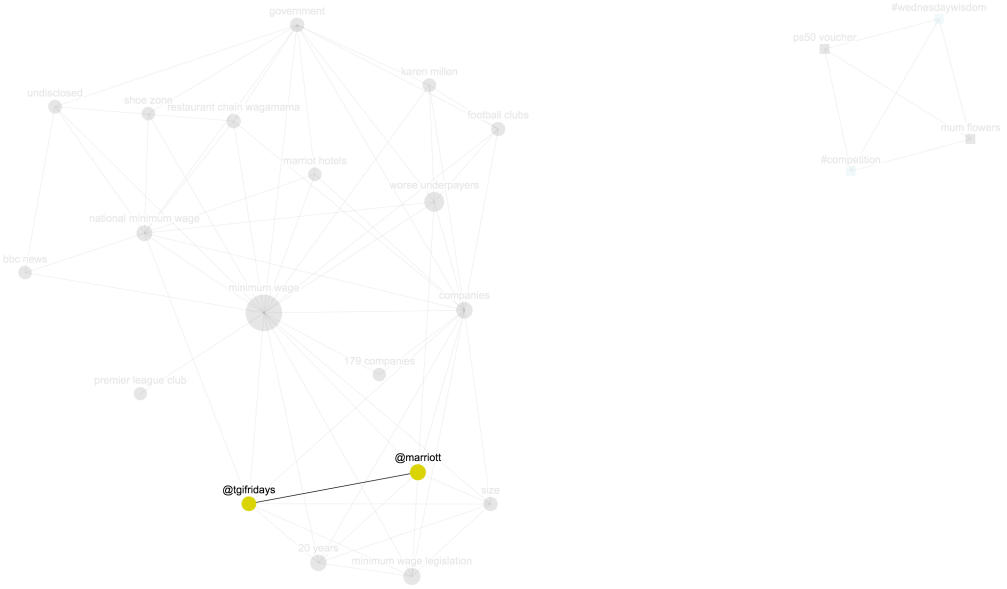

Figure H.8: Wagamama – Selected Edge

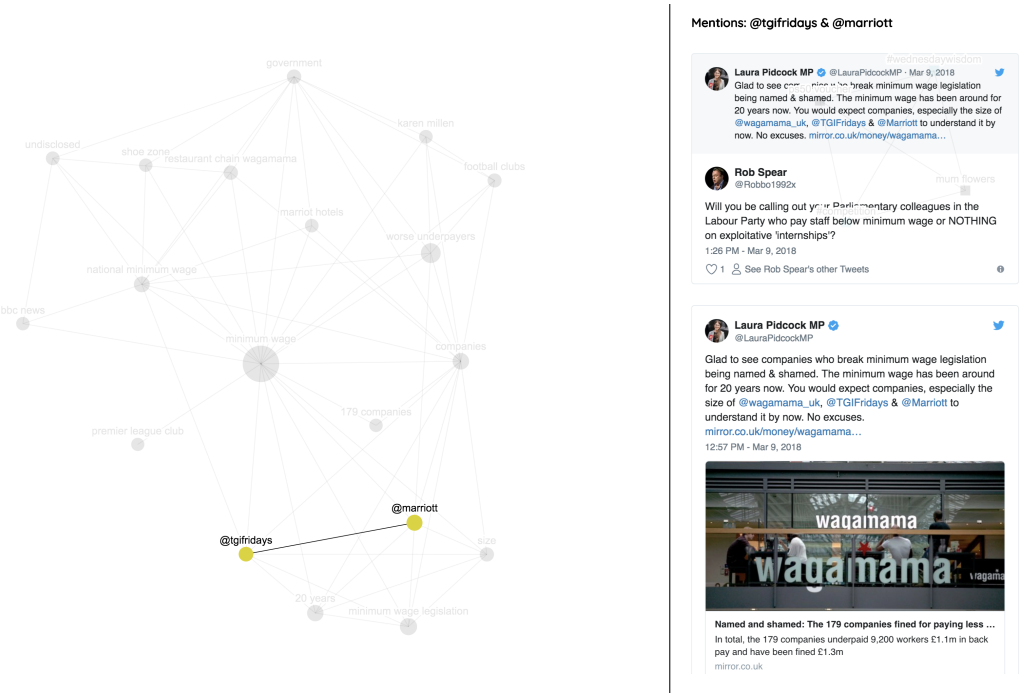

Figure H.9: Wagamama – Edge-related Mentions

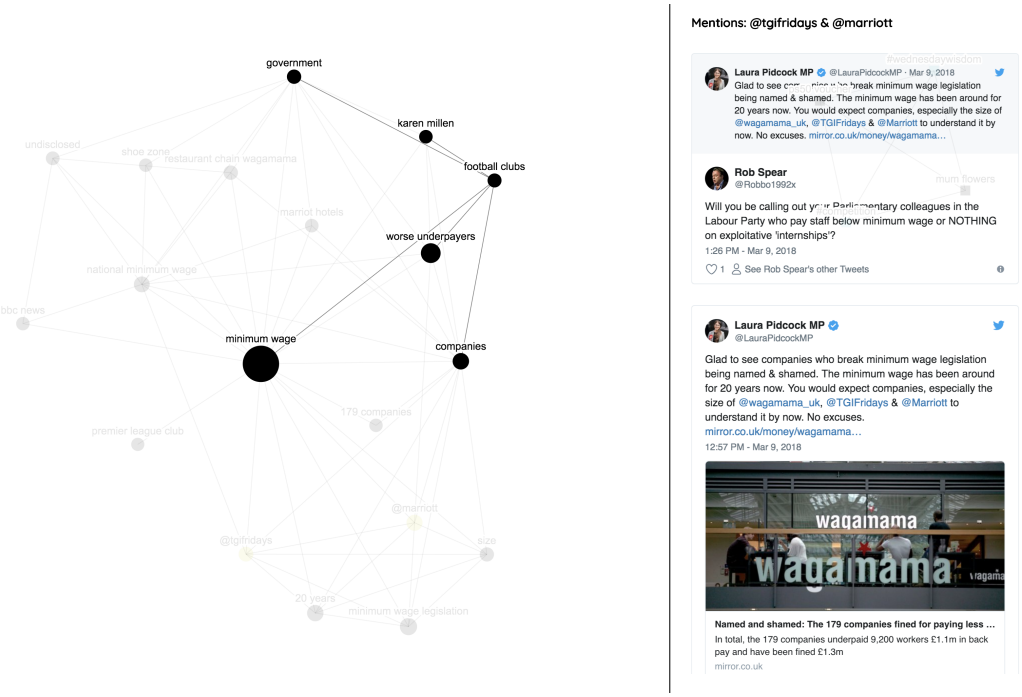

Figure H.10: Wagamama – Mentions with Another Selected Ego Network

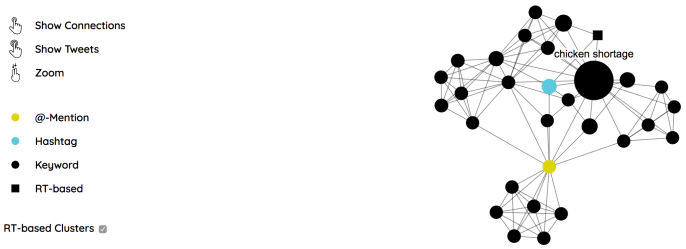

Figure H.11: KFC – Initial View

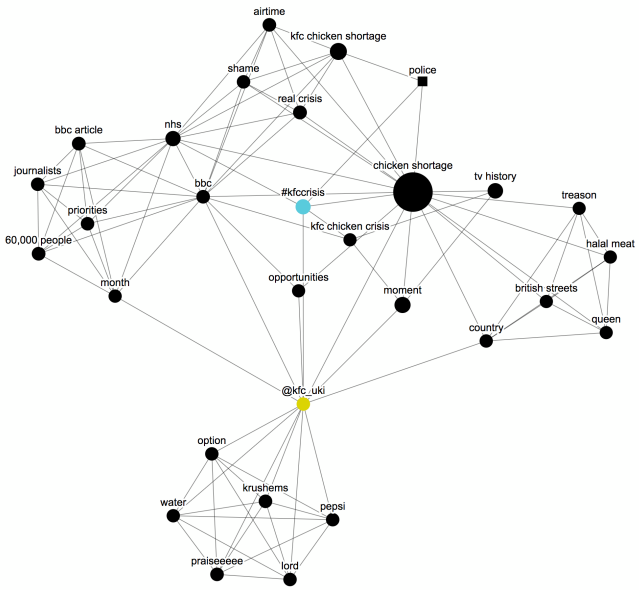

Figure H.12: KFC – Detail View

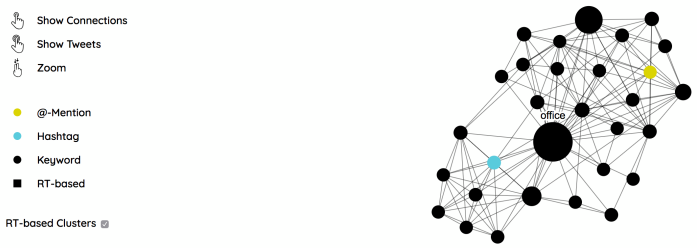

Figure H.13: Carson – Initial View

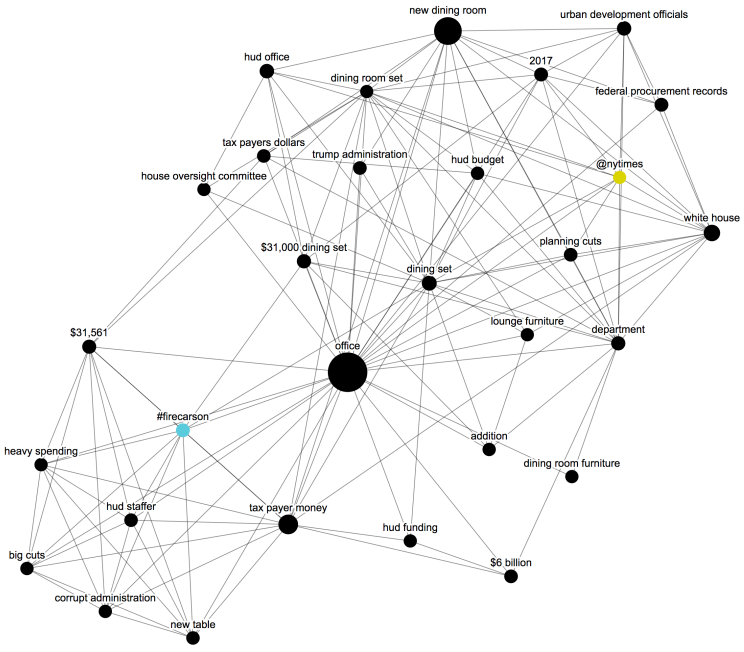

Figure H.14: Carson – Detail View

Bibliography

[1] Brandwatch. [Online]. Available: https://www.brandwatch.com/brandwatch-analytics/ (Accessed: 12 April 2018)

[2] Explosion AI. [Online]. Available: https://spacy.io/ (Accessed: 23 June 2018)

