ERASMUS UNIVERSITEIT ROTTERDAM

Erasmus University Rotterdam
Erasmus School of Economics

Master Thesis
Econometrics and Management Science
Business Analytics and Quantitative Marketing

An evaluation of semi-supervised topic modeling techniques for customer feedback understanding

Author:
Máté Váradi

Supervisor:
Prof. Dr. Richard Paap

Student number:
495556

Second assessor:
Prof. Dr. D. Fok

October 8, 2020

Abstract

Semi-supervised topic models are text mining tools that can utilize domain knowledge provided by the user to uncover the underlying themes in document collections. We apply and compare the state-of-the-art semi-supervised topic modeling techniques on corpora of short customer feedback. The methods included in the comparison are Latent Dirichlet Allocation (LDA) (Blei et al., 2003), Biterm Topic Model (BTM) (Yan et al., 2013a), SeededLDA (Jagarlamudi et al., 2012), Dirichlet Forest Latent Dirichlet Allocation (DFLDA) (Andrzejewski et al., 2009), and (Anchored) Correlation Explanation (CorEx) (Gallagher et al., 2017). We evaluate the algorithms' capability to correctly classify documents using a human-annotated subset of the customer feedback dataset, applying the F-score and Normalized Mutual Information metrics. We assess the interpretability of the resulting topics using topic coherence metrics suitable for short text; Normalized Pointwise Mutual Information (Lau et al., 2014) and the Google Titles Match score (Newman et al., 2010). We find that CorEx achieves the best performance in most aspects.

Keywords: topic modeling, document clustering, short text, Natural Language Processing, semi-supervised learning, Latent Dirichlet Allocation, topic coherence, clustering evaluation, Gibbs sampling

Contents

Acronyms

ARI Adjusted Rand Index.

BTM Biterm Topic Model.

CorEx (Anchored) Correlation Explanation.

DFLDA Dirichlet Forest Latent Dirichlet Allocation.

GMM Gaussian Mixture Model.

GSDMM Gibbs Sampling Dirichlet Mixture Model.

GTM Google Titles Match.

LDA Latent Dirichlet Allocation.

LSA Latent Semantic Analysis.

MI Mutual Information.

NLP Natural Language Processing.

NMF Non-negative Matrix Factorization.

NMI Normalized Mutual Information.

NPMI Normalized Pointwise Mutual Information.

PLSA Probabilistic Latent Semantic Analysis.

PMI Pointwise Mutual Information.

RI Rand Index.

SLDA Supervised Latent Dirichlet Allocation.

TF-IDF Term Frequency-Inverse Document Frequency.

W2V Word2Vec.

1 Introduction

In our online world, we are surrounded by short texts. It is the dominant data type on social media websites like Twitter, online Q&A sites, and so on. Customers often voice their opinion on such forums or give written feedback on products and services through online reviews. To leverage this vast amount of information we need tools that help us organize or summarize textual data automatically. An example of such a tool is topic modeling, a widely used, powerful text mining technique. Topic models are able to discover the main themes in a large and unstructured set of documents and categorize them according to the discovered themes (Blei, 2012). The field of topic modeling is constantly evolving, and it has applications in a diverse set of fields, such as software engineering, political science, and linguistics (Jelodar et al., 2019). Topic models can be particularly useful in marketing research, given their wide applicability and the amount of online data available. For example, brands can use Twitter and the service industry can use sites like Yelp to analyze reviews written about their products or services. According to Reisenbichler and Reutterer (2019), there is an ongoing trend among marketing scholars and practitioners to apply machine learning methods, such as topic models.

Among the numerous methods for topic modeling, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) is one of the most popular. However, the effectiveness of LDA and other widely used topic models is reduced when applying them on documents of shorter length (Yan et al., 2013b). Many applications require the use of short text, such as content analysis for content recommendation and event tracking on blogs, and tweets (Jipeng et al., 2019). As a consequence, some approaches are designed to deal with short documents (Lu et al., 2017; Sridhar, 2015; Yan et al., 2013a,b; Yin and Wang, 2014).

Topic modeling is an unsupervised learning problem, akin to clustering. This unsupervised nature of topic models can impede their great potential to help users understand text documents as even the state-of-the-art techniques are prone to creating topics that are not meaningful (Chang et al., 2009) or to only capture the most dominant topics of a corpus while ignoring several others. Some users of topic modeling methods may have prior expectations about the topics that appear in the corpora. For instance, Jagarlamudi et al. (2012) find that when applied to a corpus of papers from the Conference of Neural Information Processing Systems, LDA was unable to detect topics about Brain Imaging, Hardware, or Cognitive Science. Even though these areas were underrepresented in the data, such conference papers did exist. Semi-supervised topic models allow users to apply their domain knowledge to nudge the model results in the desired direction (Andrzejewski et al., 2009; Andrzejewski and Zhu, 2009; Gallagher et al., 2017; Jagarlamudi et al., 2012; Yang et al., 2015; Zhai et al., 2011). Most of the semi-supervised methods are relatively recent and their applications are also limited. Some of them may have never been tested on short text. Additionally, to our best knowledge, a thorough comparison of semi-supervised topic models has not been done yet.

Another challenge that arises from the unsupervised nature of topic models is their quantitative evaluation

for model selection. In some applications document labels are available, e.g. the topics of the documents are annotated by humans or the documents can be linked to certain categories such as hashtags in the case of tweets or forum names in the case of Reddit. Where labeled data is at hand, topic models can be compared on the task of document classification. In this case, standard metrics also used in external clustering evaluation, such as accuracy and F-scores can be applied. Alternatively, information-theoretic measures such as Normalized Mutual Information (Bouma, 2009) and Adjusted Rand Index (Hubert and Arabie, 1985) are often used. When only unlabeled data is available, perplexity or other log-likelihood based measures and various metrics from the family of coherence scores can be used. Perplexity (Wallach et al., 2009b) measures how the model performs on unseen data. Topic coherence scores measure the degree of semantic similarity between words in a topic. They are often considered as a way to measure the quality of topics.

The objective of this research is to find a suitable method to deal with short documents, which is also able to include the user's domain knowledge through a semi-supervised learning framework. To select between candidate methods, we also need to overcome the challenge of evaluating topic models. Thus, in this thesis we evaluate and compare some of the state-of-the-art unsupervised and semi-supervised topic modeling techniques on short text. The research problem has been supplied by SKIM, a marketing research agency. SKIM's clients wish to gain insights from textual data more and more often. This frequently involves having to understand short text. For this research, we use data that comes from user feedback given to a web service provider. The domain knowledge we use to guide the semi-supervised models is established by SKIM and the client and is in the form of predefined groups of terms. They expect these terms to form separate topics among the results of a topic model. The client also wishes to have information on the sizes of the resulting topics, or how frequently each topic appears in the corpus as this can be an input for strategic decisions. These details lead to the following research question:

What is the best method for SKIM to model the topics of short customer feedback?

To find relevant methods, we explore the literature for candidates, then we narrow down our search results to a select few topic models. Among our five selected models are the most widely-used topic model, LDA and a model designed for short text, the Biterm Topic Model (BTM). These two models serve as baselines to the semi-supervised methods. We run three of the most prominent semi-supervised topic models: Dirichlet Forest LDA (Andrzejewski et al., 2009), SeededLDA (Jagarlamudi et al., 2012) and Anchored Correlation Explanation (Gallagher et al., 2017).

The word *best* in the research question implies that evaluation is an important part of the research. We carefully select suitable evaluation metrics that are able to focus on different aspects of the models and to differentiate between them. To evaluate the models' performance in correctly labeling topics, we use a human-annotated subset of the user feedback data and we apply to the topic model results two variants of Normalized Mutual Information and the F_1 score for overlapping clustering. To assess topic quality and interpretability we choose the normalized variant of Pointwise Mutual Information (Newman et al., 2010)

proposed by Lau et al. (2014) and the Google Titles Match score from Newman et al. (2010). We also examine the resulting topic sizes. Lastly, it is important, that the best method is suitable for commercial applications. Therefore ease of use is also taken into account when selecting among methods. We find that based on these evaluation criteria the Anchored Correlation Explanation technique performs best.

The main contributions are twofold. First, this research provides a suitable solution for SKIM which enables them to use topic modeling techniques for text mining applications, such as understanding customer feedback. Second, we present a thorough evaluation and comparison of some of the state-of-the-art semi-supervised topic models. To our best knowledge, the set of techniques used in this thesis have not been compared before. Additionally, we apply the models to real-life data, coming from customer feedback, which is different from the standard datasets used for testing topic models. The shortness of the documents and the relative noisiness of the data may reveal characteristics of the methods that would otherwise remain undiscovered. Lastly, to assess the performance of topic models that create non-disjoint topic clusters, we introduce a novel way of calculating Normalized Mutual Information for overlapping clustering evaluation.

The rest of this thesis is structured as follows: Section 2 is a literature review, focusing on four areas relevant to this application: topic modeling in general, short text topic modeling, semi-supervised topic modeling, and topic modeling evaluation. Section 3 provides some detail about the document collection and the domain knowledge used for topic modeling. We introduce the five topic models and our evaluation metrics in Section 4. In Section 5, along with a qualitative evaluation of the model results, we present a comparison of the five models in terms of clustering performance and topic coherence, using the the selected evaluation metrics. Section 6 offers discussion points and concluding thoughts.

2 Literature

The literature review we present in this section is divided into four subsections. First, we introduce the main ideas of topic modeling. Then, we explore the literature on short text topic models and semi-supervised approaches. Finally, different evaluation approaches and metrics are discussed.

2.1 Overview of topic modeling methods

Latent Semantic Analysis (LSA) (or Latent Semantic Indexing) (Deerwester et al., 1990) is a foundational approach in text mining. When using text mining techniques, the data is often represented in a document-term matrix, where rows correspond to documents of a collection, columns correspond to the unique words of the vocabulary, and the values represent occurrence frequency, such as the term frequency-inverse document frequency (TF-IDF) score. The idea behind LSA is to decompose this matrix into two lower dimensional matrices via Singular Value Decomposition. The two matrices resulting from the decomposition correspond to a document-topic matrix and a topic-term matrix. The resulting topics, however, are difficult to interpret. Another matrix factorization based method is Non-negative Matrix Factorization (NMF) (Xu et al., 2003). Here, the resulting document-topic and topic-term matrices contain only non-negative elements. Therefore, these two matrices' elements respectively represent how much a document is associated with a topic and the degree to which a term belongs to a topic. Unlike LSA, NMF is able to handle overlapping clusters. Probabilistic Latent Semantic Analysis (Hoffmann, 1999) is a more flexible model as it models the probability of document-term co-occurrences as mixtures of Multinomial distributions. Unlike PLSA, the most popular topic model, Latent Dirichlet Allocation (LDA) (Blei, 2012) can generalize to unseen documents. LDA is a Bayesian version of PLSA, as it places a Dirichlet prior on parameters. It is estimated via variational Bayesian inference, Gibbs sampling, or the Expectation-Maximization (EM) algorithm (Jelodar et al., 2019). For LDA we have to assume that the number of topics is known a-priori and that document order does not matter. The order of words within a document and grammar structures also do not matter. This is known as the "bag-of-words" assumption. Many extensions and variations of LDA exist to relax some of its assumptions or to overcome its shortcomings, such as its disregard for word context. Jelodar et al. (2019) give an overview of some of the extensions and applications of LDA.

More recently, a number of approaches arose which use the capabilities of artificial neural networks to determine the distribution of topics. The earliest neural network-based topic model is the Replicated Softmax algorithm, introduced by Hinton and Salakhutdinov (2009). It is a generalization of the Restricted Boltzmann Machine, a two-layer undirected graphical model. Replicated Softmax inspired similar methods, such as DocNADE, a neural autoregressive topic model (Larochelle and Lauly, 2012); TopicRNN, a recurrent neural network-based method (Dieng et al., 2016); and TopicAE, a topic autoencoder (Smatana and Butka, 2019).

Since in our research, we are focusing on semi-supervised methods, the description of the techniques above

mainly serves as a demonstration of the possibilities. However, we do need a baseline method, which disregards both the shortness of the documents and the user's domain knowledge. For this purpose, we believe LDA is the most suitable candidate among the methods described above, as it is widely used and it serves as a starting point for many (semi-supervised) extensions.

2.2 Topic models for short text

In this application, we are dealing with shorter documents, generally below 500 characters. Ambiguous words in short documents are difficult to identify due to limited context. One way to deal with this sparsity is to somehow aggregate small documents, e.g. tweets by their user (Mehrotra et al., 2013). In many applications (such as ours) this is infeasible or impractical. Another way is to make stronger assumptions, for example, to assume that a short document only covers a single topic. The Gibbs Sampling Dirichlet Mixture Model (GSDMM) (Yin and Wang, 2014) is an altered LDA algorithm, which is built on this assumption. An advantage of this model is that unlike most other methods, it can infer the number of clusters automatically. However, according to Yan et al. (2013a) as a result of the assumption that one document may only contain one topic, the model tends to lose flexibility and suffers from overfitting issues. There exist variations of many of the previously mentioned techniques that tackle the sparsity problem in a new way. The Biterm Topic Model (BTM) (Yan et al., 2013a) is one of the most successful examples. This model is built on the idea that topics are groups of correlated words where the correlation is implied by word co-occurrence. BTM is a generative model similar to LDA, but it learns topics by directly modeling the generation of biterms (unordered word-pairs that co-occurred in a short document), and word co-occurrence is determined over the entire corpus. An additional method by Yan et al. (2013b) named TNMF is based on non-negative matrix factorization. TNMF learns topics directly from a term correlation matrix rather than the usual document-term matrix. Their approach is to first learn topics from term correlation data via non-negative matrix factorization, then inferring topic representations of documents by solving a non-negative least-squares problem.

Another technique specifically for short text is the Word2Vec Gaussian Mixture Model (W2V-GMM) by Sridhar (2015). It uses the word2vec word embedding algorithm (Mikolov et al., 2013) to construct a semantic vector space from the corpus, and then clusters words in the resulting vector space using a Gaussian mixture model with as many Gaussian components as topics (Jónsson and Stolee, 2015). Lu et al. (2017) introduce RIBS-TM, a neural network-based short text topic model, similar to BTM. In RIBS-TM biterm generation is preceded by the learning of the relationship between words via a simple recurrent neural network system. Furthermore, biterm construction is altered in order to filter high-frequency words that are not meaningful in topics.

Again, as our focus is on semi-supervised models, we only need to select one method from the options described above. The selected technique will serve as a baseline method which disregards domain knowledge but is able to reliably model short documents. Conveniently, some short text topic models have already been compared in the literature. Jónsson and Stolee (2015) use Twitter data to compare the

above-mentioned BTM, W2V-GMM and LDA-U, a method based on the aggregation of documents by user. In their application, BTM performed best. A thorough evaluation of short text topic models on several datasets is presented by Jipeng et al. (2019). It has no clear winner among the examined methods, but here too, BTM performs reasonably well in various aspects. Therefore, we choose BTM as our second baseline topic model.

2.3 Semi-supervised topic models

In some applications, researchers may want to incorporate domain knowledge to topic modeling or simply want to have some influence on the outcome of the methods. Mcauliffe and Blei (2008) propose a way to add document-level supervision to LDA. In supervised LDA (SLDA) (Mcauliffe and Blei, 2008) a response variable associated with each document, such as a label or an outcome variable is added to LDA. SLDA can be used for supervised tasks, such as prediction, classification, or sentiment analysis.

The method proposed by Andrzejewski and Zhu (2009), z-label LDA, gives the opportunity for the user to provide token-level supervision. That is, the user can predefine terms to only appear in certain topics. Another method by Andrzejewski et al. (2009), Dirichlet Forest LDA (DFLDA), is akin to constrained K-means clustering (Wagstaff et al., 2001). In DFLDA the user can provide must-link and cannot-link connections between word pairs. A must-link encourages the model to include both words with either high or low probability in any particular topic. A cannot-link connection between a pair of words suggests that the two words should have a low probability of coappearing in any topic. This is achieved by replacing the Dirichlet prior of LDA with a Dirichlet Forest prior. Constrained LDA by Zhai et al. (2011) works similarly to DFLDA, but must-links and cannot-links are extracted automatically from the text.

In SeededLDA, as introduced by Jagarlamudi et al. (2012), users can guide the topic model by providing sets of words, called seed words, which they believe to be representative of the topics present in the data. The Anchored Correlation Explanation (CorEx) model by (Gallagher et al., 2017) is an alternative, information-theoretic topic model. It does not rely on LDA's generative assumptions and instead uses an information-theoretic framework to learn topics. CorEx is expected to work better with short documents (Gallagher et al., 2017), which makes this method particularly relevant for our research.

CorEx, SeededLDA and DFLDA are the ones among the above-described methods which utilize user knowledge in a way that suits our application. Therefore, we focus on these three techniques. To our knowledge, a comparison of semi-supervised topic models has not been performed yet.

2.4 Topic model evaluation

According to Blei (2012) the development of evaluation methods "that match how the algorithms are used remains an open direction for topic modeling". The unsupervised nature of the methods makes model selection difficult. A different corpus or a different task may require different topic modeling

approaches. The researcher has to decide which of the many modeling assumptions are important for their goals (Blei, 2012). This subsection introduces the different aspects of topic modeling evaluation and describes relevant evaluation metrics. Furthermore, we include a summary of the metrics used in technical or applied topic modeling papers relevant to our research.

