Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Datateknik
2020 | LIU-IDA/LITH-EX-A--20/034--SE

# Anchor-based Topic Modeling with Human Interpretable Results

*Tolkningsbara ämnesmodeller baserade på ankarord*

**Henrik Andersson**

Supervisor : Lars Ahrenberg
Examiner : Eva Blomqvist

External supervisor : Leif Grönqvist

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

## Upphovsrätt



## Copyright

**Abstract**

Topic models are useful tools for exploring large data sets of textual content by exposing a generative process from which the text was produced. Anchor-based topic models utilize a separability assumption, known as the anchor word assumption, to define a set of algorithms with provable guarantees which recover the underlying topics with a run time practically independent of corpus size. Each topic is assumed to contain a word which rarely occurs in other topics, know as the topic's anchor word. A number of extensions to the initial algorithms, and enhancements made to tangential models, have been proposed which improve the intrinsic characteristics of the model making them more interpretable by humans. This thesis evaluates improvements to human interpretability due to: low-dimensional word embeddings in combination with a regularized objective function, automatic topic merging through anchor words, and utilizing word embeddings to synthetically increase corpus density. The aim is to find an anchor-based topic modeling approach which produces human interpretable results. Results show that anchor words are viable vehicles for automatic topic merging, and that using word embeddings significantly improves the original anchor method across all measured metrics. Combining low-dimensional embeddings and a regularized objective results in computational downsides with small or no improvements to the metrics measured.

# Acknowledgments

This thesis would not have been possible without the feedback, help, and encouragements given by the following people to whom I would like to direct a special thank you: Lars Ahrenberg and Leif Grönqvist for their valuable feedback as supervisors, Eva Blomqvist as the examiner of the thesis, the company and all the people at the office who supported me, and Sofia Løseth who motivated and helped me during this past half year.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

Companies today amass large amounts of data in a variety of different forms: numeric, categorical, ordinal, and textual, and it is important to be able to gauge what this data reflects. Structured data, such as the first three forms mentioned above, can easily be visualized in a variety ways since the domain of the data is known. Unstructured data, such as text, is however much harder to visualize since its length and vocabulary are unknown and possibly unbounded. Textual data is also one of the most common forms of data produced by humans since language is how we naturally reflect the world and communicate. Therefore, tools which can derive structure from textual data is of great interest.

A common tool for deriving structure from textual data is topic modeling, an approach which posits that a collection of texts is generated by a relatively small amount of topics latent within the text. Topic modeling attempts to recover these topics such that each text can be explained as a mixture of topics. This recovery process can be entirely unsupervised, meaning that the user of the tool does not have to supply any prior knowledge of the topics. Due to its unsupervised nature however, topics may not be easily interpretable by humans. The recovered topics are often presented as small collection of the most probable words within the topic, see Table 1.1 as an example.

Standard techniques for recovering topics are generally based on either a probabilistic, or an algebraic approach. Probabilistic approaches describe a generative model and attempt to recover the statistical parameters which increases the likelihood of the underlying data, these approaches include latent Dirichlet allocation (LDA) [1], and probabilistic Latent Semantic Analysis [2]. Algebraic approaches describe the data as a combination of matrices which factorize a representation of the data. These approaches make use of matrix factorization

Table 1.1: Representation of topics using the four most common words within the topic. Note that the actual topic is not recovered, only the most common words.

| Topic (Not recovered) | Most Common Words |
|---|---|
| football | soccer player game penalty |
| geology | rock ground fracture clay |
| computer engineering | programming algorithm transistor memory |

techniques such as singular value decomposition (SVD), and non-negative matrix factorization (NMF) [3].

A problem among the common recovery techniques is that they often scale poorly with large amounts of data, and can require minutes or hours to recover a single set of parameters. If the resulting topics are of poor quality, another estimation attempt with a new set of hyper-parameters may need to be completed, taking the same amount of time again. Topic models are also rarely formulated in such a way to maximize the human interpretability of their intrinsic qualities [4], requiring modified models [5] or human intervention [6] to achieve coherent results.

## 1.1 Motivation

This thesis was conducted in collaboration with a company which develops a data visualization application (referred to as "the application"). A user of the application wants to be able to easily visualize, interact with, and gain insights from data which they have collected. The goal of the application is for the user to simply be able to point at the location of their data and a set of visualizations, recommended inferences, etc. to be automatically made available. As described earlier in the chapter, structured data can often be visualized in a myriad of different ways, but unstructured data needs to be processed in some way to extract structural information. The application has limited ability to automatically extract structure from textual data and would therefore stand to gain from a topic modeling process which produces human interpretable results, and is efficient enough for interaction if required.

A relatively recent addition to the field of topic modeling are a family of models based on NMF in combination with a separability assumption [7]. This family of models, known henceforth as anchor-based models, assume that each topic contains some word which is almost entirely unique within that topic. This word is known as the topic's anchor word, or simply anchor. E.g. a possible anchor word for the topic *football* may be *offside*, or for the topic of *geology* the anchor word may be *grouting*. This separability assumption leads to methods which scale with the size of the vocabulary, as opposed to the number of documents and total number of words, while still performing on-par with established methods on a number of metrics [8]. However, not only do anchor-based models have the common drawbacks associated with human interpretability but the selected anchor words may also be unintuitive, and the uniqueness of topics produced depend highly on the anchor word recovery process.

A number of extensions have been made to the anchor word method for topic model recovery. Tandem anchors, which allow multiple words to be combined into a single anchor, allow for more intuitive anchors [9]. An anchor word recovery process based on T-distributed Stochastic Neighbor Embedding (t-SNE) has been shown to produce anchors which are more salient resulting in topics which are more unique and specific without sacrificing coherence [10]. The addition of parameter regularization has been shown to increase topic coherence, and allow for prior knowledge to be embedded [11]. The addition of meta-data in the recovery process has been shown to be able to produce sentiment sensitive topics [12]. Recent work, primarily based on knowledge from field of word-embeddings, has extended NMF-based algorithms to incorporate semantic knowledge of words to improve topic coherence for short texts. These methods make use of either a different view of the corpus based on skip gram with negative sampling (SGNS) [13], or word-embeddings learned on an external corpus [14].

Successfully combining these extensions may lead to a topic modeling method which is both efficient and human interpretable.

## 1.2 Aim

The aim of this thesis is to evaluate different anchor-based models based on their interpretability by humans. These models will either combine existing extensions to anchor-based models, or incorporate extensions to non anchor-based NMF models. The human interpretable qualities to maximize are coherence, specificity, and uniqueness of topics (these qualities and corresponding metrics are described in more detail in Chapter 2). Coherence is the quality of how easy the top words of a topic can be interpreted as a single coherent topic. Specificity is the quality of how different a topic's word distribution is from the underlying word distribution of the corpus [15]. Since coherence and specificity are local qualities of each topic, uniqueness will be used as a global quality to measure how unique topics are in relation to each other.

Initial anchor word selection based on low dimensional embeddings, and the addition of parameter regularization may improve model quality when combined. Their combined impact on model quality will be investigated. Tandem anchors can be used to iteratively improve uniqueness (and perhaps coherence) by automatically combining anchor words which produce similar topics. The initial anchors for this case may be recovered through efficient geometric methods or through t-SNE. Incorporating word-embeddings may improve coherence for anchor-based models in the same way they did for NMF-based methods for short texts, it is unclear however how this affects efficiency.

This thesis aims to investigate anchor-based topic modeling, which uses the extensions mentioned above, to estimate topic models with human interpretable results efficiently.

## 1.3 Research questions

For the following research questions, the quality of the model is measured using a set of human-correlated coherence metrics [16], specificity [15], and uniqueness.

1. How does the combination of existing extensions to the anchor method affect model quality?

   Regularization [11] and low dimensional embeddings [10] have been used to improve the quality of anchor-based topic models in the past, but have not been investigated in combination.

2. How does combining anchor words, whose resulting topic distributions are alike, affect model quality?

   Anchor-based models often produce topics which are not unique if the initial anchor are selected geometrically. Since topics are determined by their anchor words, similar topics could be merged using tandem anchors [9] resulting in increased uniqueness, and perhaps increased specificity. Since selecting anchors geometrically and estimating the topic model is efficient, an iterative optimization approach can be incorporated into the method.

3. How is model quality affected when incorporating word embeddings into the design matrix of the anchor method?

   Previous papers have shown that NMF-based model quality is improved (primarily for short texts) when word embeddings are incorporated into the design matrix [13, 14].

## 1.4 Delimitations

The quality of topic models depend on how the data is processed. This processing includes: identifying word tokens and removing stop-words, both of which are highly dependent on the underlying language of the corpus. Pre-calculated word embeddings also depend on the

language of the corpus on which they were trained. A natural delimitation is therefore to only investigate topic models for a single language, in this case English.