[3] Twitter. [Online]. Available: https://help.twitter.com/en/managing-your-account/twitter-username-rules (Accessed: 24 June 2018)

[4] Natural Language Tool Kit. [Online]. Available: https://www.nltk.org/api/nltk.tokenize.html (Accessed: 23 June 2018)

[5] D3 Data Driven Documents. [Online]. Available: https://d3js.org/ (Accessed: 05 July 2018)

[6] Twitter. [Online]. Available: https://developer.twitter.com/en/docs/twitter-for-websites/embedded-tweets/overview (Accessed: 01 August 2018)

[7] W. D. Abilhoa and L. N. de Castro, "A keyword extraction method from twitter messages represented as graphs," *Applied Mathematics and Computation*, vol. 240, pp. 308–325, 2014.

[8] J. Allan *et al.*, "Topic detection and tracking pilot study final report," in *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Arlington, VA, USA: Defense Advanced Research Projects Agency, February 1998, pp. 194–218.

[9] H. H. Alrehamy and C. Walker, "SemCluster: Unsupervised automatic keyphraseextraction using affinity propagation," in *Advances in Computational Intelligence Systems: Contributions Presented at the 17th UK Workshop on Computational Intelligence*, 1st ed., F. Chao *et al.*, Eds. Springer International Publishing AG, 2017.

[10] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[11] M. J. Bannister *et al.*, "Force-directed graph drawing using social gravity and scaling," *Computing Research Repository*, vol. abs/1209.0748, 2012.

[12] S. Beliga *et al.*, "An overview of graph-based keyword extraction methods and approaches," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 1–20, July 2015.