2.4.1 External evaluation

Evaluating the quality of topics resulting from a topic model is similar to the evaluation of clustering results. The availability of document labels makes external evaluation possible. In this case, evaluation is based on the comparison between resulting topics (clusters) and document labels (classes). A large number of different metrics exist, some adapted from classification evaluation, some from information theory. Due to this abundance of measures available we will focus on those that have been applied (recently) to topic modeling evaluation in previous research and those that were specifically designed for topic modeling tasks. Furthermore, since in our application a document may have more than one correct label, we also describe modified metrics that can deal with non-disjoint clustering, although these measures are not typically applied in topic modeling tasks.

One of the simplest metrics that is often used for topic modeling evaluation is purity (Jipeng et al., 2019; Yan et al., 2013a,b). According to Manning et al. (2008), "purity is a measure of the extent to which clusters contain a single class". Its calculation involves summing up the maximum number of data points belonging to one of the ground-truth classes in each cluster, then dividing this sum by the number of datapoints. However, using purity score for evaluation does not work well for a corpus where the distribution of topics is imbalanced. Many other metrics are based on the number of true positives, true negatives, false positives, and false negatives. Rand index (RI) simply measures the proportion of correct decisions made by an algorithm. RI is used by Zhai et al. (2011). One drawback is that here, false positives and false negatives are weighted equally. To correct for this, the Adjusted Rand Index (ARI), a corrected-for-chance version of Rand index (Vinh et al., 2010) can be used. Similarly, the F-measure or F-score, the (weighted) harmonic mean of precision and recall, can also be used to balance the contribution of false negatives in classification problems. ARI is used by Yan et al. (2013a) and Yan et al. (2013b), while the F-measure is used in Jagarlamudi et al. (2012); Fatemi and Safayani (2019) and Nugroho et al. (2015). A metric closely related to RI, but more suitable for classification problems is accuracy. It is used by Xu et al. (2003).

Mutual information (MI) is an information-theoretic metric that measures the information overlap of two random variables. In the case of topic modeling, it gives us the reduction that we get in the entropy of the clustering if we get the true document labels. It can also be looked at as the average pointwise mutual information (PMI). PMI, which refers to single events rather than the average of all possible events, is "a measure of how much the actual probability of a particular co-occurrence of events differs from what it is expected to be based on the probabilities of the individual events and the assumption of

independence" (Bouma, 2009). A closely related metric, used in Jagarlamudi et al. (2012) is the variation of information (VI)[1]. Bouma (2009) introduces normalized variants for both MI and PMI. Normalized Mutual Information (NMI) and Normalized Pointwise Mutual Information (NPMI) are less sensitive to occurrence frequency and are more easily interpretable as they have fixed lower and upper bounds. NPMI is used by Smatana and Butka (2019). NMI is used regularly in the topic modeling literature and is one of the main metrics for evaluating topic model performance (Hong and Davison, 2010; Lu et al., 2011; Nugroho et al., 2015; Mehrotra et al., 2013; Jipeng et al., 2019; Sridhar, 2015; Xu et al., 2003; Yan et al., 2013a,b; Yin and Wang, 2014). Table 1 summarizes where each mentioned metric is used.

Table 1: Overview of the use of clustering metrics in the topic modeling literature

Metric	Used in
Purity	Jipeng et al. (2019); Yan et al. (2013a); Yan et al. (2013b)
Rand Index	Zhai et al. (2011)
Adjusted Rand Index	Yan et al. (2013a); Yan et al. (2013b)
Accuracy	Xu et al. (2003)
F-score	Jagarlamudi et al. (2012)
Variation of Information	Jagarlamudi et al. (2012)
Normalized Mutual Information	Mehrotra et al. (2013); Jipeng et al. (2019); Sridhar (2015); Xu et al. (2003); Yan et al. (2013a), Yan et al. (2013b); Yin and Wang (2014)
Normalized Pointwise Mutual Information	Smatana and Butka (2019)

Topic modeling is different from the most commonly used clustering algorithms in that it results in soft instead of hard clusters. That is, for a given document a topic model may indicate more than one topic where the document is likely to belong. Typically when evaluating topic modeling results, the most probable topic is chosen per document, thereby producing hard clusters. However, this procedure can introduce bias to the evaluation. Furthermore, depending on the application, documents may also have more than one correct label.

In our application, we want to allow both clusters and ground-truth classes to overlap, meaning that a document may belong to more than one category. This needs to be considered when evaluating clustering performance, and thus we need to explore metrics that are able to handle non-disjoint clusters. Some of the above-mentioned evaluation metrics can be extended to be able to this (Amigó et al., 2009). For example, a simple extension for NMI is introduced by McDaid et al. (2011). Moreover, different ways of computing the F-score in the overlapping case are compared in N'Cir et al. (2015). The authors of this study as well as Amigó et al. (2009) conclude that the so-called BCubed metrics are the most suitable in the widest variety of possible scenarios. Bcubed precision and recall (Amigó et al., 2009) (and

[1]See Appendix A.1 for an overview of how the information theoretic concepts used in this thesis relate to each other.

consequently Bcubed F-score) are based on the similarity between clustering and class labels of all pairs of observations. For this reason, however, BCubed metrics are not efficient to compute. Lutov et al. (2019) deal with the efficient calculation of overlapping clustering metrics for large datasets.

2.4.2 Perplexity

Perplexity is a standard criterion for the comparison of different probabilistic topic models. It is generally used to find the optimal number of latent topics in LDA. Perplexity is calculated as the inverse of the geometric mean per-word likelihood (Newman et al., 2010). Details of the calculation are given by Wallach et al. (2009b). Perplexity can be interpreted as a measure of how well a probabilistic topic model fits a collection of unseen documents. However, Chang et al. (2009) showed that good model performance according to perplexity does not necessarily result in topics that are meaningful or interpretable to humans. In their study, which was the first to involve human-evaluation in topic models, subjects were asked to identify which word in a list of five topic words had been randomly switched with a word from another topic. This resulted in the counterintuitive conclusion, that in some cases humans preferred models that performed worse in terms of perplexity. As a consequence of this finding, perplexity is rarely used for evaluation in recent work.

2.4.3 Topic coherence

Topic coherence is the extent to which topic model results are interpretable to humans. Coherence scores are commonly used when evaluating different topic modeling techniques, but they do not measure how similar or dissimilar the resulting topics are, therefore they do not stand on their own.

Topic coherence is typically calculated on the top T most representative words of the topic. Newman et al. (2010) propose several metrics that evaluate the quality of a given topic in terms of its meaningfulness to humans. Their evaluation measures are compared based on how much they correlate with human judgments of topic coherence. These metrics make it possible to assess the quality of topic modeling results without human evaluations or document labels. Their most successful metric calculates the co-occurrence of word pairs in the entire corpus of the English Wikipedia. This corpus is used to estimate pointwise mutual information of each word pair among the top T terms per topic. This metric is used by Jipeng et al. (2019), Mehrotra et al. (2013) and Yan et al. (2013a). Newman et al. (2010) also introduce another promising metric that utilizes Google. The calculation involves performing a query with the top 10 most probable words of a given topic, then counting the number of occurrences of any of these 10 words in the titles of the first 100 hits of the Google search result. Mimno et al. (2011) introduced a new topic coherence score, which they have shown to correlate well with human judgments of coherence. The metric is calculated by first counting the document frequencies of words and co-document frequencies (i.e. the number of times the two words coappear in a document) of word pairs among the most representative

words within a topic. Then, logged ratios of the co-document frequencies and document frequencies are summed up. The use of this metric is motivated by the fact that word co-occurrence in documents of the corpus can imply semantic relatedness (Aletras and Stevenson, 2013). Topic coherence according to Mimno et al. (2011) is commonly used (Lu et al., 2017; Jónsson and Stolee, 2015; Gallagher et al., 2017; Sridhar, 2015; Smatana and Butka, 2019). Contrary to previous studies, Aletras and Stevenson (2013) found, however, that the metric does not correlate well with human judgments and conclude that it is sensitive to corpus size. Aletras and Stevenson (2013) suggest new coherence scores, also using Wikipedia as a reference corpus. They represent top topic words as context vectors. To create the context vector of word w they use the Wikipedia corpus to find a context window of five terms around all occurrences of word w, then they calculate the number of co-occurrences of word w and all other terms from the top topic words within the context windows. In their best performing variant, co-occurrences are weighted by NPMI (Bouma, 2009). Then, vector similarity measures such as cosine or Jaccard similarity are calculated on context vectors of a topic.

Lau et al. (2014) perform a comparison of various coherence metrics on the evaluation task of Chang et al. (2009). The metrics included in the comparison are the PMI score by Newman et al. (2010) and its normalized variant, the pairwise log conditional probability measure by Mimno et al. (2011), and the distributional similarity score by Aletras and Stevenson (2013). The highest correlations with human judgments of semantic interpretability were reached by the NPMI score and the method proposed by Aletras and Stevenson (2013).

Given the fact that short documents likely do not provide enough context for the calculation of the coherence score by Mimno et al. (2011) and the result of the above-mentioned experiments, we conclude that our application requires coherence measures that are calculated on a reference corpus, such as PMI from Newman et al. (2010) or NPMI, its normalized variant from Lau et al. (2014).

3 Data

In this section we describe the corpora that we use to extract topics from. The dataset is a large collection of short customer feedback given to SKIM's client, a web service provider. Feedback is collected at the client's website via a pop-up window over a period of 9 months. The feedbacks are given about different services, such as news, email, etc. These groups of feedbacks form separate datasets of differing sizes. Two of these datasets will be used for topic modeling: one is about the homepage and one is about the e-mail service. From now on these two datasets will be referred to as Homepage and E-mail. The documents in both datasets contain answers to a question about why the user would or would not recommend the client's service. Data is available from eight countries: Australia, Brazil, Canada, France, Hong-Kong, UK, and Taiwan. Feedback is generally written in English in four of these countries. The rest are machine translated, which usually only works reliably for European languages. Therefore, Hong-Kong and Taiwan are omitted from the analysis. Furthermore, documents that are less than four characters or two words long or contain less than three unique characters are deemed non-informative and are therefore omitted. Lastly, we also delete documents where we notice signs of failed machine translation. This leaves us with approximately 19,107 documents in the Homepage- and 35,546 in the E-mail dataset. The documents are short in length: almost always below 500 characters, but most often they are only a few words long. The distributions of document lengths are displayed in Figures 1a and 1b. There are 54 documents in both datasets that are longer than 500 characters. These are omitted from the histograms, but not from the analysis.

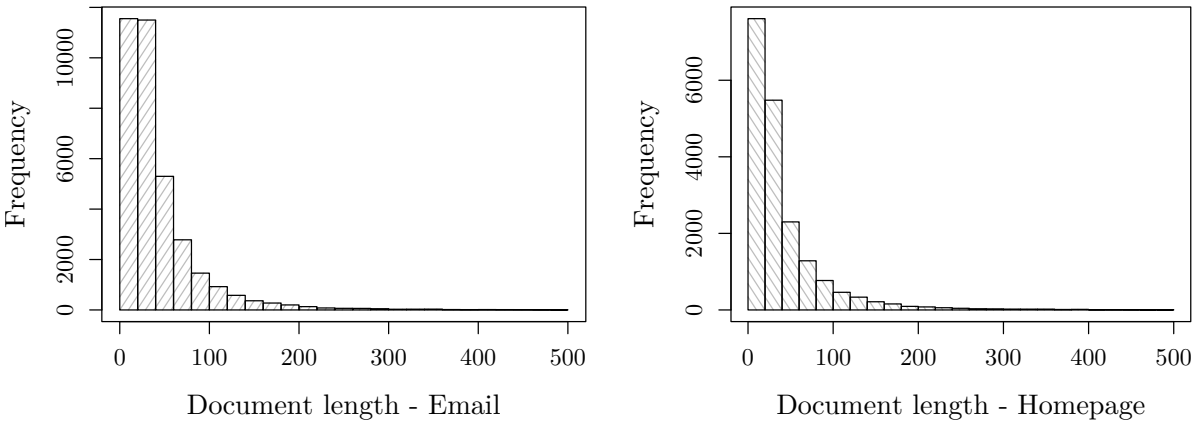


Figure 1: Frequency histogram of document length in number of characters on the Email (left) and the Homepage corpus (right).

Usually, in Natural Language Processing (NLP) the first step before analysis is bringing raw text data to a standard, analyzable format. This is referred to as text- or data preprocessing. In our application, the performed text preprocessing steps are the following. Documents are split to words (tokenization)

and made lowercase. Punctuation, white spaces, emoticons, and other "noise" is removed from the documents. The client's brand name and common English stopwords such as "a" "the" or "end" are deleted. Infrequent words, which only appear once in the corpus, are also deleted. Finally, related words are reduced to their common root (lemmatization).

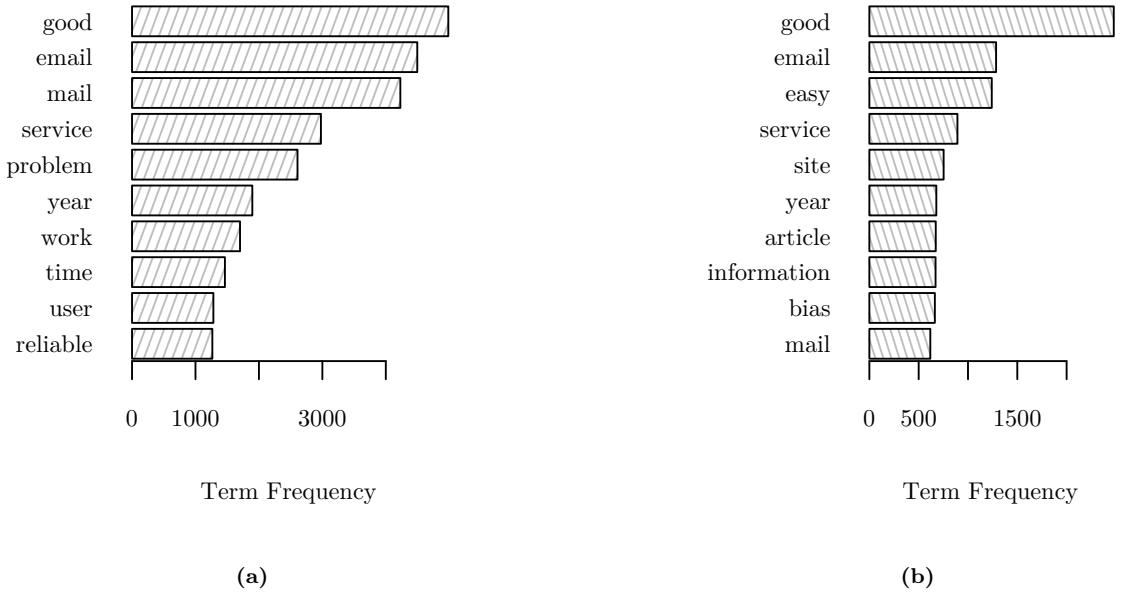


Figure 2: Top 10 terms and their frequency in the Email (left) and the Homepage corpus (right).

Figures 2a, 2b, 3a and 3b display the most frequent unigrams and bigrams of the two datasets, along with the total number of their occurrences. For privacy reasons, unigrams and bigrams containing the name of the web service provider are not displayed on these plots, even though the name appears frequently in the corpora.

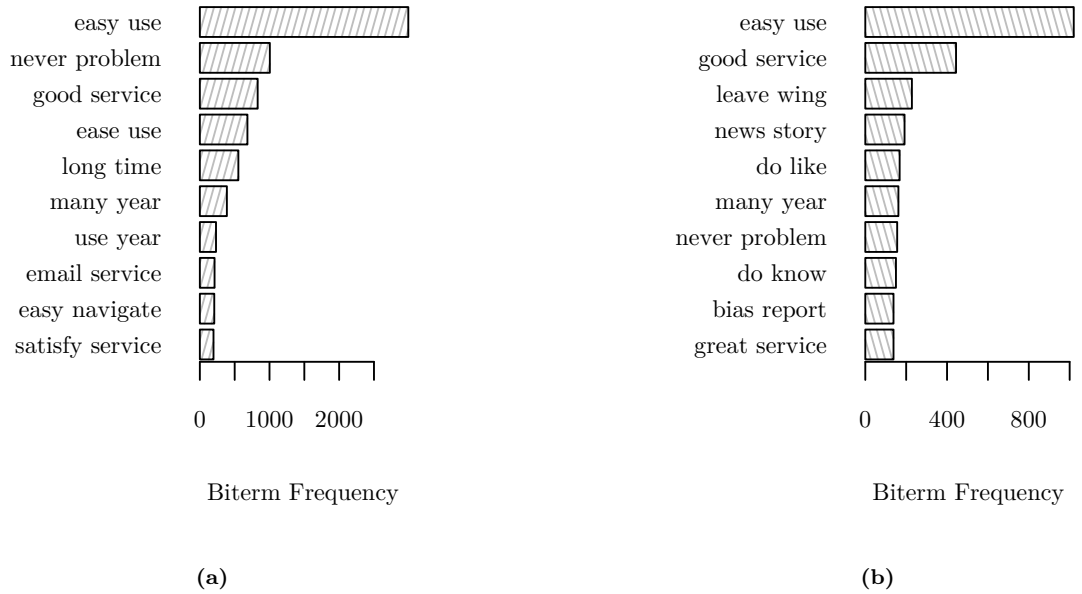


Figure 3: Top 10 biterms and their frequency in the Email (left) and the Homepage corpus (right).