The results will depend on the data used during evaluation. For reproduceability purposes the datasets chosen will mostly be publicly available datasets common within the literature. For the purposes of this thesis, it is important that the datasets are of different types, representing small/large corpora with short/long texts.

# 2 Theory

This chapter presents the theory behind non-negative matrix factorization (NMF), anchor-based topic modeling, word embeddings for short-text topic modeling, evaluation measures for topic model interpretability, and methods for selecting the topic count parameter. The first section of the chapter also introduces topic models from a general perspective, including the most popular alternative. The notation used throughout the field of topic modeling varies and there are some notational conflicts among the papers presented in this chapter. Therefore, all definitions in this thesis have been updated to reflect notation used in this thesis (see Table 2.1).

## 2.1 Topic Models

Topic modeling is a technique born out of the field of information retrieval. Information retrieval, as the name implies, deals with the ability to efficiently retrieve appropriate information from data sets. When the data set is a large text corpus a user wants to be able to submit queries and retrieve documents which match the query. If the corpus has not been indexed or processed into an appropriate form to support such queries it may be computationally intractable to match the query.

The objective of topic models is to represent text collections using short descriptions which preserve statistical relationships, and can be seen as a method of dimensionality reduction. Dimensionality reduction is a form of compression which preserves the separation between objects but changes their representation to contain more information per dimension. The new dimensions may be interpretable such that each dimension has some semantic interpretation.

One technique for dimensionality reduction is Latent Semantic Analysis (LSA) [17], which the topic models later presented build upon. LSA proposed a new indexing method of documents and words in an attempt to solve the problem in which a query, and a should-be-matching document, do not contain any overlap among words. The corpus is represented as a matrix and the technique performs a singular value decomposition of it, resulting in a $K$ dimensional space occupied by words and documents. A query can then be represented as a pseudo-document and be projected into this space; the matching documents are the documents which lie close to the pseudo-document. The authors were not interested in interpreting the $K$ dimensions of the resulting representation, focusing instead only on information retrieval. Topic models however posit that a collection of documents are generated by a small

Table 2.1: Thesis notation.

| Notation | Name | Description |
|---|---|---|
| $w$ | Word / Term | A word type. |
| $z$ | Topic | A topic. |
| $s$ | Anchor Index | Index of an anchor word. |
| $D$ | Document Count | The number of documents in the corpus. |
| $V$ | Vocabulary Size | The number of unique words within the corpus. |
| $K$ | Number of topics | The number of topics within the corpus. |
| $M$ | Descriptor Cardinality | The number of words used in the topic descriptor. |
| $H$ | Design Matrix | The BOW representation of the corpus. |
| $A$ | Topic Matrix | The word-topic distributions, such that $A_{:k} = p(w\|z = k)$. |
| $B$ | Topic-Word Matrix | The topic-word distributions, such that $B_i = p(z\|w = i)$. |
| $W$ | Document-topic Matrix | The document-topic distributions, such that $W_{:i} = p(z\|d = i)$. |
| $Q$ | Co-occurrence Matrix | Can be interpreted as $Q_{ij} = p(w_1 = i, w_2 = j)$. |
| $\bar{Q}$ | Row-normalized $Q$ | Can be interpreted as $\bar{Q}_{ij} = p(w_2 = j\|w_1 = i)$. |
| $C$ | CluWords | The CluWord representation of the vocabulary. |
| $C_{\text{TF}}$ | CluWord TF Matrix | The CluWord TF design matrix. |
| $C_{\text{TF-IDF}}$ | CluWord TF-IDF Matrix | The CluWord TF-IDF design matrix. |
| $E$ | Word-embedding | A dense word embedding. |
| $S$ | Anchor Words | The set of anchor word indices. |
| $\Omega_{\text{X}}$ | Hyperparameters | The set of hyperparameters for algorithm X. |

collection of "topics", where topics are multinomial distributions on the vocabulary used in the document collection. The decomposition of LSA can be interpreted in such a way, where the $K$ dimensions are interpreted as topics. Different topic models make different assumptions about this generation process. E.g. if LSA is interpreted as a topic model, it assumes that the $K$ topics are uncorrelated since the factorization produces vectors which are orthogonal. The representation of a document can therefore be changed from a collection of words (most likely thousands of dimensions in the smallest case) to a small collection of topics (on the order of tens or hundreds of dimensions).

One of the first proper techniques of topic modeling (before the term "topic model" was popularized) was probabilistic LSA (pLSA) [2], which is a probabilistic generative document model inspired by LSA [17]. The model was proposed as an alternative to LSA with a solid statistical foundation instead of an algebraic method with derived interpretations. pLSA defines the generative process of the corpus as follows:

1. Select a document with probability $p(d)$.
2. Select a topic with probability $p(z|d)$.
3. Generate a word with probability $p(w|z)$.

The result is a document-word pair and the topic is discarded. Likelihood maximization is used to learn the probabilistic distributions which best describe the data.

Unlike LSA, pLSA does not assume that topics are uncorrelated. Models which do not make this assumption are known as correlated topic models [18]. Topic correlations model the dependence between the topics themselves, i.e. the likelihood that two topics will co-occur. E.g. if a document is assigned the topic *baking*, then its is likelier to occur in the same context as a document about *cooking*, than a document about *politics*, or probabilistically:

$$p(z_{\text{cooking}}|z_{\text{baking}}) > p(z_{\text{politics}}|z_{\text{baking}}) \tag{2.1}$$

However, for an uncorrelated topic model no such correlations exists and the probabilistic relationship would be:

$$p(z_{\text{cooking}}|z_{\text{baking}}) \approx p(z_{\text{politics}}|z_{\text{baking}}) \approx 0 \tag{2.2}$$

No model described in this thesis will directly recover the topic correlation distributions, but the underlying assumption still affects the estimated model.

Due to a number of drawbacks with the formulation of pLSA, the latent Dirichlet allocation (LDA) [1] model was defined. The LDA model can be estimated much more efficiently, is less prone to overfitting, and can explain how unseen documents are generated. The LDA model modifies the generation process and defines the generative process for each document to be:

1. Select the number of words $N \sim$ Poisson (this step is only relevant to the generative story and the Poisson distribution is not of interest)
2. Select a distribution over topics $\theta \sim \text{Dirichlet}(\alpha)$
3. For each of the $N$ words to be generated:
   a) Select a topic $z \sim \text{Multinomial}(\theta)$
   b) Select a word $w \sim \text{Multinomial}(z, \beta)$

This generative process has much fewer parameters than the one defined by pLSA, and the number of parameters does not grow when documents are added. LDA can be seen as the seminal moment of topic modeling, where a proposed model is computationally efficient to estimate, results in a representation which performs well on downstream tasks, and has a solid and easy to interpret statistical foundation. The model requires the prior parameters $\alpha$ and $\beta$. Inferring the hidden variables $\theta$ and $z$ can be done using various statistical inference methods such a Monte Carlo simulation or variational Bayes. The parameter $K$ re-appears in LDA as the dimensionality of the Dirichlet distribution over topics, it is assumed to be known.

Similarly to LSA, the LDA model implicitly makes the assumption that topics are uncorrelated. This assumption is induced by the choice of the Dirichlet distribution as the distribution over topics. The Dirichlet distribution can be replaced with a logistic normal distribution [18] to remove the assumption, but this variation is not as common as LDA.

Initially, topic models were used in downstream tasks as efficient representations of a collection of documents. However, the popularity of LDA has lead to interest in the intrinsic properties of the topic models themselves. These properties include: observing the topics directly through some representation [19], what mix of topics a specific document contains [20], and what topic a word in a sentence is most strongly associated with [21].

Topic models generally make the assumption that the order of topics and words can be ignored, these are known as *exchangeability* assumptions. The exchangeability assumption on words means that the corpus can be represented using the bag-of-words (BOW) matrix $H$, where $H_{wd}$ is the weight of word $w$ in document $d$. This weight may simply be the number of times the word occurs in the document, known as term frequency (TF) weighting, or may be a weighting scheme which down weighs words which occur across many documents, such as term frequency-inverse document frequency (TF-IDF). The matrix $H$ is known as the design matrix, and an example of the TF weighting scheme for three small documents is shown in Figure 2.1.

