UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

WeST People and Knowledge Networks
Institute for Web Science and Technologies

Probabilistic Topic Models for Multi-Article News Comment Summarization

Bachelorarbeit

zur Erlangung des Grades eines Bachelor of Science (B.Sc.)
im Studiengang Informatik

vorgelegt von
Nico Daheim

Erstgutachter:	Prof. Dr. Steffen Staab Institute for Web Science and Technologies
Zweitgutachter:	Dr. Chandan Kumar Institute for Web Science and Technologies

Koblenz, im März 2019

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbstständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde. Die eingereichte schriftliche Fassung entspricht der auf dem elektronischen Speichermedium (CD-Rom).

	Ja	Nein
Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	☐	☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☐	☐
Der Text dieser Arbeit ist unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	☐	☐
Der Quellcode ist unter einer GNU General Public License (GPLv3) verfügbar.	☐	☐
Die erhobenen Daten sind unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	☐	☐

.....

(Ort, Datum) (Unterschrift)

Anmerkung

- If you would like us to contact you for the graduation ceremony, please provide your personal E-mail address: .....
- If you would like us to send you an invite to join the WeST Alumni and Members group on LinkedIn, please provide your LinkedIn ID : .....

Zusammenfassung

Es liegt in der Natur des Menschen, Meinung zu aktuellen Themen zu äußern. Heutzutage geschieht dies oft im Internet. Newsanbieter, zum Beispiel, bieten unter ihren Artikeln häufig eine Kommentarfunktion, die Nutzer zur Meinungsäußerung verwenden können. Dadurch bieten solche Kommentarbereiche einen Einblick in die Meinung der Öffentlichkeit, welche sich als wertvoll für Unternehmen und auch politische Entscheidungsträger darstellt. Die Masse und Diversität der Daten behindert allerdings eine schnelle Analyse. Daher finden sich in Kommentarbereichen häufig Filtermethoden, wie zum Beispiel das up- und downvoten von Kommentaren. Da solche Filtermethoden jedoch häufig ein inkorrektes Bild der Meinung widerspiegeln, werden computergestützte Lösungen benötigt. Eine solche Lösung ist die automatisierte Textzusammenfassung, die seit den 1950ern als Möglichkeit der Erstellung eines prägnanten Überblicks erforscht wird und Anwendung für Nutzerkommentare gefunden hat. In dieser Arbeit wird die automatisierte Zusammenfassung von Kommentaren unter mehreren Newsartikeln erforscht. Um einen umfassenden Überblick zu bieten, präsentiert die gewählte Methode die wichtigsten Themen in den Kommentaren in der Form eines Labels, einer Visualisierung des "sentiments" und extrahierten Kommentaren. Hierbei liegt der Fokus auf dem Gruppieren der Kommentare anhand der diskutierten Themen. Dazu wird das topic model Hierarchical multi-Dirichlet Process (HMDP), welches auch den Kontext der Kommentare miteinbezieht, mit dem Markov Cluster Algorithm (MCL) und Latent Dirichlet Allocation (LDA) verglichen. Mit einer auf einem Gold Standard, für den ein Erstellungsschema vorgestellt wird, basierenden Evaluation konnten wir zeigen, dass das HMDP die besten Ergebnisse liefert. Dies bestätigt die These, dass Kontextmodellierung das topic modeling von Kommentaren verbessern kann. Zudem wird ein Labelingalgorithmus vorgestellt, der unsupervised arbeitet.

Abstract

Expression of opinion on current topics is a part of human nature and has happened on many platforms throughout history. In times of the internet, comment sections under textual ressources are a common platform. Many news outlets, for example, offer users the possibility to comment on their articles. Their comment sections provide an insight into public opinion, which is valuable for corporations, policy makers and analysts. However, due to the scale and diversity of comments under news articles, it is not trivial to assess public opinion quickly. Thus, filtering is already commonly found in comment sections, most notably through up- or downvoting, but often fails to represent opinion correctly. This sparks a need for computational methods of filtering. Automatic text summarization has been explored since the 1950s as a way of presenting a concise overview of textual ressources and found successful application for user-contributed comments, which saves users and analysts effort. Thus, this thesis investigates extractive text summarization in the context of news article comments of multiple articles. In order to present a comprehensive overview of opinion, we choose to outline the most significant topics discussed through a label, sentiment and extracted comments. The focus of the work lies on grouping comments by an unknown number of discussed topics. The context-aware topic model Hierarchical multi-Dirichlet Process (HMDP) is compared to the Markov Cluster Algorithm (MCL) and Latent Dirichlet Allocation (LDA). An evaluation based on a gold standard, for which a creation method is outlined, shows that the HMDP performs best. This supports the thesis that context inclusion is able to improve the topic modeling of sparse comments. Furthermore, an unsupervised topic labeling algorithm is outlined as part of summary generation.

Acknowledgements

First of all, I want to thank my supervisors Prof. Dr. Steffen Staab and Dr. Chandan Kumar. Both have helped me greatly along the way of this thesis, from shaping its topic to all the constructive criticism over the course of implementation and writing. I also want to thank Jun Sun for all his help and guidance. I have enjoyed being a part of the CUTLER project together with Jun and Dr. Chandan Kumar, where I found great interest in data analysis and Natural Language Processing. Furthermore, I want to thank Dr. Zeyd Boukhers for offering consultation on topic model evaluation. Lastly, I want to thank all of my friends and family, who have supported me throughout the course of my bachelors studies, most of all my grandparents Elli and Hilarius for all the love and support.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

The Web 2.0 revolution has given users of the internet many options to voice their opinion, for example social media platforms, product reviews or comments under blog posts and news articles. This textual data can provide insights for researchers, analysts and policy makers. Figure 1 shows two comments discussing different

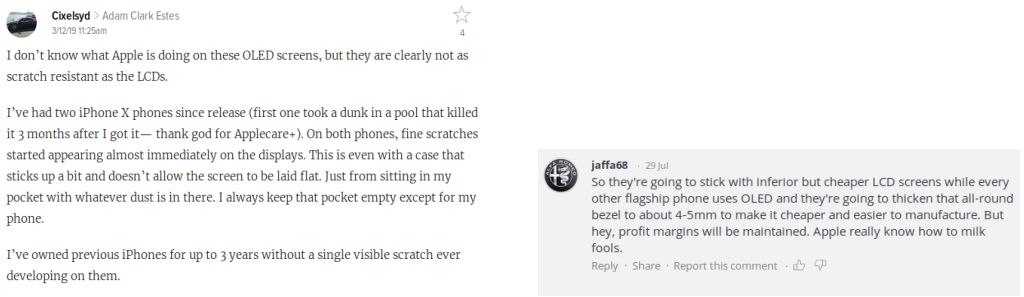


Figure 1: Two comments discussing a common topic under different articles.[3]

phone screen technologies under two different articles about iPhones® (a registered trademark of Apple). While one talks favorably about OLED screens, the other prefers LCD screens as they seem more scratch-resistant. This is valuable information and depicts public opinion about a topic. However, the right comments are not always readily available. The Huffington Post [4], for example, has reportedly received around 140,000 comments in a three-day period and news articles in the Guardian newspaper [5] have gathered 25,000 to 40,000 comments a day [1] with single news articles receiving more than a thousand comments [2, 3]. Finding the relevant comments in such a large amount of data, which can also be meaningless and redundant, is exhausting, as Edmunds and Morris have pointed in a literature review on information overload in businesses [4]. Nevertheless, filtering and summarization can provide relief [4]. Many news outlets already offer filtering for news article comments. Particularly interesting comments are highlighted after editorial selection, for example [5], which is nevertheless inherently biased on the views of the editor [5]. Intuitively, one might think that this can be alleviated when a sufficient number of, albeit non-professional, editors are involved through "Collaborative Recommendation" which is most frequently implemented in the form of up- and downvoting. This is contradicted by Chen et al. [6] whose user study showed that there appears to be almost no correlation between rating and quality. Moreover, such mechanisms inherently suffer from an rich get richer effect [2], which

[3] Sources: `https://gizmodo.com/why-i-regret-upgrading-to-an-iphone-xs-1832653430`, `https://www.express.co.uk/life-style/science-technology/991936/iPhone-X-2018-release-best-look-yet-at-Apple-s-flagship-new-render-iPhone-X-Plus`

[4] `https://www.huffingtonpost.de/`

[5] `https://www.theguardian.com/international`

might omit important comments. Thus, there clearly is a need for more effective filtering techniques.

Text summarization as a field in Natural Language Processing (NLP) aims to reduce textual resources to the necessary bits under the preservation of the overall information content. It can ensure that an analyst of public opinion only sees necessary information but is not withheld important one. This motivates the presented thesis to investigate the research problem of Text Summarization in the context of summarizing news article comments issued under multiple, topically related articles. Here, necessary bits of information are opinions on the topics discussed throughout the comments. Topic modeling and graph clustering are used to find these topics and to group comments by their targeted topic, wherein the focus of the presented work lies. This work faces two challenges. First, the broad conversational structure under multiple articles with an unknown number of topics overlapping across comments and second, the data sparsity in usually short comment. The Hierarchical multi-Dirichlet Process (HMDP) topic model is introduced to improve topic modeling by accounting for the context under which comments are created, for example the date of its creation or its conversational thread.

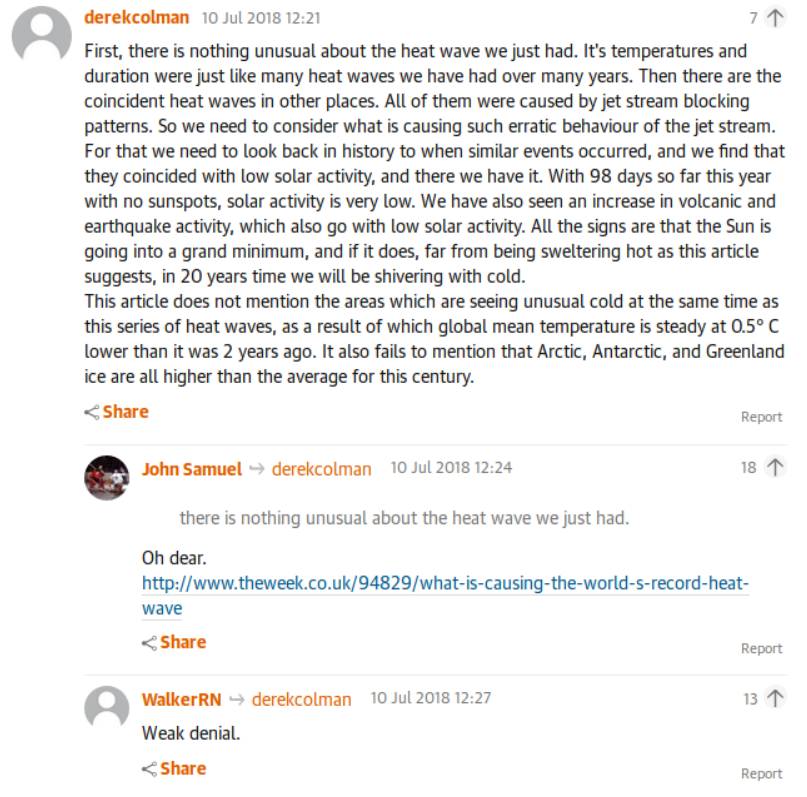


Figure 2: A comment thread snippet taken from an article about climate change of the Guardian newspaper [6]which illustrates the importance of context.

The HMDP model is especially relevant for short comments, which are typically difficult to assign topics [1]. To employ an example, consider the comment thread from an article about global warming in Figure 2. While the long comment can be clearly assigned a topic, the short ones cannot. However, with the contextual knowledge that both were issued under the article about global warming and as an answer to the comment above it can be assigned the topics of the comment it references.

The HMDP is compared to Latent Dirichlet Allocation (LDA) and the Markov Cluster Algorithm (MCL), which is graph-based and has found successful application in the clustering of comments of a single article. The comparison is based on a gold standard for which a creation process is outlined. Once comments are grouped by their topic, the most important negative and positive comments of the most important topic groups are extracted as a summary. Additionally, opinion is summarized in a visualization of sentiment. To establish importance, the ranking algorithm Maximal Marginal Relevance (MMR) is employed. Furthermore, topics are given a name through an introduced labeling algorithm.

To the best of the author's knowledge, this is the first work which explicitely aims to summarize comments under multiple news articles in a topic-driven manner. Related works either target single-article summarization [2, 3, 1, 7] or present only comment snippets [8].

1.1 Problem statement

Definition 1.1. Problem definition In this thesis, we focus on the extractive summarization of user comments under multiple, topically related news articles. The system shall be presented a set of articles $A = \{a_1, ..., a_n\}$ and each of their associated comments $C_i = \{c_1, ..., c_m\} \subseteq C$, where C denotes all considered comments. It shall then output a summary of the main topics $T\prime \subseteq T$ in the form of an extracted subset of comments $C\prime \subseteq C$. Here, T denotes the set of all topics discussed throughout the comments. The summary is required to not only present a reduced representation of the comments but also to preserve their overall meaning.

The focus of the presented work lies on the task of finding the topics discussed in comments and grouping comments by them. The main challenges which shall be adressed are mentioned in the following and considered in detail in later sections.

- An unknown number of topics.
- Data sparsity of comments.
- A broad conversational structure of comments under multiple related articles with overlapping topics, also across their comments, and a substantial temporal domain.

[6] https://www.theguardian.com/commentisfree/2018/jul/09/the-guardian-view-on-climate-change-a-global-heatwave

1.2 Thesis structure

The rest of this thesis is structured as follows. Section 2 provides an overview over related works in Text Summarization and a technical background on topic modeling, clustering and fundamental text mining techniques. Readers who are already familiar with said techniques might skip parts of section 2. Section 3 contains an overview of the chosen approach, a description of the data source and an analysis of characteristics of news article comments, on which the investigated topic modeling and clustering approaches are based. These are explained in section 4. This includes a description of how HMDP, LDA and MCL are used and how the topic labeling algorithm functions. Said methods are evaluated based on different criteria in section 5, followed by a discussion of the achieved results. The summary generation is outlined in section 6 with a description of ranking and selection and the chosen visualization. In the end, a conclusion and possibilities for future work are presented.

2 Background & Related Work

A background on text summarization, topic modeling and clustering is provided in this section. It is highlighted how topic modeling is a substantial part of extractive summarization. The problem of Text Summarization is formally introduced first and related works are outlined. Thereafter, topic modeling and clustering is introduced and a background on the models HMDP, LDA and MCL, which are used in this thesis, is provided.

2.1 Text Summarization

The research problem of Text Summarization can be defined as [9]:

Definition 2.1. Text Summarization Reducing a set of documents into a shorter version preserving overall information content and meaning [9].

In order to achieve this two approaches can be distinguished [10].

Definition 2.2. Abstractive Summarization A summary is formed by rephrasing the information of the documents [9, 11].

Definition 2.3. Extractive Summarization A summary is formed by the extraction of key segments of the documents [9, 11].

Upon comparison of the two approaches [10], a key challenge faced in abstractive summarization becomes apparent, which is aggravated by the inability of systems to "truly understand natural language" [9]. While an extractive summary reuses the information representation given by the documents by snippet selection, an abstractive summary rephrases the information. Thus, it also uses vocabulary potentially unseen in the object documents [12] which makes abstractive summarization arguably more complex [3]. Extractive summaries on the other hand often suffer from longer than average sentences being selected and a lack of coherence [9]. The latter, however, is not a concern for this work as entire comments are extracted.

This thesis makes use of techniques of extractive summarization. Hence, the research problem of abstractive summarization is not dealt with in more detail, for which the reader is referred to [13, 10, 14, 15, 11, 12, 16].

The field of extractive summarization has experienced plenty of research across different domains, especially since the 1990s [13, 10, 17]. In the following, it will first be investigated briefly in the context of single-document summarization, which has been the focus of research for a long time, and then in the context of multi-document summarization. The latter will then be considered within the domain of user-contributed content summarization and news comment summarization.

2.1.1 Single-document Summarization

Early works in document summarization have generally been concerned with single-document summarization. Cited as a first work [13, 18] in automatic text summarization in the late 1950s, Hans Peter Luhn [19] proposed a ranking scheme for sentence selection in order to automatically create literature abstracts. His work already incorporates a three-step approach common for extractive summarizers [20] and this thesis.

1. **Intermediate representation** - In order to identify important content more easily, documents are usually converted to an intermediate representation [20]. In [19], this representation is achieved by two preprocessing steps. First, words with a common stem such as "differ", "differentiate" or "different" are treated as identical, when computing their frequency. This is called stemming. Second, frequently appearing words such as "and" are removed. Such words are called stop words.
2. **Scoring** - Each text snippet is given a score indicating importance in relation to a topic in the document [20]. In [19], the presence of frequent words and their linear distance to infrequent words makes up the significance score of a sentence.
3. **Selection** - Finally, salient snippets are selected to form the summary [20]. In [19], sentences with the highest significance score are chosen.

Luhn's [19] techniques of achieving an intermediate representation are fundamental NLP techniques and find usage in this thesis.

Definition 2.4. Term Frequency Let t be a term in a document d. The weight of t, denoted as Term Frequency $\text{tf}_{t,d}$, is the number of its occurences in d [21].

A first use of Term Frequency can be found in [22]. It bases on the bag of words model [21], defined as:

Definition 2.5. Bag of words model A document d is represented by its terms t_i ignoring the exact order but preserving the number of occurences [21]. This can be represented as a matrix D with entries $\text{tf}_{t_i,d}$. [23].

Definition 2.6. Stop Words Common words which provide little value and are thus removed [21]. Let d be a document with vocabulary $V = \{w|w \in d\}$ and S the set of stop words. $V\prime = V \backslash S \subseteq V$ denotes the vocabulary of d after stop word removal. While sets of common vocabulary exist[7], stop words are often domain specific [21].

Definition 2.7. Stemming and Lemmatization Both reduce words "to a common base form" [21]. Stemming usually uses heuristics to remove the end of a word. Lemmatization can include context information by a dictionary, for example that "better" has the lemma "good" [21, 24].

[7] `https://www.nltk.org/book/ch02.html`

Furthermore, we want to quickly define what the term "ranking" describes in this context, since it constitutes an important concept for this thesis.

Definition 2.8. Ranking In the context of text summarization, ranking can be defined using a scoring function and a linear order relation. Let S be a set of text snippets. The scoring function

$$f : S \to \mathbb{R} \tag{1}$$

assigns a real number to any snippet in S. Based on f, snippets can be ranked following a linear order relation defined as

$$s_i \geq s_j \Leftrightarrow f(s_i) \geq f(s_j) \quad \forall s_i, s_j \in S. \tag{2}$$

Luhn's work [19] was extended by Edmundson [25] in 1969, who established that not only frequent words but also cue words and phrases in cue locations such as title or conclusion are significant. This combination of multiple features was influential to later research involving machine mearning [20], which constitutes the majority of recent research in single-document summarization [13]. Especially neural network approaches have been studied in the previous years, e.g. in the works of Svore et al. [26]. Often the aim is also not extractive but abstractive summarization [27, 12, 28, 16]. For more information on single-document summarization the reader is referred to [13, 9, 20].