3.1 Domain knowledge

In semi-supervised topic modeling, domain knowledge needs to be supplied to the algorithms in the form of lists of words that are representative of certain topics. These lists are called by various names in the literature. We adopt the terminology from Jagarlamudi et al. (2012) and refer to them as "seed sets". A seed set is made up of "seed words".

In this application, SKIM's client wants to guide the topic modeling results toward topics that are of interest to them. To achieve this, we use a "codebook" created by SKIM. It contains frequently appearing words and word combinations from the documents, organized according to the client-supplied topics. The codebook was already available before work on this research has started. We tokenized, filtered, and lemmatized these words and used them as seed sets in this thesis. 32 seed sets are available for the Email- and 25 for the Homepage dataset. Five examples are provided in Table 2. As it is shown, seed sets may differ in length.

Table 2: Seed set examples

Topic	Seed words
Advertisements	ad adblock advert advertise advertisement banner block click commercial promotion
Attachments	attachments download pdf photo picture
Inbox	inbox correspondence exchange box letter mail mailbox message outbox subject thread
Customer service	help customer feedback assistance support
Storage	archive backup capacity cloud database loss memory server size space storage volume

3.2 Document labels

Humans with domain knowledge (SKIM employees) annotated a small random subset of both datasets (approximately 1000 documents per dataset) to obtain "ground-truth" document labels. These labeled subsets can be used as a test set when we evaluate topic modeling performance. The labels used for document classification correspond to the same topics that were used as seed sets, as described in the previous section. Some examples are displayed in Table 3. Multi-topic assignments are common and are generally more common for longer documents. A few documents are not classified into any of the topics, as in some cases even the annotators cannot find suitable document labels. This can be due to typos, faulty machine-translation, or other noise in the text that the data preprocessing steps did not account for. These documents are discluded from the model evaluation procedure.

Table 3: Document label examples

Dataset	Document text	Document label(s)
Email	*"I'm totally satisfied with your service"*	Overall image
Email	*"Needs user interface improvement"*	Layout
Email	*"Quite simply the application is superb but there are too much ads"*	App, Advertisements
Homepage	*"Easy to use"*	Usability
Homepage	*"Too biased about Brexit. Too much news about royal family."*	Bias
Homepage	*"Good news and up to date information"*	Reliability/quality/-accuracy; Up-to-date

The test datasets containing document labels serve as the basis for clustering evaluation in Section 5.3, while the domain knowledge described in Section 3.1 will be used for the semi-supervised topic models, which we introduce in the following section.

4 Methodology

In this section, we introduce the topic models we apply. The focus is on semi-supervised methods, namely SeededLDA, Dirichlet-Forest LDA, and Anchored Correlation Explanation. Additionally, as baseline methods, we include the most standard topic model, Latent Dirichlet Allocation, as well as a topic model specifically created for short text, the Biterm Topic Model. The metrics used for evaluation are also described in this section.

4.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a generative probabilistic model built on the fundamental assumption of topic models, that each document consists of a mixture of topics, and each topic consists of a collection of words. In the generative process of LDA, documents are created according to this assumption to infer the topics of a corpus. Let D be the number of documents; N_d the number of words in document d; W the vocabulary size, or the number of distinct words in a collection of documents (a corpus); and K the number of latent topics. Furthermore let $\mathbf{w} = \left\{\mathbf{w^{(1)}}, \mathbf{w^{(2)}}, ..., \mathbf{w^{(D)}}\right\}$ denote the corpus, where $\mathbf{w^{(d)}} = \left(w_1^{(d)}, w_2^{(d)}, ..., w_{N_d}^{(d)}\right)$ is the set of words in document d and $w_i^{(d)}$ is the i-th word of document d. Topic assignments are denoted as $\mathbf{z} = \left\{\mathbf{z^{(1)}}, \mathbf{z^{(2)}}, ..., \mathbf{z^{(D)}}\right\}$, where $\mathbf{z^{(d)}} = \left(z_1^{(d)}, z_2^{(d)}, ..., z_{N_d}^{(d)}\right)$ is the vector of topic assignments in document d. For instance, $z_i^{(d)} \in \{1, ..., K\}$ is the index of the topic used to generate the word $w_i^{(d)}$ [2]. Let θ denote the $D \times K$ matrix of document-topic proportions and ϕ the $K \times W$ matrix of word probabilities in topics. The entry in the d-th row and k-th column of the document-topic matrix, $\theta_{d,k}$ expresses the probability that document d belongs to topic k. An element of the topic-word matrix, $\phi_{k,w}$ gives the probability of word w occurring in topic k. Consequently, the elements of both these matrices are between 0 and 1. Rows of θ, θ_d are K-dimensional random vectors that sum up to one, while rows of ϕ, ϕ_k are W-dimensional random vectors that also sum up to 1. The generative process for a document collection $\mathbf{w}$ is as follows:

1. Choose $\theta_d \sim \text{Dir}(\alpha)$ for each document $d \in \{1, ..., D\}$
2. Choose $\phi_k \sim \text{Dir}(\beta)$ for each topic $k \in \{1, ..., K\}$
3. For each word $w_i,\ i \in \{1, ..., N_d\}$ in each document $d \in \{1, ..., D\}$:
   (a) Choose a topic $z_i \sim \text{Multinomial}(\theta_d)$
   (b) Choose a word $w_i \sim \text{Multinomial}\left(\phi_{z_i}\right)$

$\text{Dir}(\alpha)$ and $\text{Dir}(\beta)$ denote Dirichlet distributions with corpus-level hyperparameters α and β. The Dirichlet distribution samples over a probability simplex and is conjugate to the Multinomial distribution. The value of α sets a prior to document-topic proportions. A lower value will result in fewer topics being

[2] The subscript $^{(d)}$ is only used when it is important to differentiate between documents. Otherwise, it is omitted for the sake of more easily readable notation.

assigned to each document, while a high value means that the distribution of documents among topics will be more spread out. Therefore a higher α generates documents that are more similar to each other. The prior for the word-topic distribution is given by β. Similarly, a low value means that each topic will be composed of only a few prevalent words, while a higher value will encourage wider usage of the vocabulary to model topics. Usually, symmetric Dirichlet priors are used in LDA, such that $\alpha = \alpha_1 = ... = \alpha_K$ and $\beta = \beta_1 = ... = \beta_W$. In the symmetric case, α and β can be simply represented by scalars instead of vectors. The use of symmetric priors implies the assumptions about the corpus that all topics have the same chance of appearing in each document and that all words have the same chance of appearing in each topic. While this assumption can be acceptable in some situations, Wallach et al. (2009a) showed that the combination of an asymmetric α and a symmetric β prior usually performs best. This strategy can be used to account for power-law word usage and in situations where we expect certain topics to be more common than others.

Along with the values of hyperparameters α and β, the number of topics K has to be set a-priori. The vocabulary size W is known, as we observe the corpus $\mathbf{w}$. The variables of interest to us, the topic assignments $\mathbf{z}$, topic proportions in documents θ, and word proportions in topics ϕ are latent. Posterior inference boils down to solving the following joint posterior density:

$$p(\theta, \phi, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{p(\theta, \phi, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{p(\mathbf{w}|\alpha, \beta)} \tag{1}$$

This distribution is intractable to compute (Darling, 2011), but approximate inference techniques are available. The two main methods are variational Expectation-Maximization (Blei et al., 2003) and collapsed Gibbs sampling (Griffiths and Steyvers, 2004). In this thesis, we use the latter estimation technique for LDA, as the subsequent semi-supervised models, which are based on LDA also employ collapsed Gibbs sampling. We now describe the estimation procedure using Griffiths and Steyvers (2004), Steyvers and Griffiths (2007), Heinrich (2005), and Darling (2011).

Gibbs sampling is a Markov Chain Monte Carlo technique which can be used to approximate a multivariate probability distribution indirectly, without having to compute its density, by sampling from lower-dimensional parameter distributions conditioned on all other parameter values. This sampling is done sequentially until the target distribution is approximated (Casella and George, 1992). The procedure starts with a burn-in period to remove the effect of the parameters used for initialization. The classical Gibbs sampler would involve sampling from the full conditional distributions of θ, ϕ and z_i. The prior distributions θ and ϕ, however, can be integrated (or "collapsed") out. This approach is referred to as collapsed or Rao-Blackwellised Gibbs sampling (Heinrich, 2005). In collapsed Gibbs sampling for LDA we wish to compute

$$p\left(z_i|\mathbf{z}_{-i}, \alpha, \beta, \mathbf{w}\right), \tag{2}$$

where $\mathbf{z}_{-i}$ denotes all the topic assignments except for z_i. Following the rules of conditional probability we get:

$$p\left(z_i|\mathbf{z}_{-i},\alpha,\beta,\mathbf{w}\right)=\frac{p\left(z_i,\mathbf{z}_{-i},\mathbf{w}|\alpha,\beta\right)}{p\left(\mathbf{z}_{-i},\mathbf{w}|\alpha,\beta\right)}\propto p\left(z_i,\mathbf{z}_{-i},\mathbf{w}|\alpha,\beta\right)=p(\mathbf{z},\mathbf{w}|\alpha,\beta), \tag{3}$$

which can be as expressed as:

$$\begin{aligned}p(\mathbf{z},\mathbf{w}|\alpha,\beta)&=\iint p(\mathbf{z},\mathbf{w},\theta,\phi|\alpha,\beta)\mathrm{d}\theta\mathrm{d}\phi\\&=\iint p(\phi|\beta)p(\theta|\alpha)p(\mathbf{z}|\theta)p\left(\mathbf{w}|\phi_z\right)\mathrm{d}\theta\mathrm{d}\phi\\&=\int p(\mathbf{z}|\theta)p(\theta|\alpha)\mathrm{d}\theta\int p\left(\mathbf{w}|\phi_z\right)p(\phi|\beta)\mathrm{d}\phi.\end{aligned} \tag{4}$$

Let $n_{d,k}$ be the number of times topic k is assigned to any word in document d and $n_{k,w}$ the number of times word w is assigned to topic k . Furthermore, let the subscript $^{(-i)}$ mean that the i-th token is left out of the calculation of counts. Given that the Dirichlet distribution is conjugate to the Multinomial distribution, the first term can be written

$$\begin{aligned}\int p(\mathbf{z}|\theta)p(\theta|\alpha)\mathrm{d}\theta&=\int\prod_i\theta_{d,z_i}\frac{1}{\mathrm{B}(\alpha)}\prod_k\theta_{d,k}^{\alpha_k}\mathrm{d}\theta_d\\&=\frac{1}{\mathrm{B}(\alpha)}\int\prod_k\theta_{d,k}^{n_{d,k}+\alpha_k}\mathrm{d}\theta_d\\&=\frac{\mathrm{B}\left(\left(\sum_k n_{d,k}\right)+\alpha\right)}{\mathrm{B}(\alpha)},\end{aligned} \tag{5}$$

where $\mathrm{B}(\alpha)$ is the multivariate Beta function $\mathrm{B}(\alpha)=\frac{\Pi_k\Gamma(\alpha_k)}{\Gamma\left(\sum_k\alpha_k\right)}$. The result is a Dirichlet distribution, whose parameter is given by the sum of the number of words generated by each topic in document d plus α. The second term of (4) is given

$$\begin{aligned}\int p\left(\mathbf{w}|\phi_z\right)p(\phi|\beta)\mathrm{d}\phi&=\int\prod_d\prod_i\phi_{z_{d,i},w_{d,i}}\prod_k\frac{1}{\mathrm{B}(\beta)}\prod_w\phi_{k,w}^{\beta_w}\mathrm{d}\phi_k\\&=\prod_k\frac{1}{\mathrm{B}(\beta)}\int\prod_w\phi_{k,w}^{\beta_w+n_{k,w}}\mathrm{d}\phi_k\\&=\prod_k\frac{\mathrm{B}\left(\left(\sum_w n_{k,w}\right)+\beta\right)}{\mathrm{B}(\beta)},\end{aligned} \tag{6}$$

which is also a Dirichlet distribution, whose parameter is given by the sum of the number of words generated by topic k across all documents plus β. Combining equations (5) and (6) yields

$$p(\mathbf{z}, \mathbf{w}|\alpha, \beta) = \prod_d \frac{\mathrm{B}\left(\left(\sum_k n_{d,k}\right) + \alpha\right)}{\mathrm{B}(\alpha)} \prod_k \frac{\mathrm{B}\left(\left(\sum_w n_{k,w}\right) + \beta\right)}{\mathrm{B}(\beta)}. \tag{7}$$

Utilizing the chain rule and the rules of conditional probability, we can now derive the equation for the the Gibbs sampling process of LDA [3]:

$$\begin{aligned} p\left(z_i|\mathbf{z}^{(-i)}, \mathbf{w}\right) &= \frac{p(\mathbf{w}, \mathbf{z})}{p\left(\mathbf{w}, \mathbf{z}^{(-i)}\right)} = \frac{p(\mathbf{z})}{p\left(\mathbf{z}^{(-i)}\right)} \frac{p(\mathbf{w}|\mathbf{z})}{p\left(\mathbf{w}^{(-i)}|\mathbf{z}^{(-i)}\right) p\left(w_i\right)} \\ &\propto \prod_d \frac{\mathrm{B}\left(\left(\sum_k n_{d,k}\right) + \alpha\right)}{\mathrm{B}\left(\left(\sum_k n_{d,k}^{(-i)}\right) + \alpha\right)} \prod_k \frac{\mathrm{B}\left(\left(\sum_w n_{k,w}\right) + \beta\right)}{\mathrm{B}\left(\left(\sum_w n_{k,w}^{(-i)}\right) + \beta\right)} \\ &\propto \frac{\Gamma\left(n_{d,k} + \alpha_k\right) \Gamma\left(\sum_k (n_{d,k}^{(-i)} + \alpha_k)\right)}{\Gamma\left(n_{d,k}^{(-i)} + \alpha_k\right) \Gamma\left(\sum_k (n_{d,k} + \alpha_k)\right)} \frac{\Gamma\left(n_{k,w} + \beta_w\right) \Gamma\left(\sum_w (n_{k,w}^{(-i)} + \beta_w)\right)}{\Gamma\left(n_{k,w}^{(-i)} + \beta_w\right) \Gamma\left(\sum_w (n_{k,w} + \beta_w)\right)} \\ &\propto \left(n_{d,k}^{(-i)} + \alpha_k\right) \frac{n_{k,w}^{(-i)} + \beta_w}{\sum_w (n_{k,w}^{(-i)} + \beta_w)}. \end{aligned} \tag{8}$$

Finally, we can estimate the distribution of words in each topic and the distribution of topics in each document simply using the count variables $n_{d,k}$ and $n_{k,w}$ in the following two equations:

$$\phi_{k,w} = \frac{n_{k,w} + \beta_w}{\sum_{w(} n_{k,w} + \beta_w)} = \frac{n_{k,w} + \beta}{\sum_w (n_{k,w}) + W\beta}, \tag{9}$$

$$\theta_{d,k} = \frac{n_{d,k} + \alpha_k}{\sum_k (n_{d,k} + \alpha_k)} = \frac{n_{d,k} + \alpha}{\sum_k (n_{d,k}) + K\alpha}, \tag{10}$$

where the second equalities hold if symmetric priors are used.

4.2 Biterm Topic Model

The main idea behind the Biterm Topic Model (Yan et al., 2013a) is that word co-occurrence in documents implies correlation between words. This idea is also utilized in LDA by modeling the generation of words in each document, but this approach is sensitive to the sparsity of the data in the case of short documents. Instead, BTM is built on aggregated word co-occurrence patterns of the corpus. The word biterm denotes an unordered word combination co-occurring in a predefined context (e.g.,in a short term sequence, such as a tweet). For example, a tweet with three words generates the following biterms:

$$(w_1, w_2, w_3) \Rightarrow \{(w_1, w_2), (w_2, w_3), (w_1, w_3)\},$$

where the order does not matter. In our application, given the shortness of the documents, all pairs of words co-occurring in any document are considered biterms. Let the corpus contain B biterms $\mathbf{b} = \{b_1, b_2, ..., b_B\}$ with $b_i = (w_{i1}, w_{i2})$. The prevalence of topics in the entire corpus is represented by the

[3] For a complete derivation see Heinrich (2005) and Carpenter (2010)

K-dimensional Multinomial distribution $\Theta = \{\Theta_1, \Theta_2, ..., \Theta_K\}$ with $\sum_{k=1}^{K} \Theta_k = 1$ and $\Theta_k = p(z = k)$. The meaning of $\mathbf{z}$, K, W, α, β and ϕ remain the same as in Section 4.1 : $\mathbf{z}$ denotes topic assignments, K the number of hidden topics, W the length of the vocabulary, α and β are hyperparameters of the Dirichlet priors, and ϕ is the $K \times W$ matrix of topic-word proportions. Here too, symmetric priors are used, thus α and β are scalar valued.