## 2.2 Anchor Method for Topic Modeling

The anchor method for topic model estimation is an efficient way of recovering the multinomial word distribution for each topic, represented as the topic matrix, $A$. This method

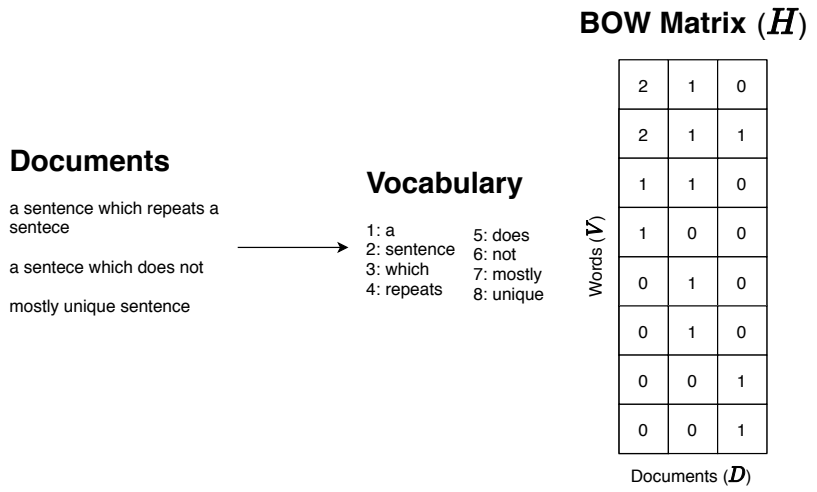



Figure 2.1: Example of the BOW representation (with TF weighting) of three documents with a vocabulary of eight words.

produces a model which is referred to as an anchor-based model, while the method for estimating the model is referred to as the anchor method. The method is based on two assumptions: (1) the corpus can be factored into two non-negative matrices, and (2) each topic contains a word which has near zero probability in any other topic. Anchor-based topic models are built on NMF (assumption 1), with a separability assumption (assumption 2) added to guarantee efficient estimation. NMF is matrix factorization technique involving three matrices with non-negative values: $A \in \mathbf{R}_+^{V \times K}$, $W \in \mathbf{R}_+^{K \times D}$, and $H \in \mathbf{R}_+^{V \times D}$, with $(V + D)K \ll VD$, such that:

$$AW \approx H \tag{2.3}$$

Where $V$ is the size of the vocabulary, and $D$ is the number of documents.

The paper which gives NMF its name applied this factorization technique to estimate topic models [3]. The factorization technique was proposed as a more natural way of decomposition when compared to contemporary methods, since the result is a strictly additive combination of parts. With the standard weighting schemes described previously, the matrix $H$ is non-negative by definition. NMF-based topic modeling posits that the matrix $A$ can be interpreted as a topic matrix, in which each column represents a topic, rows represent words, and cells reflect how strongly the word is associated with the topic. $A$ is normalized by column such that each column can be interpreted as a conditional distribution on a topic. See Figure 2.2 for a graphical representation of the matrix factorization. It has also been shown that pLSA solves NMF if Kullback-Leibler (KL) divergence minimization is the objective [22], suggesting that NMF-based topic modeling has statistical merit.

Finding the non-negative matrices $A$ and $W$ which factorize $H$ is NP-hard[1] [23]. However, by making a separability assumption (see Definition 2.2.1), recovering $A$ becomes solvable in polynomial time with provable guarantees [7].

**Definition 2.2.1. Anchor word assumption** - Each topic distribution contains a word (known as the topic's anchor word) with non-zero probability only in that topic distribution.

[1] An alternative factorization method, singular value decomposition (SVD), can be used but requires that each document is generated by only a single topic.

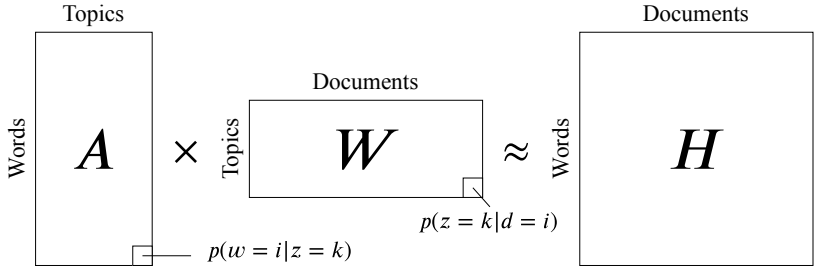



Figure 2.2: Factorization view used NMF topic modeling.

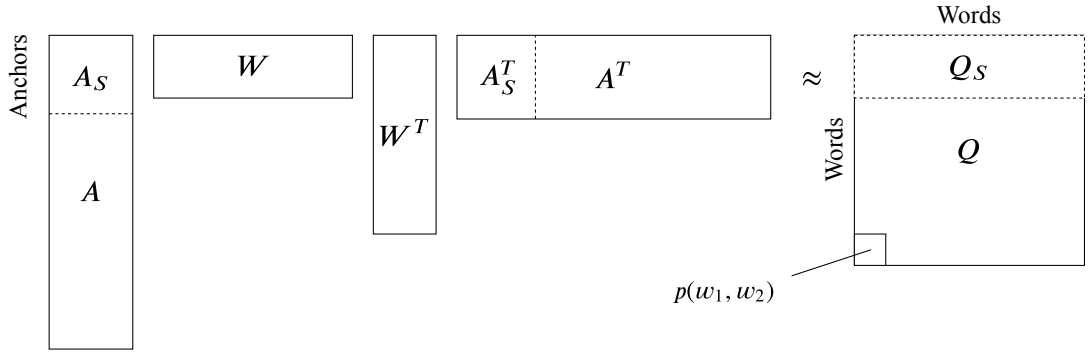



Figure 2.3: Factorization view used in the anchor word method. Same $A$ and $W$ as in Figure 2.2.

The anchor word assumption allows for recovering $A$ efficiently from the Gram matrix of $H$, denoted $Q$. This matrix should be constructed such that its expectation is given by:

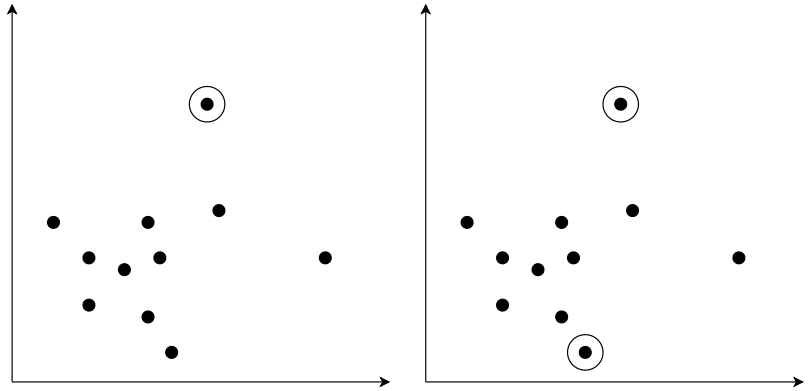
$$\mathrm{E}[Q] = \frac{1}{D}\sum_{d=1}^{D} AW_d W_d^T A^T \tag{2.4}$$

and can be interpreted as the joint probability on words, $Q_{ij} = p(w_1 = i, w_2 = j)$. The reason for this slightly different factorization is that it results in the ability to "read off" the values of $WW^T A^T$ in the first $K$ rows of $Q$ ($Q_S$), and then use those values to recover the entirety of $A$ [7] (see Figure 2.3 for a graphical representation of the factorization). This is due to the anchor word assumption and the knowledge of which words are the anchors. Unfortunately, the algorithms presented in the original paper did not turn out to be practical due to the computational complexity required to find the anchor words and the sensitivity to noise in the recovery process. The follow-up paper resolved both of these drawbacks and presented a set of algorithms which find anchor words quickly, and a recovery method more robust to noise [8].

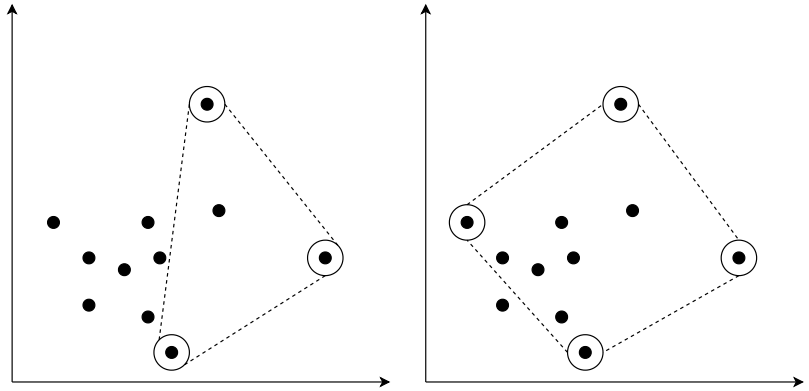
NMF, and subsequently the anchor method, do not utilize a process which implicitly assumes that topics are uncorrelated. Unlike say LSA which utilizes SVD as its matrix factorization method, a method where the basis vectors have to be orthogonal. The anchor method does not recover the topic correlation matrix.