[13] Beniesta. (2018, February). *Tweet Number 966342169731637249*. [Online]. Available: https://twitter.com/Beniesta_/status/966342169731637249 (Accessed: 12 September 2018)

[14] K. Bennani-Smires *et al.*, "Embedrank: Unsupervised keyphrase extraction using sentence embeddings," *Computing Research Repository*, vol. abs/1801.04470, 2018.

[15] D. M. Blei *et al.*, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, Mar. 2003.

[16] V. D. Blondel *et al.*, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[17] P. Bojanowski *et al.*, "Enriching word vectors with subword information," *Computing Research Repository*, vol. abs/1607.04606, July 2016.

[18] S. P. Borgatti *et al.*, *Analyzing Social Networks*, 2nd ed. Thousand Oaks, CA, USA: SAGE Publications, Inc., 2018.

[19] M. Bostock. (2016, 18 September). blocks.org. [Online]. Available: https://bl.ocks.org/mbostock/1747543 (Accessed: 23 June 2018)

[20] A. Bougouin *et al.*, "Topicrank: Graph-based topic ranking for keyphrase extraction," in *Sixth International Joint Conference on Natural Language Processing*, ser. IJCNLP '13. Asian Federation of Natural Language Processing / ACL, 2013, pp. 543–551.

[21] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the 7th International Conference on World Wide Web*, ser. WWW '98. Amsterdam, Netherlands: Elsevier Science Publishers B. V., April 1998, pp. 110–117.

[22] P.-I. Chen and S.-J. Lin, "Automatic keyword prediction using google similarity distance," *Expert Systems with Applications: An International Journal*, vol. 37, no. 3, pp. 1928–1938, 2010.

[23] A. Clauset *et al.*, "Finding community structure in very large networks," *Physical Review E*, vol. 70, p. 066111, Dezember 2004.

[24] S. Danesh *et al.*, "SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction," in *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, ser. SEM '15, June 2015, pp. 117–126.

[25] T. Dunning and E. Friedmann, *Practical Machine Learning: A New Look at Anomaly Detection*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, June 2014.

[26] S. Dworkin. (2018, February). *Tweet Number 968665971832565760.* [Online]. Available: https://twitter.com/funder/status/968665971832565760 (Accessed: 12 September 2018)

[27] S. Emmons *et al.*, "Analysis of network clustering algorithms and cluster quality metrics at scale," *PLOS ONE*, vol. 11, no. 7, pp. 1–18, July 2016.

[28] G. Ercan and I. Cicekli, "Using lexical chains for keyword extraction," *Information Processing & Management*, vol. 43, no. 6, pp. 1705–1714, November 2007.

[29] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, Dec. 2004.

[30] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, 1st ed. New York, NY, USA: Cambridge University Press, 2007.

[31] Fineline Interiors. (2018, March). *Tweet Number 971347428950663168*. [Online]. Available: https://twitter.com/FinelineOldham/status/971347428950663168 (Accessed: 12 September 2018)

[32] C. Florescu and C. Caragea, "A position-biased pagerank algorithm for keyphrase extraction," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence and the 29th Innovative Applications of Artificial Intelligence Conference*, ser. IAAI '17. Menlo Park, CA, USA: Association for the Advancement of Artificial Intelligence, February 2017, pp. 4923–4924.

[33] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, February 2010.

[34] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, Jan. 2007.

[35] B. Friedman. (2018, February). *Tweet Number 968684607829880833*. [Online]. Available: https://twitter.com/BFriedmanDC/status/968684607829880833 (Accessed: 12 September 2018)

[36] C. E. Germán Aquino, Waldo Hasperué and L. Lanzarinin, "A novel, language-independent keyword extraction method," XVIII Congreso Argentino de Ciencias de la Computación, October 2013.

[37] M. Grineva *et al.*, "Extracting key terms from noisy and multitheme documents," in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 661–670.

[38] R. Gummadi. (2018, 30 January). *Instagram Graph API Launches and Instagram API Platform Deprecation.* facebook for developers. [Online]. Available: https://developers.facebook.com/blog/post/2018/01/30/instagram-graph-api-updates/ (Accessed: 25 April 2018)

[39] Y. HaCohen-Kerner, "Automatic extraction of keywords from abstracts," in *Knowledge-Based Intelligent Information and Engineering Systems*, V. Palade *et al.*, Eds. Berlin/Heidelberg, Germany: Springer International Publishing AG, 2003, pp. 843–849.