The generative process of BTM is similar to LDA, but here, instead of each document, it is the whole corpus that consists of a mixture of topics. Moreover, instead of words, biterms are generated. For a visual comparison, see Figure 5. Formally, the process for BTM is as follows:

1. Choose $\Theta \sim \text{Dir}(\alpha)$
2. Choose $\phi_k \sim \text{Dir}(\beta)$ for each topic $k \in \{1, ..., K\}$
3. For each $b_i \in \mathbf{b}$
   (a) Choose a topic $z_i \sim \text{Multinomial}(\Theta)$
   (b) Choose words $w_{i,1}, w_{i,2} \sim \text{Multinomial}(\phi_{z_i})$

From the above procedure it follows that the probability of biterm b_i conditioned on Θ and ϕ can be written as

$$\begin{aligned} p\left(b_i|\Theta,\phi\right) &= \sum_k p\left(w_{i,1}, w_{i,2}, z_i = k|\Theta,\phi\right) \\ &= \sum_k p\left(z_i = k|\Theta_k\right) p\left(w_{i,1}|z_i = k, \phi_{k,w_{i,1}}\right) p\left(w_{i,2}|z_i = k, \phi_{k,w_{i,2}}\right) \\ &= \sum_k \Theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}}, \end{aligned} \tag{11}$$

where the last line follows from the definitions $\Theta_k = p(z = k)$ and $\phi_{k,w} = p(w|z = k)$. Integrating over Θ and ϕ gives the probability of b_i given the hyperparameters α and β:

$$p\left(b_i|\alpha,\beta\right) = \iint \sum_k \Theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \mathrm{d}\Theta \mathrm{d}\phi. \tag{12}$$

Taking the product of the probabilities of all B biterms of the document collection given the hyperparameters leads to the likelihood over the entire corpus:

$$p(\mathbf{b}|\alpha,\beta) = \iint \prod_{i=1}^{B} \sum_k \Theta_k \phi_{k,w_{i,1}} \phi_{k,w_{i,2}} \mathrm{d}\Theta \mathrm{d}\phi. \tag{13}$$

We are interested in estimating the topic proportion vector Θ and the topic-word proportion matrix ϕ. This is done via Gibbs sampling, similar to the approximate inference for LDA in Griffiths and Steyvers

(2004). The Gibbs sampling equation is

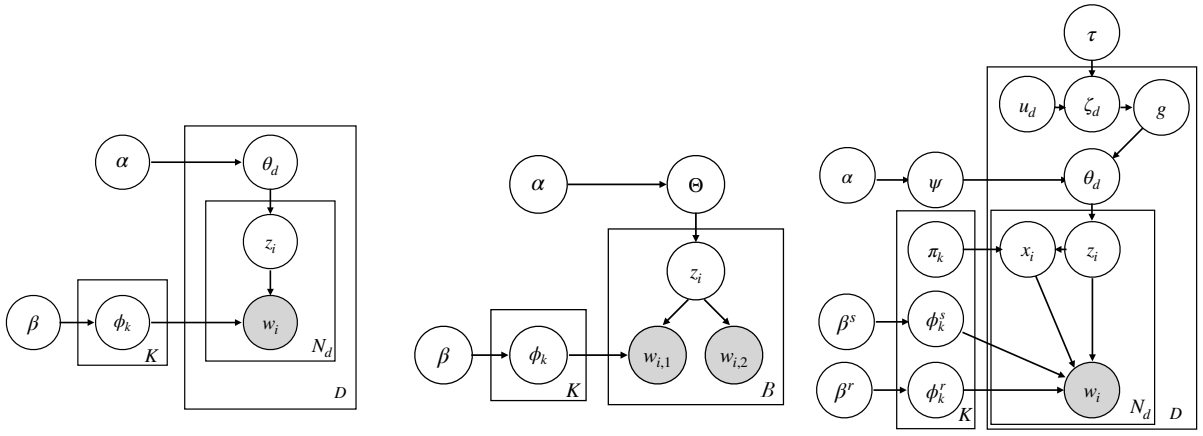
$$p\left(z_i = k|\mathbf{z}_{-i}, \mathbf{b}\right) \propto \left(m_k^{(-i)} + \alpha\right) \frac{\left(m_{k,w_{i,1}}^{(-i)} + \beta\right)\left(m_{k,w_{i,2}}^{(-i)} + \beta\right)}{\left(\sum_w m_{k,w}^{(-i)} + W\beta\right)^2}, \tag{14}$$

where m_k is the number of times topic k is assigned to any biterm and $m_{k,w}$ is the number of times topic k is assigned to word w. The subscript $^{(-i)}$ again denotes the exclusion of biterm b_i from counting. The Gibbs sampling algorithm iterates through the following procedure: a topic is drawn for each biterm of the corpus according to (14), then the counts m_k, $m_{k,w}$ and $m_{k,w}^{(-i)}$ are updated. After convergence of the Gibbs sampler, the final counts can be used to estimate the elements of Θ and ϕ:

$$\phi_{k,w} = \frac{m_{k,w} + \beta}{\sum_w (m_{k,w}) + W\beta}, \tag{15}$$

$$\Theta_k = \frac{m_k + \alpha}{B + K\alpha}. \tag{16}$$

Derivations for (14), (15), and (16) are included in the supplementary material of Yan et al. (2013a). Furthermore, as BTM does not model documents, we need extra calculations to get the probability that document d belongs to topic k. This can be inferred from the estimated model and the formula is derived in Yan et al. (2013a).



(a) **(b)** **(c)**

Figure 4: Graphical representation of LDA (left), BTM (middle) and SeededLDA (right).

4.3 SeededLDA

SeededLDA (Jagarlamudi et al., 2012) is a semi-supervised LDA extension. This approach allows the user to provide S sets of seed words representative of the corpus to guide the topic discovery process. These "seed sets" correspond to the sets of words described in Section 3.1. In our application $S = K$, meaning that seed words are used for all hidden topics. We build on this equality in the description of the method, and the generative process is therefore slightly simplified as compared to the one given in Jagarlamudi et al. (2012). To describe SeededLDA, we use the notation introduced in Section 4.1. Two aspects of LDA's generative process are modified to infuse the topic model with the user's domain knowledge.

First, topic-word proportions are upgraded by the use of seed words. In SeededLDA, instead of being defined as a Multinomial distribution ϕ_k over words, topics are defined as a mixture of two Multinomial distributions: a "seed-topic" distribution ϕ^s_k and a regular topic-word distribution ϕ^r_k. These Multinomial distributions are generated from Dirichlet priors (Step 1a and 1b of the generative process below). The seed topic distribution can only generate words from the seed set for topic k, which is provided by the user. Table 1 of Section 3.1 provides examples of seed sets. For example, the seed topic distribution corresponding to the topic *attachments* can only generate the five words given in the second row of the table. The seed topic distribution over words is inferred by the model. The regular topic-word distribution is interpreted in the same way as ϕ_k in LDA. Note that, while the regular topic-word distribution ϕ^r can be represented as a $K \times W$ matrix, where rows are W-dimensional random variables, this is not the case for the seed topic distribution, as the seed sets may differ in length. In the generative process, a Bernoulli trial with parameter π_k determines whether the seed topic distribution or the regular topic-word distribution is used (Step 2e/ii-iv). The parameter π_k is chosen from a uniform random distribution (Step 1c).

Second, seed sets are used to improve the document-topic distribution. To first give an overview of how this is achieved, we transfer the information of the seed sets to documents containing their words. Then for each document, document-topic proportions are drawn through a two step process: first we sample a seed set g. Then, we use the corresponding group-topic distribution (or seed set-topic distribution) ψ_g as an asymmetric prior in a Dirichlet distribution to generate document-topic proportions θ_d. The intuition behind this procedure is to encourage the similarity of document-topic distributions of documents that are likely to be about the same seed topics.

The group-topic distribution ψ relates seed sets to regular topics through a Multinomial distribution. As in our case, there are as many seed sets as regular topics ($K = S$), the rows of the $K \times K$ matrix ψ, ψ_k represent the topic distribution of the k-th seed set or group (these two terms are used synonymously in this section). These distributions are generated from a Dirichlet prior (Step 1d).

For each document, the list of allowed seed sets is represented by the binary vector u_d of length S, whose

elements take the value 1 if the document contains words from the given seed topic and 0 otherwise [4]. As an example, imagine that we have three topics, each with a corresponding set of seed words: *politics*, *science*, and *sports*. Their seed words are ("president"; "government"), ("research"; "vaccine"), and ("football"; "stadium"). Then, the u_d vector corresponding to the document "President reveals plans for new football stadium" will be (1,0,1). u_d is observable, but it is part of the generative process of SeededLDA (Step 2a).

Together with hyperparameter τ, u_d is used to sample a group from the allowed list of seed sets for each document. τ is the usual parameter vector or scalar (in the symmetric case) of a Dirichlet distribution. The authors use $\tau = 1$, which is equivalent to a uniform distribution over a $K-1$ probability simplex. A document-group distribution ζ_d is sampled from a Dirichlet distribution with mean τu_d (Step 2b). Then, a group g is chosen from ζ_d (Step 2c). Finally, the group-topic distribution corresponding to group g, ψ_g is used as a parameter vector in the Dirichlet distribution used to generate the length K vector of document-topic probabilities θ_d (Step 2d). θ_d then generates topics for each word of a document, just like in LDA (Step 2e/i). The generative process is as follows (Jagarlamudi et al., 2012):

1. For each topic $k \in \{1, ..., K\}$
   (a) Choose regular topic $\phi_k^r \sim \text{Dir}(\beta_r)$
   (b) Choose seed topic $\phi_k^s \sim \text{Dir}(\beta_s)$
   (c) Choose $\pi_k \sim \text{Beta}(1, 1)$
   (d) Choose group-topic distribution $\psi_k \sim \text{Dir}(\alpha)$
2. For each document $d \in \{1, ..., D\}$
   (a) Observe binary vector u_d
   (b) Choose a document-group distribution $\zeta_d \sim \text{Dir}(\tau u_d)$
   (c) Choose a group variable $g \sim \text{Mult}(\zeta_d)$
   (d) Choose $\theta_d \sim \text{Dir}(\psi_g)$
   (e) For each word $w_i, i \in \{1, ..., N_d\}$
      (i) Choose a topic $z_i \sim \text{Mult}(\theta_d)$
      (ii) Choose an indicator $x_i \sim \text{Bern}(\pi_{z_i})$
      (iii) if x_i is 0
      Choose a word $w_i \sim \text{Mult}(\phi_{z_i}^r)$
      (iv) if x_i is 1
      Choose a word $w_i \sim \text{Mult}(\phi_{z_i}^s)$.

A visual summary of the process can be seen in Figure 4c. For inference, a collapsed Gibbs sampler is used, similar to the one introduced in Section 4.1.

[4] If no seed words are found then u_d is a vector of ones

4.4 Dirichlet Forest LDA

Dirichlet Forest Latent Dirichlet Allocation (DFLDA) (Andrzejewski and Zhu, 2009) is a semi-supervised topic model based on LDA. It incorporates domain knowledge via a modification in the generation of word proportions in topics. Domain knowledge is expressed using Must-Link and Cannot-Link primitives, two concepts borrowed from the constrained clustering literature. A Must-Link connection between two words influences the model to generate them by the same topic, while a Cannot-Link connection promotes their generation by separate topics. Formally, **Must-Link** (i, j) is defined as follows: two words i, j have similar probability (whether it is large or small) within any topic, i.e., $\phi_{k,i} \approx \phi_{k,j}\ \forall k \in \{1, ..., K\}$. Conversely a **Cannot-Link** (i, j) means that within any topic, two words i, j should not have a large probability at the same time, but the probabilities are allowed to both be small.

Must-Links are encoded using a Dirichlet tree. The Dirichlet tree distribution, similarly to Dirichlet, is conjugate to the Multinomial distribution. It is a generalization of the Dirichlet distribution, such that it allows the variables (in this case, words) in a Dirichlet distribution to be dependent on each other, which is required to encode Must-Links. An example of a Dirichlet tree can be seen in Figure 5a. Here, the parameter $\eta \geq 1$ sets the strength of the domain knowledge. Domain knowledge is "turned on" for $\eta > 1$, while $\eta = 1$ is equivalent to LDA. Similarly to SeededLDA, domain knowledge is not enforced: Must-Links and Cannot-Links are user-preferences rather than hard constraints.

While Must-Links are transitive [5], Cannot-Links are not. Cannot-Links are organized in graphs, where the nodes represent words and the edges represent Cannot-Link relations. When represented together, the two constraint types are organized in a Cannot-Link graph, where nodes either correspond to a Must-Link closure or a word without any Must-Link connections. Let R denote the number of connected components in the full graph and $r \in \{1, ..., R\}$ the subgraphs within the full graph. Let $Q^{(r)}$ be the number of maximal cliques [6] in the complement graph [7] of the r-th connected component.

A Dirichlet tree is shown in Figure 5b. A Dirichlet Forest consists of $\prod_{r=1}^{R} Q^{(r)}$ Dirichlet trees since each has a different subtree out of the $Q^{(r)}$ possibilities at every branch $r \in \{1, ..., R\}$. Consequently, Dirichlet trees are uniquely identified by a vector of indices $\mathbf{q} = (q^{(1)}, ..., q^{(R)})$ where $q^{(r)} \in \{1, ..., Q^{(r)}\}$. Dirichlet trees $\mathbf{q}_k$ are sampled per topic from the DirichletForest(β, η) prior, $p(\mathbf{q}_k) = \prod_{r=1}^{R} p(q_k^{(r)})$ (Step 2 of the generation process). Dirichlet trees are then used to sample topic-word proportions (Step 3). Further details are given in Andrzejewski et al. (2009), in Section 3. The generation process can be summarized as follows:

1. Choose $\theta_d \sim \text{Dir}(\alpha)$ for each document $d \in \{1, ..., D\}$
2. Choose $\mathbf{q}_k \sim \text{DirichletForest}(\beta, \eta)$ for each topic $k \in \{1, ..., K\}$

[5] Must-Link (i, j) and Must-Link (j, l) imply Must-Link (i, l).
[6] A maximal clique is a subgraph (clique) where none of the vertices are part of another clique.
[7] A complement graph is created by removing the edges of a graph while filling in its missing edges.

3. Choose $\phi_k \sim \text{DirichletTree}(\mathbf{q}_k)$ for each topic $k \in \{1, ..., K\}$

4. For each word $i \in \{1, ..., N_d\}$ in each document $d \in \{1, ..., D\}$

   (a) Choose a topic $z_i \sim \text{Multinomial}(\theta_d)$

   (b) Choose a word $w_i \sim \text{Multinomial}\left(\phi_{z_i}\right)$,

The complete generative model is

$$p\left(\mathbf{w}, \mathbf{z}, \mathbf{q}_{1:K} | \alpha, \beta, \eta\right) = p\left(\mathbf{w} | \mathbf{q}_{1:K}, \mathbf{z}, \beta, \eta\right) p(\mathbf{z}|\alpha) \prod_k p\left(\mathbf{q}_k\right), \tag{17}$$

where $\mathbf{q}_{1:K} = \mathbf{q}_1, ..., \mathbf{q}_K$. Inference is performed via collapsed Gibbs sampling, similarly to Section 4.1. However, in this case besides $\mathbf{z}$ we need to sample $\mathbf{q}_{1:K}$ as well. The sampling equations and the formulas for estimating $\phi_{k,w}$, and $\theta_{d,k}$ after convergence are given in Andrzejewski et al. (2009), in Section 4.

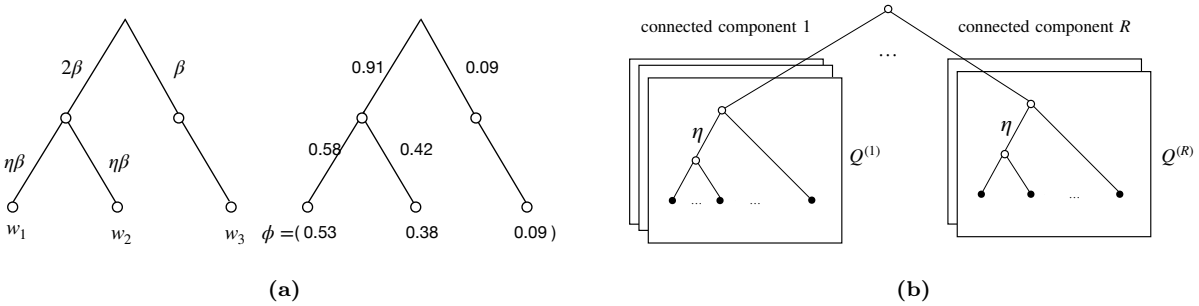


Figure 5: Illustrations taken from Andrzejewski et al. (2009): A Dirichlet tree encoding a Must-link between w_1 and w_2 and a corresponding sample ϕ (left). Dirichlet trees in a Dirichlet Forest (right). Full dots represent Must-link closures or words without Must-link connections.

4.5 Anchored Correlation Explanation

Anchored Correlation Explanation (CorEx) (Gallagher et al., 2017) is a recent alternative topic modeling approach that does not rely on the generative assumptions of the previously discussed techniques. Instead, it uses an information-theoretic framework to find maximally informative topics. While LDA-based methods model topics as distributions over words, CorEx models topics as binary vectors of length W, the length of the vocabulary. The elements of this vector represent whether words of the vocabulary are present in the topic or not. In CorEx, each word can only be associated with a single topic.