To construct the Gram matrix $Q$ in an appropriate manner, the method described in the supplemental material of the follow-up paper can be used [24].

To find the anchor words a row-normalized version of $Q$, which can be interpreted as the conditional distribution on words, $\bar{Q}_{ij} = p(w_2 = j|w_1 = i)$, is used. The row vectors of $\bar{Q}$ are randomly projected to a lower dimensional subspace for efficiency. The algorithm then selects the $K$ vectors which maximize the volume of a polygon, spanned by the vectors, within the subspace. Such a maximization process can be done efficiently using a stabilized Gram-Schmidt process. A two dimensional visualization of the algorithm is shown in Figure 2.4



(a) The initial step selects two points, the one furthest from the origin, and the point furthest from the previously selected point.



(b) The following steps iteratively select $K-2$ points which maximize the volume (area in the figure) of the polygon spanned by the points. Note that the polygon selected is not necessarily the convex hull.

Figure 2.4: Visualization of the `FastAnchorWords` algorithm for a two dimensional random projection. This projection is normally selected much larger ($\approx$ 1000 dimensions).

with $K = 4$. The intuition behind this algorithm is that anchor words will only co-occur with a small number of other words, and therefore will end up as extreme points in the vector space spanned by the rows of $\bar{Q}$. The algorithm is called `FastAnchorWords` as in the original paper. Unfortunately, this method typically picks anchors which are non-salient ("eccentric" anchors), which in turn produces topics similar to the underlying word distribution. This is because there will always be a large collection of extremely rare words which do not anchor any particular topic. To alleviate this drawback a hyperparameter which disregards words with very low document frequency is introduced, called the anchor threshold.

The recovery method presented in the original paper only used the $K$ rows $Q$ corresponding to the anchor words to recover $A$, making it sensitive to noise. In fact, $Q$ is often so noisy that the original recovery algorithm totally breaks down for small data sets, as noted in the supplemental material of the follow-up paper [24]. To achieve a method more robust to noise, the authors instead attempt to recover the topic-word matrix $B$ using all rows of $\bar{Q}$. Since $B_{ik}$ can be interpreted as $p(z = k|w = i)$ we can recover $A$ by using Bayes' rule. This process assumes that each row $\bar{Q}_i$ is a convex combination of $\bar{Q}_S$ and the corresponding row $B_i$ (see Equation 2.5), resulting in $V$ constrained minimization problems.

$$\bar{Q}_i \approx \sum_{s \in S} B_{is} \bar{Q}_s \tag{2.5}$$

If the objective function of the minimization process is selected as the euclidean distance, then the recovery process is done in $O(KV^2 + K^2VT)$ time, where $T$ is the average iterations required by the exponentiated gradient minimization process. The objective function which is minimized by the exponentiated gradient algorithm is:

$$B_i = \arg\min_{B_i} ||\bar{Q}_i - B_i\bar{Q}_S||_2 \tag{2.6}$$

The high-level algorithm to recover $A$ (and $B$) from $H$ can be seen in Algorithm 1[2]. The algorithm can be modified in three ways: (1) modifying the co-occurrence statistic (directly, or indirectly through changing $H$), (2) changing the method for finding anchors, and (3) changing the objective function minimized by `Recover`.

**Algorithm 1:** AnchorModel

**Data:** $H \in \mathbf{R}_+^{V \times D}$
**Result:** $A \in \mathbf{R}^{V \times K}$, $B \in \mathbf{R}^{V \times K}$
1 $Q \longleftarrow$ `Cooccurrence`$(H)$
2 $\vec{p} \longleftarrow$ row-normalization factor of $Q$
3 $\bar{Q} \longleftarrow \frac{Q}{\vec{p}}$
4 $S \longleftarrow$ `FastAnchorWords`$(\bar{Q}, \Omega_A)$
5 $A, B \longleftarrow$ `Recover`$(\bar{Q}, S, \vec{p}, \Omega_R)$

The major contribution of the anchor method algorithm is that its computational complexity only depends on the parameters $K$ and $V$ after $Q$ has been estimated. Since $Q$ is a corpus statistic it does not change unless the underlying corpus is changed, meaning that it only needs to be calculated once and all subsequent model estimations become independent of corpus size.

### Extensions to the Anchor Method

Since anchor words are unique to each topic they can be seen as a label for the topic. Such a label however would only be interpretable if the chosen word is salient [25], which the anchors chosen through the method previously described may not be. An attempt to remedy this problem is to use a non-linear embedding designed for visualization, instead of the random projection used in `FastAnchorWords`. One such embedding is T-distributed Stochastic Neighbor Embedding (t-SNE) [26] which has been shown to produce more salient anchors, more coherent topics, and more unique topics [10]. The dimensionality of the embedding is generally selected to be small, between 2 and 4, which means that finding the convex hull can be done efficiently using *QuickHull* [27]. Contrary to the previous method, the convex hull is found exactly instead of approximately through a greedy algorithm. The intuition to why this method works well is that an embedding like t-SNE does not aim to preserve the magnitude of vector distances, it is instead designed to visualize the data in a meaningful way in low dimensions. This leads to extreme points being words which separate the data but are also not overtly rare, since in a visualization they should be interpretable. This method of finding anchors removes the hyperparameter $K$ since the convex hull of the embedded vectors is found exactly and varies with dimensionality, corpus, and random initialization of t-SNE. In the original paper, the authors do not investigate replacing random projection with t-SNE and running the greedy algorithm to find $K$ anchors. t-SNE is not as fast as a random projection but significant speed improvements have been made in recent years [28]. It is also unclear if the cardinality of the convex hull is correlated with the "optimal" value of $K$.

[2]Note that the algorithm does not recover $W$, this is done by fitting a LDA model with static topics given by $A$.

Sometimes a topic may be better captured by two or more words instead of a single anchor word. An anchor word which is a combination of multiple words is called a tandem anchor [9], and can be added as additional rows in $Q$. No modifications to the `Recover` algorithm have to be made. A tandem anchor, $\vec{s}$, for a set of words, $G$, is constructed as the harmonic mean of their corresponding rows in $Q$:

$$\vec{s}_i = \sum_{w \in G} \left( \frac{Q_{wi}^{-1}}{|G|} \right)^{-1} \tag{2.7}$$

This method was proposed in the context of interactive topic modeling where users were allowed to modify, combine, add, or remove anchor words in an attempt to improve the topic model. It was found that tandem anchors not only add interpretability to anchors themselves, but also improve the quality of the estimated topic model.

A common method within machine learning is the use of parameter regularization to avoid overfitting, and/or embed prior knowledge. This can be done by adding a regularization term to the objective function of a learning problem. The addition of Beta-regularization to the objective used in the `Recover` function has been shown to increase topic coherence [11]. Beta regularization is derived from using a Dirichlet prior common in LDA models. To optimize this new objective `L-BFGS` is used instead of the exponentiated gradient algorithm, and convergence of $B$ is checked by measuring the L2-norm between the estimations. The new objective function is:

$$\begin{gathered} B_i = \arg\min_{B_i} ||\bar{Q}_i - B_i \bar{Q}_S||_2 - \lambda \sum_{s_k \in S} \log \text{Beta}(A_{ik}; a, b) \\ a = \tfrac{x}{V} + 1,\ b = \tfrac{(V-1)x}{V} + 1,\ x > 0 \end{gathered} \tag{2.8}$$

This objective function is dependent on the value of $A$, which is the matrix that is to be recovered. To solve this issue, $A$ can be calculated from the value of the previous estimation of $B$. The regularization term can then be re-formulated as [29]:

$$\begin{gathered} \sum_{s_k \in S} (a-1) \log(T_i B_{ik}) + (b-1) \log([TB]_k - T_i B_{ik}) + (2-a-b) \log([TB]_k) \\ T = [T_1, \ldots, T_V] \\ T_i = \sum_{v=1}^{V} Q_{iv} \end{gathered} \tag{2.9}$$

To check converge, the current estimation, $B^{(i+1)}$, is checked against the preceeding estimation, $B^{(i)}$:

$$||B^{(i+1)} - B^{(i)}||_2 \leqslant 0.1 \tag{2.10}$$

## 2.3 Word Embeddings for Short-text Topic Modeling

A vocabulary can be represented using a vector space representation which encodes some relatedness between words as closeness in the vector space [30]. The rows of the matrices $Q$ and $\bar{Q}$ are such representations in which words that co-occur in the underlying corpus are close; these row-vectors are somewhat sparse and have a large dimensionality. A neural word embedding is a vector space representation of a vocabulary in which word vectors are dense and have relatively low dimensionality. Word embeddings, represented as matrix $E$, contains a dense low-dimensional row vector, $\vec{e}$, for each word in the vocabulary. These representations are able to capture semantic regularities between words, e.g. words such as *queen* and *king* are close within the vector space. Contrary to the co-occurrence representation, these words do not frequently occur next to each other, but they do occur in similar contexts. Semantic word embeddings even allow for solving analogy tasks using vector arithmetic, e.g. $\vec{e}_{king} - \vec{e}_{man} + \vec{e}_{woman} \approx \vec{e}_{queen}$.