2.1.2 Multi-document Summarization

This thesis is concerned with multi-document summarization. It has emerged in the mid 1990s [13] and seeks to summarize multiple documents in a single summary. Thus, the general definition of Text Summarization can be refined for Multi-Document Summarization.

Definition 2.9. Multi-document Summarization Reducing multiple documents into one shorter version preserving overall information content and meaning.

In extractive summarization, the set of source documents $D = \{d_1, d_2, ..., d_n\}$ is reduced to a summary $S = \{s_i\}$, where $s_i \subseteq d_1 \vee ... \vee s_i \subseteq d_n$ for all extracted snippets. Extracted snippets are often sentences but may also be entire comments, for example. The sources may have overlapping information but may also be contradictory [13, 29]. This brings forth new challenges, which do not occur in single-document summarization, as pointed out in [29].

- A higher degree of redundancy. [29]
- The presence of a temporal dimension. [29]
- A higher compression ratio. [29]
- An amplified co-reference problem [29], which describes whether two expressions target a common entity [30].

Cited as the pioneering work [13] in multi-document summarization, McKeown and Radev [31, 32] proposed an *abstractive* summarization tool called SUMMONS [8] which aims to summarize news articles on a certain event [32]. It is made up of two main components. First, a content planner fills handmade templates [13] with information from a knowledge base [31]. It then arranges and connects them based on a number of operators, sets of heuristics, e.g. a contradiction operator indicating differing information. Operators thereby influence the importance of templates, based on which they are selected [31]. In the end, a linguistic component arranges the information using words from a lexicon and grammar. However, also because of the need for handcrafted templates, SUMMONS is hard to generalize for other domains [13]. Furthermore, the claim made in [31] that extractive approaches would be unfitting for multi-document summarization has been disproven [13].

Radev himself et al. [33] have done so with MEAD which summarizes multiple news articles in as they note human-like quality [33]. It builds on top of a topic detection and tracking system called CIDR [34] which groups news articles into broad topic clusters similar to [2, 3, 5, 1]. As the notion of topic cluster will be recurring throughout the thesis, it is briefly defined in the following and refined to our use case in later sections.

Definition 2.10. Topic cluster Documents which target a common topic can be grouped together in clusters which each describe one topic and are called topic clusters. Such clusters can either be soft, meaning that a document can belong to multiple clusters, or hard, meaning that each document only belongs to a single cluster. The former is sometimes also referred to as fuzzy clustering [1]. Topic clustering denotes the act of establishing such clusters.

CIDR is centroid-based [13, 35]. Similar documents are grouped together based on similarity to the cluster centroid. New clusters can be opened up if a similarity threshold is not reached with any existing centroid [34]. For each topic cluster, the sentences closest to the centroid words are selected as the summary sentences [33].

Definition 2.11. Cluster centroid A vector which constitutes the mean of a cluster of vectors.

In CIDR, similarity is measured by cosine similarity [34].

Definition 2.12. Similarity Measure Based on [36], a similarity measure can be defined as a function $f : \Omega \times \Omega \to [0, 1]$, where Ω denotes a set of objects, and for which hold:

1. $f(A, B) = 1 \Leftrightarrow A = B$ [36]
2. $f(A, B) < 1 \Leftrightarrow A \neq B$ [36]
3. $f(A, B) = 0 \Leftrightarrow$ A and B share no commonalities. [36]

[8] SUMMarizing Online NewS articles

Furthermore, the following intuition should be noted:

4. $f(A, B) > f(A\prime, B\prime) \Leftrightarrow$ A and B share more commonalities than A$\prime$ and B$\prime$ [36].

In CIDR [34], Ω is the set of Term Frequency-Inverse Document Frequency (TF-IDF) vector representations defined as follows.

Definition 2.13. Term Frequency-Inverse Document Frequency Let t be a term and $d \in D$ be a document, where D denotes the set of all documents.

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \cdot \text{idf}_t \text{ [21]} \tag{3}$$

idf_t denotes the Inverse Document Frequency (IDF) of t in D and $\text{tf}_{t,d}$ the Term Frequency of t in d. The TF-IDF score is high for a term t occuring often in only few documents and low for a term contained in (nearly) all documents [21].

Definition 2.14. Inverse Document Frequency Let D be the set of documents with $|D| = N$. The IDF of a term t is defined as:

$$\text{idf}_t = log(\frac{N}{\text{df}_t}) \tag{4}$$

df_t denotes the Document Frequency of t.

Definition 2.15. Document Frequency Let D be a set of documents and t be a term. Then

$$\text{df}_t = |\{d \in D | t \in d\}| \tag{5}$$

, i.e. "the number of documents in the collection that contain a term t" [21]

When a word is represented as a TF-IDF vector v, every entry v_i denotes the TF-IDF weight of a term in the vocabulary of the document collection. The underlying concept first presented by Salton et al. [37] in 1975 is accordingly defined as:

Definition 2.16. Vector Space Model Each document $d \in D$ is represented by an alternative representation $v = (v_1, ..., v_n)$, where each vector element v_i represents the weight of a term contained in the document collection D [37].

Such vector representations are common as they allow linear algebra operations, such as cosine similarity calculation.

Definition 2.17. Cosine Similarity A function defined for non-zero vectors $v, w \in \Omega$, where Ω denotes an inner product space defined as follows for two column vectors.

$$cosim : \Omega \times \Omega \rightarrow [-1, 1] \tag{6}$$

$$cosim(v, w) = \frac{v^T w}{||v||||w||} \tag{7}$$

$||\cdot||$ denotes the Euclidian norm $||v|| = \sqrt{\sum_1^M v_i^2}$ for a vector v with M entries. In NLP, v and w are usually positive. In this case, $cosim$ is a similarity measure as defined in definition 2.12 and has the signature:

$$cosim(v, w) : \Omega \times \Omega \rightarrow [0, 1] \tag{8}$$

Recent works in the domain usually use Latent Dirichlet Allocation (LDA) [2, 3, 5, 38], Markov model-based algorithms [39] or k-means clustering [35, 3] to group documents by topics. Many [40, 41, 42, 43] also target the applications of neural networks for extractive multi-document summarization.

2.1.3 Summarization of user-contributed comments

A popular way of user-contribution are comments under textual and non-textual resources, such as news articles, product descriptions or videos [2]. As we will see later, such comments differ from other textual resources due to their sparsity, noise and fluctuating length. Many works in the field of user-contributed content summarization have focused on summarizing opinion and sentiment [2]. Notably, Wang et al. [44] have proposed a solution to *Latent Aspect Rating Analysis*, which is described as finding the latent opinion of commenters on different aspects of the reviewed objective [44]. The proposed regression model bases on an Aspect Segmentation Algorithm, which "assign[s] each sentence to the aspect that shares the maximum term overlapping with this sentence" [44]. However, in [44] aspects are not seen as latent but assumed to be known beforehand [44].

Closer related to this thesis, Khabiri et al.'s [5] take on summarizing YouTube comments models the previously presented three-step approach [20]. Similarity-based k-means clustering and topic clustering based on LDA are compared. To tackle data sparsity, the number of topics of a comment is restricted to one. For ranking, term importance-based and precedence-based schemes are compared. In the latter, a random walk [9] is performed on a graph established based on comment similarity [5]. After ranking the top k comments are selected. An evaluation of the different combinations of clustering and ranking schemes is conducted by letting 5 subjects rate the first 50 comments of 30 videos by spur of interest and informativeness [5]. Based on this, Normalized Discounted Cumulative Gain (NDCG) is calculated. It is concluded that LDA coupled with a precedence-based ranking yields the best results [5].

The approach chosen in [5] shares many commonalities with existing news comment summarization efforts [2, 3] which are presented in the following.

[9] see definition 2.30

2.1.3.1 Summarization of news article comments

The procedure of topic clustering and ranking comments within each cluster before selection has been adopted throughout works in news comment summarization [2, 3, 1, 7]. Ma et al. [2] formalize this as *Topic-driven Reader Comments Summarization* (TORCS). Furthermore, they formalize the relationships between news articles and comments as follows.

- **news-news** - Two news articles report on the same topic. [2]
- **intra-comment-comment** - Comments propagate ideas of comments under the same news article [2].
- **intra-news-comment** - Comments propagate ideas of the news article they are issued under [2].
- **inter-comment-comment** [2] - Comments propagate ideas from comments under a different article [2].
- **inter-news-comment** - Comments are influenced by other news articles [2].

Based on these relationships, two LDA-based topic models are defined. Both treat an article as a master document and each of its comments as slave documents. They are compared to a bisecting clustering algorithm called CLUTO [10]. In the Master-Slave Topic Model (MSTM), topics of comments are derived from the topics of the associated article and drawn based on its topic-word distribution which is inferred using LDA [2]. The Extended Master-Slave Topic Model (EXTM) allows topics of comments to stem from extended topics not derived from the associated article [2]. A switch drawn from a binomial distribution determines the origin of a topic. Both models are compared using perplexity. It is concluded that EXTM significantly outperforms MSTM [2]. Furthermore, both topic models outperform CLUTO in terms of topic cohesion and topic diversity. For ranking comments, Maximal Marginal Relevance (MMR) [45] and a combination of Rating and Length are compared. MMR as used in this thesis is defined in subsection 6.1. Both methods are compared in combination with the different clustering approaches in a user study. 15 comments extracted as a summary for each of 50 randomly selected articles are rated by subjects by topic cohesion, topic diversity and news relatedness. MMR is reported to have yielded the best results, especially in combination with EXTM [2].

Ma et al.'s [2] and Khabiri et al.'s [5] works have served as the basis of Llewellyn et al.'s [3] research, who investigated different methods of clustering and ranking. A baseline unigram approach, where comments were assigned to one of the 14 most frequent terms, is compared to cosine distance [11] clustering, k-means clustering and LDA. In an evaluation based on a gold standard of comments of one article grouped by annotators, only LDA beats the baseline by micro-averaged F-Score [3]. For ranking, the methods used in Khabiri et al. [5] and Ma et al. [2] are applied.

[10] `http://glaros.dtc.umn.edu/gkhome/views/cluto`
[11] Defined as 1 - [Cosine Similarity]

For each method, human-produced summaries of comments under articles of the Guardian newspaper are compared against three comments selected from each associated topic cluster. PageRank offers the best results followed by MMR [3]. It is also proposed that including sentiment [3] might improve a summary.

A slightly different approach was chosen by researchers of the University of Sheffield [46, 1, 39, 47, 48, 7]. The method used in their series of papers uses graph-based clustering and is inspired by works in argument mining [46]. The notions of issue, viewpoint and assertion are distinguished in comments and build a first approach [46]. Assertions are described as propositions of commenters which "[express] a viewpoint" about an issue [46]. A graph representation is derived containing each notion as nodes and relationships between them as edges. Lastly, algorithms to produce summaries of one issue and a complete graph are presented in theory. The latter is executed manually and compared to a gold standard summary. In a following paper [1], this is translated to a version of the Markov Cluster Algorithm (MCL) [49] which is able to automatically determine the number of topic clusters. Comments are graph nodes between which edges are established if a threshold similarity is exceeded. Similarity is measured by a linear regression model of multiple similarity measures. Positive training samples are made up of comments which quote the same paragraph of an article of the Guardian newspaper [1]. Once the hard topic clusters are identified, they are labeled with a graph-based unsupervised labeling approach based on Hulpus et al. [50]. An LDA model is used to discover five topics for each cluster. Concepts for the ten most likely words of each cluster form a graph using DBPedia [12] which is merged. The centrality of each graph forms the cluster label. Finally, an evaluation involving human gold standards shows that the graph-based approach outperforms LDA approaches and that the created labels are often meaningful [1]. In a follow-up [39], it is constituted that a summary shall outline opinion on main issues. Based on this, a three-step, task-based evaluation process for summarization is established including a questionnaire and group discussion and investigated in a pilot evaluation of a prototype system [39]. Based on this, the prototype system is refined resulting in *The SENSEI Overview of Newspaper Readers' Comments* [7]. Users are presented a pie chart containing the labeled topics [48], a selected extract for each topic and the possibility to click a pie chart wedge representing a topic to find more comments on it [7, 48]. The usage of a chart to summarize the discussed topics is supported by a study of preferences in summary design [51]. A combination of chart and side-by-side summary indicating (dis-)agreement is concluded to be favorable [51]. The SENSEI system [7], however, does not include sentiment but presents a highest ranked sentence per topic. A ranking of sentences is established based on closeness to the centroid of the topic cluster [48]. Additionally, a supervised model for cluster labeling [48] is presented which outperforms a TF-IDF-based baseline with statistical significance [48]. Again, a linear regression model is used to extract labels. Gold standard is produced from human-made summaries of news article comments. The gold standard used throughout the works

[12] `https://wiki.dbpedia.org/`

is publicly available in the SENSEI corpus [13] comprised of 18 articles, associated comments and annotations [47] .

All of the afore-mentioned works target the summarization of comments under a single article. Only [8] explicitely targets the summarization of comments under multiple articles. However, it does not target a topic-driven summarization and focusses on being lightweight. News articles and comments are retrieved for a query using heuristics. Then, comment snippets are presented after being ranked by K-L divergence [14].

2.1.4 Evaluation of Automatic Summaries

Evaluation means of summarization efforts can be categorized as being either **extrinsic** or **intrinsic** [52]. Intrinsic evaluation techniques often make use of reference summaries used as a gold standard [52]. However, creating such a reference summary is tedious and expertise-needy [53]. Even then, agreement between annotators is not guaranteed [13] and an ideal solution to the summarization problem does not exist [13]. Furthermore, while corpusses containing human reference summaries exist, for example for the DUC [15] summarization tasks, they are commonly domain specific and, thus, not generally applicable. Nevertheless, when reference summaries can be used there exist different evaluation metrics such as the ROGUE [54] package which has become de-facto standard [13]. It consists of multiple measures which calculate overlap between reference and evaluated summary. The ROGUE-N co-occurence statistics, for example, measure the ratio of overlapping N-grams and the ROGUE-L score measures the longest common subsequence [54].

Due to the tediousness of intrinsic evaluation of an entire summarization system and the restricted availability of gold standards, it is common to evaluate single components of such a system on their own. In related works this is often carried out by evaluating topic clustering [2, 3, 5, 1] or ranking steps [3, 5]. Furthermore, user studies are frequently used in related works [2, 3, 5]. While such commonly involves subjects scoring the performance of different steps of the summarization effort [2, 3, 5, 1], there also exist approaches to modeling evaluation as a game for both entire summaries [52] and used technologies. In [23, 55], for example, a game where subjects identify intruder words is used to measure topic model performance. Furthermore, user studies can be conducted in a task-based fashion [33], such as outlined in [39] where a task-based evaluation was conducted on a prototype system. User involvement in general is important in extrinsic evaluation which "measures the efficiency and acceptability of the generated summaries in some task" [52].

[13] `http://sensei.group.shef.ac.uk/sensei/corpus.html`

[14] for more information refer to `https://nlp.stanford.edu/IR-book/html/htmledition/extended-language-modeling-approaches-1.html` [21]

[15] `https://www-nlpir.nist.gov/projects/duc/index.html`

2.2 Topic Modeling & Clustering

This section introduces topic modeling & clustering which have been key elements of many related summarization systems [33, 2, 3, 5, 38, 39, 35, 3]. For summarizing news article comments, LDA [2, 3] and MCL [39] have shown good performance and are, hence, investigated and outlined in the following. Additionally, the Hierarchical Dirichlet Process (HDP) [56] model and a context-aware extension thereof, the Hierarchical multi-Dirichlet Process (HMDP) [23], is introducted.
The goal of topic modeling can be described as finding the latent topics discussed in a document or a set of documents [23]. A topic model does not require background knowledge about how topics are distributed in the documents and expressed with words [23]. This enables a number of use cases, as pointed out by Kling [23]. Representing documents by their topic can increase computational efficiency. A query for documents can be limited to documents covering a broad topic of the query string. Similarly, a machine learning algorithm can be applied to only the topic words of a document instead of the entire document [23]. This is a form of dimension reduction [57], where a high dimensional text consisting of many tokens is represented by a representation with fewer tokens and lower dimension, which maintains semantic properties [57]. Topic models are not restricted to documents and have also been used to find genetic patterns, for example [58]. Topic models are initially not concerned with grouping which is, nevertheless, enabled by the document-topic distribution.

Clustering algorithms such as the MCL, on the other hand, intent to group datapoints into groups containing similar data [59]. Here, the notion of similarity is broad and not immediately concerned with the notion of *topic*. Use cases of clustering are thus very general and not restricted to textual data, as well. Just like topic models, they have been successfully applied in bioinformatics to detect protein families [60].
All presented topic models and clustering approaches base on the bag-of-words assumption [23]. Since they base on advances in probability theory, a common background is established in the following which is largely based on [61] and [23]. Readers familiar with these concepts might skip the following subsection.

2.2.1 Preliminaries in Probability Theory

Definition 2.18. Random Variable a measurable function X with the signature

$$X : \Omega \to \mathbb{R} \tag{9}$$

, where Ω denotes the set of possible outcomes called sample space. A random variable assigns a value to each of the outcomes in the sample space. Discrete random variables can take on only countably many values, whereas continuous random variables can take on an uncountably infinite number of values.

To employ an example, let random variable X_c describe a coin toss. X_c is clearly discrete as the set of possible outcomes contains heads and tails. For the sake of illustration, let us define the assigned value as 10$ won if it matches a guess or lost otherwise in the following. For a random variable X, a probability distribution exists, defining the probability for each measurable set of outcomes $x \in \mathcal{A}$ to be observed.

Definition 2.19. Probability distribution a function with the signature

$$P : \mathcal{A} \to [0, 1] \tag{10}$$

, where $\mathcal{A}$ denotes a σ-Algebra on Ω and for which hold

$$P(A) \geq 0 \quad \forall A \subset \Omega \tag{11}$$

$$P(\Omega) = 1 \tag{12}$$

$$P(\cup_{j=1}^{\infty} A_j) = \sum_{j=1}^{\infty} P(A_j) \quad \forall \text{ disjunct } A_1, A_2, ... \subset \mathcal{A} \tag{13}$$

P is also called probability measure on $\mathcal{A}$ and the triple $(\Omega, \mathcal{A}, P)$ is called probability space.

Discrete random variables have discrete distributions commonly defined as Probability Mass Function (PMF) [23].

Definition 2.20. Probability Mass Function For a discrete random variable $X : \Omega \to \{x_1, x_2, ..., x_n\}$, the PMF is defined as:

$$f : \{x_1, x_2, ..., x_n\} \to [0, 1] \tag{14}$$

$$f(x_i) = \begin{cases} P(X = x_i) & x_i \in \{x_1, x_2, ..., x_n\} \\ 0 & \text{else} \end{cases} \tag{15}$$

For our random variable X_c, the PMF describes the probabilities of winning or losing 10$. In a fair toss, both would be equal to 0.5. Since a continuous random variable can take on an uncountably infinite number of values, the probability of a one-point set occuring equals zero in continuous distributions. Continuous distributions are commonly defined by Probability Density Functions (PDF) [23].