First, let us discuss some information-theoretic concepts [8]. Let X be a discrete random variable. $\text{H}(X)$ denotes the entropy of X, while the mutual information of two random variables X_1 and X_2 is given by $\text{I}(X_1; X_2) = \text{H}(X_1) + \text{H}(X_2) - \text{H}(X_1, X_2)$. The multivariate generalization of mutual information is total

[8] A more complete overview of these concepts can be found in Appendix A.1

correlation. The total correlation (TC) of a group of random variables X_G is defined as

$$\mathrm{TC}\left(X_G\right) = \sum_{i \in G} \mathrm{H}\left(X_i\right) - \mathrm{H}\left(X_G\right) = D_{\mathrm{KL}}\left(p\left(X_G = x_G\right) \| \prod_{i \in G} p\left(X_i = x_i\right)\right), \tag{18}$$

where $D_{\mathrm{KL}}\left(P \mid\mid Q\right)$ denotes the Kullback-Leibler Divergence between probability distributions P and Q, and is calculated as $D_{\mathrm{KL}}\left(P \mid\mid Q\right) = \sum_{x \in X} P(x) \log_2\left(\frac{P(x)}{Q(x)}\right)$. Thus, the second term in (18) stands for the Kullback-Leibler divergence between the probability of observing the group of random variables X_G and the maximum entropy product approximation of this probability, which is given by the product of the marginal probabilities of observing $X_i, i \in G$, the random variables included the group G.

Total correlation measures total dependence. When conditioning on another random variable Y, total correlation can be expressed as:

$$\mathrm{TC}\left(X_G \mid Y\right) = \sum_{i \in G} \mathrm{H}\left(X_i|Y\right) - \mathrm{H}\left(X_G|Y\right). \tag{19}$$

Therefore the reduction in total correlation when conditioning on Y is

$$\mathrm{TC}\left(X_G; Y\right) = \mathrm{TC}\left(X_G\right) - \mathrm{TC}\left(X_G|Y\right) = \sum_{i \in G} \mathrm{I}\left(X_i; Y\right) - \mathrm{I}\left(X_G; Y\right). \tag{20}$$

Now we return to the context of topic modeling. Given the difference between CorEx and the previous methods, we must introduce some new notation, but the meaning of W, D, and K remains the same. Let $Y_1, ..., Y_K$ represent binary latent topics and X_G a group of words. The group of words corresponding to topic k is denoted X_{G_k}. Note that, it follows from (19) that $\mathrm{TC}\left(X_G \mid Y\right) = 0$ means that a topic explains all the dependencies in X_G.

The goal of CorEx is to use latent topics to maximally explain the dependencies of words in documents. Therefore we maximize the following expression:

$$\max_{G_k, p\left(y_k|x_{G_k}\right)} \sum_{k=1}^{K} \mathrm{TC}\left(X_{G_k}; Y_k\right). \tag{21}$$

The objective function (21) can be rewritten using (20) and introducing an indicator variable $\gamma_{i,k}$, which takes the value 1 if word i is in topic k (i.e. $i \in G_k$) and 0 otherwise. Groups G_k are not allowed to overlap, that is, each unique word of the corpus should belong to a single topic. This appears as a constraint on $\gamma_{i,k}$ in the optimization problem:

$$\begin{aligned} &\max_{\gamma_{i,k}, p(y_k|x)} \sum_{k=1}^{K} \left(\sum_{i=1}^{W} \gamma_{i,k} I\left(X_i : Y_k\right) - I\left(X; Y_k\right) \right) \\ &\text{s.t.} \gamma_{i,k} = \mathbb{1}\left[k = \arg\max_{\bar{k}} I\left(X_i; Y_{\bar{k}}\right)\right], \end{aligned} \tag{22}$$

where $\mathbb{1}$ denotes an indicator function. Thus, words are assigned to the topic with which their mutual information is the highest. The constraint is relaxed to smooth the optimization, and the second line of (22) is replaced with a softmax function. This constraint relaxation yields a set of update equations that are iterated until convergence. After the relaxation $\gamma_{i,k} \in [0, 1]$, and at iteration t its update becomes:

$$\gamma_{i,k}^{t} = \exp\left(\lambda^{t}\left(I\left(X_i; Y_k\right) - \max_{\bar{k}} I\left(X_i; Y_{\bar{k}}\right)\right)\right), \tag{23}$$

where λ is the sharpness of the softmax function. It is initially set at a small value, then it is gradually increased to impose a hard constraint. The remaining update equations are given in Gallagher et al. (2017), in Section 2.2. The iterative update scheme is similar to the Expectation-Maximization (EM) algorithm, and its parameter initialization is random. Therefore it is recommended to run the model several times and choosing the result with the highest total correlation value.

Document-topic matrices are generally sparse, and CorEx exploits this in the optimization by treating the corpus as a $D \times W$ binary matrix X. The elements $X_{d,i}$ take the value 1 if word i appears in document d any number of times and 0 otherwise. This binarization may be a disadvantage of the method when applied to long documents, but it is unlikely to have a negative effect in the case of short documents.

The Correlation Explanation method (described so far) lends itself well to a semi-supervised extension. This is done by adding constraints to the optimization problem, which can anchor certain words into a specific topic. Anchor words are input by the user, similar to how seed words are used in SeededLDA. When anchoring is used, the optimization is constrained so that $\gamma_{i,k} = \nu_{i,k}$, where $\nu_{i,k} \geq 1$ is a parameter controlling the strength of the supervision. In practice, we use the same strength for all i, k pairs, thus the hyperparameter ν controls domain knowledge strength for the entire model. Moreover, unlike DFLDA, CorEx is able to handle overlapping sets of seed words. This can be useful when a user wants to separate a larger topic into two. For example, a user may use the anchor words "book", "room", "hotel" and "book", "shelf", "read" to separate a topic that contains the word "book" in both of its senses.

In LDA-based methods, the most representative words of a topic can be found by looking at the words with the largest probability estimates $\phi_{k,w}$ in each topic. In the Anchored Correlation Explanation model this can be done by ranking the mutual information terms $I(X_i; Y_k)$. Similarly, document-topic proportions, denoted as $\theta_{d,k}$ in LDA can also be calculated in CorEx by calculating the probability of the latent factor Y_k conditioned on $X_{d,*}$, the row of the document-term binary matrix corresponding to document d. Note, however, that LDA is a generative model, meaning that it estimates a probability distribution over documents. CorEx, on the other hand, is a discriminative model that estimates the probability that a document belongs to a certain topic, given the document's words.

4.6 Evaluation metrics

To provide a good assessment of the performance of the proposed methods, we evaluate them from two different aspects: topic coherence and how well the resulting document clusters correspond to document labels.

4.6.1 Coherence metrics

First, we assess topic coherence. According to Quan et al. (2015) some of the traditional coherence scores may work well for long documents but can be less favorable for short texts. This is due to their limited number of word co-occurrences, which is often used for the calculation of coherence scores (e.g. in Mimno et al. (2011)). Therefore we opt for a measure that uses an external corpus for the calculation of coherence and performs well in the latest studies. Our choice is the NPMI-score, the normalized variant of PMI from Newman et al. (2010), as used by Lau et al. (2014) [9]:

$$\text{NPMI}(\mathbf{w}_k) = \frac{1}{T(T-1)} \sum_{1 \leq i < j \leq T} \frac{\ln \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\ln p\,(w_i, w_j)}, \tag{24}$$

where $\mathbf{w}_k = (w_1, w_2, .., w_T)$ represents the top T most probable words for topic $k \in \{1, ..., K\}$. The probabilities $p\,(w_i, w_j)$, $p\,(w_i)$, and $p\,(w_j)$ are estimated on the English Wikipedia corpus.

Furthermore, we also use a search engine-based scoring method from Newman et al. (2010). This metric uses the entire internet as an external data source via Google searches. First, the T most probable words from a topic are queried (separated by "+" signs) on Google. An example search would be: *interface+layout+look+format+navigation+view+presentation+ui+design+screen.* Topics are scored by counting the number of matches between the most representative words of the topic and the titles of the first 100 query results. Formally, the Google-titles-match (GTM) score is given:

$$\text{GTM}(\mathbf{w}_k) = \sum_{i=1}^{T} \sum_{j=1}^{|V|} \mathbb{1}[w_i = v_j], \tag{25}$$

where $v_j, j = 1, ..., |V|$ are all the terms of the titles of the (maximum) top 100 query results and the indicator function $\mathbb{1}$ is responsible for counting matches.

In coherence evaluation, the scores that a model receives are the two averages of the NPMI and GTM scores over all K topics. Higher scores mean better performance.

[9] Adhering to the original work of Bouma (2009), we use natural logarithms instead of binary logarithms to calculate information theoretic measures, such as mutual information. Changing the base of the logarithm does not result in relevant changes in the interpretation of the metrics, but it does change the unit of information measurement.

4.6.2 Clustering metrics

Second, we evaluate the document clusters that result from topic models, using human-annotated document labels. We think that external evaluation is particularly useful as topic coherence measures only focus on individual topics and not the result of the topic model as a whole. In Section 2.4 we described a number of external evaluation methods for topic modeling. From those we choose variations of Normalized Mutual Information (NMI) and of the F_1 score.

Let $\Omega = \{\omega_1, ..., \omega_K\}$ be the resulting set of K document clusters (topics), and $\mathbb{C} = \{c_1, ..., c_L\}$ be L document classes that result from the human annotation process described in Section 3.2. The size of the test set, or the number of documents that make up our clusters and classes is denoted by D_e. Normalized Mutual Information can be calculated by dividing mutual information by joint entropy. The formal definition according to Bouma (2009) is given:

$$\begin{aligned} \mathrm{NMI}(\Omega, \mathbb{C}) = & \frac{\mathrm{I}(\Omega; \mathbb{C})}{\mathrm{H}(\Omega, \mathbb{C})} = \frac{\sum_{i,j} p(\omega_i, c_j) \ln \frac{p(\omega_i, c_j)}{p(\omega_i)p(c_j)}}{-\sum_{i,j} p(\omega_i, c_j) \ln p(\omega_i, c_j)} = \\ & \frac{\sum_{i,j} \frac{|\omega_i \cap c_j|}{D_e} \ln \frac{D_e |\omega_i \cap c_j|}{|\omega_i||c_j|}}{-\sum_{i,j} \frac{|\omega_i \cap c_j|}{D_e} \ln \frac{|\omega_i \cap c_j|}{D_e}}, \end{aligned} \tag{26}$$

where $\mathrm{I}(\Omega; \mathbb{C})$ denotes the mutual information between Ω and $\mathbb{C}$, $\mathrm{H}(\Omega, \mathbb{C})$ denotes joint entropy. $|\mathbb{C}|$ denotes the cardinality of a set $\mathbb{C}$ and $p(\omega_i, c_j)$ represents the probability that documents from class c_j end up in cluster ω_i, which can be calculated simply by dividing the size of the intercept of ω_i and c_j by the number of documents D_e. Instead of joint entropy one can use the mean, maximum or other functions of $\mathrm{H}(\Omega)$ and $\mathrm{H}(\mathbb{C})$ for normalization. NMI ranges between 0 and 1, where 1 corresponds to a perfect match between Ω and $\mathbb{C}$, while 0 means the clustering is random with respect to class membership.

Since in our application sets within both the clusters Ω and classes $\mathbb{C}$ are allowed to overlap, meaning that documents may belong to more than one cluster and/or class, we use the NMI variation introduced in McDaid et al. (2011), which allows such non-disjoint sets. They propose the use of the maximum as the normalization function in the calculation of overlapping NMI:

$$\mathrm{NMI}_{\mathrm{o}}(\Omega, \mathbb{C}) = \frac{\mathrm{I}(\Omega; \mathbb{C})}{\max[\mathrm{H}(\Omega) + \mathrm{H}(\mathbb{C})]}. \tag{27}$$

However, the calculation of mutual information $\mathrm{I}(\Omega; \mathbb{C})$ and entropies $\mathrm{H}(\Omega)$ and $\mathrm{H}(\mathbb{C})$ is more involved in the non-disjoint case and is given in detail in Appendix A.2. As an alternative, we propose another calculation method for NMI, which involves a uniform random sampling of multiple cluster and class assignments. The resulting set of clusters $\mathbb{C}^{'}$ and classes $\Omega^{'}$ are disjoint, hence each document has a single corresponding cluster and class assigned. The random sampling is repeated to create n_r and n_q disjoint sets of both $\Omega^{'}_r$ and $\mathbb{C}^{'}_q$ respectively. Then, the average NMI of all pairs of these sets is calculated

according to (26). Formally, we calculate stochastic overlapping NMI as:

$$\mathrm{NMI_s}(\Omega, \mathbb{C}) = \frac{1}{n_r n_q} \sum_r \sum_q \frac{\mathrm{I}(\Omega_r^{'}; \mathbb{C}_q^{'})}{\mathrm{H}(\Omega_r^{'}, \mathbb{C}_q^{'})}. \tag{28}$$

Our second metric used for clustering evaluation is based on the F_1-score, defined as $F_1 = \frac{2PR}{P+R}$ (Derczynski, 2016), where P and R denote precision and recall respectively:

$$\begin{aligned} P =& \frac{|\text{ true positives }|}{|\text{ true positives}| + |\text{false positives }|} \text{ and} \\ R =& \frac{|\text{ true positives }|}{|\text{ true positives}| + |\text{false negatives }|}. \end{aligned} \tag{29}$$

A related measure designed for overlapping clusters is the BCubed F_1-score (Amigó et al., 2009). First consider two measures comparing the cluster and class assignments of two documents d_i and d_j, multiplicity precision and multiplicity recall:

$$\begin{aligned} P^{(\text{mult})}(d_i, d_j) &= \frac{\mathrm{Min}\left(|\Omega\left(d_i\right) \cap \Omega\left(d_j\right)|, |\mathbb{C}\left(d_i\right) \cap \mathbb{C}\left(d_j\right)|\right)}{|\Omega\left(d_i\right) \cap \Omega\left(d_j\right)|}, \\ R^{(\text{mult})}(d_i, d_j) &= \frac{\mathrm{Min}\left(|\Omega\left(d_i\right) \cap \Omega\left(d_j\right)|, |\mathbb{C}\left(d_i\right) \cap \mathbb{C}\left(d_j\right)|\right)}{|\mathbb{C}\left(d_i\right) \cap \mathbb{C}\left(d_j\right)|}. \end{aligned} \tag{30}$$

where $\Omega\left(d_i\right)$ and $\mathbb{C}\left(d_i\right)$ are respectively the sets of clusters and true classes associated with document d_i. BCubed precision and recall are simply averages of these measures over all document pairs:

$$\begin{aligned} P^{(\text{BCubed})} &= \binom{D_e}{2}^{-1} \sum_i \sum_{j>i} P^{(\text{mult})}(d_i, d_j), \\ R^{(\text{BCubed})} &= \binom{D_e}{2}^{-1} \sum_i \sum_{j>i} R^{(\text{mult})}(d_i, d_j), \end{aligned} \tag{31}$$

where D_e is the number of documents in the test set. Then we can calculate the BCubed F_1 score simply as:

$$F_1^{(\text{BCubed})} = \frac{2P^{(\text{BCubed})}R^{(\text{BCubed})}}{P^{(\text{BCubed})} + R^{(\text{BCubed})}}. \tag{32}$$

4.7 Implementation

To run the topic models described above, we used the available open-source Python and `R` implementations of the methods. For BTM, we used the `R` package `BTM` [10]. We ran LDA and SeededLDA in the Python

[10] https://cran.r-project.org/web/packages/BTM/index.html

package called `GuidedLDA` [11]. LDA was run in the `topicmodels` `R` package too, [12] as this implementation offers some additional functionality, such as the ability to access the log-likelihood values in each iteration and to modify control parameters of the Gibbs sampling algorithm. For DFLDA and CorEx we worked with the Python implementations published by the authors [13] [14].

Before we ran the models, we cleaned the datasets and performed the text preprocessing steps described in Section 3, using `R`. Then, the data was transformed into suitable formats for the different model implementations. Clustering metrics and the GTM coherence score were implemented in `R`. For the calculation of GTM, we needed to scrape Google search results. For this purpose we used import.io, [15] a web data integration platform that can convert unstructured web data into a structured format. For the calculation of NPMI coherence score we built on the code provided by the authors of Lau et al. (2014) [16]. As our reference corpus, we used a freely available sample of the English Wikipedia from Kaggle, [17] consisting of 30,477 Wikipedia articles. Visualizations were also created in `R`.

Hyperparameter settings play an important role in the outcome of the models, therefore for each method we performed a grid search to find the best hyperparameter combinations. For the four LDA-based models this means finding the hyperparameters that maximize the approximate log-likelihood of the model. In the case of CorEx we maximize total correlation. In the case of LDA and BTM the hyperparameters to optimize for are α and β. In DFLDA, besides α and β there is an additional hyperparameter, the domain knowledge strength, η. SeededLDA also uses α and β. The GuidedLDA package we used provides a slightly modified implementation of SeededLDA, as the user can set the strength of the domain knowledge $0 \leq \pi \leq 1$, which will have an impact on whether seed topics or regular topics are used to generate words (Step 2e/ii of the generative process in Section 4.3). Thus, we look for the optimal value of π as well. In CorEx only the strength parameter $\nu \geq 1$ needs to be optimized. For each model, the grid search was performed over numerous combinations of parameter settings. For the LDA-based models, only symmetric priors were tested.