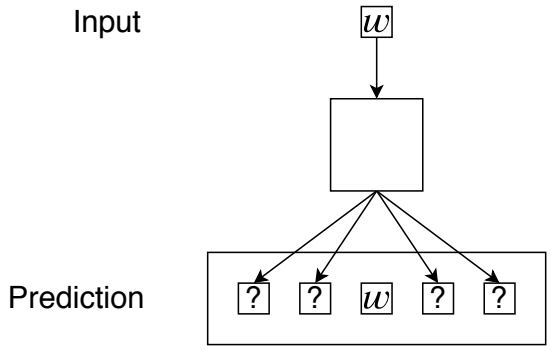



Figure 2.5: Graphical view of skip-gram prediction problem.

Neural word embeddings were initially learned by training a neural network on a skip-gram task [31]. The task gets its name from the prediction problem known as skip-gram, in which, given a word, the correct surrounding words are to be predicted (see Figure 2.5).

Negative sampling was introduced shortly afterwards, giving rise to the skip gram with negative sampling (SGNS) task. This modification of the skip-gram objective punishes the model when it predicts words which occur often according to the underlying distribution of words (a "noise" distribution) [32]. The new task resulted in much better word embeddings when the noise distribution was scaled appropriately. Further improvements have been made which capture the semantic relationships between words even better [33, 34].

Short-text corpora are document collections where the average document length is very short, e.g. tweets found on Twitter which limits the total number of characters to 280. Due to the short document lengths, the design matrix $H$ becomes extremely sparse, which results in a very noisy ground truth from which to estimate a topic model. Word embeddings learned using SGNS have been shown to perform an implicit matrix factorization [35]. This matrix factorization view of word embeddings have been used to improve coherence of NMF-based topic models on short-text corpora [13]. The SGNS view of the corpus can be used to pad the design matrix with words which have similar semantic meaning within the corpus. This method does not require pre-calculated word vectors learned on a separate corpus, but also does not get the semantic benefits gained when word embeddings have been trained on a very large set of documents.

High quality pre-calculated word embeddings learned using a corpus such as Wikipedia can be used to create a semantic vocabulary, $C$, of pseudo words (called CluWords in the original paper), by representing each word as a vector of its cosine similarity to every other word in the word embedding [14]. A threshold $\alpha$, known as the cosine threshold, is used to filter words which are too dissimilar from the representation.

$$C_{ij} = \begin{cases} \cos(\vec{e}_i, \vec{e}_j) & \text{if } \cos(\vec{e}_i, \vec{e}_j) \geqslant \alpha \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

The matrix $C$ has the same dimensionality as $Q$, but instead of capturing corpus co-occurrence it captures semantic closeness. The CluWords can be used to create two new design matrices which are much denser than the original design matrix, the term frequency matrix:

$$C_{\text{TF}}^T = H^T C \in \mathbf{R}_+^{D \times V} \tag{2.12}$$

and the TF-IDF matrix:

$$C_{\text{TF-IDF}}^T = C_{\text{TF}}^T \operatorname{diag}(\text{IDF}(C)) \in \mathbf{R}_+^{D \times V} \tag{2.13}$$

where the IDF of the CluWord matrix is defined as:

$$\text{IDF}(C) = \log\left(D \Big/ \sum_{1 \leqslant d \leqslant D} \left[ H_{\mathbf{B}}^T C^T \odot \frac{1}{H_{\mathbf{B}}^T C_{\mathbf{B}}^T} \right]_d \right) \in \mathbf{R}^V \tag{2.14}$$

This formulation of CluWord IDF is equivalent to the one presented in the original paper but more concise. $H_{\mathbf{B}}$ refers to the logical version of $H$, where any value greater than 1 is set to 1 (same for $C_{\mathbf{B}}$). The $\odot$ operator signifies the Hadamard product, i.e. element-wise multiplication. The matrix within the brackets is a $D \times V$ matrix, where $D$ is the number of documents, and $V$ is the size of the vocabulary. Each element of this matrix is the weight of the CluWord in the document scaled by the number of its constituents which appear in the document. The columns of this matrix are summed to create a $V$ dimensional vector of IDF values for each word.

Note that the term frequency matrix $C_{\text{TF}}$ is a denser version of the original design matrix $H$, and that both of the newly defined design matrices are non-negative. No papers, to our knowledge, have investigated anchor models with TF-IDF design matrices, and it is unclear whether the co-occurrence statistic used by the anchor method is valid when based on TF-IDF instead of TF.

## 2.4 What Makes a Topic Model Interpretable

There is no objective definition of what makes a topic model interpretable by humans, but certain metrics and intuitions can be used in an attempt to create a subjective definition. For this thesis, the selected qualities of an interpretable model are:

- **Coherence** - A coherent representation which when observed should naturally reflect a theme in the underlying text.
- **Specificity** - The topic should not summarize the entire corpus, each topic should reflect a specific theme within a subset of documents in the corpus.
- **Uniqueness** - Each topic should capture a unique theme not captured by other topics.

These qualities have all been used within the literature to find models which correlate well with human judgement, and a number of different metrics exist to measure these qualities.

### Topic Descriptor

Clearly, the interpretability of a topic model depends on how it is presented. A number of visualization tools have been developed and investigated in an attempt to figure out how to present topic models to the user [25, 36]. This thesis will not investigate any visualization techniques except for the top $M$ words representation necessary for certain metrics, known as the topic descriptor. These words are often selected according to their probability within the topic, but other orderings which may correlate better with human judgement exists. One such ordering is *relevance* [36], which is a combination of the word probability within the topic and *lift* [37]:

$$r_{\text{rel}}(w, k|\lambda) = \lambda \log(A_{wk}) + (1 - \lambda) \log \underbrace{\frac{A_{wk}}{p(w)}}_{lift} \tag{2.15}$$

Where $A_{wk}$ is the probability of word $w$ in topic $k$, and $p(w)$ is the probability of $w$ according to the underlying word distribution. The optimal value for $\lambda$ was determined to be 0.6 in the original study [36], suggesting that ordering by lift aligns better with human judgement.

Another topic descriptor, inspired by the TF-IDF weighting scheme, is defined as [38]:

$$r_{\text{TF-IDF}}(w,k) = A_{wk} \log \frac{A_{wk}}{\left(\prod_{k'=1}^{K} A_{wk'}\right)^{\frac{1}{K}}} \tag{2.16}$$

and has been shown to produce more coherent descriptors [39].

The choice of descriptor affects some of the following metrics such as coherence and uniqueness. It is therefore important to clearly state the topic descriptor used when evaluating the metric, and also to use the topic descriptor which is to be used for later visualization. Updating the model in an attempt to improve the metrics for one descriptor may disimprove the same metrics for another descriptor.

The standard probability based topic descriptor for a topic is ordered by the corresponding column in the topic matrix:

$$r(w,k) = A_{wk} \tag{2.17}$$

### Coherence

Topic models have historically been evaluated using statistical or extrinsic measures, either by evaluating performance on downstream tasks [40, 41], or by measuring predictive likelihood on a held-out data set [42]. These measures avoid looking under the hood of the topic model and, for the case of predictive likelihood, have been shown to negatively correlate with human interpretability [4]. In response to this discovery the task of finding an automatic evaluation metric which correlates well with human judgement was introduced [43]. These metrics are collectively known as coherence measures since they aim to predict how coherent the words in the topic descriptors are.

The process of finding coherence metrics generally involve performing large scale user studies where the users have to perform some task indicating the quality of a topic. The results are compared with the coherence measures to compute how well the metrics correlate with human judgement. The tasks range from simply rating the topics on how coherent they feel, to tasks such as word intrusion where a user has to determine which word in a topic does not belong.

Most popular coherence measures are based on word co-occurrence using either the original (underlying) corpus [5] or an external corpus such as Wikipedia [43]. In general, using an external corpus results in stronger correlation with human ratings [16]. The main coherence metrics are measured by aggregating the similarity of all words in the topic descriptor for each topic. These include:

1. $C_{\text{UMass}}$ - An asymmetrical measure based on log conditional probability where co-occurrence is calculated using document frequency [5].
2. $C_{\text{UCI}}$ - A point-wise mutual information (PMI) based measure using term co-occurrences estimated using a sliding window [43].
3. $C_{\text{NPMI}}$ - A measure which represents words as vectors using normalized PMI (NPMI) [44] (estimated in the same way as 2) and a similarity measure using cosine similarity [45].