Definition 2.21. Probability Density Function a function with the signature

$$f : \Omega \to \mathbb{R} \tag{16}$$

for which hold

$$f(x) \geq 0 \; \forall \; x \in \Omega \text{ , else } 0 \tag{17}$$

$$\int_{\Omega} f(x)dx = 1 \tag{18}$$

Based on the PDF, the distribution function of a continuous random variable X can be defined as:

Definition 2.22. Continuous distribution

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(x)dx \tag{19}$$

Random variables can be combined to form a joint distribution. This models the probability that each of the random variables takes on a certain set of outcomes. A dice roll described by the random variable Y_d added to the coin toss can be modeled with a joint distribution, for example.

Definition 2.23. Joint distribution For two random variables X, Y the joint distribution is defined to be in the discrete case:

$$P(X = x_i, Y = y_i) := P(\{X = x_i\} \cap \{Y = y_j\}) \tag{20}$$

and to be

$$f_{X,Y} = f_{Y|X}(y|x)f_X(x) = f_{X|Y}(x|y)f_Y(y) \tag{21}$$

in the continuous case.

In the latter, $f_{Y|X}(y|x) = f_{X|Y}(x|y)$ denotes the conditional probability distribution of Y given $X = x$ and X given $Y = y$. This concept is also known as the *posterior distribution* and important in probabilistic topic modeling [58], as it shows a distribution when a certain set of outcomes is already known. Suppose the dice is rolled after the coin toss and multiplies win or loss by the number of eyes present. The conditional distribution of Y_d given X_c then describes the probability of winning or losing 10, 20,..., 60$ depending on the known coin toss.

Definition 2.24. Conditional distribution Let X, Y be two discrete random variables. The conditional distribution of Y given X is given by:

$$f_{Y|X}(Y = y|X = x) = \frac{P(X = x \cap Y = y)}{P(X = x)} \tag{22}$$

$f_X(x)$ and $f_Y(y)$ denote the marginal distribution of X and Y. This can be used to factor out variables as is often desirable in probabilistic models with many variables [23]. While for the conditional distribution of Y under X the value of X is known, in the marginal distribution it is unknown.

Definition 2.25. Marginal distribution Let X and Y be two random variables. The marginal distribution of X in the discrete case is given by:

$$f_X(x) = \sum_y P(X = x, Y = y) \tag{23}$$

and in the continuous case by:

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)dy \tag{24}$$

For probabilistic topic models, two distributions are of special interest; the multinomial distribution and the Dirichlet distribution. As the multinomial distribution is a multivariate version of the binomial distribution the latter is introduced first.

Definition 2.26. Binomial distribution Let n denote the number of trials, p the probability of a successful trial and X the random variable, which denotes the number of successes. Then $X \sim Bin(n, p)$ with PMF

$$P(X = k) = \binom{n}{k} p^k (1-p)^{(n-k)}. \tag{25}$$

Tossing our coin 10 times, for example, is described by a discrete random variable $X_{d'} \sim Bin(10, 0.5)$ if the toss is fair.

Definition 2.27. Multinomial distribution Let K be the number of categories with probabilities $p_1, p_2, ..., p_K$ and $n_1, n_2, ..., n_K$ their counts.

$$P(n_1, n_2, ..., n_K) = \frac{(\sum_{i=1}^{K} n_i)!}{n_1! \cdots n_K!} \cdot p_1^{n_1} \cdots p_K^{n_K} \tag{26}$$

The Dirichlet distribution is related to the multinomial distribution as it can be used as a prior distribution for its parameters [23]. A prior distribution expresses a *prior belief* about how data is distributed. Moreover, the Dirichlet distribution is a *conjugate prior* of the multinomial [62], meaning that if the prior distribution is Dirichlet-distributed, then so is the *posterior distribution*.

Definition 2.28. Dirichlet distribution

$$P(\theta_1, ..., \theta_K | \alpha_1, ..., \alpha_K) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_K)}{\Gamma(\sum_{k=1}^{K} \alpha_K)} \cdot \prod_{k=1}^{K} \theta^{\alpha_k - 1} \tag{27}$$

It holds $\theta_K \in [0, 1]^K \ \forall k = 1, ..., K$ and $\sum_{k=1}^{K} \theta_k = 1$ [23]. Furthermore, smaller α_i induce sparsity while larger α_i induce a smoothed distribution. If $\alpha_i = \alpha_j \ \forall i, j \in \{1, ..., K\}$, the prior is symmetric and favors no θ_i over the other [23]. The PDF can then be rewritten as

$$P(\theta_1, ..., \theta_K | \alpha) = \frac{\Gamma(\alpha)^K}{\Gamma(K \cdot \alpha)} \cdot \prod_{k=1}^{K} \theta_k^{\alpha - 1} \tag{28}$$

If α, referred to as the concentration parameter [23], equals 1, the Dirichlet distribution equals the uniform distribution.

In the definition of the Dirichlet distribution, Γ denotes the gamma function defined as follows.

Definition 2.29. Gamma function Let $z > 0$. Then

$$\Gamma(z) = \int_0^\infty x^{z-1} \cdot e^{-x} \tag{29}$$

defines the gamma function. For $n \in \mathbb{N}$, the gamma function equals the factorial of n, i.e. $\Gamma(n) = n!$.

Coming back to distributions in general, joint distributions can also be defined by stochastic processes [58] as is the case in LDA, HDP or HMDP.

Definition 2.30. Stochastic process A family of random variables

$$X = (X_0, X_1, X_2, ...) \tag{30}$$

on a common finite set S, also called the state space. Stochastic processes commonly model events on a timeline. Our example of rolling a dice after a coin toss is a simple stochastic process, for example.

For the MCL algorithm, a specific type of stochastic process is of importance; the Markov chain.

Definition 2.31. Markov chain A stochastic process X is called Markov chain, if for all $n \geq 1$ and for all $s, x_0, x_1, ..., x_{n-1} \in S$ holds:

$$P(X_n = s | X_0 = x_0, X_1 = x_1, ..., X_{n-1} = x_{n-1}) = P(X_n = s | X_{n-1} = x_{n-1}) \tag{31}$$

That is, the current state is only dependent on the state immediately before it.

A Markov Chain can be represented by a weighted graph $G = (V, E)$ with a set of *vertices* or *nodes* V and a set of *edges* or *arcs* E. Each vertex represents a state, hence $V = S$, and each edge holds the (one-step) transition probability $P(X_{n+1} = j | X_n = i)$ for the vertices i and j it connects. This notion can also be described by a matrix known as the $|V| \times |V|$ *transition matrix* $\mathbb{P}$.

Definition 2.32. Transition matrix The transition matrix $\mathbb{P}$ of a Markov chain holds the transition probabilites $p_{ij} = P(X_{n+1} = j | X_n = i)$ for all pairs of vertices $(i, j) \in V$. It is stochastic, i.e. :

$$p_{ij} \geq 0 \quad \wedge \quad \sum_{j=1}^{|V|} p_{ij} = 1 \quad \forall i, j \in V \tag{32}$$

For the associated Markov graph, the sochasticness translates to the outgoing edges of a vertex v summing up to 1.

While the above single-step transition matrix shows short-term behaviour of a Markov chain, the long-term behaviour is described by the n-step transition matrix $\mathbb{P}_n = \mathbb{P}(m, m+n) = \mathbb{P}^n$; following the Chapman-Kolmogorov equation $\mathbb{P}(m, m+n+r) = \mathbb{P}(m, m+n) \cdot \mathbb{P}(m+n, m+n+r)$.

2.2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a topic model introduced by Blei et al. [62] in 2003. It bases on two assumptions:

1. Topics are usually described by few key terms which are thus very likely to occur when a document talks about the topic [57]. Crain et al. [57] illustrate this with the difficulty a player faces in the game of Taboo® (a registered trademark of Hasbro) where players have to describe a word without using a set of taboo words. This can be modeled as a "conditional probability that an author will use a term given the topic the author is writing about" [57], the topic-word distribution.
2. Documents are assumed to be created in a certain generative process [58, 63]. This process can be modeled as a stochastic process and inferred based on the seen documents [58].

LDA requires the number of topics K to be known a priori. Furthermore, every document considers each of the K topics [58] which are characterized by a topic-word distribution [64]. For each document w in a corpus D with $|D| = M$, the generative process as described in [62] and outlined by Kling [23] is defined as follows.

1. Choose the number of words from a Poisson distribution

$$N \sim Poi(\lambda) \tag{33}$$

2. For every of the K topics, draw a topic-word distribution ϕ_k from a Dirichlet distribution with parameter β

$$\phi_k \sim Dir(\beta) \tag{34}$$

3. Draw a document-topic distribution θ_i from a Dirichlet distribution with parameter α

$$\theta_i \sim Dir(\alpha) \tag{35}$$

4. For each of the N words w_{ij}, draw a topic z_{ij} from θ_i, then draw a word w_{ij} from $\phi_{z_{ij}}$

$$z_{ij} \sim Mult(\theta_i) \tag{36}$$

$$w_{ij} \sim Mult(\phi_{z_{ij}}) \tag{37}$$

The generative process "defines a joint probability distribution over both the observed and hidden random variable" [58]. The documents and their words are observed, while the actual document-topic and topic-word distributions are hidden [58]. Based on this, the posterior distribution of the model given the documents can be inferred [58]. The posterior distribution given a document w is given by

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{38}$$

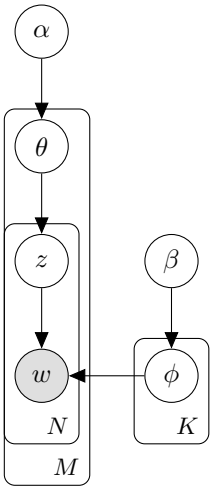


Figure 3: Plate notation of the LDA model as adopted from Kling [23]. Variables with grey background are observed, while such with white background are hidden. Plates indicate repitition.

as outlined in [62]. Calculating the numerator, "the joint distribution of all the random variables" [58], is achievable. Calcularing the denominator, "the marginal probability of the observations, which is the probability of seeing the observed corpus under any topic model" [58], is not since this would equal going through all possible models [58, 23]. For K topics and a corpus of L distinct words, this is a space of size K^L [58, 23]. Thus, approximative methods are used in practice. Popular methods are Gibbs sampling and variational inference [62, 58, 23]. The LDA implementation used in this thesis employs the latter. It appromixates the hidden parameters of the model by assuming their independence and, thereby, modeling it as an approximative distribution. More precisely, collapsed variational inference is used "where the multinomial parameters are marginalised out" [23]. For more information on inference, the reader is referred to [62, 58, 23, 57]

LDA is simple and powerful topic model [58] but needs the number of topics to be known a priori, which is usually not the case for real-world data [64] and leads us to the next two models, HDP and HMDP.

2.2.3 Hierarchical Dirichlet Processes

The Hierarchical Dirichlet Process (HDP) is a Bayesian nonparametric topic model first presented by Teh et al. [56]. It is also based on the two underlying ideas of LDA; how a topic is best described by certain essential words [57] and how document creation can be described with an inferable generative process [62]. In HDP, however, the generative process is different. The a priori restriction of the topic space made in LDA is lifted as it does not rely on finite-dimensional [65] Dirichlet distribution

priors. Instead, the Dirichlet Process (DP) [66] is used. Thus, the explorable topic space is of infinite size [56] and K does not need to be known beforehand. In DP mixture models[16], new groups, in our case topics, can be opened up if necessary [56]. The notion of *Dirichlet Process* shall be defined in the following. The Dirichlet process [66] is a stochastic process. It is a distribution of probability measures and can be thought of as an infinite-dimensional generalization of the Dirichlet distribution. It is commonly illustrated with a metaphor called the Chinese Restaurant Process (CRP) [67] where a chinese restaurant with an unbounded number of tables is considered. Customers, denoted by θ_i [56], enter the restaurant one by one. They sit at an occupied table with a probability proportional to the number of people already sitting at it denoted by m_k. Every customer, however, can also open up a new table with a probability γ. γ is, therefore, also called scaling parameter. In a topic model, it influences the number of topics [23]. As outlined in [23], a random variable Z can be defined which indicates the table a person chooses with the distribution:

$$P(Z = k) = \begin{cases} \frac{m_k}{(\sum_{j=1}^{K} m_j)+\gamma} & \text{for existing tables} \\ \frac{\gamma}{(\sum_{j=1}^{K} m_j)+\gamma} & \text{for a new table } k = K+1 \end{cases}. \tag{39}$$

Here, K denotes the number of tables already occupied. For different views such as the Stick-Breaking construction refer to [56, 23, 65]. Formally, the Dirichlet process is defined as follows.

Definition 2.33. Dirichlet process Let G be a probability measure over a measurable space θ. G is distributed according to a Dirichlet process with base measure H and scaling parameter γ, write $G \sim DP(\gamma, H)$, if for any partition $(A_1, ..., A_N)$ of H [66, 56, 23]:

$$(G(A_1), ..., G(A_N)) \sim Dir(\gamma H(A_1), ..., \gamma H(A_N)) \tag{40}$$

As outlined, DP mixture models can open up new clusters when necessary [56]. Then, the cluster parameters are selected dependening on a common base distribution G_0. However, such base distributions are often smooth. This is a problem when components should be shared as the probability of drawing the same component twice would then equal zero [56, 23]. That is, no topic could be chosen twice. Therefore, G_0 is itsself drawn from a DP mixture which couples "document-specific measures on the topic space" [23]. This way, components can be shared by groups [56] and, thus, topics between documents.

As the HDP model uses multiple Dirichlet Processes it is natural to extend the metaphors used to explain the DP to explain its generative process. The CRP is extendend to the Chinese Restaurant Franchise (CRF) [56]. Instead of a single restaurant, multiple franchise restaurants are considered. Every restaurant stands for a document-level Dirichlet Process. Furthermore, all of the restaurants share an infinite set of dishes, the gobal menu. This corresponds documents sharing a set of

[16] Mixture models assume "grouped observations [...] generated by mixtures of multiple latent distributions" [23], in our case documents consisting of mixtures of topics [23].

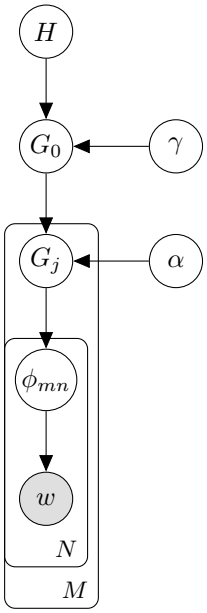


Figure 4: Plate notation of the HDP model as adopted from Kling [23] and Teh et al. [56]. Variables with grey background are observed, while such with white background are hidden. Plates indicate repitition.

topics in the HDP topic model. Customers correspond to words. They enter the restaurant and choose a table as in the CRP [23]. Again, a table is chosen with probability $\frac{m_k}{(\sum_{j=1}^{k} m_j)+\alpha}$ proportional to the number of people already sitting there. A new one is opened up with probability $\frac{\alpha}{(\sum_{j=1}^{k} m_j)+\alpha}$ dependent on the scaling parameter α. When a new table is opened in the CRF, a single dish is chosen which is served to all customers who opt to sit at the table [56]. The dish is chosen with a probability proportional to the number of tables across all restaurants already serving it [23]. A new dish is chosen according to scaling parameter γ. Let n_k denote the number of tables serving dish k across all restaurants. Again, a random variable X can be defined which indicates the probability that dish k is chosen. It is distributed as follows [68].

$$P(X = k) = \begin{cases} \frac{n_k}{(\sum_{j=1}^{K} n_j)+\alpha} & \text{for dishes, which have already been served} \\ \frac{\gamma}{(\sum_{j=1}^{K} n_j)+\gamma} & \text{for a new dish } k = K + 1 \end{cases} \tag{41}$$

, where K denotes the number of unique dishes already served.

Accordingly, the generative process of the HDP is defined as [56, 23]:

1. Sample the global topic measure G_0 from a Dirichlet Process with concentration parameter γ and Dirichlet prior $H \sim Dir(\beta)$.

$$G_0 \sim DP(\gamma, H) \tag{42}$$

2. Draw a document-specific measure G_m for every document from a Dirichlet Process with scaling parameter α and G_0 as base measure

$$G_m \sim DP(\alpha, G_0) \tag{43}$$

3. For every word w_{mn}, in document m draw a multinomial topic-word distribution ϕ_{mn} from G_m and the word w_{mn} from ϕ_{mn}

$$\phi_{mn} \sim G_j \tag{44}$$

$$w_{mn} \sim \phi_{mn} \tag{45}$$

As for LDA, there exist different approximative inference schemes for HDP as the actual inference of the parameters is intractable. Gibbs sampling and variational inference schemes can be tailored to fit the infinite probability measures in the HDP [23]. Again, the reader is referred to [56, 23] for Gibbs sampling and [23] for variational inference.

2.2.4 Hierarchical multi-Dirichlet Process

The previously introduced topic models have only considered the textual data of documents [23]. However, especially for social data, there exists contextual data which surrounds documents, e.g. timestamps or authorship information [23], which can aid in finding the latent topics. In the following, the Hierarchical multi-Dirichlet Process (HMDP) model developed by Kling [23] is introduced which allows for context inclusion by extending the HDP [56] for arbitary contexts [23].

In the HMDP, context is described by context variables can have four different natures [23].

- Discrete context variables do not have a pre-defined ordering [23] and include, for example, an article identifier specifying the article associated with a comment.
- Linear and continuous context variables can be ordered [23] and include, for example, a timestamp.
- Spherical context variables are commonly geographical locations [23].
- Cyclic context variables are commonly of temporal nature [23]. Cyclic typically refers to daily, weekly, monthly or annual cycles [23].

For a review of other context-aware models, the reader is referred to [23].
In the HMDP model, context variables are treated independently and serve as a storage of "the location of each document in the context space" [23]. An example of such a *context space* is a timeline, where the corresponding *context variable* can be the timestamp of a comment. Documents with similar context variables in a context space are grouped into *context groups*. Thus, every document is in exactly one context group per context space. Context groups are furthermore related to *context*

clusters. For discrete context variables, the relationship is 1:1 where a context groups parent is the context cluster of the same index. For linear and cyclic variables, it is 1:N where the parents of the context group are now not only the context cluster with the same index but also its adjacent clusters. The benefit of context clusters is that they "are associated with topic distributions which can be arbitrarily mixed to obtain the prior for the documents of a context group" [23]. That is, documents of the same context group are believed to have a similar topic distribution. This is not possible in the standard Dirichlet Process and is better illustrated when the definition of the *Multi-Dirichlet Process* [69] (MDP) is compared to the one of a standard Dirichlet Process.

Recall the definition of the DP. $G \sim DP(\gamma, H)$ is dependent on the scaling parameter γ and the base measure H. In the MPD however, not only one base measure and scaling parameter is considered but a set of parent measures $G_1, ..., G_P$ with associated scaling parameters $\alpha_1, ..., \alpha_P$. With

$$A = \sum_{p=1}^{P} \qquad \eta_p = \frac{\alpha_p}{A} \tag{46}$$

the MDP can be alternatively understood as a DP with base measure $H = \sum_{p=1}^{P} \eta_p G_p$, i.e. "the weighted sum of parent distributions, and concentration parameter A" [69], and, thus, as $DP(A, H)$ [69]. Formally the MDP is defined as follows.