Naturally, looking for optimal hyperparameters requires that the models have converged. According to Darling (2011), Gibbs sampling theoretically guarantees convergence, but the number of iterations required is difficult to determine. One way to check convergence is by calculating the difference in the average parameter values of the model for the first 10% and for the last 50% of the Gibbs draws after a certain number of iterations (Geweke, 1992). Another approach is calculating the difference in the log-likelihood of the model at different iterations.

[11] https://pypi.org/project/guidedlda/
[12] https://cran.r-project.org/web/packages/topicmodels/index.html
[13] http://pages.cs.wisc.edu/ andrzeje/research/df_lda.html
[14] https://github.com/gregversteeg/CorEx_topic
[15] https://www.import.io/
[16] https://github.com/jhlau/topic_interpretability
[17] https://www.kaggle.com/kevinlu1248/wikipedia-articles-csv-2017

5 Results

In this section we report the results of applying topic models to corpora of customer feedback. We start by discussing convergence and optimal model hyperparameter combinations. For the parameter of CorEx we conduct additional analysis in Section 5.4. In Section 5.1 we evaluate the resulting topics qualitatively, by looking at the most representative words per topic. We compare the methods in terms of topic coherence and clustering performance in Sections 5.2 and 5.3. Finally, in 5.5 we examine the resulting topic sizes and look at how the results change for LDA and DFLDA when using asymmetric Dirichlet priors.

To ensure convergence, a high number of iterations (20,000) was used in the four LDA-based methods. Running the model again on an even higher number of iterations (25,000) does not result in an increased log-likelihood value for any of the models in any of the datasets, therefore we can conclude that collapsed Gibbs sampling has reached convergence. A convergence plot for LDA can be seen in Figure 10 of Appendix A.3. The convergence of CorEx is also confirmed, as its implementation includes a stopping condition based on the objective function value.

The resulting optimal sets of hyperparameters from the grid search described in 4.7 are displayed in Table 4. The grid search of CorEx revealed interesting results and is therefore dealt with in more detail in Section 5.4. The resulting α and β priors of the LDA-based methods are similar within the datasets. Interestingly, the grid search for SeededLDA suggests the use of only modest domain knowledge strength.

Table 4: Optimal hyperparameter combination of each model

	Dataset	
	Email	Homepage
LDA	α=50/K, β=0.001	α=30/K, β=0.001
BTM	α=10/K, β=0.01	α=50/K, β=0.01
SeededLDA	α=50/K, β=0.01, π=0.5	α=30/K, β=0.001, μ=0.5
DFLDA	α=50/K, β=0.001, η=2000	α=30/K, β=0.01, η=7000
CorEx	ν=10	ν=10

5.1 Qualitative evaluation

As a first step in the evaluation process we look at the resulting topics represented by their most probable words. We analyze the top 10 words of topics resulting from the five different topic models. Randomly selected topics of the Email and the Homepage datasets are displayed in Tables 5 and 6 respectively.

LDA yields a few coherent topics, such as Topic 4 and 26 of the Email- and Topic 21 of the Homepage dataset, which seem to be about competitors, junk/spam mails, and biased reporting respectively. These topics also roughly correspond to some of the seed topics we provided to the semi-supervised algorithms.

LDA's other displayed topics are less easy to interpret.

At first sight, BTM seems to produce less coherent and more generic topics. Topic 12 of the Email dataset is clearly about advertisements, but other topics are harder to recognize. It is noticeable that BTM's topics are made up of more common terms. In fact, the top ten terms of BTM aggregated over all topics are significantly more frequent than the top words of LDA: compared to LDA, BTM's terms appear in the corpus 2.65 times more often in the Email and 2.16 times more often in the Homepage dataset.

Table 5: Top 10 words in each model for randomly selected topics of the Email dataset

	Topic 4	Topic 5	Topic 7	Topic 12	Topic 22	Topic 26
LDA	[competitor name]	mail	use	mail	email	much
	good	send	easy	happy	address	spam
	like	message	find	love	first	lot
	think	receive	comfortable	experience	provider	still
	little	box	decade	website	personal	come
	system	important	super	extremely	ever	inbox
	other	quickly	userfriendly	deliver	main	advertise
	[competitor name]	late	reasonably	informative	primary	filter
	compare	delivery	timely	wonderful	switch	junk
	[competitor name]	fail	reasonably	relatively	client	block
BTM	account	email	email	ad	email	email
	email	mail	use	many	account	list
	folder	delete	mail	much	use	contact
	mail	one	like	add	year	address
	unread	select	good	page	password	mail
	login	easy	account	advertise	address	name
	possibility	address	get	advertisement	experience	change
	first	also	year	like	time	search
	feature	time	work	annoy	change	access
	create	click	one	mail	since	delete
SeededLDA	user	security	easy	recommend	easy	easy
	friendly	privacy	use	email	use	use
	easy	account	like	friend	reliable	good
	interface	datum	love	use	find	storage
	use	hack	navigate	mail	secure	free
	good	issue	good	like	convenient	space
	convenient	breach	simple	people	simple	access
	find	concern	email	know	mail	email
	reliable	year	access	already	safe	security
	mail	service	layout	reason	navigate	capacity

Continued on next page

Table 5 continued

	Topic 4	Topic 5	Topic 7	Topic 12	Topic 22	Topic 26
	email	use	email	recommend	app	interface
	send	since	message	know	storage	like
	receive	feel	delete	friend	application	new
	can	ever	folder	people	file	layout
DFLDA	important	start	read	already	space	nice
	lose	comfortable	find	other	attachment	look
	quickly	decade	manage	will	capacity	format
	unable	worry	go	everyone	big	old
	arrive	disappoint	hard	think	option	version
	month	begin	difficult	ask	server	navigation
	easy	security	interface	hate	nothing	storage
	use	information	layout	comment	give	space
	user	privacy	look	discussion	just	capacity
	ease	hack	format	replys	score	server
CorEx	friendly	datum	navigation	forum	reason	size
	simple	set	view	crappy	special	unlimited
	navigate	safety	presentation	subscriber	complain	memory
	simplicity	check	ui	[client product]	like	archive
	handle	breach	design	indonesia	really	cloud
	facility	info	screen		think	large

Moving on to semi-supervised models, some of SeededLDA's topics match the seed topics very well: topic 5 of the Email dataset is about security and privacy, topic 2, 8, 14, 17 and 21 of the Homepage dataset are respectively about the client's overall image, advertisements, biased reporting, gossip, and surveys. These topics all appear among our seed topics, in exactly this order. However, according to the seed topics, in the Email dataset topics 7,22 and 26 should respectively be about the *layout* of the website, *finance*, and *memory/storage*. Instead, we receive general topics with some seed words from the relevant seed topics, such as "layout" and "navigate" in topic 7. We can also see that the terms "easy" and "use" appear in all three of these topics. This pair of words is certainly a very commonly used combination, but they are used as seed words in the *usability* topic and should therefore ideally make up one large topic. A positive feature of SeededLDA is that it in topic 12 of the Email dataset (and also in an undisplayed topic of the Homepage dataset) it picks up a coherent topic that is not part of the seed topics. The topic is about recommendations, and it generally results from respondents directly answering the survey's question about whether they would recommend the client's service to anyone. The existence of this topic proves that SeededLDA is able to learn topics that are prevalent in the corpus but not included in the provided domain knowledge.

Table 6: Top 10 words in each model for randomly selected topics of the Homepage dataset

	Topic 2	Topic 8	Topic 14	Topic 17	Topic 21	Topic 22
LDA	email	mail	email	site	bias	comment
	can	slow	new	just	leave	security
	account	also	send	love	report	racist
	even	spam	friend	nothing	political	hate
	[competitor name]	message	feature	current	medium	datum
	open	little	account	country	material	privacy
	still	etc	hack	government	wing	section
	phone	lot	address	serve	right	post
	app	sometimes	already	guy	liberal	personal
	stop	system	provider	policy	fact	especially
BTM	messenger	even	email	like	news	bias
	get	email	get	use	page	site
	use	article	use	email	article	news
	back	ad	account	news	read	report
	version	load	security	now	interest	leave
	last	news	phone	since	good	see
	now	stop	contact	time	home	political
	cam	cycle	time	search	email	government
	bring	type	year	good	world	like
	big	can	send	make	change	make
SeededLDA	good	many	bias	news	survey	good
	service	ad	leave	much	like	service
	great	much	news	story	get	email
	excellent	advert	report	like	pop	quality
	satisfy	add	political	celebrity	stupid	great
	email	advertise	site	fake	answer	news
	site	like	wing	gossip	want	information
	happy	annoy	government	article	question	content
	experience	pop	liberal	real	annoy	site
	work	advertisement	material	bias	keep	excellent

Continued on next page

Table 6 continued

	Topic 2	Topic 8	Topic 14	Topic 17	Topic 21	Topic 22
	email	time	good	like	nothing	many
	mail	get	service	want	search	ad
	account	pop	year	one	reason	advertise
	address	keep	great	see	know	advert
DFLDA	inbox	survey	long	other	engine	spam
	recommend	stupid	love	answer	informative	advertisement
	cause	questionnaire	website	reliable	result	click
	personal	work	bad	[competitor name]	little	block
	day	personal	excellent	[competitor name]	perfect	bait
	step	matter	provider	[competitor name]	point	advertiser
	good	ad	bias	gossip	survey	recommend
	service	advertise	leave	celebrity	questionnaire	will
	year	many	racist	tabloid	stupid	finance
	great	advert	political	kardashian	force	ask
CorEx	long	spam	wing	royal	poll	anyone
	bad	advertisement	right	hollywood	participate	see
	excellent	click	liberal	actor	fill	want
	website	block	report	garbage	answer	friend
	provider	much	opinion	kim	rid	anything
	last	bait	propaganda	royalty	skip	phone

In DFLDA the order of the topics does not match the order of the seed topics. This model also detects the theme about recommendations in topic 12 of the Email dataset. Some topics correspond well to seed topics, such as topic 26 of Email and topics 8, 21, and 22 of Homepage, which are about *layout*, *survey*, *search engine*, and *advertisements* respectively. Interestingly, even though these topics are coherent and recognizable, their top terms are not always the same as the ones used as seed words. Overall, DFLDA's topics seem rather coherent, but the signs of having applied domain knowledge are more subtle than in the case of the other two semi-supervised methods.

The opposite is true for CorEx, where topics correspond exactly to the seed topics we used. Topics 4, 5, 7, and 26 of the Email corpus and topics 2, 8, 14, 17, and 21 are all perfectly recognizable topics from the seed topics. The order of the topics is the same and the top terms match our seed words. We can see that topic 4 of the Email dataset is about the previously mentioned common topic, *usability*, and for CorEx, the words "easy" and "use" appear in this topic only. This is of course no surprise, as CorEx only assigns one topic to each word. CorEx's results are not perfect nonetheless. According to the seed topics, topic 12 of the Email corpus should be about comments made by users on the website. The top terms

from CorEx's topic 12 do contain the seed words associated with the *comments* topic, but among them also appear terms that are probably unrelated, such as "Indonesia". The fact that only nine words are assigned to this topic suggests that it is quite small. In fact, this topic probably does not actually exist in the Email corpus, but CorEx enforced its creation, resulting in a small and meaningless cluster. Topic 22 is a similar case in both datasets - the topic should be about finance, one of the client's products, but instead, the topic becomes quite generic and difficult to interpret. Again, this is presumably due to the fact that *finance* is seldom mentioned in any of the documents. CorEx's lack of flexibility is also demonstrated by the lack of a *recommend* topic. Although topic 22 of the Homepage dataset contains the term "recommend", it is in combination with the word "finance". However, these two words only appear together in one out of 19,107 documents of the Homepage corpus. To conclude, CorEx appears to use domain knowledge very well when the implied topics are indeed present in the data but is a rather inflexible method which produces meaningless clusters when the seed topics do not exist in the corpus.

5.2 Coherence evaluation

We now evaluate the interpretability of the resulting topics, using the topic coherence metrics described in Section 4.6.1. Coherence evaluation is performed on the entire corpus. Table 7 contains the average NPMI topic coherence and average GTM scores [18] calculated on the top 10 words per topic. The best values along each metric are displayed in bold.

The NPMI values achieved by the models are noticeably low. For example, the topics consisting of top terms *ad, advertise, many, advert, spam* and *help, customer, support, feedback, assistance* only score 0.11 and 0.07 in NPMI respectively, even though they seem perfectly coherent. The reason for this could be that the English Wikipedia is not suitable as a reference corpus for our datasets. If however, this limitation affects all topics equally, then the relative NPMI scores can still be a good measure to compare topic model performance.

There is no clear winner in coherence evaluation. CorEx achieves the best NPMI values, but it performs worse in terms of GTM. SeededLDA achieves relatively high scores in all metrics in both datasets. Interestingly, BTM achieves higher GTM scores than LDA and some semi-supervised methods, even though in Section 5.1 BTM's topics seemed the least interpretable. This is possibly caused by BTM's use of more common words among its top ten terms, consequently producing more matches in Google search results. Furthermore, this is the scenario where coherence evaluation may be misleading. BTM's topics may have been coherent, but they were also very similar to each other. This aspect of intertopic similarity is not measured by coherence scores.

Concerned that 10 words may be too long to form a meaningful Google query, we also ran all coherence scores on only the top 5 words. These results can be seen in Table 10 of Appendix A.3. SeededLDA and CorEx perform best in this setup. The drastic improvement of CorEx as compared to Table 7 might be

[18] Google queries for the calculation of the GTM score were run on 2020.07.22

explained by the fact that, as we have seen in Section 5.1, CorEx's top few words usually mainly consist of the seed words provided for a given topic, which, naturally form a coherent cluster. The subsequent words, however, are in some cases, only very loosely connected. When only the top 5 words are taken, these "bad" topics cannot ruin average coherence as much.

Table 7: Coherence scores of each model, using the top 10 words per topic

	Email		Homepage	
	NPMI	GTM	NPMI	GTM
LDA	0.039	55.516	0.036	51.917
BTM	0.048	**138.750**	0.025	75.125
SeededLDA	0.055	111.568	0.044	**83.292**
DFLDA	0.042	71.097	0.028	49.12
CorEx	**0.062**	63.438	**0.053**	54.210

5.3 Clustering evaluation

Now we analyze and compare the different topic models' ability to correctly classify documents. This is done by calculating the selected clustering evaluation metrics from Section 4.6.2 on the annotated test corpora introduced in Section 3.2.

For clustering evaluation, document-topic proportion matrices first need to be transformed into binary label matrices for each model. To do this, the following procedure was used. In each document shorter than 500 characters, we take the two topics with the highest probability estimates. If the probability estimate exceeds the pre-determined threshold of 5%, the document is classified into the topic. In documents that are at least 500 characters long, we take the top three topics per document before applying the same threshold-based classification. NMI_s was calculated on 50 non-overlapping samples of both clusters and class labels.

Clustering evaluation results can be seen in Table 8. The best value along each metric is displayed in bold. There is little variation in NMI_o, therefore we draw no conclusion based on this metric. CorEx achieves the best NMI_s in both datasets and the best and second-best $F_1^{(\text{BCubed})}$ score in the Email and Homepage datasets respectively. Surprisingly, the two baseline methods often outperform SeededLDA. Overall, CorEx produces the best results in clustering evaluation.

Table 8: Clustering evaluation metrics for each model

	Email			Homepage		
	NMI_o	NMI_s	$F_1^{(\text{BCubed})}$	NMI_o	NMI_s	$F_1^{(\text{BCubed})}$
LDA	**0.970**	0.217	0.090	0.936	0.239	0.254
BTM	0.947	0.185	0.297	0.937	0.212	0.233
SeededLDA	0.954	0.169	0.296	**0.947**	0.199	0.194
DFLDA	0.945	0.224	0.255	0.938	0.251	**0.291**
CorEx	0.956	**0.296**	**0.410**	0.943	**0.361**	0.281

5.4 Analysis of the CorEx strength parameter

While looking for the optimal domain knowledge strength parameter ν for the Anchored Correlation Explanation model we noticed that the method's objective function value monotonically increases as we increase ν. Even when far exceeding the range for ν provided by the authors of the method we found no upper limit to this increase. Therefore in this section, we investigate whether a higher strength parameter value also corresponds to better performance in terms of clustering evaluation and topic coherence.

The results for the Email dataset are summarized in Figure 6. It is evident that after a certain value of ν, further increases do not result in better clustering performance as measured by the NMI_s and $F_1^{(\text{BCubed})}$ scores. Topic coherence as measured by the Google Titles Match score decreases as we increase ν. The results on the Homepage dataset can be seen in Figure 11 in Appendix A.3. There, we see some increase in coherence, but not in clustering performance. For this reason, we set ν at 10 for both datasets.