The $C_{\text{UMass}}$ metric for a sorted topic descriptor is defined as [5, 16]:

$$C_{\text{UMass}} = \frac{2}{M(M-1)} \sum_{i=2}^{M} \sum_{j=1}^{i-1} \log \frac{p(w_i, w_j) + \epsilon}{p(w_j)} \tag{2.18}$$

Where $M$ is the descriptor cardinality. The word probabilities, $p(w_j)$, and joint probabilities, $p(w_i, w_j)$, are calculated as the document frequency of words in the original or an external

corpus. Generally, this metric uses the original corpus according to its initial definition. This metric has been shown to correlate with human ratings [5], is popular in the literature, but does not correlate as well as the other metrics mentioned [16]. The metric is heavily dependent on the size of the reference corpus to identify coherent topics, but can generally be used to identify incoherent topics [39]. The parameter $\epsilon$ is added to avoid taking the logarithm of zero and should be small ($\approx 10^{-12}$) [46].

$C_{UCI}$ was the first measurement shown to correlate with human ratings. It was found through a comparison of 15 metrics derived from the field of Natural Language Processing (NLP) whose corresponding metrics have been shown to correlate with lexical similarity [43]. For a topic descriptor the metric is defined as:

$$C_{UCI} = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \text{PMI}(w_i, w_j) \tag{2.19}$$

$$\text{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)} \tag{2.20}$$

The $\epsilon$ parameter is the same as in the $C_{UMass}$ coherence metric. The metric does not depend on the order of the words in the topic descriptor as $C_{UMass}$ does. The word probabilities are calculated using a sliding window, as opposed to document frequency for $C_{UMass}$. In its initial definition the sliding window was selected to be 10 [43] but further evaluation has shown stronger correlation using larger window sizes $\geqslant 50$ [16]. The $C_{UCI}$ metric was shown to perform better when PMI was replaced by NPMI[3] [45].

$$\text{NPMI}(w_i, w_j) = \frac{\text{PMI}(w_i, w_j)}{-\log p(w_i, w_j)} \tag{2.21}$$

The $C_{NPMI}$ metric (different from $C_{UCI}$ with NPMI) is based on distributional semantics using NPMI weighted word vectors [45]. Each element $\vec{w}_{ij}$ of a word vector $\vec{w}_i$ is the NPMI weight between word $w_i$ and word $w_j$. The features of this vector space are selected as the $M$ most probable topic words, resulting in $M$ dimensional vectors. The metric is calculated as the mean of the cosine similarities between the vector representations of words in the topic descriptor:

$$C_{NPMI} = \frac{2}{M(M-1)} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \cos(\vec{w}_i, \vec{w}_j) \tag{2.22}$$

$$\cos(\vec{w}_i, \vec{w}_j) = \frac{\vec{w}_i \cdot \vec{w}_j}{||\vec{w}_i||_2 ||\vec{w}_j||_2} \tag{2.23}$$

$$\vec{w}_{ij} = \text{NPMI}(w_i, w_j) \tag{2.24}$$

The $C_{NPMI}$ metric can be modified to use any other word vector representation, such as word embeddings learned from neural networks. Coherence measures where word embeddings are used instead of the NPMI vectors have had positive results [47, 39] but are not as common within the literature.

The impact of topic cardinality on coherence measures are generally ignored but have been shown to affect the metrics [48]. A proposed solution to this is to calculate an aggregate measure across different values of $M$.

[3] NPMI was investigated due to its usage in collocation extraction [44].

### Specificity

The specificity of a topic measures how different a topic is from the underlying word distribution. This metric is evaluated by measuring the KL divergence between the word-topic distribution and the underlying word distribution of the corpus [15]. Topic specificity is defined as:

$$\text{TS} = \frac{1}{K}\sum_{k} D_{\text{KL}}(A_{:k}||p_H) \tag{2.25}$$

Where $A_{:k}$ is the word distribution of topic $k$, and $p_H$ is the word distribution of the underlying corpus.

Originally, the metric was designed to identify "junk topics", i.e. topics which are incoherent and do not provide the user with any valuable information. The coherence metrics presented earlier have become the conventional metrics for measuring coherence, but a drawback is that they only evaluate the topic descriptor. A topic which simply reflects the underlying distribution may have good coherence but clearly would not be a topic which reflects a distinct theme.

### Uniqueness

The evaluation metrics so far have been local to each topic, measuring how coherent or specific a topic is. Maximizing such metrics can easily be done by simply repeating the best topic $K$ times. A perfect topic model should find $K$ *different* topics, which requires a global measure of uniqueness across topics. Since each topic is a distribution over words it is possible to measure their similarities using statistical measures. A global measure of dissimilarity can be defined as [10]:

$$\text{TD} = \max_{1\leqslant k\leqslant K} ||\frac{1}{K}\sum_{k'} A_{:k} - A_{:k'}||_2 \tag{2.26}$$

Where $A_{:k}$ is the word distribution of topic $k$. Distance between topic distributions is measured using euclidean distance, but any other distance metric would be valid.

Since topics are presented to the user using their topic descriptors, a natural measure of non-uniqueness is the Jaccard similarity (JS) between the descriptors [39]:

$$\text{JS} = \frac{2}{K(K-1)}\sum_{j=2}^{K}\sum_{i=1}^{j-1}\frac{|T_i \cap T_j|}{|T_i \cup T_j|} \tag{2.27}$$

Where $T_i$ are the $M$ words in the descriptor of topic $i$. Since $0 \leqslant \text{JS} \leqslant 1$, uniqueness can be defined as $1 - \text{JS}$.

## 2.5 Determining the Number of Topics

All topic models presented in this chapter assume that the number of latent topics, $K$, is known. This assumption is far from reasonable for a number of reasons:

- The corpus may contain "hidden" topics, unknown beforehand.
- The model may not be able to recover all topics deemed semantically unique by a human.
- Topic modeling is often used to reveal the underlying topics, meaning the user has no knowledge of what topics the corpus contains beforehand.

To alleviate this problem for LDA models a number of different metrics have been proposed [49, 50, 51]. These metrics attempt to measure how well separated the topics recovered by the model are. The assumption is that the natural number of topics can be identified when increasing or decreasing $K$ leads to worse separation. Either because decreasing $K$ forces the model to "spread" the removed topic across all other topics, or because increasing $K$ forces the model to split a topic into new topics which are alike. This method is inspired by previous work on trying to automatically find the natural size of ontologies [52].

The first two metrics simply measure the pairwise correlation of topics by measuring the cosine similarity [49], or the Jensen-Shannon divergence (JSD) [50], between the column vectors of $A$. The average pair-wise correlation distance (or similarity), according to some similarity metric $s$, is defined as:

$$\text{CD}_s = \frac{2}{K(K-1)} \sum_{j=2}^{K} \sum_{i=1}^{j-1} s(A_{:i}, A_{:j}) \tag{2.28}$$

The similarity metrics used in this thesis are cosine similarity and a similarity version of JSD ($1 - \text{JSD}$ since $0 \leqslant \text{JSD} \leqslant 1$). This metric should increase when a bad topic split or merge has been performed.

The metrics described above only make use of the topic distribution defined by $A$. However, the LDA model also estimates the document-topic distribution $W$, which can also be utilized when determining the natural number of topics [51]. If all topics are well separated then the column vectors of $A$ are orthogonal and their L2-norms are the singular values of the SVD of $A^T$. If these topics describe the corpus well then the singular values should be proportional to the magnitude of each topic in the corpus. The topic magnitudes can be calculated as $L \times W^T$, where $L$ is a vector containing the length of each document. The metric is defined as the symmetric KL divergence between the singular values of $A$, $\sigma_A$, and the topic magnitudes of the corpus:

$$\text{Arun} = D_{\text{KL}}\left(\sigma_A || L \times W^T\right) + D_{\text{KL}}\left(L \times W^T || \sigma_A\right) \tag{2.29}$$

This metric should reach its minimum around the optimal number of topics.

The metrics for determining the natural number of topics are all defined for the LDA model, which is not a correlated topic model. It is unclear whether the methods also apply to correlated topic models, such as the ones estimated using NMF or the anchor method. This is because the topics of a correlated topic model are not inherently well separated and therefore the metrics described above may not converge or reach an optima.

# 3 Method

This chapter introduces the data sets used for estimation and evaluation, the method for selecting hyperparameters, the method for incorporating word embeddings into the anchor-based estimation process, and the merging strategies for combining topics using tandem anchors.