Definition 2.34. Multi-Dirichlet Process [69] Let $G_1, ..., G_P$ be probability measures on a standard Borel space $(\Theta, \mathcal{B})$ with associated scaling parameters $\alpha_1, ..., \alpha_P$. A probability measure G over $(\Theta, \mathcal{B})$, which for all finite measurable partitions $(A_1, ..., A_r)$ of Θ yields a Dirichlet-distributed random vector

$$(G(A_1), ..., G(A_r)) \sim Dir(\sum_{p=1}^{P} \alpha_p G_p(A_1), ..., \sum_{p=1}^{P} \alpha_p G_p(A_r) \tag{47}$$

is called Multi-Dirichlet Process, write $MDP(\alpha_1, ..., \alpha_P, G_1, ..., G_P)$.

Translating this a comparison of HMDP and HDP, the distribution of topics in a document in the HMDP is not only influenced by a single base measure but by a mixture of measures depending on contextual information. The generative process in the HMDP is defined according to [23] as follows.

1. Draw a global topic distribution G_0 from a DP with a symmetric Dirichlet distribution H over the topic space as base measure:

$$G_0 \sim DP(\gamma, H) \qquad H = Dir(\beta) \tag{48}$$

2. For every context space f of the F context spaces and every of the C_f context clusters of the context space, draw a topic distribution for each context cluster j:

$$G_j^c \sim DP(\alpha_0, G_0) \tag{49}$$

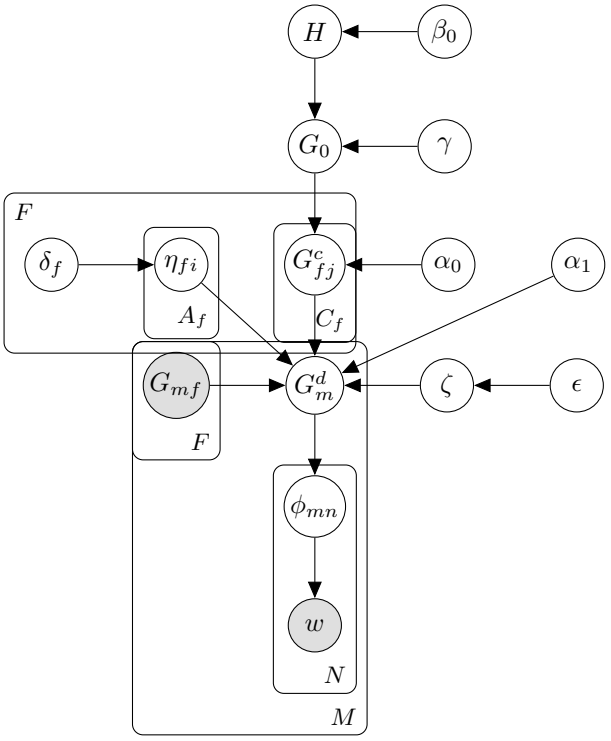


Figure 5: Plate notation of the HMDP model as adapted from Kling [23]. Variables with grey background are observed, while such with white background are hidden. Plates indicate repitition. See Appendix for a table of variables.

3. Context spaces can have different strengths on the document topics. For each context space, the strength of influence is stored in ζ, which is drawn from a Dirichlet distribution:

$$\zeta \sim Dir(\epsilon) \tag{50}$$

4. For each context cluster within a group draw a multinomial η_{fi} from a symmetric Dirichlet distribution with parameter δ_f, which governs its influence:

$$\eta_{fi} \sim Dir(\delta_f) \tag{51}$$

5. For every document m a document specific topic distribution is sampled from a multi-Dirichlet Process. Here $\alpha\boldsymbol{\xi\eta}$ is shorthand for the vector of mixing proportions and $\boldsymbol{G^c}$ for the cluster-specific base measure of parent clusters.

$$G^d_m \sim MDP(\alpha_1\boldsymbol{\xi\eta}, \boldsymbol{G^c}) \tag{52}$$

6. For every of the N_m words of m, a multinomial topic-word distribution ϕ_{mn} is drawn from the document's topic distribution G^d_m, from which a word w_{mn} is then drawn.

$$\phi_{mn} \sim G^d_m \tag{53}$$

$$w_{mn} \sim \phi_{mn} \tag{54}$$

The HMDP implementation used in this thesis [17] employs collapsed variational inference and and requires a truncation K of topics.

2.2.5 Markov Cluster Algorithm

Algorithm 1 The Markov Cluster algorithm as outlined in [49]

Require: (normalized) square Matrix M of order n, power parameter p, inflation parameter r
  repeat
    Expand: $M \leftarrow M^p$
    Inflate: $m_{ij} \leftarrow \frac{m_{ij}^r}{\sum_{k=1}^{n} m_{kj}^r}$
  until M is (nearly-) idempotent
  Read clusters from M

The Markov Cluster Algorithm [49] (MCL) is different from the previously presented models. It is a graph-clustering algorithm and seeks to find groups in graphs which are believed to naturally occur [49]. It is not focused on the specific notion of *topics* or documents in general. Nevertheless, its viability for the task of topic clustering has been shown [1]. Furthermore, the MCL is able to determine the number of clusters automatically [49]. The fundamental idea of the MCL is best described with a quote. Let $G = (V, E)$ be a graph.

> "A random walk [18] in G that visits a dense cluster will likely not leave the cluster until many of its vertices have been visited" [49].

In the MCL, a random walk is interpreted as simulating flow within a graph. Flow in the natural groups of a graph will remain strong, while flow between the natural groups will vanish over time revealing borders between groups [49]. To illustrate this, let us consider a metaphor used by van Dongen [49]. Imagine a car driver driving around aimlessly between a set of crossings and turns denoting vertices via streets denoting edges and through different districts denoting natural groups. Inside a district there are many crossings connected by viable roads, while districts themselves are connected only by very few roads. Naturally the driver will remain within a district for a substantial amount of time.

In order to find groups, the graph is first transformed into a Markov graph. The MCL algorithm then works on the the (single-step) transition matrix M associated with the Markov graph. Flow is simulated by Markov transition where the transition matrix is repeatedly raised to the power of the parameter p which is called *expansion*. For comment clustering, it has been reported that $p > 2$ results in too few

[17] `https://github.com/ckling/promoss`
[18] A finite and time-reversible markov chain [70]

clusters [1]. However, repeated Markov transitions alone are not sufficient for clustering, as the Markov process lacks the exhibition of a cluster structure [49]. Thus, a second operation enabled by an *inflation operator* is added and conversely called *inflation*. Both expansion and inflation promote flow in natural groups [49]. The inflation operator is based on the Hadamard product $A \circ B$ given by the element-wise multiplication of A and B as $(A \circ B)_{ij} = (A)_{ij}(B)_{ij}$. When applied to two identical matrices as in the MCL, this is equivalent to raising each element of the matrix to the power of two. In the MCL, it is not restricted to the power of two but variably determined by the inflation parameter r. Furthermore, a normalization step ensures the stochasticness of M at the end of every iteration. An inflation parameter $r \in (0, 1)$ induces column-homogeneity of M and $r \in (1, \infty)$ increases inhomogeneinity [49]. A Matrix M being column-homogeneous means that the non-zero values of every column are equal [49]. In practice, $r \geq 2$ has been reported to result in too many clusters [1]. The inflation operator is formally defined [49] as follows.

Definition 2.35. Inflation operator Γ_r Let $M \in \mathbb{R}^{k \times l}$. The inflation operator $\Gamma_r : \mathbb{R}^{k \times l} \to \mathbb{R}^{k \times l}$ with power coefficient r is defined by [49]:

$$(\Gamma_r M)_{ij} = \frac{m_{ij}^r}{\sum_{k=1}^{n} m_{kj}^r} \tag{55}$$

Both steps, expansion and inflation, are in theory repeated until M is idempotent, i.e. does not change anymore. In practice, a limited of maximum iterations is used [1]. In the end, the clusters can be read of the rows of the matrix M. To sum it up, the algorithm consists of two operations: **Expansion**, which allows flow between groups, and **inflation**, which promotes flow within groups and limits flow across groups [49, 1]. A maintained implementation is available [19] which uses pruning for performance reasons.

2.2.6 Overview

Table 1: Properties fulfilled by introduced models. [1] Implicitely.

Model	Able to learn no. of groups	Context-awareness
LDA	-	-
HDP	✓	-
HMDP	✓	✓
MCL	✓	(✓)[1]

Distinction can be made in different senses. LDA [62], HDP [56] and HMDP [23] are categorized as topic models [62, 23, 57, 23] and the MCL [49] as a graph-clustering

[19] `https://micans.org/mcl/`

algorithm [49]. LDA requires the number of topics to be known a priori [62]. All other models can determine it automatically [56, 23, 49]. LDA [62] and HDP [56] only consider textual information. MCL and HMDP can be used to include context. The MCL does not explicitely use contextual information but context can implicitely be included into the Markov graph build-up by similarity measures based on datapoint relationships, for example whether two comments are in the same thread [1]. The HMDP [23] explicitely models context information in its generative process [23].

2.3 Linear Regression

The Markov Cluster Algorithm implementation used in [1] and reimplemented in this thesis uses a linear regression model to weight edges between two comment nodes by their similarity. Linear regression in the context of Machine Learning is a supervised learning model [59]. It uses a training set of m observations $\{X_i\}$ with $i = 1, ..., m$ and associated target values $\{t_i\}$ with the goal to predict t for a new observation x with the trained model. From a probabilistic point of view this is equivalent to modeling the distribution of conditional probabilities $P(t|x)$ [59] which "minimize[s] the expected value of a suitably chosen loss function" [59]. Linear regression can be such that it considers a linear combination of input variables to make predictions [59].

$$y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n \text{ where } x = (x_1, ..., x_n)^T \text{ and } w = (w_0, ..., w_n)^T \quad (56)$$

This can be extended to a linear combination of nonlinear basis functions $\phi_j(x)$ [59]

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathrm{x}) \quad (57)$$

The purpose of training is to find the parameters w and thereby the equation which fits the training set of observations and target values best. The set of observations can be represented as a matrix $A \in \mathbb{R}^{m \times n}$ where each column represents an input variable and each row one observation. The set of target values can be represented as a column-vector $b \in \mathbb{R}^m$, where each element b_i represents the labeled target value of the observation of row i of A. Finding the parameter values $w \in \mathbb{R}^n$ then equals solving the equation system $Aw = b$. However, often holds true $rk(A) \neq rk(A, b)$[20] which means that the equation system does not have a solution [71]. Thus, the objective is to find a set of parameters which approximates the equation system best, i.e. minimizes the residual $Aw - b$. This can be formulated as the Least Squares problem [71].

[20] The rank of a matrix is defined as the dimension of the image [71] or the maximum number of linearly independent rows.

Definition 2.36. Least Squares Problem Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. The problem of finding $w \in \mathbb{R}^n$ such that

$$\sum_{i=1}^{m} |b_i - (Aw)_i|^2 \to min \quad \text{or} \quad min_{w \in \mathbb{R}^n} ||b - Aw||_2^2 \tag{58}$$

is called Least Squares Problem and $w = \text{argmin}_{w \in \mathbb{R}^n} ||b - Aw||_2^2$ is called Least Squares solution [71]. $w \in \mathbb{R}^n$ then also solves the *normal equation*

$$A^T Aw = Ab \tag{59}$$

The Least Squares Problem can also be formulated such that w follows certain constraints. For example, the Non-negative Least Squares Problem constrains w to be positive. The least squares solution is then

$$w = \underset{w \in \mathbb{R}^n}{\text{argmin}} ||b - Aw||_2^2 \text{ where } w \geq 0. \tag{60}$$

Arbitrary bounds for w can be defined accordingly.

3 Approach

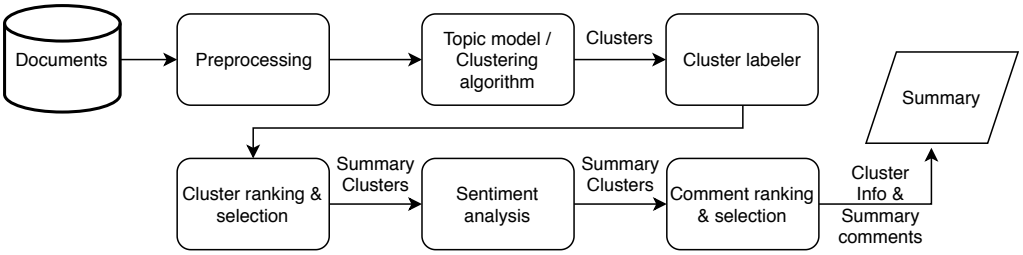


Figure 6: Outline of the approach used for summary generation.

In the previous section we saw that there are plenty of methods for extractive sumarization. However, there is a common denominator [20], a three-step approach which is also found in related works targeting single-article comment summarization [2, 3, 7]. The same three-step approach consisting of topic clustering, ranking and selection is used in this thesis. The definition of topic clustering from section 2 shall be refined for our chosen approach in the following.

1. **Topic clustering** - In this step, we cluster comments by their topic. Hard clustering is performed, where each comment belongs to exactly one cluster, which tackles data sparsity [72]. Each comment is assigned to its most significant topic. Topics have a distribution over documents, so that we can express the cluster of each topic $t_i \in T$ as the comments $c \in C$ for which t_i is most likely.

$$C^{t_i} = \left\{ c \in C | t_i = \underset{t \in T}{\operatorname{argmax}} P(t|c) \right\} \tag{61}$$

   In order to cluster comments by topic, we compare the topic models HMDP and LDA and the graph-clustering algorithm MCL.

2. **Ranking of comments and clusters** - Ranking comments establishes how significant a comment is in respect to its topic. Therefore, comments are only ranked within each topic cluster found in the previous step. As outlined in section 2, ranking bases on a scoring function and a linear order relation. Each cluster is ranked based on the number of comments contained in it and each comment within a cluster is ranked based on its Maximal Marginal Relevance. A higher score indicates a higher significance.
3. **Selection** The ten highest ranked clusters and their highest ranked comment of each positive and negative sentiment are selected as summary.

Selection requires the sentiment of a comment to be known. Therefore, we use sentiment analysis before selection. Furthermore, a short label is given to each topic cluster after topic modeling and clustering. In the following, the tasks of topic modeling and clustering are described in detail. Later stages of summary generation,

namely ranking, sentiment analysis, selection and visualization can be found in section 6.

This thesis and the following sections focus on topic clustering, since the challenges of an unknown number of topics and topics overlapping across comment sections of different articles are significant in multi-article summarization. Moreover, we want to target the problem of data sparsity with context inclusion in the HMDP.

3.1 Data source

In order to execute the outlined approach, we considered different datasets upon their fulfillment of domain and metadata requirements as follows.

- Multiple, topically related, articles.
- Time of every articles publication.
- A sufficient number of comments of each article.
- Metadata for comments:
  - Link between article and comment.
  - Timestamp of publication.
  - Thread identifier.

Three existing datasets have found consideration. SoLSCSum [73] which contains 25,633 comments issued under 157 articles with annotations and gold standard summaries was unavailable as of the start of this thesis. Second, the SENSEI annotated corpus [47] contains 18 articles and associated comments from the Guardian newspaper with groupings by topic, based on which human-made summaries were produced. However, the dataset and reference summaries target single-article summarization. Thus, comments are also not grouped across articles which an evaluation of our methods requires. Therefore, the publicly available Yahoo News Annotated Comments Corpus [74] [21] is used throughout this thesis. It contains 522,000 comments issued in 140,000 conversational threads under articles of Yahoo news. Many are topically related on political questions. Furthermore, timestamps and thread identifieres are contained. While articles are not included in the first place, a link of each comment to its article by headline and url exists. To enforce the constraint of articles being related by topic, three subsets of the YNACC were compiled. Each contain varying amounts of articles and comments to enable a study at different scales. For every article, the content and date of publication was scraped from the website using a rule-based scraper. First, the HTML of each page was retrieved and then, the content within relevant element tags was retrieved using the Python library Scrapy [22] and its xPath [23] implementation. Combined with the associated

[21] https://github.com/cnap/ynacc
[22] https://github.com/scrapy/scrapy
[23] https://www.w3.org/TR/xpath/all/

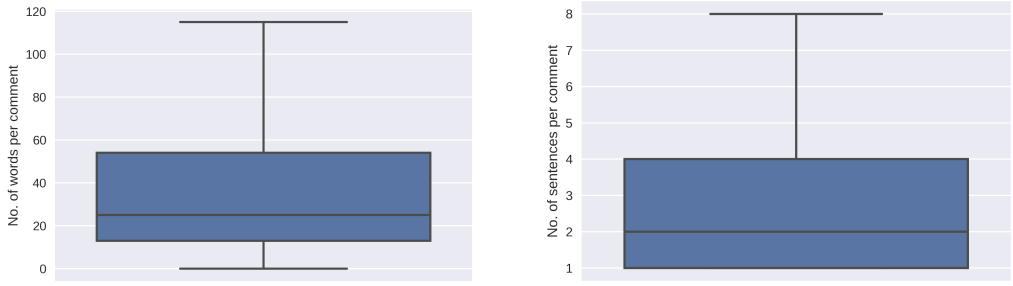
Table 2: Overview over the three YNACC subsets.

	#1	#2	#3
No. of articles	10	17	38
No. of comment threads	37	46	934
No. of comments	259	312	6195
Date range	19/4 - 5/5/2016	3/4 - 5/5/2016	2/4 - 6/5/2016
Topic domain	narrow	broad	broad

comments from the YNACC, this results in a sufficient dataset. However, as the YNACC does not contain any grouping, one was manually produced as described in subsection 5.1. In the following, the three subsets are referred to as dataset #1, dataset #2 and dataset #3 ascending by their size for convenience.

3.2 Characteristics of news article comments

Now that the skeleton of the approach and datasets are established, we will consider characteristics of news comments in order to be able to formulate sound topic clustering methods. A more thorough evaluation is found in [74]. Therefore, a subset of the YNACC [24] [74] annotated by experts is analyzed. It contains 23,383 comments from 696 articles. A list of annotations can be found in the Appendix. As outlined in



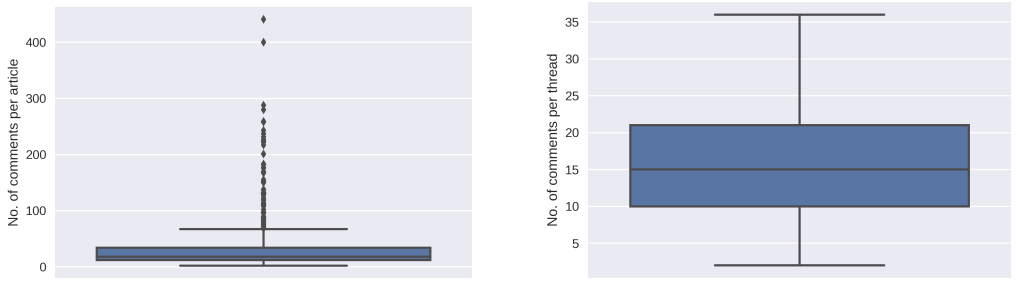
(a) Boxplot of the no. of words per comment

(b) Boxplot of the no. of sentences per comment

Figure 7: Description of the comment length for comments in the YNACC subset annotated by experts.