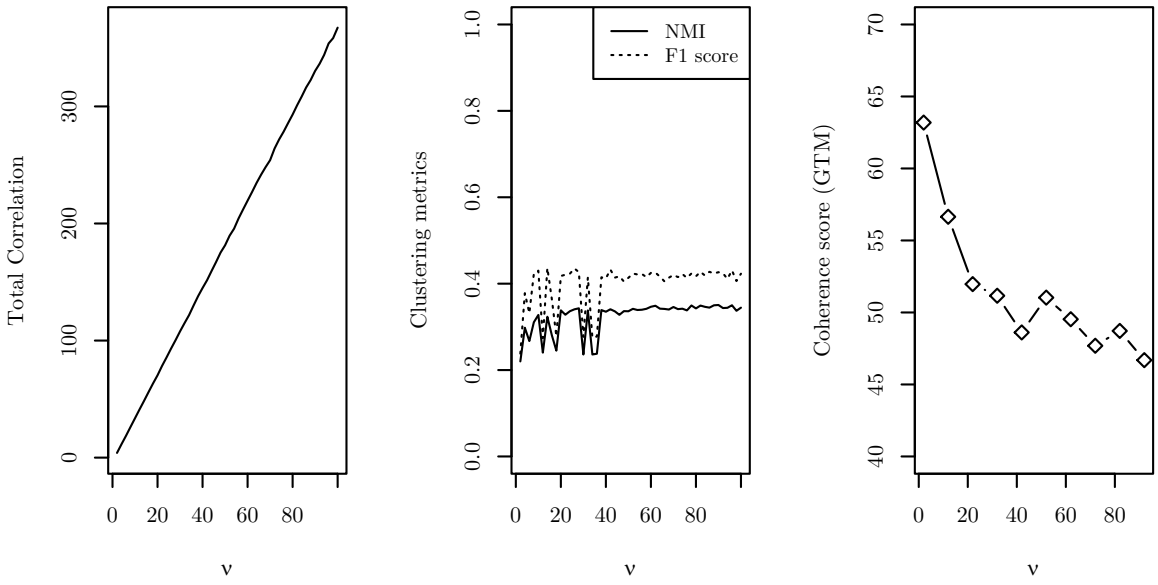


Figure 6: Total correlation (left), NMI_s and $F_1^{(\text{BCubed})}$ scores, (center), and the Google Titles Match score as a function of the CorEx domain knowledge strength parameter ν in the Email dataset.

5.5 Topic sizes and asymmetric priors

An important result for the client is to see how frequently certain topics appear in customer feedback. In this application, it is reasonable to expect that some topics are more common than others. For example, topic 2 of the seed topics is about the overall image of the client, which customers are expected to mention more often than a certain functionality or client product. Topic sizes are visualized in Figure 7 via heatmaps, where darker cell colors represent larger topics. Topic sizes inferred from the test sets are also shown as an estimate of correct sizes. These test topic sizes can only be directly compared to CorEx and SeededLDA, as only these models' topics follow the same order (i.e. the order of the seed topics). For the other models the figure is mainly useful to gauge how much variation there is in the resulting topic sizes.

According to the test data, the previously mentioned topic 2 and topic 4, a topic about usability, should generally be the largest topics in both datasets. CorEx correctly captures that these topics are large, but it also severely overestimates the sizes of some other topics, mainly topics 17, 22, and 23 in the Email and 15, 22, and 23 in the Homepage corpus. These are generally clusters where the CorEx algorithm picked up a different structure in the data than what was intended according to the seed topics. An example was already discussed in Section 5.1: topic 22 of the Email dataset should be about finance but instead collects documents with the word "recommend". BTM has sufficient variation in topic sizes, but its topic order does not match the order of the seed topics, therefore we cannot compare the sizes using the heatmaps only.

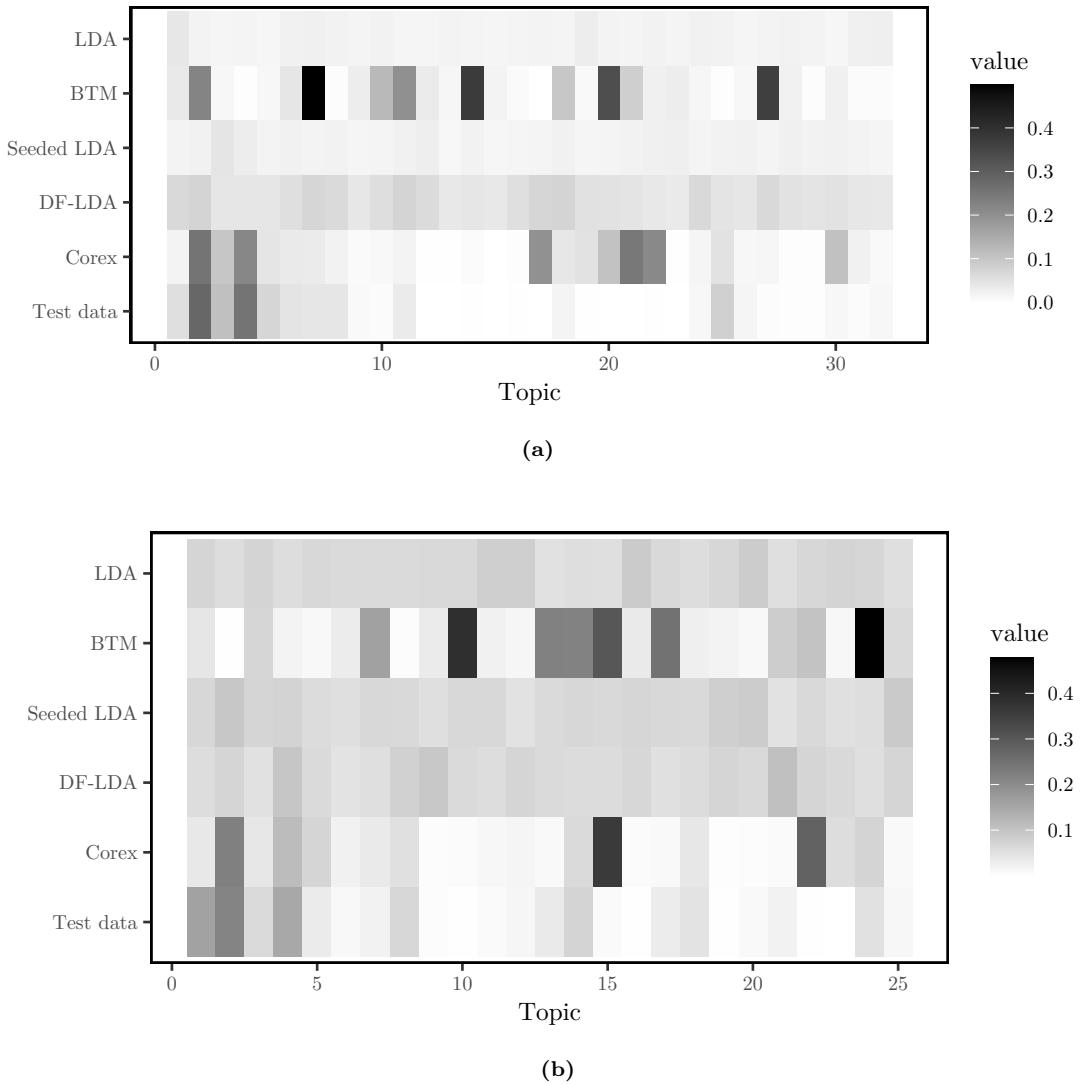
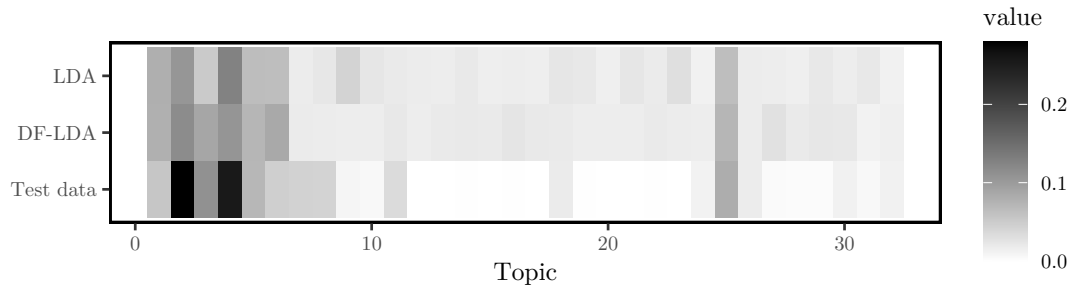


Figure 7: Resulting topic sizes for the five models and for the test set in the Email (top) and Homepage (bottom) datasets. Cell colors indicate the proportion of documents belonging to a topic.

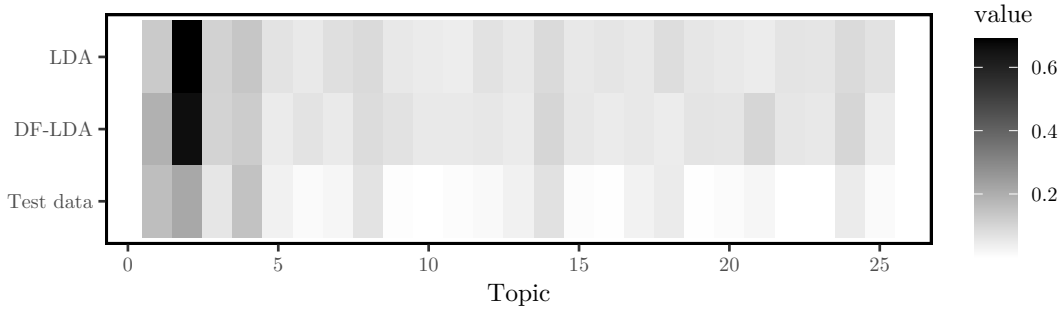
We can see that the topic sizes of LDA, SeededLDA, and DFLDA have little variation. This is clearly a problem. However, conveniently, through LDA prior parameter α we can manipulate the models' document-topic distributions. Specifically, varying topic sizes can be encouraged by the use of asymmetric priors. Instead of a scalar for α, we input a K-vector, whose values can be influenced by the "true" topic sizes from the test set. This provides an additional level of model (semi-)supervision.

LDA and DFLDA support the use of asymmetric priors. Running these models [19] results in topic sizes that resemble the sizes indicated by the test set much more, as it can be seen in Figure 8.

[19]The reported results correspond to an α vector of (50, 60, 50, 60, 50, 50, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 50, 45, 45, 45, 45, 45, 45, 45)/K in the Email dataset and (35, 40, 30, 35, 25, 25, 25, 30, 25, 25, 25, 25, 25, 30, 25, 25, 25, 25, 25, 25, 25, 25, 30, 25)/K in the Homepage dataset.



(a)



(b)

Figure 8: Topic sizes for LDA and DFLDA with asymmetric priors and for the test set in the Email (top) and Homepage (bottom) datasets.

Note that, when asymmetric priors are used, the topic order will correspond to the α vector provided, which corresponds to the order of the seed sets. However, in case of LDA and DFLDA it is possible that the k-th topic of a model has the same size as the k-th seed topic, but their contents are different. Therefore, in order to see if the revised topic sizes actually translate into better model performance, we rerun clustering evaluation for the two models with asymmetric priors. The resulting scores can be seen in Table 9. Numbers are displayed in bold if there has been improvement relative to the models' counterparts with symmetric priors. It appears that the Email corpus benefits more from the use of asymmetric priors. The $F_1^{(\text{BCubed})}$ score only improves in the case of LDA, but that does not mean that the underlying clustering structure remains unchanged. Using asymmetric priors results in a 23 percentage point average increase in BCubed recall and an 11.7 percentage point average decrease in BCubed precision. Thus, depending on the application and the cost of false positives versus false negatives, asymmetric priors can be both favorable or unfavorable.

Table 9: Clustering evaluation of LDA and DFLDA with asymmetric priors

	Email			Homepage		
	NMI_o	NMI_s	$F_1^{(\text{BCubed})}$	NMI_o	NMI_s	$F_1^{(\text{BCubed})}$
LDA	0.962	**0.220**	**0.154**	0.936	0.203	0.208
DFLDA	**0.964**	**0.247**	0.196	**0.940**	0.216	0.216

6 Conclusion and Discussion

The goal of this research was to find the best method for SKIM to model the topics of short customer feedback. We were focusing on semi-supervised topic models, and found that some of these methods can successfully utilize domain knowledge to guide topic modeling results in a desired direction. The technique that appears to be the best suited for our application is CorEx, as it performs best in clustering evaluation and performs fairly well in terms of topic coherence while producing roughly correct topic sizes. As CorEx is different from the other, LDA-based methods we used, there could be many reasons behind its good performance. One reason may be that the assumption that words can only belong to a single topic makes the use of domain knowledge more straightforward. Another likely reason is that the binarization of the document-term matrix is better suited for topic modeling on short documents. A positive feature of CorEx which was unused in our application is its ability to treat overlapping seed topics. CorEx's main drawback is its inflexibility, which is revealed when it is used with seed topics that occur infrequently in the data.

Concerning practical implications and the user-friendliness of the existing implementations of the methods, we found that the best performing method, CorEx also has the most well-maintained and most easy-to-use Python implementation among the semi-supervised models. We ran into numerous problems while setting up the code for DFLDA and SeededLDA on Windows, therefore those methods had to be run on macOS. For this reason, we could not include time comparisons of the algorithms, but we can state that DFLDA was the slowest of the methods and CorEx was by far the fastest. Therefore, we recommend CorEx as the best existing method for SKIM to find the topics of customer feedback. However, due to the method's inflexibility, we recommend that the user-supplied domain knowledge be precise and custom-made for the corpus. A suggested method to do this is to first run a more flexible method, such as LDA, which can find the most dominant topics of a corpus. Then LDA's results can be used in combination with SKIM and the client's domain knowledge when creating seed topics for CorEx.

Our research is of course not without limitations. The classification procedure we applied for clustering evaluation is rather arbitrary and the use of the same threshold for all the models may unfairly favor one model over another, depending on the document-topic distributions that the models return. For example, since the probability estimates in the document-topic matrices of LDA and SeededLDA were in general lower than elsewhere, using a larger threshold resulted in much lower F_1 scores for these methods. Another issue related to clustering evaluation is the use of overlapping clustering metrics. Topic modeling is very well suited for overlapping clustering evaluation, given the fundamental assumption of topic modeling, that documents are made as a distribution over a number of topics. Despite this, overlapping clustering metrics are not commonly used in the topic modeling literature, and therefore there may exist better metrics than the ones used in this research. Naturally, we could have simply classified each document to the topic with the highest probability estimate (as is common in the literature) and when annotating the test set we could have assigned the documents to the (subjectively) best topic. However, we believe

that this would have caused even more unreliability, as the resulting labels would hardly correspond to the *ground-truth*, rather than to a single possible correct clustering solution. Lastly, we have to mention here that the test sets on which the reported metrics are calculated are rather small.

Regarding coherence evaluation, NPMI values seem surprisingly small. We hypothesized that this can be the consequence of the unsuitability of the Wikipedia corpus for this application. Another reason could be that the sample of the English Wikipedia we downloaded is too small or is a non-representative sample. It would be interesting to recalculate NPMI using a more informal, large, internet-based corpus such as the iWeb corpus. The Google Titles Match score is an interesting and intuitive measure, but to our knowledge, it has not been tested outside the original paper of Newman et al. (2010). It is possible for instance, that an update to the Google algorithm could bias the results. Another possible limitation is related to data cleanness. Despite applying a meticulous data cleaning procedure, some noise remained in the data, such as typos and foreign words from partly mistranslated documents. However, it is uncertain whether this is truly an issue. According to Chaudhary (2020) adding typos or other random noise to a document before text mining tasks can even augment the robustness of the applied NLP model.

There are a number of avenues besides using a different reference corpus that remain unexplored. First, in the data cleaning procedure, we opted for word lemmatization instead of stemming. We did so, simply because lemmas are easier to read, as stemming often results in meaningless words. However, in some applications stemming works better, therefore it might be worthwhile to repeat our analysis with stemmed words. Moreover, we did not vary the number of hidden topics. This is justified by building on domain knowledge in this application, thus the number of hidden topics can be treated as known. However, it could be useful to allow for slightly more topics than what the domain knowledge suggests and see if the free topics, for which seed sets are not used form coherent topics or whether their existence improves the model's performance in terms of overall coherence or model fit.

We found optimal hyperparameters of the models via a grid search, where we looked at which hyperparameter combination achieves the best objective function value for a given method. However, we already established that these objective function values, such as the log-likelihood in LDA-based models and total correlation in CorEx do not necessarily correspond to better models in terms of human interpretability nor classification ability. It would be reasonable therefore to use coherence scores as an alternative metric to optimize for in grid search. However, the process of calculating a coherence score on a reference corpus for a large number of hyperparameter combinations per method is an arduous task. Coherence metrics that do not require a reference corpus such as the one in Mimno et al. (2011) would be much easier to calculate, but their performance on such short documents is likely to be dissatisfactory.

We hypothesized that CorEx's power comes from its suitability to handle short text. SeededLDA shows promise as well, as it performs strongly in clustering evaluation and the model can reliably use domain knowledge. It, however, lacks the capability to create topics of differing sizes. One way to correct for this is to derive and implement an improved SeededLDA model, that can handle asymmetric priors. An

even better improvement could be achieved by further modifying SeededLDA's generative process. BTM seemed to produce sufficiently variable topic sizes and is able to deal with short documents via a relatively simple modification of LDA's generative procedure. It might be possible to derive a method that joins BTM's and SeededLDA's modifications to the generative process, thereby combining the essence of the two algorithms. This would result in a semi-supervised LDA variant for short text. This is the most important direction of further research that we can identify.

References

Aletras, N. and Stevenson, M. (2013). Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 13–22.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486.

Andrzejewski, D. and Zhu, X. (2009). Latent dirichlet allocation with topic-in-set knowledge. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 43–48. Association for Computational Linguistics.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th annual international conference on machine learning*, pages 25–32.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Carpenter, B. (2010). Integrating out multinomial parameters in latent dirichlet allocation and naive bayes for collapsed gibbs sampling. *Rapport Technique*, 4:464.