[40] D. Y. Haribhakta, "Big data, text categorization and topic modelling," in *Big Data Analytics*, P. Kulkarni *et al.*, Eds. PHI Learning Private Limited, 2016, vol. 1, ch. 5, pp. 76–91.

[41] K. S. Hasan and V. Ng, "Automatic keyphrase extraction: A survey of the state of the art," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. 1x. Baltimore, MD, USA: Association for Computational Linguistics, June 2014, pp. 1262–1273.

[42] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: Association for Computing Machinery, 1999, pp. 50–57.

[43] A. Hollocou *et al.*, "Improving pagerank for local community detection," *Computing Research Repository*, vol. abs/1610.08722, October 2016.

[44] L. Hong and B. D. Davison, "Empirical study of topic modeling in twitter," in *Proceedings of the First Workshop on Social Media Analytics*, ser. SOMA '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 80–88.

[45] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 216–223.

[46] J. Hurlock and M. L. Wilson, "Searching twitter: Separating the tweet from the chaff," in *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*. Menlo Park, CA, USA: Association for the Advancement of Artificial Intelligence, July 2011.

[47] G. Ignatov and R. Mihalcea, *Text Mining: A Guidebook for the Social Sciences*, 1st ed. Thousand Oaks, CA, USA: SAGE Publications, Inc., 2017.

[48] P. W. III and F. Wang, "Size matters: A comparative analysis of community detection algorithms," *Computing Research Repository*, vol. abs/1712.01690, 2017.

[49] A. Java *et al.*, "Why we twitter: Understanding microblogging usage and communities," in *Procedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop 2007*. Cham, Switzerland: Springer International Publishing AG, August 2007, pp. 56–65.

[50] E. Jónsson and J. Stolee, "An evaluation of topic modelling techniques for twitter," April 2016.

[51] B. S. Khan and M. A. Niazi, "Network community detection: A review and visual survey," *Computing Research Repository*, vol. abs/1708.00977, August 2017.

[52] S. G. Kobourov, "Spring embedders and force directed graph drawing algorithms," *Computing Research Repository*, vol. abs/1201.3011, 2012.

[53] H. Kwak *et al.*, "What is twitter, a social network or a news media?" in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 591–600.

[54] A. Lancichinetti *et al.*, "Finding statistically significant communities in networks," *PLOS ONE*, vol. 6, no. 4, pp. 1–18, April 2011.

[55] J. Leskovec *et al.*, "Statistical properties of community structure in large social and information networks," in *Proceedings of the 17th International*

Conference on World Wide Web, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 695–704.

[56] ——, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 631–640.

[57] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, vol. Short Papers. Baltimore, MA, USA: Association for Computational Linguistics, June 2014, pp. 30–308.

[58] J. Li *et al.*, "Keyword extraction based on tf/idf for chinese news document," *Wuhan University Journal of Natural Sciences*, vol. 12, no. 5, pp. 917–921, September 2007.

[59] Z. Li *et al.*, "Keyword extraction for social snippets," in *Proceedings of the 19th International Conference on World Wide Web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 1143–1144.

[60] Y. Liu *et al.*, "Comparison of two schemes for automatic keyword extraction from medline for functional gene clustering," in *Proceedings of the IEEE Computational Systems Bioinformatics Conference*, ser. CSB '14. Washington, DC, USA: IEEE Computer Society Press, 2004, pp. 394–404.

[61] Z. Liu *et al.*, "Clustering to find exemplar terms for keyphrase extraction," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '09, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009, pp. 257–266.

[62] H. M. Lynn *et al.*, "Swiftrank: An unsupervised statistical approach of keyword and salient sentence extraction for individual documents," *Procedia Computer Science*, vol. 113, pp. 472–477, 2017, the 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017) / The 7th International Conference on Current

and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2017) / Affiliated Workshops.

[63] O. Medelyan and I. H. Witten, "Domain-independent automatic keyphrase indexing with small training sets," *Journal of the American Society for Information Science and Technology*, vol. 59, no. 7, pp. 1026–1040, May 2008.

[64] R. Mihalcea and D. Radev, *Graph-based Natural Language Processing and Information Retrieval*, 1st ed. New York, NY, USA: Cambridge University Press, 2011.

[65] R. Mihalcea and P. Tarau, "TextRank: Bringing order into texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '04. Stroudsburg, PA, USA: Association for Computational Linguistics, July 2004.