Figure 7 a) and b), comments are usually short and sparse with a median word count of 25 and a median sentence count of 2. 75% of comments were not longer than 4 sentences or 54 words. The median vocabulary size of comments is 19 words. Nevertheless, heavy outliers with far more than 100 words exist. The number of comments issued per article fluctuates heavily with a standard deviaton of over 46. In

[24] https://github.com/cnap/ynacc



(a) Boxplot of the no. of comments per article (b) Boxplot of the no. of comments per thread

Figure 8: No. of comments per article and thread.

the dataset, each article has between 2 and 441 comments. Comments are grouped in subdialogues which are also called threads and solidify a reply-relationship. Such threads form a significant grouping with a median number of 15 and a maximum number of 48 comments contained. A large share of threads contain an argumen-

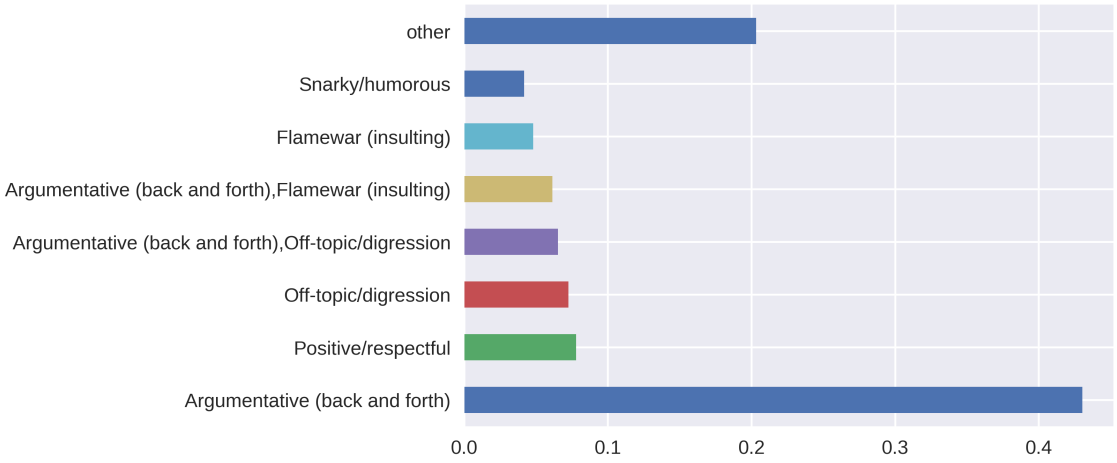


Figure 9: Conversation types within a thread.

tative debate, as shown in Figure 9. Typically, such argumentative debates share a common topic. Hence, the thread relationship of two comments can be a determining factor for whether they share a topic. This supports the intra-comment-comment relationship introduced by Ma et al. [2], where comments echo ideas from other comments. The majority of comments is not issued to the general audience but rather as a reply to a specific comment, as outlined in Figure 10, reinforcing the idea that commenters are inspired by other comments. Nevertheless, such inspiration does not only need to be taken from a comments' article and its commentsphere. Only about 60% of the comments were labeled as on-topic with the article or il-

legible of which no distinction was made in the labeling process [25]. A third of the comments were labeled as off-topic with the article. It is conceivable that a substantial amount of these comments talk about a related topic found in related articles and their comments which make up the relationships inter-comment-comment and inter-news-comment in [2]. Thus, the idea that the conversational structure under comments of topically related articles is very broad with heavily overlapping topics is corroborated. Still, the article of a comment seems to be at least indicative of its topic.

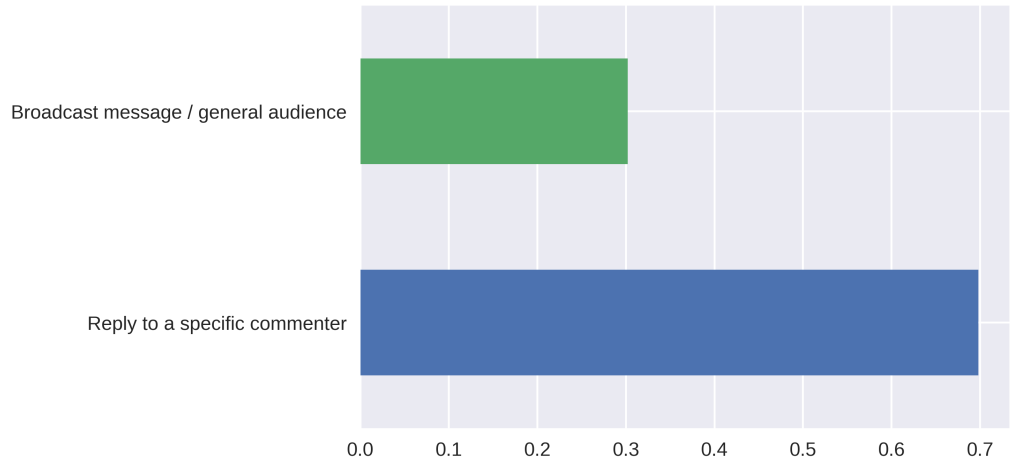


Figure 10: The intended audience of comments.

[25] https://github.com/cnap/ynacc/blob/master/rater-guidelines.pdf

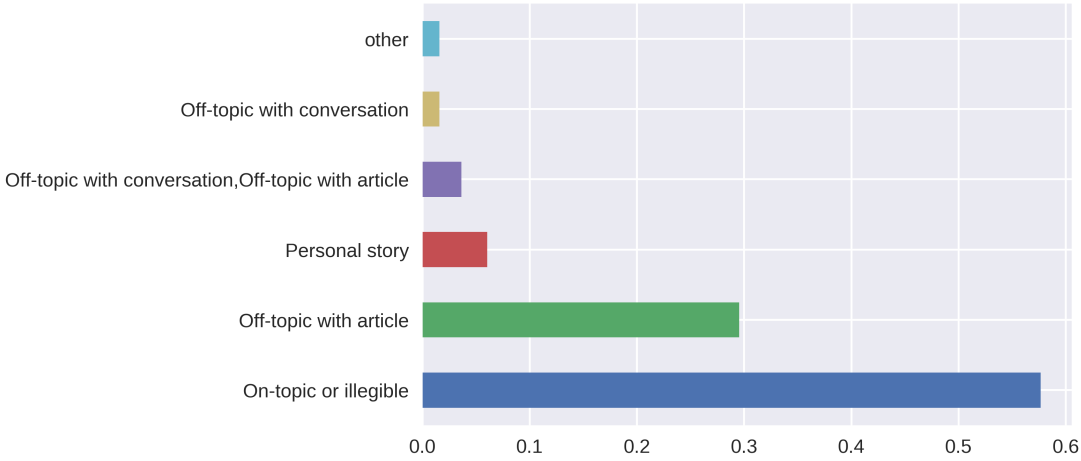


Figure 11: Topic of comments.

4 Topic Modeling & Clustering

News article comments are generally of unstructured nature in regards to their topic. This hinders readers [46] and summarization systems. For a user, clustering comments by common topics makes it possible to gain an overview over the discussed topics and their reception in terms of e.g. popularity and polarity quickly. Furthermore, it enables selective reading as a user can choose topics of interest. For an (extractive) summarization system, the benefit is two-fold. First and foremost, it aids in ensuring the key requirement of preserving the overall information content. If comments are selected independent of their topic, a ranking algorithm might omit an important topic completely. Often times ranking algorithms favor long comments over short ones [3]. To employ a simplified example, if a set of comments under news articles contains two dominant topics of which one is discussed in lengthy comments and one in rather short comments, the latter might be spared completely if comments are ranked and selected independent of their topic. Furthermore, the structuring allows efficient use of query-optimizing ranking mechanisms. This thesis uses such a mechanism, Maximal Marginal Relevance [45] which maximizes query-relevance while minimizing redundancy based on similarity measurement. Without any structure, however, it is not clear what the query should be. For one article it might be the title but for multiple ones? Then again, using the title might omit topics different from it. In contrast, when comments are grouped by topic, the query can be restricted to the topic. The most likely words of the topic-word distribution in a topic model can form the query and comments are ranked within each topic cluster. This brings forth performance as a second advantage of grouping. Summary clusters are selected before ranking comments in only selected clusters which ensures that only comments considered for a summary are ranked. Additionally, ranking algorithms such as MMR [45] or LexRank [11] usually require similarity measurement between every considered document. Here, ranking comments only within clusters requires less operations than ranking them across clusters. To engage the previous simplified example, let the N comments be divided into topic A consisting of K comments and topic B consisting of M comments with $N = M + K$. $\binom{N}{2}$ similarities have to be measured to compare all of the N comments [26]. When comparing comments only within the two topics, $\binom{K}{2} + \binom{M}{2}$ measurements have to be taken. As $\binom{N}{2} > \binom{K}{2} + \binom{M}{2}$ holds for positive real numbers M, K[27], partioning is beneficial performance-wise. Thus, it is clearly advantageous for the summarization effort to cluster comments by a common topic.

Nevertheless, there are a number of challenges, as outlined in Figure 12, which form the basis of model selection.

[26] As the similarity of an object to itsself is 1 as defined in 2.12 it is sufficient to compare every possible pair other than a comment and itsself.

[27] A proof can be found in the Appendix

1. **Unknown no. of topics** - The number of topics discussed under an article is generally unknown [1].
2. **Sparsity** - As shown in subsection 3.2, comments are usually sparse, limiting the number of exploitable co-occurences between words for topic modeling [72] and hampering similarity measurement. Clustering short comments is generally difficult [1].
3. **Broad conversational structure** - A set of multiple articles and their comments can have a large temporal domain and overlapping topics across articles and comments.

Figure 12: Main challenges faced in topic clustering in this thesis.

Related works in single-article comment summarization have commonly employed two types of models. Topic Models due to the probabilistic model of document creation and their ability to find words highly descriptive of topics and graph-clustering which models relations between comments through similarity. More precisely, Latent Dirichlet Allocation [5, 2, 3, 1] and the Markov Cluster Algorithm [1] have been applied successfully. Both have outperformed other approaches such as Cosine Distance Clustering [3] or k-means Clustering [3, 5]. Therefore, it is natural to examine their applicability for the multi-article domain of the presented work. However, Latent Dirichlet Allocation is unable to learn the number of topics. When comments under a single article are modeled, an estimate of the topic number beforehand might be feasible. For multiple articles, such an assumption is presumably too restrictive. Thus, alternatives which are able to determine the number of topics are considered. We choose the afore-mentioned MCL and the nonparametric topic model Hierarchical multi-Dirichlet Process, which is also context-aware. Nevertheless, LDA serves as a baseline to which the MCL the HMDP are compared due to its proven performance.

In order to tackle data sparsity in LDA, both comments and articles are modeled. This provides auxiliary information and co-occurences which has shown to be performance enhancing [1]. Additionally, it bases on the assumption that comments pick up on paragraphs and ideas from articles which should, hence, be included in the model of document creation. The document-topic distribution of articles is not relevant and unused, since only comments are considered in the clustering. All three methods perform hard clustering in this thesis. In both topic models, comments are assigned their most likely topic, maximizing $P(t_i|c)$ for comment c and topic t_i under its document-topic distribution. Restricting the number of topics of a comment to one is also used to tackle data sparsity [72] and therein common to related works [1, 2, 3, 5]. Furthermore, the texts are preprocessed in the same fashion for all models. Punctuation is removed, words are lowercased, stop words are re-

moved and words are stemmed. The NLTK stop list of english words and the NLTK implementation of the Porter stemmer [75] are used.
The LDA implementation used in this thesis is available at `https://github.com/ckling/promoss` as part of the PROMOSS package. In the following, it is described how the MCL and HMDP are used and why the HMDP is introduced.

4.1 Markov Cluster Algorithm

The graph-based Markov Cluster Algorithm has outperformed LDA for clustering comments of one article in [1]. Hence, their method is examined in this thesis. Unlike in both topic models, articles are not considered but only comments. Let $C = \{c_i\}_{i=1...N}$ be the set of comments of all articles. The Markov graph is built up with each comment c_i as a node. Edges are established based on comment similarity. Each comment c_i is compared to each other comment c_j with a set of seven similarity measures [28] The graph is thus of the form (C, E) with $E \subseteq C \times C$ denoting the set of edges. In the following, let v_i denote the TF vectors of, t_i the set of terms and n_i the set of Named-Entities of c_i. The used similarity measures are as follows.

1. Cosine similarity of TF vector representations.
2. Cosine similarity of TF-IDF vector representations.
3. A modified version of cosine similarity -

$$\text{cosim}_{\text{mod}} = \begin{cases} \frac{v_i \cdot v_j}{5} & \text{if } v_i \cdot v_j \leq 5 \\ 1 & \text{else} \end{cases} \tag{62}$$

4. $$\text{dice}(c_1, c_2) = 2\frac{|t_1 \cap t_2|}{|c_1| + |c_2|} \tag{63}$$

5. $$\text{jaccard}(c_1, c_2) = \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|} \tag{64}$$

6. $$\text{Named-Entity Overlap}(c_1, c_2) = \frac{|n_1 \cap n_2|}{|n_1 \cup n_2|} \tag{65}$$

7. Thread-relationship which returns 1 if two comments c_i and c_j are in the same thread and 0 otherwise.

However, a certain degree of redundancy can be expected in a set of such closely related measures. Therefore, in addition, a restricted version of graph build-up is investigated which only uses cosine similarity based on TF-IDF vectors and thread-relationship. The latter is a form of implicite metadata modeling which addresses

[28] In the original paper [1] eight were used, however, the reply-relationship could not be replicated based on the dataset used in this thesis.

data sparstiy based on the notion that comments in the same thread likely target the same topic. The edge weight between two comments indicates their topical similarity. In both cases it is determined by a linear combination

$$e(c_1, c_2) = \sum_i \lambda_i s_i \tag{66}$$

of all considered similarity measures s_i equal to a linear regression model which is trained to obtain the weights λ_i. As in [1], an edge is added when a certain threshold is reached which is reportedly beneficial. Once the graph is established, it serves as input to the Markov Cluster Algorithm as developed by van Dongen [49]. In the end, the comment clusters can be read off the rows of the matrix [1]. The overall algorithm of MCL in the context of this thesis is, thus, as outlined in [1].

Algorithm 2 The Markov Cluster algorithm as used in this thesis and outlined in [1]

Require: A set of comments $C = \{c_i\}_{i=1...N}$, a square Matrix M of order N, power parameter p, inflation parameter r, maximum number of iterations iter, threshold

for $c_i \in C$ **do**
  for $c_j \in C$ **do**
    if $i = j$ **then**
      $m_{ij} \leftarrow 1$
    else if $e(c_i, c_j) \geq$ threshold **then**
      $m_{ij} \leftarrow e(c_i, c_j)$
    else
      $m_{ij} \leftarrow 0$
    end if
  end for
end for
repeat
  Expand: $M \leftarrow M^p$
  Inflate: $m_{ij} \leftarrow \frac{m_{ij}^r}{\sum_{k=1}^n m_{kj}^r}$
until iteration = iter
Read clusters from M
return Comment clusters

A substantial amount of training data is necessary to train the regression model. Therefore, comment pairs from the same and from differing topic clusters were collected from the gold standard used for evaluation. Target values are 1 for positive instances of the same and 0 for negative instances of different topic clusters. Herein this thesis deviates from [1], where the target values were 0 for negative instances and between $[0.5, 1]$ for positive instances. In total, 2524 positive and 7476 negative samples were collected. The linear regression models for both graph build-ups were

initially trained using the standard scikit-learn linear regression model. However, this obtained negative weights which are unwanted in the reproduction. Formulating the problem as non-negative least squares also produced zero values. Thus, a lower bound of 0.1 was set for the weights. Both least squares models were implemented using the SciPy[29] [76] implementation. The obtained weights can be found alongside the implementation in the Jupyter notebook "mcl_regression.ipynb".

4.2 Hierarchical multi-Dirichlet Process

The Hierarchical multi-Dirichlet Process model [23] is a nonparametric topic model and able to determine the number of topics automatically. Furthermore, contextual information is included in its generative process. Hence, the social context in which comments are created in the real world can be considered. In the introduction we have already seen an example where this is beneficial to topic modeling and how it can alleviate the challenging problem of data sparsity which is most evident in very short comments. A lack of sufficient textual data makes such comments difficult to assign topics to [1] if their context is not included. With the HMDP, we are able to include the context associated with comments. It is modeled as a prior belief about the topic distribution of documents with similar metadata through a context-specific prior [23]. Three types of context spaces are explored in this thesis based on certain assumptions.

1. **Timestamp of comment creation** - Context clusters within a timeline group comments created in the same timeframe. This models the idea that comments under news articles are influenced by currently trending topics and other comments and articles created shortly before them. Comments under an article about an election, for example, likely refer to current events in the local political environment. While it is also possible to model cycles, e.g. of topics present on weekdays and topics present on weekends [23], these are generally not expected to exist in news article comments.
2. **Article identifier** - Comments are grouped by their associated article based on a numeric identifier. The analysis of the YNACC shows that around 60% of comments broadly targeted their articles topic. Thus, it is reasonable to assume that a large number of comments under an article are likely to refer to the its topic. Moreover, comments echo ideas from their associated article as outlined in [2].
3. **Thread identifier** - Comments are grouped within threads indicated by a numeric identifier. Since such subdialogues constitute a reply-relationship between comments, it is likely that comments within the same conversational thread target similar topics as in the example in the introduction.

In addition to the consideration of metadata, the HMDP model is trained on articles as well as comments. The combination of modeling comments, articles and associ-

[29] `http://www.scipy.org/`

ated metadata seeks to tackle data sparsity and to model the broad conversational structure of comments under multiple news articles. With it, the relationships between comments and articles as outlined by Ma et al. [2] are targeted. Modeling articles supports the idea that comments spread information from their associated and from other articles. For this thesis, the HMDP implementation as developed by Kling [23] available at `https://github.com/ckling/promoss` as part of the PROMOSS package is used.

4.3 Cluster Labeling

Algorithm 3 The topic labeling algorithm used in this thesis.