Casella, G. and George, E. I. (1992). Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Chaudhary, A. (2020). A visual survey of data augmentation in nlp. `https://amitness.com/2020/05/data-augmentation-for-nlp`.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain monte carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

Darling, W. M. (2011). A theoretical and practical implementation tutorial on topic modeling and gibbs sampling. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 642–647.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Derczynski, L. (2016). Complementarity, f-score, and nlp evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266.

Dieng, A. B., Wang, C., Gao, J., and Paisley, J. (2016). Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*.

Fatemi, M. and Safayani, M. (2019). Joint sentiment/topic modeling on text data using a boosted restricted boltzmann machine. *Multimedia Tools and Applications*, 78(15):20637–20653.

Gallagher, R. J., Reing, K., Kale, D., and Ver Steeg, G. (2017). Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to the calculations of posterior moments. *Bayesian statistics*, 4:641–649.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235.

Heinrich, G. (2005). Parameter estimation for text analysis. Technical report, Technical report.

Hinton, G. E. and Salakhutdinov, R. R. (2009). Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614.

Hoffmann, T. (1999). Probabilistic latent semantic idexing. In *Proc. 1999 Int. Conf. on Research and Development in Information Retrieval (SIGIR'99)*, page 79.

Hong, L. and Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Jagarlamudi, J., Daumé III, H., and Udupa, R. (2012). Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., and Zhao, L. (2019). Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211.

Jipeng, Q., Zhenyu, Q., Yun, L., Yunhao, Y., and Xindong, W. (2019). Short text topic modeling techniques, applications, and performance: a survey. *arXiv preprint arXiv:1904.07695*.

Jónsson, E. and Stolee, J. (2015). An evaluation of topic modelling techniques for twitter.

Klejdysz, J., Lumsdaine, R. L., and van der Wel, M. (2018). Shifts in ecb communication: a text mining approach.

Larochelle, H. and Lauly, S. (2012). A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716.

Lau, J. H., Newman, D., and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W. (2016). An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 5(1):1608.

Lu, H.-Y., Xie, L.-Y., Kang, N., Wang, C.-J., and Xie, J.-Y. (2017). Don't forget the quantifiable relationship between words: Using recurrent neural network for short text topic discovery. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lu, Y., Mei, Q., and Zhai, C. (2011). Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval*, 14(2):178–203.

Lutov, A., Khayati, M., and Cudré-Mauroux, P. (2019). Accuracy evaluation of overlapping and multi-resolution clustering algorithms on large datasets. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–8. IEEE.

MacKay, D. J. and Mac Kay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to information retrieval*. Cambridge university press.

Mcauliffe, J. D. and Blei, D. M. (2008). Supervised topic models. In *Advances in neural information processing systems*, pages 121–128.

McDaid, A. F., Greene, D., and Hurley, N. (2011). Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*.

Mehrotra, R., Sanner, S., Buntine, W., and Xie, L. (2013). Improving lda topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 889–892.

Meilă, M. (2003). Comparing clusterings by the variation of information. In *Learning theory and kernel machines*, pages 173–187. Springer.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the conference on empirical methods in natural language processing*, pages 262–272. Association for Computational Linguistics.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108. Association for Computational Linguistics.

Nugroho, R., Molla-Aliod, D., Yang, J., Zhong, Y., Paris, C., and Nepal, S. (2015). Incorporating tweet relationships into topic derivation. In *Conference of the Pacific Association for Computational Linguistics*, pages 177–190. Springer.

N'Cir, C.-E. B., Cleuziou, G., and Essoussi, N. (2015). Overview of overlapping partitional clustering methods. In *Partitional Clustering Algorithms*, pages 245–275. Springer.

O'callaghan, D., Greene, D., Carthy, J., and Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657.

Phan, X.-H., Nguyen, L.-M., and Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web*, pages 91–100.

Quan, X., Kit, C., Ge, Y., and Pan, S. J. (2015). Short and sparse text topic modeling via self-aggregation. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Reisenbichler, M. and Reutterer, T. (2019). Topic modeling in marketing: recent advances and research opportunities. *Journal of Business Economics*, 89(3):327–356.

Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., Albertson, B., and Rand, D. G. (2014). Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082.

Röder, M., Both, A., and Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408.

Smatana, M. and Butka, P. (2019). Topicae: A topic modeling autoencoder. *Acta Polytechnica Hungarica*, 16(4).

Sridhar, V. K. R. (2015). Unsupervised topic modeling for short texts using distributed representations of words. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 192–200.

Steyvers, M. and Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of machine learning research*, 11(Oct):2837–2854.

Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al. (2001). Constrained k-means clustering with background knowledge. In *Icml*, volume 1, pages 577–584.

Wallach, H. M., Mimno, D. M., and McCallum, A. (2009a). Rethinking lda: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009b). Evaluation methods for topic models. In *Proceedings of the 26th annual international conference on machine learning*, pages 1105–1112.

Watanabe, S. (1960). Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82.

Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273.

Yan, X., Guo, J., Lan, Y., and Cheng, X. (2013a). A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1445–1456.

Yan, X., Guo, J., Liu, S., Cheng, X., and Wang, Y. (2013b). Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *proceedings of the 2013 SIAM International Conference on Data Mining*, pages 749–757. SIAM.

Yang, S., Lu, W., Yang, D., Yao, L., and Wei, B. (2015). Short text understanding by leveraging knowledge into topic model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1232–1237.

Yin, J. and Wang, J. (2014). A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242.

Zhai, Z., Liu, B., Xu, H., and Jia, P. (2011). Constrained lda for grouping product features in opinion mining. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 448–459. Springer.

A Appendices

A.1 Information theory basics

We continually use and refer to information theoretic concepts throughout this thesis. This appendix gives an overview of the most frequently used information theoretic concepts of this thesis and how they relate to each other. The section relies predominantly on MacKay and Mac Kay (2003).

Let X and Y represent discrete random variables. The fundamental concept in information theory is entropy. It quantifies the uncertainty of a random variable. The entropy of X is defined as:

$$\mathrm{H}(X) = -\sum_{x \in \mathcal{X}} p\left(x\right) \log_2 p\left(x\right), \tag{33}$$

where x and $\mathcal{X}$ respectively denote a realization and the support set of random variable X. Traditionally in information theory, binary logarithms are used, but the base of the logarithm can be varied depending on the application. Conditional entropy of Y given X is given by

$$\mathrm{H}(Y \mid X) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \tag{34}$$

and joint entropy by

$$\mathrm{H}(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(x, y). \tag{35}$$

A relation between these two concepts is given by the chain rule of conditional entropy : $\mathrm{H}(Y \mid X) = \mathrm{H}(X, Y) - \mathrm{H}(X)$. We can express mutual information as:

A measure of mutual dependence between two random variables X and Y is given by mutual information. It is defined as:

$$\mathrm{I}(X; Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right). \tag{36}$$

We can use the entropy of X and Y along with their joint entropy for an alternative definition of mutual information:

$$\mathrm{I}(X; Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X, Y) \tag{37}$$

Applying the chain rule of conditional entropy, this gives the following equalities:

$$\begin{aligned} \mathrm{I}(X;Y) &= \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &= \mathrm{H}(X) - \mathrm{H}(X \mid Y) \\ &= \mathrm{H}(Y) - \mathrm{H}(Y \mid X) \\ &= \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &= \mathrm{H}(X,Y) - \mathrm{H}(X \mid Y) - \mathrm{H}(Y \mid X) \\ &= \mathrm{I}(Y;X). \end{aligned} \tag{38}$$

A closely related measure to mutual information is the variation of information (Meilă, 2003). It measures the distance between X and Y and can be expressed as:

$$\begin{aligned} \mathrm{VI}(X;Y) &= 2\mathrm{H}(X,Y) - \mathrm{H}(X) - \mathrm{H}(Y) \\ &= \mathrm{H}(X) + \mathrm{H}(Y) - 2\mathrm{I}(X,Y) \\ &= \mathrm{H}(X,Y) - I(X,Y) \\ &= \mathrm{H}(X \mid Y) + \mathrm{H}(Y \mid X). \end{aligned} \tag{39}$$

A summary of the relations between the concepts described above is given in Figure 9.

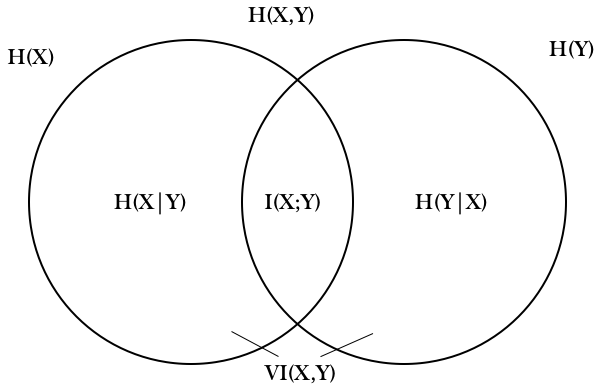


Figure 9: Information diagram illustrating how the concepts of entropy, joint entropy, mutual information, and variance of information relate to each other.

The normalized variant of mutual information, normalized mutual information is used frequently throughout this thesis. In Bouma (2009), normalized mutual information is defined as:

$$\mathrm{NMI}(X,Y) = \frac{\mathrm{I}(X;Y)}{\mathrm{H}(X,Y)}. \tag{40}$$

Another concept used in this thesis is total correlation. It is a multivariate generalization of mutual information, as it measures the dependence among n random variables. Total correlation is defined as:

$$\mathrm{TC}\left(X_1, X_2, \ldots, X_n\right) = \left[\sum_{i=1}^{n} \mathrm{H}\left(X_i\right)\right] - \mathrm{H}\left(X_1, X_2, \ldots, X_n\right), \tag{41}$$

where $\mathrm{H}\left(X_1, X_2, \ldots, X_n\right)$ corresponds to the joint entropy of the n random variables (Watanabe, 1960).

A.2 Calculation of overlapping NMI

The following is a summary of the calculation steps described in McDaid et al. (2011). We have n items, a clustering result $\Omega = \{\omega_1, ..., \omega_K\}$ and ground-truth item labels $\mathbb{C} = \{c_1, ..., c_L\}$. Let X and Y denote $n \times K$ and $n \times L$ binary matrices of cluster and class memberships respectively. This means that $X_{i,k}$ takes the value 1 if item i is in cluster k according to the clustering algorithm and 0 otherwise. Similarly, $Y_{i,l}$ takes the value 1 if item i is in class l and 0 otherwise. To compare cluster k and class l we define the following counts:

$$\begin{aligned}
a &= \sum_{i=1}^{n} \left[X_{\mathrm{ik}} = 0 \wedge Y_{il} = 0\right] \\
b &= \sum_{i=1}^{n} \left[X_{ik} = 0 \wedge Y_{il} = 1\right] \\
c &= \sum_{i=1}^{n} \left[X_{ik} = 1 \wedge Y_{il} = 0\right] \\
d &= \sum_{i=1}^{n} \left[X_{ik} = 1 \wedge Y_{il} = 1\right].
\end{aligned}$$

Furthermore let us define the function $h(x, y) = -x \log_2 \frac{x}{y}$. Then, the lack of information between cluster k and class l is given:

$$\mathrm{H}^*(X_k|Y_l) = \begin{cases} \mathrm{H}\left(X_k \mid Y_l\right) = \mathrm{H}\left(X_k, Y_l\right) - \mathrm{H}\left(Y_l\right) & \text{if } h(a,n) + h(d,n) \geq h(b,n) + h(c,n) \\ = h(a,n) + h(b,n) + h(c,n) + h(d,n) & \\ -h(b+d,n) - h(a+c,n) & \\ & \\ h(c+d,n) + h(a+b,n) & \text{otherwise.} \end{cases} \tag{42}$$

When we want to compare entire matrices, instead of only columns we need

$$\mathrm{H}(X|Y) = \sum_{k \in \{1,\ldots K\}} \mathrm{H}(X_k \mid Y) = \sum_{k \in \{1,\ldots K\}} \min_{l \in \{1,\ldots L\}} \mathrm{H}^*\left(X_k \mid Y_l\right). \tag{43}$$

Entropy is calculated as

$$\mathrm{H}(X) = \sum_{k=1}^{K} \mathrm{H}\left(X_k\right) = \sum_{k=1}^{K} \left(h(\sum_{i=1}^{n} \left[X_{ik} = 1\right], n) + h(\sum_{i=1}^{n} \left[X_{ik} = 0\right], n) \right). \tag{44}$$

$\mathrm{H}(Y)$ and $\mathrm{H}(Y|X)$ are calculated similarly. Finally, mutual information is given by

$$\mathrm{I}(X;Y) = \frac{1}{2}[\mathrm{H}(X) - \mathrm{H}(X|Y) + \mathrm{H}(Y) - \mathrm{H}(Y|X)], \tag{45}$$

which is the the average of the two possible ways in which we can calculate mutual information according to its definition given both matrices. Now we can calculate overlapping NMI as:

$$\mathrm{NMI_o}(\Omega, \mathbb{C}) = \frac{\mathrm{I}(\Omega;\mathbb{C})}{\max[\mathrm{H}(\Omega) + \mathrm{H}(\mathbb{C})]}. \tag{46}$$

A.3 Additional results

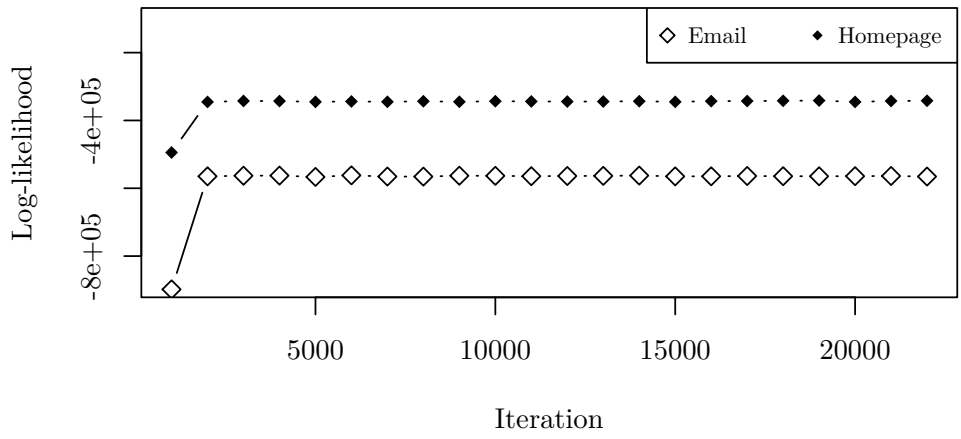


Figure 10: Convergence plot for LDA.

Table 10: Coherence scores of each model, using the top 5 words per topic

	Email		Homepage	
	NPMI	GTM	NPMI	GTM
LDA	0.056	87.250	0.045	98.174
BTM	0.074	141.452	0.041	105.217
SeededLDA	0.083	**147.031**	0.05	**114.917**
DFLDA	0.067	108.845	0.035	94.917
CorEx	**0.085**	111.72	**0.093**	110.080

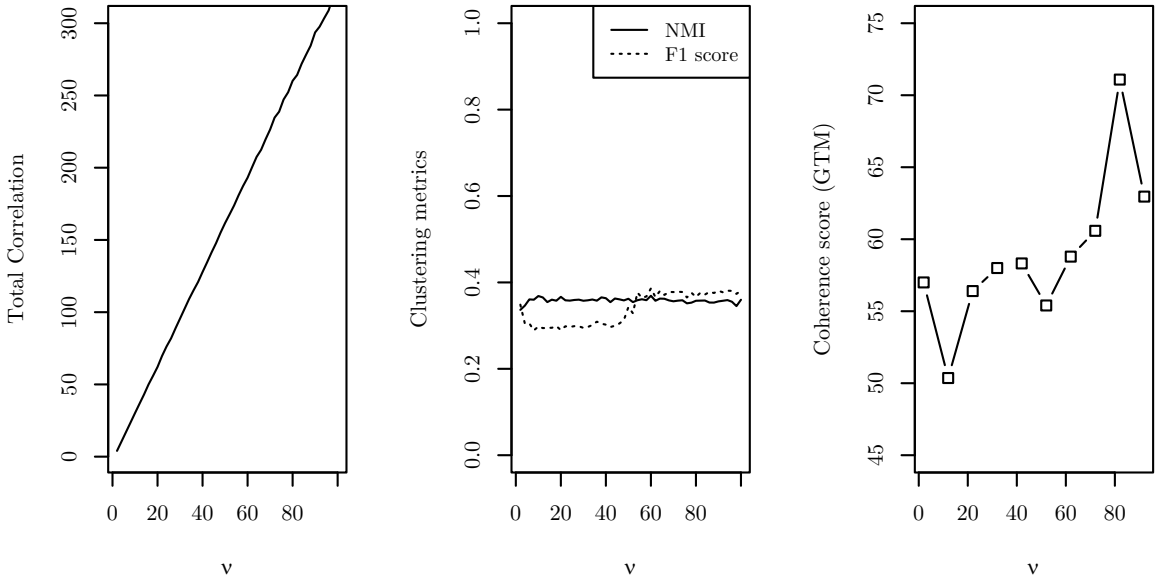


Figure 11: Total correlation (left), NMI_s and $F_1^{(\text{BCubed})}$ scores, (center) and the Google Titles Match score as a function of the CorEx domain knowledge strength parameter ν in the Homepage dataset.