## 3.1 Corpora

The corpora selected for this thesis were a combination of publicly available data sets, common within the literature on topic modeling, and data sets collected using Twitter's public APIs. These data sets were meant to capture a variety of different types and sizes of textual data, such as:

- Formal long-form documents, both large and small collections (NYT and NIPS).
- Informal short-text document (Twitter).
- Informal medium-length documents (NG20).

Topic modeling requires pre-processed data to achieve valuable results. This generally includes removing stop words, removing adverbs, filtering based on frequency, filtering based on document length etc. These pre-processing steps are generally corpus dependent, therefore a minimal selection of common pre-processing steps were selected for this thesis. Since the anchor method scales quadratically with vocabulary size it is important to restrict the number of word types in the post-processed data sets. All data sets were pre-processed to filter out:

- 318 stop words[1].
- Low frequency words occurring in less than 0.1% of documents.
- Words which contained digits, underscores, "non-word characters" (regex pattern `\W`), or were shorter than three letters[2].

[1] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
[2] Full regex: `\b[^\W\d_]\b{3,}`

Table 3.1: Corpus information before pre-processing. Word types were counted using the default tokenizer in `CountVectorizer` for the Twitter and 20 Newsgroups corpus. ADL denotes average document length.

| Corpus | Documents | Word Types | ADL | Source |
|---|---|---|---|---|
| NYT | 300,000 | 102,660 | 331.8 | [55] |
| NIPS | 1,500 | 12,419 | 1288.2 | [55] |
| Twitter | 1,000,000 | 288,153 | 13.8 | statuses/filter API |
| NG20 | 18,846 | 134,410 | 182.7 | scikit-learn |

Table 3.2: Corpus information after pre-processing. Word types were counted using the regex described earlier. ADL denotes average document length.

| Dataset | Documents | Word Types | ADL |
|---|---|---|---|
| NYT | 299,399 | 20,460 | 273.2 |
| NIPS | 1,491 | 11,911 | 1244.8 |
| Twitter | 464,065 | 7,655 | 9.6 |
| NG20 | 17,496 | 8,080 | 73.9 |

- Documents of size less than 5 after pre-processing.

The pre-processing was partly performed using the `CountVectorizer` class in the *scikit-learn* [53] library (version 0.22.1). The stop word list was selected since it was available by default in the library [54]. Information of all data sets pre-, and post-processing is available in Tables 3.1 and 3.2 respectively.

The NYT corpus, consisting of articles published in the New York Times, as well as the NIPS corpus, consisting of papers published at the Neural Information Processing Systems conference, were collected in BOW format, publicly provided by UC Irvine [55]. Both data sets contained documents written in formal English with high average document length. The data sets were also topical by nature. News articles generally deal with a small subset of topics, such as local or world politics, economy, or culture. NIPS papers all deal with topics within a specific field, with terminology overlap among the articles.

The NG20 data set, consisting of messages published to 20 different newsgroups, was provided by the scikit-learn library in a format which excludes headers, footers, and quotes from the documents. Newsgroups were a precursor to internet forums, a place where users could hold discussions around specific topics. This topical property of newsgroups make them suitable for topic modeling tasks involving downstream classification, since every document is associated with a specific topic. This thesis only used the data set for evaluating the topic models themselves, not their performance on the classification task.

The Twitter data set was collected using the statuses/filter API[3], selecting tweets categorized as English. The tweets were collected between 2020-03-02 and 2020-03-08, and only 20% of tweets published were recorded. Words were restricted to ones consisting of only ASCII and alphabetical characters, the hashtag symbol was stripped, and all words were lowercased. The *Spacy*[4] tokenizer was used to tokenize the tweets. Filtering words by minimum document frequency affected this data set particularly hard, reducing the number of documents by 72%, and the vocabulary size by 99.6%. Because of this the minimum document frequency was changed from 0.1% to 0.01% for this data set, resulting in signficantly less filtering, but still reducing the number of documents by more than 50%. Discussion of this is postponed to Chapter 5. The Twitter data set was not topical by design, but world news or current events may have induced topics across many authors.

[3]https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter
[4]https://spacy.io/

Table 3.3: The parameter settings used in the gensim LDA estimation process.

| Parameter | Value |
|---|---|
| chunksize | 2000 |
| passes | 1 |
| batch | False |
| alpha | symmetric |
| eta | None |
| decay | 0.5 |
| offset | 1 |
| eval_every | 10 |
| iterations | 50 |
| gamma_threshold | 0.001 |
| minimum_probability | 0.01 |

## 3.2 Baselines

For comparison against non anchor-based topic models three baselines were used, LDA [1] estimated using the *gensim*[5] [56] library, NMF [3] estimated using scikit-learn, and CluWords [14] also estimated using scikit-learn. The anchor method with no enhancements was also included as a baseline, referred to as the unmodified anchor method.

### LDA

The parallelized LDA implementation[6] of the gensim library was based on an online version of variational Bayes [57] designed to handle massive document collections. Gensim was picked to estimate the LDA model since it was one of the most popular LDA implementations in the Python ecosystem. The default parameters of the implementation were preferred (see Table 3.3 for a complete list of relevant parameter settings). The hyperparameters "iterations" and "passes" were increased four fold for the NIPS data set and two fold for the NG20 data set. This was due to the corpus having few documents, which resulted in poor document convergence with default parameter values.

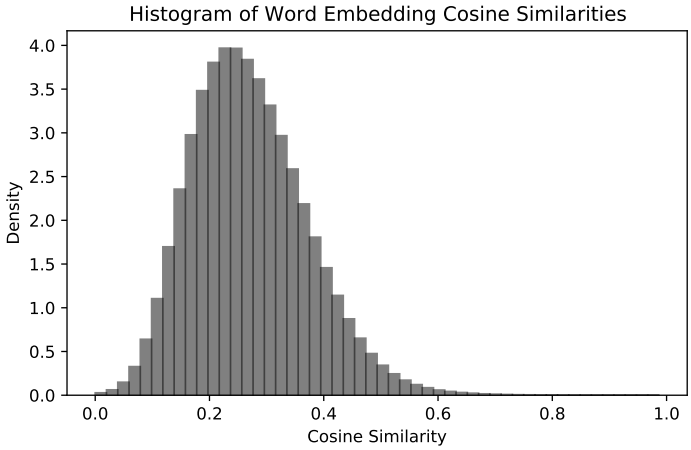
### NMF

Scikit-learn implemented two solvers for NMF, one based on coordinate descent [58], and the other based on multiplicative updating [59]. The default solver is the one based on coordinate descent and was therefore the one used in this thesis. The solver was run using the default parameters (see Table 3.4 for a complete list of the relevant parameters). Note that both the anchor method and these solvers attempted to solve the same problem. However, the factorization matrices estimated were not expected to be identical (or even similar) since the NMF of a given matrix is not unique, and both methods only find an approximation.

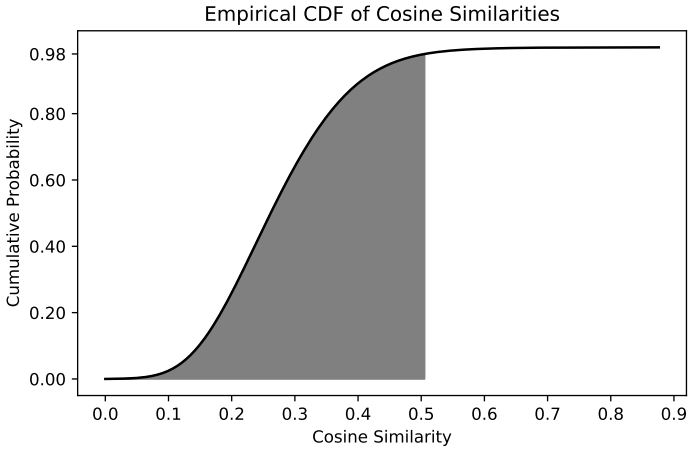
### CluWords

The CluWords baseline only required an NMF solver whose input was the $C_{\text{TF-IDF}}$ design matrix. The word embeddings used for creating the CluWord vocabulary were the publicly available[7] fastText vectors of dimension 300, trained on the Wikipedia 2017 corpus. Words which occurred in the vocabulary of the original corpus but did not exist in the embedding space were assigned a unit vector in the CluWord vocabulary, i.e. out-of-vocabulary words

[5]https://radimrehurek.com/gensim/
[6]https://radimrehurek.com/gensim/models/ldamulticore.html
[7]https://fasttext.cc/docs/en/english-vectors.html




(a) Histogram of the pair-wise absolute cosine similarities between the randomly sampled word embedding vectors.




(b) Empirical CDF with $\approx$ 2% of the most similar words in the unshaded region.