[66] T. Mikolov *et al.*, "Efficient estimation of word representations in vector space," *Computing Research Repository*, vol. abs/1301.3781, January 2013.

[67] J. Mothe *et al.*, "Community detection: Comparison of state of the art algorithms," in *Computer Science and Information Technologies*, ser. CSIT '17, September 2017, pp. 125–129.

[68] L. P. MP. (2018, March). *Tweet Number 972063867424661504*. [Online]. Available: https://twitter.com/LauraPidcockMP/status/972063867424661504 (Accessed: 12 September 2018)

[69] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices." *Physical Review E*, vol. 74 3 Pt 2, p. 036104, September 2006.

[70] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[71] A. Y. Ng *et al.*, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA,

US: MIT Press, 2001, pp. 849–856.

[72] NHS Million. (2018, February). *Tweet Number 967139937597513731.* [Online]. Available: https://twitter.com/NHSMillion/status/967139937597513731 (Accessed: 12 September 2018)

[73] Y. Ohsawa *et al.*, "Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor," in *Proceedings of the Advances in Digital Libraries Conference*, ser. ADL '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 12–18.

[74] T. Opsahl *et al.*, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Networks*, vol. 32, no. 3, pp. 245–251, 2010.

[75] M. Orcutt. (2013, 10 December). *The Many Tongues of Twitter.* MIT Technology Report. [Online]. Available: https://www.technologyreview.com/s/522376/the-many-tongues-of-twitter/ (Accessed: 03 May 2018)

[76] G. K. Orman *et al.*, "On accuracy of community structure discovery algorithms," *Computing Research Repository*, vol. abs/1112.4134, 2011.

[77] G. K. Palshikar, "Keyword extraction from a single document using centrality measures," in *Pattern Recognition and Machine Intelligence*, A. Ghosh *et al.*, Eds. Berlin/Heidelberg, Germany: Springer International Publishing AG, 2007, pp. 503–510.

[78] C. Pasquier, "Single document keyphrase extraction using sentence clustering and latent dirichlet allocation," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, ser. SemEval '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 154–157.

[79] J. Pennington *et al.*, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '14. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.

[80] S. Petrov *et al.*, "A universal part-of-speech tagset," *Computing Research Repository*, vol. abs/1104.2086, 2011.

[81] S. Petrović *et al.*, "Can twitter replace newswire for breaking news?" in *Proceedings of the 7th International AAAI Conference on Weblogs and Social Media*, ser. ICWSM '13. Menlo Park, CA, USA: Association for the Advancement of Artificial Intelligence, 2013.

[82] C. Politics. (2018, February). *Tweet Number 968675386761629696.* [Online]. Available: https://twitter.com/CNNPolitics/status/968675386761629696 (Accessed: 12 September 2018)

[83] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences - ISCIS 2005*, p. Yolum *et al.*, Eds. Berlin/Heidelberg, Germany: Springer International Publishing AG, 2005, pp. 284–293.

[84] N. Pudota *et al.*, "A new domain independent keyphrase extraction system," in *Digital Libraries*, M. Agosti *et al.*, Eds. Berlin/Heidelberg, Germany: Springer International Publishing AG, 2010, pp. 67–78.

[85] U. N. Raghavan *et al.*, "Near linear time algorithm to detect community structures in large-scale networks." *Physical Review E*, vol. 76 3 Pt 2, p. 036106, 2007.

[86] A. Ritter *et al.*, "Open domain event extraction from twitter," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12. New York, NY, USA: ACM, 2012, pp. 1104–1112.

[87] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[88] D. Sarkar, *Text Analytics with Python: A Practical Rea-World Approach to Gaining Actionable Insights from Your Data*, 1st ed. New York, NY, USA: Apress Media LLC, 2016.

[89] H. Sayyadi *et al.*, "Event detection and tracking in social streams," in *In Proceedings of the International Conference on Weblogs and Social Media*, ser. ICWSM '09. Menlo Park, CA, USA: Association for the Advancement of Artificial Intelligence, March 2009.

[90] J. Schneider, "Topic modeling based on keywords and context," *Computing Research Repository*, vol. abs/1710.02650, 2017.

[91] D. Scott *et al.*, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[92] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Conference on Computer Vision and Pattern Recognition,*, ser. CVPR '97, San Juan, Puerto Rico, June 1997, pp. 731–737.