Require: A set of comments $C^{t_i} = \{c_i\}_{i=1...N}$ which have been clustered in the same topic cluster of topic t_i, ϕ the topic-word distribution of t_i, W the set of the K most likely words by ϕ.
  for $c \in C^{t_i}$ **do**
    $L \leftarrow L \cup \{\text{n-grams contained in } c\}$
  end for
  return $\text{argmax}_{l \in L} \left(|l \cap W| + \sum_{w \in (l \cap W)} P(w|\phi) \right)$

Labeling of topic clusters makes it possible to present a concise overview over a topic. This is beneficial to selective reading as a user can quickly see which topics are of interest. For topic labeling, this thesis chooses a lightweight approach which does not make use of external knowledge but uses information provided by comments and topic models. It bases on the fact that the topic-word distribution of a topic assigns high probabilities to words representative of it. Furthermore, it is assumed that comments provide n-grams which make up for a label of their associated topic. In order to combine both notions, n-grams, in the case of the thesis bi- and trigrams, are extracted and filtered by whether they have an intersection with the top K words of the topic-word distribution. In the end, the n-gram with the largest intersection is chosen as a label for the topic cluster. If there exist multiple such n-grams, the one with the largest sum of word probabilities under the topic-word distribution is chosen. As the MCL does not provide such a distribution, one can be obtained indirectly by training a topic model with a topic number of 1 on the comments of the identified cluster which is then used to find labels according to the afore-mentioned approach. Let L denote the set of n-grams in the comments of a topic cluster which are label candidates. Let $W = \{w_1, ..., w_k\}$ be the set of the K most likely words under the topic-word distribution ϕ and, hence, $P(w_i|\phi) \geq P(w_j|\phi)$ for all $w_i \in W, w_j \notin W$. Then a selected label $l \in L$ fulfills

$$l = \underset{l \in L}{\text{argmax}} \left(|l \cap W| + \sum_{w \in (l \cap W)} P(w|\phi) \right). \tag{67}$$

A proof can be found in the Appendix. In this thesis, the intersection is calculated between candidates and the $K = 10$ most likely words of the topic. Bi- and trigrams are collected and filtered using NLTK. Before collection, punctuation and stop words are removed in comments and words are lowercased. Stemming is left out for the labels as it decreases readability. However, in order to obtain the probability of each word from the topic models topic-word distribution, a stemmed representation of each word is used as for the topic models stemming is performed. Again, the NLTK stop list of english words and Porter stemmer [75] implementation are used.

5 Topic Modeling & Clustering Evaluation

5.1 Approach

The evaluation is restricted to an evaluation of topic clustering ability, on which the thesis is focused. Comparing the topic models LDA and HMDP to the graph-clustering algorithm MCL extrinsically relies on a grouped gold standard. Ideally, evaluation should base on a large dataset. However, we were unable to find a sufficient dataset with gold standard grouping and annotating a large dataset in its entirety is infeasible in the restricted timeframe of the thesis. Thus, a subset of 100 of 6195 comments contained in dataset #3 was annotated and grouped into five broad topic clusters. The annotation scheme was as follows. Five comments with clearly distinct topics were chosen as seed comments. Then, random samples of the entire set were taken. If a comment was clearly about the same topic as one of the seed comments, it was grouped to it. If the topic could not be determined directly, e.g. for a short comment such as "This is a great idea", additional information in form of the conversational thread was taken into account. If, based on this additional info, the topic was one of the seeds comments topics, the comment was grouped to the seed comment. Otherwise a new sample was taken.

All models were trained on only the 100 grouped comments as well as on the entire dataset. The 100 annotated comments were in both cases used for evaluation based on the metrics BCubed Recall, BCubed Precision and BCubed F-measure, which were used to compare LDA and MCL in [1]. The BCubed measures fulfill all four constraints for extrinsic cluster evaluation metrics set in [77]. These are cluster homogeinity, completeness, rag bag which attributes for noisiness and size vs. quantity [77]. Recall and precision are calculated on a per item basis and then averaged. While arbitrary weights for datapoints are possible, each one is given the same weight in our evaluation. Precision and recall are defined according to [78] as follows. Let N be the number of comments, $c_i \in C^{t_i} = \{c \in C | t_i = \text{argmax}_{t \in T} P(t|c)\}$ be a comment and its associated topic cluster and G_i the gold standard cluster of comment c_i.

Definition 5.1. BCubed Precision

$$\text{Precision} = \sum_{i=1}^{N} \frac{\text{Precision}(c_i)}{N} \tag{68}$$

$$\text{Precision}(c_i) = \frac{|\text{correctly clustered comments in } C^{t_i}|}{|C^{t_i}|} \tag{69}$$

Definition 5.2. BCubed Recall

$$\text{Recall} = \sum_{i=1}^{N} \frac{\text{Recall}(c_i)}{N} \tag{70}$$

$$\text{Recall}(c_i) = \frac{|\text{correctly clustered comments in } C^{t_i}|}{|G_i|} \tag{71}$$

F-measure is then calculated according to [79].

Definition 5.3. F-measure

$$F_\beta = \frac{(\beta^2 + 1) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{72}$$

$\beta > 1$ favors recall, $\beta = 1$ calculates the harmonic mean and $\beta < 1$ favors precision.

For evaluation using only the 100 annotated comments for training, $\beta = 1$ and the number of inferred and gold standard clusters are chosen to be equal. For the MCL, such restrictions were achieved based on empirical evaluation of different inflation parameters. In the setting where all comments are clustered, on the other hand, $\beta = 0.5$ is chosen, since the number of inferred groups differs significantly from the number of groups in the gold standard, which is necessary for a natural evaluation. Therefore, it is expected that comments, which where grouped together in the coarsely grouped gold standard, are grouped in multiple, more fine-granulated groups. Thus, a higher number of false negatives [30] is expected which is penalized in recall. Precision is given a higher weight with $\beta = 0.5$ as advised in [79], based on the notion that clustering comments dissimilar by topic together should be penalized stronger than separating topics into more fine-granulated clusters.

In addition, both topic models are compared using Perplexity. 30% of documents are held-out for testing in the corpus D_{test} and the model is trained on the other 70% of documents in corpus D_{train}. Perplexity describes, how likely new observations are on the held-out documents with the trained model which tests the hypothesis of the generative process of document creation. Lower perplexity indicates a higher likelihood of new documents and better performance [23].

Definition 5.4. Perplexity

$$\text{perplexity}(\text{D}_{\text{test}}) = \exp\left(-\frac{\sum_{j=1}^{M}\sum_{i=1}^{N_i} \log p(w_{ji})}{\sum_{j=1}^{M} N_i}\right) \tag{73}$$

Evaluation of the HMDP includes an evaluation of context space influence carried out by inspection of context space weights. This provides information about the indicativeness of the considered forms of metadata.

[30] Comments clustered together by gold standard but not the model.

5.2 Results

5.2.1 MCL

In order to find out whether the results reported in [1] translate well for summarizing comments of multiple articles, the approach in [1] was replicated and the MCL applied on all three datasets. While algorithm and graph build-up are generally fast on the two smaller datasets, their combination took 1h35m on dataset #3. Different inflation parameter values, which can range from 1.2 for a coarse clustering to 5.0 for a fine granulated clustering, were tried out.

Table 3: Distribution of the number of threads contained in clusters found on dataset #3 by the MCL as replicated from [1] for different inflation parameters.

	2.0	1.2
No. of topics	923	574
Standard deviation	0.24	1.78
Minimum	1.0	1.0
25%-Quantile	1.0	1.0
50%-Quantile	1.0	1.0
75%-Quantile	1.0	2.0
Maximum	7.0	8.0

The starting value was set to 2.0 as advised by van Dongen in [31]. The MCL identified 923 clusters for 6195 comments. Upon inspection of the clustering structure, it became apparent that almost all clusters contained only a single thread. Inflation parameter 1.2 still resulted in 574 clusters and a median number of one thread per cluster. This version of the MCL generally appears to overfit the thread structure of comments through thread-relationship measure. In contrast, the version presented in this thesis, which only uses cosine similarity based on TF-IDF vectors and thread-relationship of two comments for graph build-up, finds 142 clusters for inflation parameter 2.0 and results in a more natural clustering. Therefore, this version is meant with 'MCL' for the remaining parts of the thesis.

5.2.2 Comparison of HMDP, LDA and MCL

When compared only on the 100 gold standard comments, HMDP and MCL outperform the baseline LDA significantly. Furthermore, MCL has a slightly higher recall, whereas HMDP provides a higher precision. Combining both measures in the F_1 measure, results in HMDP having an edge over the MCL. When trained on all comments, the results differ greatly from before. HMDP and MCL again outperform LDA. HMDP has high precision, followed by LDA and MCL with significant

[31] `https://micans.org/mcl/`

Table 4: Performance on 100 gold standard comments. The number of topics in topic models was restricted to 5 and the inflation parameter of the MCL was set to 1.55 to obtain 5 groups corresponding to the gold standard.

	HMDP	LDA	MCL
Precision	0.44	0.34	0.38
Recall	0.65	0.49	0.69
F_1-measure	0.52	0.40	0.49

Table 5: Performance on dataset #3 using 100 gold standard comments for evaluation. Here, $F_{0.5}$ was calculated as outlined above. The number of topics was restricted to 80 in LDA and HMDP and the inflation parameter i=1.808 used in the MCL to obtain 80 clusters. In both topic models, words occuring at least 20 times were kept.

	HMDP	LDA	MCL
Precision	0.78	0.67	0.5
Recall	0.27	0.2	0.64
$F_{0.5}$-measure	0.57	0.45	0.52

distance each. Recall, however, is high for the MCL but generally lower for HMDP and LDA. The HMDP again performs better than LDA in terms of recall. In terms of $F_{0.5}$-measure, the HMDP can be made out as the best performing model. To make sense of the drastic difference between the results obtained in both settings, we want to take a closer look at the clustering structure.

Table 6: Distribution of the number of comments contained in clusters found on dataset #3.

	HMDP	LDA	MCL
Standard deviation	51.6	198.7	362.4
Minimum	25.0	3.0	1.0
25%-Quantile	48.0	15.5	3.0
50%-Quantile	59.0	34.0	6.0
75%-Quantile	87.25	65.0	17.75
Maximum	297.0	1673.0	2699.0

As outlined in Table 6, the MCL clustered 4592 of the 6195 comments in the two largest clusters with the largest cluster containing 2699 comments and the second largest cluster containing 1893 comments. The median size of clusters was 6 in the MCL, 59 in the HMDP and 34 in LDA. The standard deviation of cluster sizes was $\tilde{3}62$ for the MCL, 52 for HMDP and 199 for LDA. This suggests that both topic models, especially HMDP, yield a generally smoother distribution of comments across

clusters than the MCL. Furthermore, it can be observed that, throughout all models, topic clusters contain comments from multiple articles and multiple threads as outlined in Figure 14.

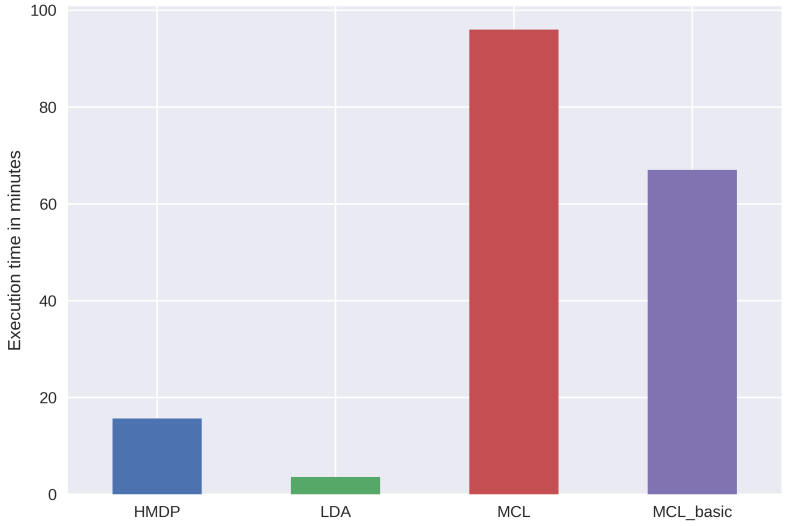
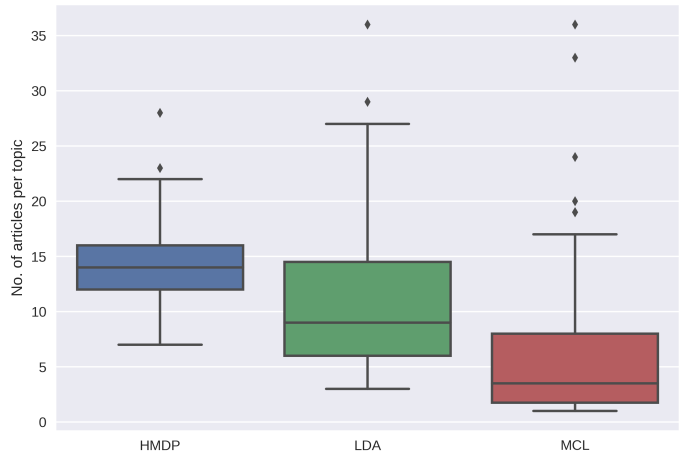
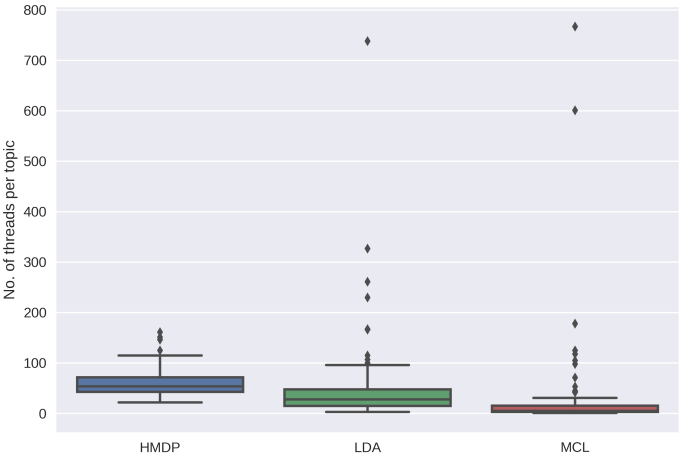


Figure 13: Execution times of the different models on dataset #3. Both MCL versions include graph build-up. MCL_basic denotes the version with cosine similarity and thread-relationship and MCL the version as used in [1]. For HMDP and LDA words occuring 20 times or more were kept and the topic number restricted to 80.

A comparison of execution times on dataset #3 shows that both topic models are faster than the MCL, which requires the similarity measurement of $\binom{n}{2}$ pairs of comments for graph build-up, which amounts to 19,185,915 pairs for 6195 comments.



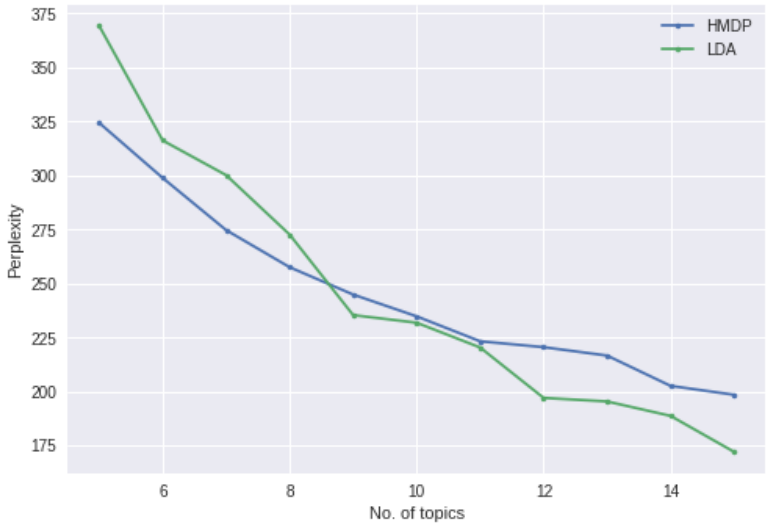
(a) Boxplot of the no. of articles per topic



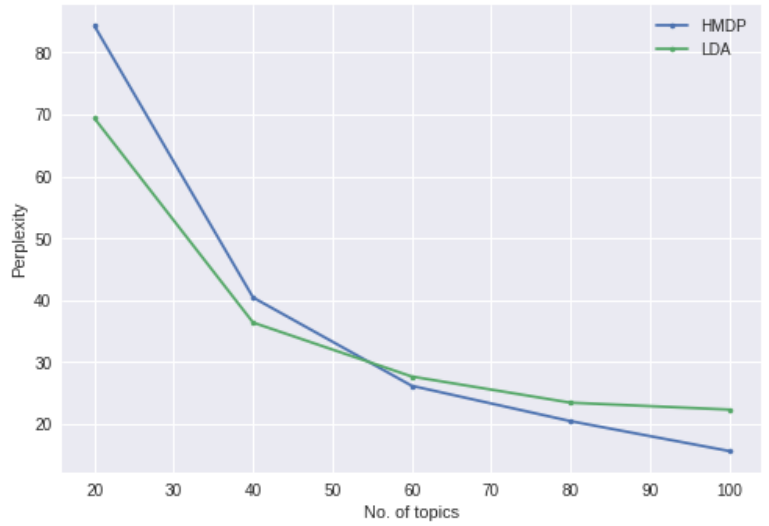
(b) Boxplot of the no. of threads per topic

Figure 14: Description of the number of articles and threads contained in topic clusters found on dataset #3. The number of topics / groups was restricted to 80.

5.2.3 Comparison of HMDP and LDA



(a) Perplexity of HMDP and LDA on dataset #1. Words occuring at least 2 times were kept.



(b) Perplexity of HMDP and LDA on dataset #3. Words occuring at least 30 times were kept.

Figure 15: Perplexity of HMDP and LDA on datasets of different sizes. On both datasets 30% of the documents were held-out for testing 70% used for training.

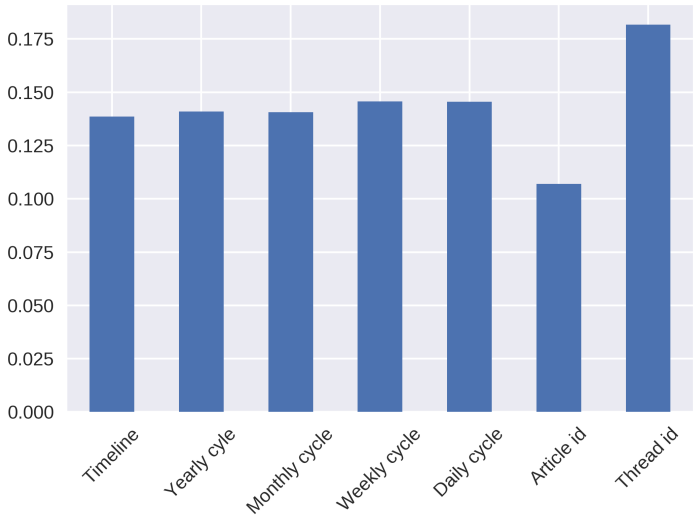
HMDP and LDA do not differ greatly in terms of perplexity on all datasets.

HMDP	LDA
wall border illeg build would hire work built employ mexico	wall build built tunnel china fenc border street reduc hotel
white black student union organ group racist supremacist colleg univers	student union colleg campu univers white ethnic member histor european
right ralli protest assault peopl free speech troubl person caus	ralli protest assault speech shove event troubl attend crowd disrupt
black white live matter peopl dont racist say privileg race	racism blm definit racist base race group superior belief discrimin
illeg alien tax money billion feder employ state dollar paid	illeg cost dollar billion tax state level taxpay alien feder
muslim islam kill christian religion amal europ clooney sharia murder	muslim christian islam terrorist thousand radic non gay rape death
women man men woman use bathroom room rape bruce transgend	woman men women bathroom room bruce daughter molest transgend restroom
liber lie media left conserv truth brain dead wing agenda	liber conserv speak polit left came realiz elect socialist today

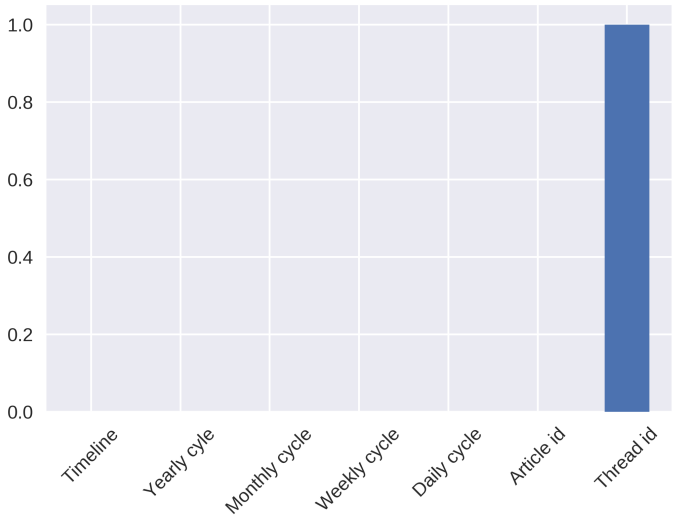
Table 7: Aligned subset of the 80 topics found out by HMDP and LDA on dataset #3 outlined by the 10 most likely words by topic-word distribution.

Comparison of the top words by topic-word distributions of HMDP and LDA shows that both find sensible and coherent topics across all datasets. The above table illustrates inferred topics of dataset #3, which is mainly concerned with the broad topic "Donald Trump" in the context of the general election 2016 in the United States of America.

5.2.4 Context influence in HMDP



(a) Context influence on dataset #1



(b) Context influence on dataset #2 and #3

Figure 16: Context space influence in the HMDP model.

In the HMDP model, the influence of each context space is stored in the variable ζ [23]. A higher context space weight is associated with a higher influence and indicativeness. The distribution of context weights is fairly smooth in the model trained on the small dataset #1 with the thread-relationship being most influential. The thread-relationship is the sole influential form of metadata, when the HMDP is

trained on the larger datasets #2 and #3.

5.2.5 Topic labeling

Topic words	#1	#2	#3	#4
wall border illeg build would hire work built employ mexico	border wall	build wall	build border wall	great wall china
white black student union organ group racist supremacist colleg univers	white student	white student	white student union	white student union
right ralli protest assault peopl free speech troubl person caus	speech right	trump rally	free speech right	going trump rally
black white live matter peopl dont racist say privileg race	black white	black lives	black lives matter	black lives matter
illeg alien tax money billion feder employ state dollar paid	illegal aliens	state local	illegal aliens cost	state local level
muslim islam kill christian religion amal europ clooney sharia murder	christian islam	non muslims	muslim horde amal	5 billion muslims
women man men woman use bathroom room rape bruce transgend	women men	mens room	man wearing women	according police report
liber lie media left conserv truth brain dead wing agenda	media lie	left wing	liberal left agenda	fit racial narrative

Table 8: Topic labels for HMDP topics infered on dataset #3. The methods #1 and #3 used the approach outlined in subsection 4.3 where the intersection of bi- / trigram and top topic words by topic word distribution was maximized. Columns #2 and #4 equal the most frequent bi- and trigram after stop word removal which is used as a baseline.

The presented algorithm often yields descriptive labels with a sufficient amount of bi- and trigrams. Upon manual inspection, the labels generated by this approach are fairly similar to the labels found by the baseline approach using the most frequent bi- or trigram after stop word removal. Nevertheless, the baseline approach more often results in insensible labels such as "great wall china" in the first or "according police report" in the row before the last. Sometimes, however, the labels found by the approach outlined in subsection 4.3 are also not completely descriptive of the topic such as in the row before the last where the labels indicate a topic about men and women and possibly clothing but fail to highlight that it is about the issue of transgender people and bathroom laws. Furthermore, noise in comments reduces the descriptiveness of the labels. Trigram labels appear to be more often descriptive than bigram labels, such as in the second row where "white student" fails to outline the problem of student unions which "white student union" does. Furthermore, stop

word removal seems to improve the quality of approaches #2 and #4 significantly but did not seem to have a strong effect on #1 and #3.

5.3 Discussion

The results show that the good results reported for the MCL version with seven similarity measures as reported in [1] could not be replicated. The thread-relationship of two comments was too influential and the clustering structure nearly mirrored the thread structure. Two differences between [1] and the presented thesis conceivably contribute to the different results. First and foremost, the domain of the works is different. While [1] targets the summarization of comments under a single article, this thesis aims to summarize comments under multiple articles. In the latter the conversational structure is broader, because topics not only overlap within comments of one article but of multiple articles. It is conceivable that a strong influence of the thread structure on clustering yields good results for summarizing comments under a single article, especially when threads under an article target fairly distinct topics, but less so for multiple articles. For multiple articles, a thread-heavy grouping appears unnatural as topics overlap across articles. On a different note, the regression model in this thesis differs from the one in [1]. Targets were obtained by a gold standard with target values of 0 and 1 and not by scraping comments quoting equal passages in an article with target values 0 for negative and in the range of 0.5 and 1 for positive samples. This might have contributed to a larger thread influence than in [1]. With a smaller thread-influence, as in the MCL which uses only cosine similarity and the thread relationship with weights of 1.7 and 0.1, the MCL resulted in a more natural grouping. In this setting it was compared to LDA and HMDP. Here, we could reproduce the results of [1], where the MCL outperformed LDA. Nevertheless, the MCL was surpassed in terms of F-measure and Precision by the HMDP. LDA was even surpassed across all measures. This comes to no surprise, since the context-aware generative model of the HMDP provides sounder model for comment creation, which is heavily influenced by context, than LDA. The improved results show the benefit of modeling context. The fact that the MCL, where such is indirectly included in the Markov graph build-up of the MCL, is able to outperform LDA corroborates this. The high precision of the HMDP indicates a higher topic separation compared to the other models. Especially, when compared to the MCL on the entire dataset #3 this becomes apparent. MCL has a very high recall, but an observation of the cluster structure reveals nearly 75% of the comments being clustered in the two largest clusters. This indicates that the MCL is less able to separate topics on a large set of comments. This might be explained by the effects of data sparsity in comments on similarity measurement.

As far as the approach chosen for evaluation is concerned, it is apparent that clustering an entire dataset but only measuring performance on an annotated subset is indicative. Still, it is limitated and requires thorough manual observation. A drawback of the method is the differing number of clusters between gold standard and

inferred grouping, which likely results in a higher amount of false negatives. Therefore, using $F_{0.5}$-measure to emphasize precision over recall appears reasonable.

To conclude the model comparison, our results indicate that the HMDP provides the best performing model for the clustering of news article comments with an unknown number of discussed topics, for which modeling context was shown to be benificial. Furthermore, annotating a subset of a large dataset appears viable when results are manually observed, as well.

In regards to topic model evaluation, the fact that BCubed-metrics clearly show a better performance of HMDP than LDA while Perplexity does not, highlights the restricted sufficiency held-out likelihood measurements for topic modeling evaluation. This has also been reported in studies such as [55] and is an important conclusion for future work. Additional evaluation means than Perplexity, such as gold standard-based evaluation or user involvement as presented in [55], should be considered.

Coming back to the HMDP, the evaluation of metadata influence showed that the thread-relationship was most influential. This supports that its strong influence in the MCL as presented in [1] can be beneficial when summarizing a single article. Moreover, it supports the validity of the assumption that comments within a thread share a common topic. However, it is surprising that article-relationship and timestamp had no influence when the large corpus was modeled. For timeline influence, an explanation might be that there are too many overlapping topics at a certain time for the HMDP to find borders between context groups, especially in a broad topic domain. In a narrower topic domain, such as the small dataset #1, the timely evolution of topics might be more substantial which is indicated by its associated context influence. The expected absence of cycles was corroborated. The significantly low influence of a comments article relationship might be explained two-fold. On the one hand, there might be too many articles which are closely related in terms of their topic to infer differences between comments from different articles. On the other hand, comments under an article might talk about too many different topics to be a sensible grouping. This would also support the idea that users bring in new information from related sources such as other articles or comments [2].

Lastly, it could be outlined that our topic labeling approach outlined in subsection 4.3 can produce descriptive labels when presented with a sufficient amount of data. However, the expressiveness of our results in cluster labeling is limited, since a formal evaluation of the labeling algorithm was not conducted.

6 Summary Generation

6.1 Ranking & Selection

As in [2] and [1], clusters are selected by the number of comments they contain. Thus, the scoring function is defined as $f(C^{t_i}) = |C^{t_i}|$, i.e. assigns each cluster the number of its comments. The ten largest clusters are selected for summary.

Comments are selected as a summary of each selected cluster. The objective of ranking comments is to find the comments which represent the topic of the cluster concisely. Thus, a ranking mechanism should rank comments highest which are closest to the topic of the cluster without being redundant. As Maximal Marginal Relevance (MMR) [45] aims for both, it is selected for this thesis. MMR in the context of the presented thesis is defined according to [45].

Definition 6.1. Maximal Marginal Relevance Let $C^{t_i} = \{c \in C | t_i = \text{argmax}_{t \in T} P(t|c)\}$ be a set of comments sharing a topic $t_i \in T$, where T denotes the set of all topics. Let S be a set of already selected comments, q denote a set of words representing t_i and s_1 and s_2 be two similarity measures. Then MMR is defined as:

$$\text{MMR} = \underset{c_i \in C^{t_i} \setminus S}{\text{argmax}} \left[\lambda \left(s_1(c_i, q) - (1 - \lambda) \max_{c_j \in S} s_2(c_i, c_j) \right) \right] \tag{74}$$

Here, λ decides about the diversity of the ranking effort [45]. It is easy to see that for $\lambda = 1$, the ranking is entirely based on similarity to the query. A larger λ possibly returns more redundant information close to the query, while a smaller λ returns a sample of information around the query [45]. That is, a smaller λ returns a more diverse set of comments while a larger λ returns comments very close to the topic representation q.

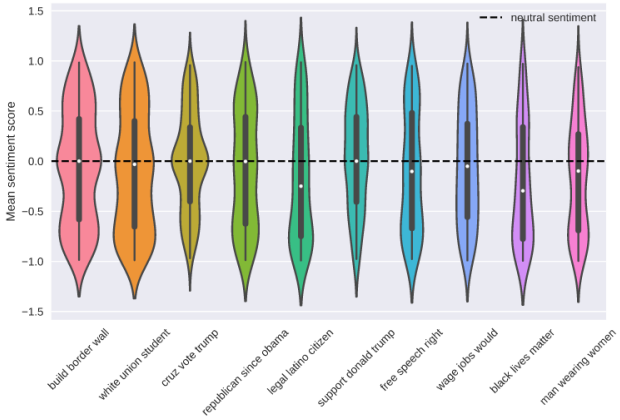
In this thesis, we choose $\lambda = 0.8$ as reported suitable for comments in [2]. Furthermore, we choose cosine similarity of TF-IDF vector representations for both similarity measures. For topic representation, the most likely words by topic-word distribution of a topic model are used, because it assigns high probabilities to words likely to be present in a document when it is known to talk about the topic. Thus, such words are descriptive of the topic. Again, for the MCL such a distribution does not exist but a topic model can be trained on the comments (and articles) of each topic cluster found by MCL with a topic number of 1 to obtain a topic-word distribution. In the end, highest scored comments of positive and of negative sentiment are chosen as a summary of each selected topic cluster.

6.2 Sentiment analysis

Selection requires the sentiment of each considered comment to be known. Therefore, polarity of comments is calculated using the pretrained VADER [80] sentiment

analysis model of NLTK. Each comment is assigned a valence score indicating positive, neutral or negative sentiment. For more details on sentiment analysis and the VADER model, the reader is kindly referred to [80].

6.3 Visualization



	positive	negative
build border wall	You can fix the immigration issue tomorrow, enforce E-Verify, and imprison business owners who knowingly hire illegals. Work would dry up for illegals, and they would resort to crime, or go home, to pay their bills. The fence is a great idea. The concept that a wall doesn't work is silly. It works great for prisons, should we have prisons without walls? It's just that a wall doesn't work in a vacuum, you need officers patrolling that wall, motion sensors, cameras, ect. This is 2016, we can patrol our border.	I'm tired of hearing about "this new study and that new study" about illegal immigration and the border. I live on the border and spend lots of time on BOTH sides. I am Mexican (legal) and have lots of friends and relatives just across the border. Those idiots who conduct these so called "studies" have NO CLUE and certainly don't know as much about the border as those of us who live here and are in Mexico on a regular basis. The fact is that a good, well planned wall WILL be very effective. The wall between Tijuana and San Ysidro (just south of San Diego) has already proved that. The border wall and related technology south of Yuma, Arizona is also quite effective. These do not stop ALL illegal aliens, but they DO stop the vast majority of them. In my opinion, the Trump wall WILL substantially reduce the flow of illegal aliens. If the wall is built and the U.S. Federal Government institutes E-verify, ends sanctuary cities, fines employers who hire illegal aliens and ICE deports any illegal alien who is identified on a day-to-day basis, the illegal alien problem will slowly but surely disappear. Also, the Feds need to STOP letting the so-called "Central American" refugee children\families enter and stay in the USA. The whole thing is a SCAM.
white union student	black student unions aren't racist organizations.	Why is it OK for colleges to have a Black Student Union, but a White Student Union is called racist?
cruz vote trump	Once again, "demolishes" as the colorful verb to show the illogical bias this writer has against Trump. Trump absolutely murdered everyone in Florida, an won 99 delegates. Cruz took a state that he was prime to take already, and got 36 delegates. Now we have New York coming up on the 19th, and Trump is more than primed to take all 95 delegates. Cruz supporters, or rather anti-trump crowds, are just grasping at straws at this point. They don't even realize a big chunk of people voting for Cruz are just anti-trump and give not one #$%$ about Cruz. Forcing him into the nomination for the general election is just saying they want to lose. The same can be said for any other candidate. Trump will most certainly run in the general election, GOP or third party, and he has taken away enough GOP voters to his side in order to cripple their chance to have any chance in the 2016 election. They have two choices, Trump or Hillary. That's all the choice they ever had. They do not have the voters to elect someone class president at this stage in the game.	The problem with your statement is that polls actually show only 25% of Sanders supporters won't vote for Clinton and a much higher percentage of GOP voters will refuse to support trump. The numbers are meaningless anyway because loud whiners always cry when there candidate loses and the overwhelming majority end up voting party line anyway because the alternative in a general election setting is extremely unappealing. You are a Trump troll obviously.

Figure 17: A sample summary of dataset #3 created using HMDP, the oulined labeling and MMR with $\lambda = 0.8$. The table is cut off for brevity; the entire page shows all 10 selected topics.

In accordance with the results of a study of summary design for online debates carried out by Sanchan et al. [51], a combination of chart and side-by-side summary in

the form of a table is chosen. We choose the violin plot as chart, since it combines the information of a boxplot with a probability density estimate.The distribution of sentiment across comments is plotted with the chart indicating how many comments were assigned the topic in general and by polarity. Since all selected topics are plotted against each other and the violin areas are scaled by the number of contained comments, this enables a quick comparison between the significance of topics. Furthermore, one can get a brief overview over the overall reception of it. For each topic, the comments selected as outlined in the previous subsection are shown in a table. This allows for a quick overview of positive and negative aspects.

7 Conclusion & Future Work

In the presented thesis, we studied the problem of multi-article news comment summarization and focused on the challenging problem of clustering comments by topic, which is an essential component of many extractive summarization systems. To the best of the author's knowledge, it is novel in summarizing comments under multiple articles in a topic-driven manner. For the task of topic clustering, three models were compared. The Hierarchical multi-Dirichlet Process (HMDP) topic model was introduced, because it allows context inclusion to tackle data sparsity in comments. It was compared to the Markov Cluster Algorithm (MCL) using the parametric topic model Latent Dirichlet Allocation (LDA) as baseline.

Evaluation was restricted to topic clustering and carried out based on a gold-standard with comments grouped by topic. A process to form grouped gold standards based on a number of seed comments to which randomly sampled comments are clustered was outlined. In accordance to this process, 100 of 6195 comments in a used dataset were annotated. This allowed an evaluation using a large corpus without complete annotation. Nevertheless, this approach is limited due to a differing number of clusters in gold standard and clustered corpus. To counteract, precision was emphasized over recall and the clustering structure manually observed. It turned out that the HMDP performed best by F-measure, followed by the MCL and LDA when using both only the gold standard and the entire dataset for training. Moreover, it provided the most sensible clustering structure. The thesis that context inclusion can improve topic modeling and clustering in sparse comments is, thus, supported. Three different types of contexts were evaluated, namely the thread-relationship, article-relationship and timestamp of a comment. The thread-relationship was most influential in topic modeling as shown by an inspection of the context weights inferred by the HMPD model. Furthermore, our evaluation highlighted drawbacks of held-out likelihood measurements for topic models, since Perplexity did not show any performance differences between HMDP and LDA, while gold standard-based evaluation showed significant differences.

An unsupervised approach to topic clusters labeling was outlined which utilizes n-grams extracted from the comments and their importance in accordance to the topic-word distribution of topic models. With sufficient data, this method yielded descriptive labels which, nevertheless, was not validated with a formal evaluation. Furthermore, the ranking algorithm Maximal Marginal Relevance was used to maximize topic-relevance of summary comments. The summary generation results in a violin plot of sentiment and the highest ranked comments of negative and positive sentiment extracted for each of the most important topics.

The results of this thesis open many possibilities for future work. The steps of ranking, cluster labeling and visualization could be evaluated formally for the summarization of comments of multiple articles, which was not included in this thesis. Furthermore, methods of opinion summarization were included based on sentiment

analysis and can be explored further. Comment topics typically have a large temporal nature, where opinion evolves over time. This can be included included in a summary as is indicated in Appendix. The applicability of the results is generally not restricted to news article comments, as well. It would be interesting to see, how they translate to other sorts of textual data in context such as Tweets. A summary, where multiple social input streams are combined would be an interesting challenge, as well.

The topic clustering methods of this thesis can be developed further, as well. New context spaces can be explored with the HMDP such as a reply-relationship or geospatial context spaces of e.g. geographically close comments, which could not be included based on this thesis's dataset. Moreover, hyperparameter optimization and an optimization of the number of context clusters for timely and geospatial contexts with e.g. an Infinite Gaussian mixture model [32] can be investigated. The MCL can be optimized, as well, by tackling data sparsity in the similarity measurement between comments by including external knowledge bases, for example.