[93] Shuangyong Song *et al.*, "Detecting keyphrases in micro-blogging with graph modeling of information diffusion," in *13th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, ser. PRICAI '14, D.-N. Pham and S.-B. Park, Eds. Cham, Switzerland: Springer International Publishing AG, December 2014, pp. 26–38.

[94] M. R. Siegel and L. J. Stephens, *Schaum's Outline of Theory and Problems of Statistics*, 4th ed. New York, NY, USA: McGraw Hill Companies, April 2011.

[95] S. S. Sonawane and P. A. Kulkarni, "Graph based representation and analysis of text document: A survey of techniques," *International Journal of Computer Applications*, vol. 96, no. 19, pp. 1–8, June 2014.

[96] V. K. R. Sridhar, "Unsupervised topic modeling for short texts using distributed representations of words," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, ser. VS@NAACL-HLT '15. Stroudsburg, PA, USA: Association for Computational Linguistics, 2015, pp. 192–200.

[97] A. O. Steinskog *et al.*, "Twitter topic modeling by tweet aggregation," in *Proceedings of the 21st Nordic Conference of Computational Linguistics*.

Linköping, Sweden: Linkoping University Electronic Press, May 2017, pp. 77–86.

[98] P.-N. Tan *et al.*, *Introduction to Data Mining*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

[99] The New York Times. (2018, February). *Tweet Number 966292688931782656*. [Online]. Available: https://twitter.com/nytimes/status/966292688931782656 (Accessed: 12 September 2018)

[100] ——. (2018, February). *Tweet Number 968840807531909120*. [Online]. Available: https://twitter.com/nytimes/status/968840807531909120 (Accessed: 12 September 2018)

[101] A. Trask *et al.*, "sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings," *Computing Research Repository*, vol. abs/1511.06388, 2015.

[102] G. Tsatsaronis *et al.*, "SemanticRank: Ranking keywords and sentences using semantic graphs," in *Proceedings of the 23rd International Conference on Computational Linguistics*, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1074–1082.

[103] Walsh. (2018, March). *Tweet Number 971158635714940930*. [Online]. Available: https://twitter.com/Waaalsh_/status/971158635714940930 (Accessed: 12 September 2018)

[104] M. Wang *et al.*, "Community detection in social networks: An in-depth benchmarking study with a procedure-oriented framework," in *Proceedings of the 41st Very Large Data Bases*, ser. VLDB '15, vol. 8, no. 10. VLDB Endowment, 2015, pp. 998–1009.

[105] R. Wang *et al.*, "Corpus-independent generic keyphrase extraction using word embedding vectors," in *Software Engineering Research Conference*, 2014, p. 39.

[106] X. Wang *et al.*, "LeadLine: Interactive visual analysis of text data through event identification and exploration," in *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, ser. VAST '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 93–102.

[107] C. Wu *et al.*, "Machine learning-based keywords extraction for scientific literature," *Journal of Universal Computer Science*, vol. 13, no. 10, pp. 1471–1483, October 2007.

[108] S. Wu. (2015, 10 July). *Understanding the Force*. Medium. [Online]. Available: https://medium.com/@sxywu/understanding-the-force-ef1237017d5 (Accessed: 28 July 2018)

[109] X. Yan *et al.*, "A biterm topic model for short texts," in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 1445–1456.

[110] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, Jan. 2015.

[111] Z. Yang *et al.*, "Keyword extraction by entropy difference between the intrinsic and extrinsic mode," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 19, pp. 4523–4531, 2013.

[112] W. Yin and H. Schütze, "Learning word meta-embeddings," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. Long Papers. Stroudsburg, PA, USA: Association for Computational Linguistics, August 2016.

[113] C. Zhang *et al.*, "Automatic keyword extraction from documents using conditional random fields," in *Journal of Computational Information Systems*, vol. 4, no. 3, 2008, pp. 1169–1180.

[114] D. Zhou *et al.*, "Unsupervised event exploration from social text streams," *Intelligent Data Analysis*, vol. 21, no. 4, pp. 849–866, August 2017.

[115] Y. Zuo *et al.*, "Word network topic model: a simple but general solution for short and imbalanced texts," *Knowledge and Information Systems*, vol. 48, no. 2, pp. 379–398, August 2016.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

This thesis is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text.

Felix Heck
Stuttgart, November 2018