[32] as recommended by Kling in `https://github.com/ckling/promoss`

References

[1] A. Aker, E. Kurtic, A. R. Balamurali, M. L. Paramita, E. Barker, M. Hepple, and R. J. Gaizauskas, "A graph-based approach to topic clustering for online comments to news," in *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings* (N. Ferro, F. Crestani, M. Moens, J. Mothe, F. Silvestri, G. M. D. Nunzio, C. Hauff, and G. Silvello, eds.), vol. 9626 of *Lecture Notes in Computer Science*, pp. 15–29, Springer, 2016.

[2] Z. Ma, A. Sun, Q. Yuan, and G. Cong, "Topic-driven reader comments summarization," in *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012* (X. Chen, G. Lebanon, H. Wang, and M. J. Zaki, eds.), pp. 265–274, ACM, 2012.

[3] C. Llewellyn, C. Grover, and J. Oberlander, "Summarizing newspaper comments," in *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* (E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, eds.), The AAAI Press, 2014.

[4] A. Edmunds and A. Morris, "The problem of information overload in business organisations: a review of the literature," *Int J. Information Management*, vol. 20, pp. 17–28, 2000.

[5] E. Khabiri, J. Caverlee, and C. Hsu, "Summarizing user-contributed comments," in *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011* (L. A. Adamic, R. A. Baeza-Yates, and S. Counts, eds.), The AAAI Press, 2011.

[6] B. Chen, J. Guo, B. L. Tseng, and J. Yang, "User reputation in a comment rating environment," in *KDD*, pp. 159–167, ACM, 2011.

[7] A. Funk, A. Aker, E. Barker, M. L. Paramita, M. Hepple, and R. J. Gaizauskas, "The SENSEI overview of newspaper readers' comments," in *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings* (J. M. Jose, C. Hauff, I. S. Altingövde, D. Song, D. Albakour, S. N. K. Watt, and J. Tait, eds.), vol. 10193 of *Lecture Notes in Computer Science*, pp. 758–761, 2017.

[8] G. Raveendran and C. L. Clarke, "Lightweight contrastive summarization for news comment mining," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, (New York, NY, USA), pp. 1103–1104, ACM, 2012.

[9] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," 2010.

[10] U. Hahn and I. Mani, "The challenges of automatic summarization," *Computer*, vol. 33, pp. 29–36, Nov. 2000.

[11] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *CoRR*, vol. abs/1109.2128, 2011.

[12] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *CoNLL*, pp. 280–290, ACL, 2016.

[13] D. Das and A. F. T. Martins, "A survey on automatic text summarization," 2007.

[14] G. Carenini and J. C. K. Cheung, "Extractive vs. nlg-based abstractive summarization of evaluative text: The effect of corpus controversiality," in *INLG*, The Association for Computer Linguistics, 2008.

[15] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A graph based approach to abstractive summarization of highly redundant opinions," in *COLING*, pp. 340–348, Tsinghua University Press, 2010.

[16] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *CoRR*, vol. abs/1705.04304, 2017.

[17] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, (New York, NY, USA), pp. 406–407, ACM, 2001.

[18] K. Ježek and J. Steinberger, "Automatic text summarization (the state of the art 2007 and new challenges)."

[19] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 1958.

[20] A. Nenkova and K. R. McKeown, "A survey of text summarization techniques," in *Mining Text Data*, pp. 43–76, Springer, 2012.

[21] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[22] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM J. Res. Dev.*, vol. 1, pp. 309–317, Oct. 1957.

[23] C. C. Kling, *Probabilistic models for context in social media*. PhD thesis, University of Koblenz and Landau, Germany, 2016.

[24] T. Korenius, J. Laurikkala, K. Järvelin, and M. Juhola, "Stemming and lemmatization in the clustering of finnish text documents," in *CIKM*, pp. 625–633, ACM, 2004.

[25] H. P. Edmundson, "New methods in automatic extracting," *J. ACM*, vol. 16, no. 2, pp. 264–285, 1969.

[26] K. M. Svore, L. Vanderwende, and C. J. C. Burges, "Enhancing single-document summarization by combining ranknet and third-party sources," in *EMNLP-CoNLL*, pp. 448–457, ACL, 2007.

[27] U. Khandelwal, "Neural text summarization," 2016.

[28] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *EMNLP*, pp. 379–389, The Association for Computational Linguistics, 2015.

[29] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz, "Multi-document summarization by sentence extraction," in *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, (Stroudsburg, PA, USA), pp. 40–48, Association for Computational Linguistics, 2000.

[30] W. M. Soon, H. T. Ng, and C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational Linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[31] K. McKeown and D. R. Radev, "Generating summaries of multiple news articles," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, (New York, NY, USA), pp. 74–82, ACM, 1995.

[32] D. R. Radev and K. R. McKeown, "Generating natural language summaries from multiple on-line sources," *Comput. Linguist.*, vol. 24, pp. 470–500, Sept. 1998.

[33] D. R. Radev, H. Jing, and M. Budzikowska, "Centroid-based summarization of multiple documents: sentence extraction utility-based evaluation, and user studies," *CoRR*, vol. cs.CL/0005020, 2000.

[34] D. R. Radev, V. Hatzivassiloglou, and K. R. McKeown, "A description of the cidr system as used for tdt-2," 1999.

[35] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, (New York, NY, USA), pp. 299–306, ACM, 2008.

[36] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, (San Francisco, CA, USA), pp. 296–304, Morgan Kaufmann Publishers Inc., 1998.

[37] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, pp. 613–620, Nov. 1975.

[38] R. Arora and B. Ravindran, "Latent dirichlet allocation based multi-document summarization," in *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, AND '08, (New York, NY, USA), pp. 91–97, ACM, 2008.

[39] E. Barker, M. L. Paramita, A. Funk, E. Kurtic, A. Aker, J. Foster, M. Hepple, and R. J. Gaizauskas, "What's the issue here?: Task-based evaluation of reader comment summarization systems," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.* (N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik,

B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, eds.), European Language Resources Association (ELRA), 2016.

[40] Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, "Ranking with recursive neural networks and its application to multi-document summarization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2153–2159, AAAI Press, 2015.

[41] Y. Zhang, M. Er, R. Zhao, and M. Pratama, "Multi-view convolutional neural networks for multi-document extractive summarization," *IEEE TRANSACTIONS ON CYBERNETICS*, vol. PP, 11 2016.

[42] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, "Graph-based neural multi-document summarization," 06 2017.

[43] P. Ren, Z. Chen, Z. Ren, F. Wei, L. Nie, J. Ma, and M. de Rijke, "Sentence relations for extractive summarization with deep neural networks," *ACM Trans. Inf. Syst.*, vol. 36, pp. 39:1–39:32, Apr. 2018.

[44] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: A rating regression approach," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, (New York, NY, USA), pp. 783–792, ACM, 2010.

[45] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 335–336, ACM, 1998.

[46] E. Barker and R. J. Gaizauskas, "Summarizing multi-party argumentative conversations in reader comment on news," in *Proceedings of the Third Workshop on Argument Mining, hosted by the 54th Annual Meeting of the Association for Computational Linguistics, ArgMining@ACL 2016, August 12, Berlin, Germany*, The Association for Computer Linguistics, 2016.

[47] E. Barker, M. L. Paramita, A. Aker, E. Kurtic, M. Hepple, and R. J. Gaizauskas, "The SENSEI annotated corpus: Human summaries of reader comment conversations in on-line news," in *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pp. 42–52, The Association for Computer Linguistics, 2016.

[48] A. Aker, M. L. Paramita, E. Kurtic, A. Funk, E. Barker, M. Hepple, and R. J. Gaizauskas, "Automatic label generation for news comment clusters," in *INLG 2016 - Proceedings of the Ninth International Natural Language Generation Conference, September 5-8, 2016, Edinburgh, UK* (A. Isard, V. Rieser, and D. Gkatzia, eds.), pp. 61–69, The Association for Computer Linguistics, 2016.

[49] S. van Dongen, *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[50] I. Hulpus, C. Hayes, M. Karnstedt, and D. Greene, "Unsupervised graph-based topic labelling using dbpedia," in *WSDM*, pp. 465–474, ACM, 2013.

[51] N. Sanchan, K. Bontcheva, and A. Aker, "Understanding human preferences for summary designs in online debates domain," *Polibits*, vol. 54, pp. 79–85, 2016.

[52] M. Hassel, *Evaluation of Automatic Text Summarization - A practical implementation*. Licentiate thesis, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, May 2004.

[53] M. Zopf, M. Peyrard, and J. Eckle-Kohler, "The next step for multi-document summarization: A heterogeneous multi-genre corpus built with a novel construction approach," in *COLING*, pp. 1535–1545, ACL, 2016.

[54] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. ACL workshop on Text Summarization Branches Out*, p. 10, 2004.

[55] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei, "Reading tea leaves: How humans interpret topic models," in *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds.), pp. 288–296, Curran Associates, Inc., 2009.

[56] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, 2004.

[57] S. P. Crain, K. Zhou, S.-H. Yang, and H. Zha, *Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond*, pp. 129–161. Boston, MA: Springer US, 2012.

[58] D. M. Blei, "Probabilistic topic models," *Commun. ACM*, vol. 55, pp. 77–84, Apr. 2012.

[59] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[60] A. Enright, S. Van Dongen, and C. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic acids research*, vol. 30, pp. 1575–84, 05 2002.

[61] R. Rockenfeller, "University of Koblenz-Landau, Lecture notes: Stochastik," 2018.

[62] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[63] H. M. Wallach, "Topic modeling: beyond bag-of-words," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006* (W. W. Cohen and A. Moore, eds.), vol. 148 of *ACM International Conference Proceeding Series*, pp. 977–984, ACM, 2006.

[64] H. M. Wallach, *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

[65] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the dirichlet distribution and related processes," tech. rep., 2010.

[66] T. S. Ferguson, "A bayesian analysis of some non-parametric problems," *The Annals of Statistics*, vol. 1, 03 1973.

[67] D. Aldous, "Exchangeability and related topics," in *École d'Été St Flour 1983*, pp. 1–198, Springer-Verlag, 1985. Lecture Notes in Math. 1117.

[68] E. Xing, "Carnegie Mellon University, Lecture notes: Probabilistic Graphical Models, Lecture 20 March 31, 2014: Dirichlet Process and Hierarchical DP," 2014.

[69] C. C. Kling, J. Kunegis, S. Sizov, and S. Staab, "Detecting non-gaussian geographical topics in tagged photo collections," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, (New York, NY, USA), pp. 603–612, ACM, 2014.

[70] L. Lovász, "Random walks on graphs: A survey," in *Combinatorics, Paul Erdős is Eighty* (D. Miklós, V. T. Sós, and T. Szőnyi, eds.), vol. 2, pp. 353–398, Budapest: János Bolyai Mathematical Society, 1996.

[71] T. Götz, "University of Koblenz-Landau, Lecture notes: Numerik," 2017/2018.

[72] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, "Improving topic models with latent feature word representations," *TACL*, vol. 3, pp. 299–313, 2015.

[73] M.-T. Nguyen, C.-X. Tran, V. Tran, and M.-L. Nguyen, "Solscsum: A linked sentence-comment dataset for social context summarization," 10 2016.

[74] C. Napoles, J. Tetreault, E. Rosata, B. Provenzale, and A. Pappu, "Finding good conversations online: The yahoo news annotated comments corpus," in *Proceedings of The 11th Linguistic Annotation Workshop*, (Valencia, Spain), pp. 13–23, Association for Computational Linguistics, April 2017.

[75] M. F. Porter, "Readings in information retrieval," ch. An Algorithm for Suffix Stripping, pp. 313–316, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[76] E. Jones, T. Oliphant, P. Peterson, *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [http://www.scipy.org/; accessed 2019/28/02].

[77] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Inf. Retr.*, vol. 12, no. 4, pp. 461–486, 2009.

[78] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chains," in *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pp. 563–566, 1998.

[79] N. Chinchor, "Muc-4 evaluation metrics," in *Proceedings of the 4th Conference on Message Understanding*, MUC4 '92, (Stroudsburg, PA, USA), pp. 22–29, Association for Computational Linguistics, 1992.

[80] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* (E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, eds.), The AAAI Press, 2014.

8 Appendix

A Appendix

A.1 Variables of the HMDP topic model

As adopted from Kling [23].

Table 9: Variables of the hierarchical multi-Dirichlet process topic model [23].

M	Number of documents
N_m	Number of words in document m
F	Number of context spaces
C_f	Number of context clusters for context space f
A_f	Number of context groups with common parents in context space f
β_0	Concentration parameter of the topic prior
H	Space of all possible topic multinomials, with Dirichlet prior β_0
G_0	Global measure on the topic space (includes topics and their weight)
G^c_{fj}	Measure on the topic space for cluster j of context space f
G^d_m	Document-specific measure on the topic space
γ	Scaling parameter of the global Dirichlet process (DP)
α_0	Scaling parameter for the context cluster DPs
α_1	Scaling parameter for the document MDPs
ϕ_{mn}	Topic drawn for w_{mn}
w_{mn}	Word n for document m
g_{mf}	Context group of document m in context space f
ζ	Mixing weights for the context spaces
η_{fi}	Mixing weights for the group i in context space f
ε	Concentration parameter of the Dirichlet prior on ζ
δ_f	Concentration parameter of the Dirichlet prior on η in context space f

A.2 Annotations of the YNACC subset annotated by experts

Table 10: The annotations for each comment in the YNACC subset annotated by experts as outlined in [74] and available under `https://github.com/cnap/ynacc`.

sdid	Subdialog id, identifies the comment thread of a comment.
commentindex	Position of the comment in the thread.m
headline	Headline of the article the comment was issued under.
url	URL of the article.
guid	Encrypted user id.
commentid	Comment id.
timestamp	Timestamp of when the comment was issued.
thumbs-up	No. of upvotes.
thumbs-down	No. of downvotes.
text	Comment text.
parentid	ID of the first comment in the subdialogue.
constructiveclass	Whether a comment was constructive or not.
`sd_agreement`	(Dis-) agreement throughout the subdialogue.
`sd_type`	Type of converstation in the subdialogue.
sentiment	Sentiment label of the comment.
tone	Tone of the comment.
commentagreement	(Dis-) agreement of the comment.
topic	Whether the comment was on- or off-topic
intendedaudience	Broadcast or reply.
persuasiveness	Whether the comment was persuasive or not.

A.3 Proof of claim in Section 4

Proof. Let $M, K > 0$ and $N = M + K$.

$$\binom{N}{2} = \binom{M+K}{2} = \frac{1}{2}(M+K)(M+K-1) \tag{75}$$

$$= \frac{1}{2}(M^2 + K^2 + 2MK - M - K) \tag{76}$$

$$> \frac{1}{2}(M^2 + K^2 - M - K) = \frac{1}{2}\left(M(M-1) + K(K-1)\right) \tag{77}$$

$$= \binom{M}{2} + \binom{K}{2} \tag{78}$$

□

A.4 Proof of claim in Section 4.3

Proof. Let X be a discrete random variable denoting a draw of a word from the set of words W, the sample space, denoting the words representing topic t. Then $X \sim \phi$ is distributed according to the multinomial distribution ϕ which denotes the topic-word distribution of the topic model. Let L be the set of labels and $x \in L$ be a label which solely has the largest intersection $x \cap W$ and thus $x = \text{argmax}_{l \in L} |l \cap W|$. Let $y \in L$ be a label with $|x \cap W| > |y \cap W| \Leftrightarrow |x \cap W| = |y \cap W| + a$ with $a \geq 1$. Suppose y was chosen as a label, then:

$$y = \underset{l \in L}{\text{argmax}} \left(|l \cap W| + \sum_{w \in (l \cap W)} P(w|\phi) \right) \tag{79}$$

and thus:

$$|y \cap W| + \sum_{w \in (y \cap W)} P(w|\phi) \geq |x \cap W| + \sum_{w \in (x \cap W)} P(w|\phi) \tag{80}$$

$$\Leftrightarrow y \cap W| + \sum_{w \in (y \cap W)} P(w|\phi) \geq a + |y \cap W| + \sum_{w \in (x \cap W)} P(w|\phi) \tag{81}$$

$$\Leftrightarrow \sum_{w \in (y \cap W)} P(w|\phi) \geq a + \sum_{w \in (x \cap W)} P(w|\phi) \tag{82}$$

$$\tag{83}$$

However, as $a > 1 \wedge 1 \geq \sum_{w \in (x \cap W)} P(w|\phi) > 0$ [33] this would be equivalent to:

$$\sum_{w \in (y \cap W)} P(w|\phi) > 1 \tag{84}$$

[33] It is strictly greater 0 as otherwise the n-gram x could not have occured under the model with topic-word distribution ϕ.

which is trivially false as $\sum_{x \in W} P(X = x) = P(W) = 1$. Therefore it holds:

$$|x \cap W| + \sum_{w \in (x \cap W)} P(w|\phi) > |y \cap W| + \sum_{w \in (y \cap W)} P(w|\phi) \tag{85}$$

Thus, y can not have been the chosen label if it fulfills the equation. Instead, x would be the chosen label and every label which fulfills the equation

$$x = \underset{l \in L}{\operatorname{argmax}} \left(|l \cap W| + \sum_{w \in (l \cap W)} P(w|\phi) \right) \tag{86}$$

fulfills

$$x = \underset{l \in L}{\operatorname{argmax}} |l \cap W|. \tag{87}$$

Therefore, the labels found by the outlined algorithm fulfill said equation. □

A.5 Topic evolution over time

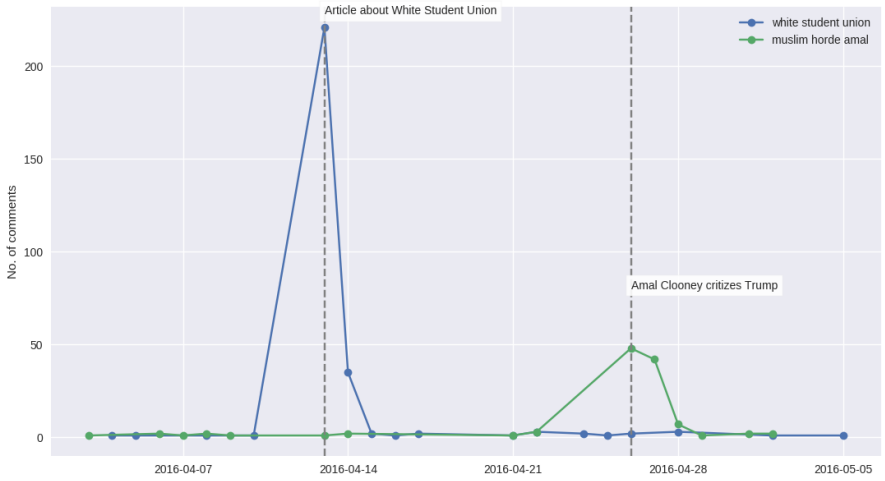


Figure 18: Evolution of two topics of dataset #3 over time with notable events marked.

This figure illustrates how the timely evolution of topics can be visualized. This is especially interesting, when there are multiple notable events and articles on a topic over time.