Figure 3.1: Graphs used for selecting the cosine threshold selection.

were simply represented as themselves. In the original paper, the threshold used during the vocabulary construction was set to 0.4 in order to capture $\approx$ 2% of the most similar words [14]. To determine the threshold for the word embeddings used in this thesis, the word vectors were randomly sampled and the pair-wise cosine similarities were collected (see Figure 3.1a for a histogram of the cosine similarities sampled). The threshold was set to 0.5, according to the empirical cumulative density function, to capture $\approx$ 2% of the most similar words (see Figure 3.1b).

### Unmodified Anchor Method

The anchor method required the following hyperparameters to be selected: anchor threshold, subspace dimension for random projection, and recovery tolerance. Previous literature gave some guidance as to the magnitude of these parameters but give no framework for selecting them for any given corpus. The first two parameters, anchor threshold, and subspace dimension, were formulated such that they become less corpus dependent.

The anchor threshold controlled which words were eligible in the anchor word selection process. In previous work, the anchor threshold was set to a discrete value, such as 3 [10] or 500 [60], controlling how many documents a word has to appear in to be an eligible an-

Table 3.4: The parameter settings used by the scikit-learn NMF solver.

| Parameter | Value |
|---|---|
| init | None |
| solver | cd |
| beta_loss | frobenius |
| tol | 0.0001 |
| max_iter | 200 |
| random_state | None |
| alpha | 0 |
| l1_ratio | 0 |
| shuffle | False |

chor word. For this thesis the anchor threshold was formulated as a proportion, e.g. an anchor threshold of 90% meant words which occurred in 90% of documents were eligible as anchor words. Topics produced by the anchor method, especially when topic count was small, depended highly on the value of this threshold [11]. A low threshold resulted in eccentric anchor words, while a high threshold resulted in words for which the anchor word assumption did not hold. Anchors which were too eccentric also broke the anchor word assumption, since they did not belong to any particular topic. The threshold was clearly corpus dependent, which is why it was formulated as a proportion instead of a set discrete value.

To select an appropriate subspace dimension for the random projection, used by the `FastAnchorWords` algorithm, the `johnson_lindenstrauss_min_dim` function (available in the scikit-learn library) was used. The function is based on the Johnson–Lindenstrauss lemma, which for given number of samples, $V$, and a distortion rate $\epsilon$, gives the minimum dimensionality, as:

$$\text{dimensionality} \geqslant \frac{4 \log V}{\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}} \tag{3.1}$$

This parameter had previously been selected around 1000 [60], which for vocabulary of size 3,000 would indicate a distortion rate of $\approx$ 28%. However, the applicability of two dimensional t-SNE embeddings as a projection space may indicate that this distortion rate could be much higher, which would lead to better performance.

The recovery tolerance is recommend to be set small, between $10^{-6}$ [24] and $10^{-10}$ [60]. This parameter greatly impacts the time of the estimation process but may not greatly affect the outcome, and should therefore be selected as large as possible within the range. This parameter was set to $10^{-6}$ for all experiments in this thesis.

## 3.3 Design Matrix with Word Embeddings

The process of creating the CluWord vocabulary used with the anchor method was the same as the one used for the baseline described earlier in the chapter. Co-occurrence estimation has the computational complexity of $O(Dd_{\text{ADL}}^2)$, where $d_{\text{ADL}}$ denotes the average document length. The threshold parameter used when creating the CluWord vocabulary greatly affects the resulting design matrix density, and therefore greatly increases $d_{\text{ADL}}$. For the experiments made in this thesis the threshold was set higher than 0.5, such that the co-occurrence calculation could be performed in a reasonable time.

The TF design matrix generated by the CluWord vocabulary was normalized such that the smallest non-zero value was 1. This was done to avoid negative results from the co-occurrence estimation process. To incorporate the word embeddings into the anchor method, the normalized TF design matrix, $C_{\text{TF}}$, replaced the BOW matrix, $H$, as the input of Algorithm 1.

## 3.4 Regularization with t-SNE-anchors

Anchor-based optimization with regularization and anchor words selected using t-SNE also required a set of hyperparameters. The implementation used the t-SNE embedding available in *openTSNE* [61]. The objective function was minimized using the *scipy* [62] library.

The t-SNE method for selecting anchors required a subspace dimension to be selected. The library used for calculating the embedding, openTSNE, breaks down for dimensions higher than 2 since the optimization methods used were not designed for embedding in higher dimensions. Estimating a t-SNE embedding, even for a lower dimensionality such as 2, was expensive in the context of an efficient method. Therefore, no dimensionality higher than 2 was evaluated for the experiments.

The Beta regularization required two hyperparameters to be set, the prior ($x$) and the regularization coefficient ($\lambda$). The parameter $x$ controls the characteristics of the Beta distribution, as described in the Theory chapter (see Definition 2.8), while $\lambda$ controls the amount of regularization to be applied. When either parameter is set to 0 no regularization is performed. The prior parameter, $x$, was set to 1 as in the original paper [11]. This thesis presents results for different values of $\lambda$ for all data sets.

To minimize the Beta regularized objective function, the `SLSQP` algorithm was used, available in the scipy library. `SLSQP` is a constrained and bounded optimization method, allowing for the stochasticity constraint of the $B$ matrix to be defined. The original paper [11] used `L-BFGS` to optimize the objective, which is not a constrained optimization method. It is unclear how the constraints were imposed for the method, therefore a constrained method was used for this thesis which likely came at the expensive of longer estimation times.

The original paper also did not investigate Beta regularization for L2 objective, instead using the KL objective found in the anchor method paper [8]. The KL objective results in much longer estimation times when compared to the L2 objective, and has since been excluded in later papers presenting the anchor method [63]. In this paper, Beta regularization was applied to the L2 objective.

## 3.5 Tandem Anchor Optimization

Optimization using tandem anchors was performed by merging the anchor words of previously estimated topics. The process for selecting the natural number of topics, described in the Theory chapter, used correlation distance as the guiding metric. The same tactic was used for this optimization process, where the objective was to go from $K$ to $K'$ topics by merging similar topics. The pair-wise correlation distances were calculated using cosine similarity, resulting in a list of topic pairs sorted by correlation. To perform the merging of topics, two strategies were employed to go from $K$ to $K'$ topics:

1. `Unique` - Each pair of topics were merged, avoiding pairs where either topic has already appeared in a merge. See Algorithm 2 for the full algorithm.

2. `Many` - Each pair of topics were merged such that, if one of the topics had already been included in an upcoming topic merge, then the other topic's anchor was also added to the planned merge. If both topics were included in different merges, then the two disjoint sets of anchor words were merged into one set. See Algorithm 3 for the full algorithm.

The first merging algorithm, `Unique`, resulted in tandem anchors which only consisted of pairs of the previously available anchors. This meant that tandem anchors would only include two real words within the vocabulary after the initial merging phase. But tandem anchors, which appears as "pseudo words" within the vocabulary after the initial merging phase, may have been merged with real words or with each other in subsequent merging phases. Note that if there were four topics which were exactly the same they would result

in two merged topics, as opposed to a single topic, in a single merge phase. Therefore the strategy could reduce $K$ by at most half in a single merge phase.

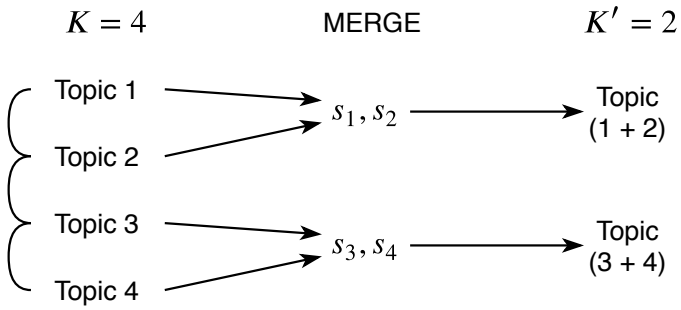
The second merging strategy, `Many`, was more general and had no limit as to how many words could be included in a tandem anchor for a single merge phase. For the example given above, a result could be a single new topic consisting of all anchors which generated the four alike topics (see Figure 3.2 for a visual comparison of the difference). This strategy could reduce the number of topics to 1 for any initial value of $K$. The `Unique` strategy would have needed at least two merging phases to combine the four topics into a single new topic. At first glance the `Unique` strategy may have seemed worse than `Many` since it was more limited. However, it is important to note that tandem anchors containing many words were less likely to fulfill the anchor word assumption. This was because the more words which were included in the tandem anchor, the less unique the actual tandem anchor became. Such a property may have resulted in better results for the more limited strategy. Other strategies for merging anchors surely existed but were not investigated.

**Algorithm 2:** `Unique` strategy for merging anchors.

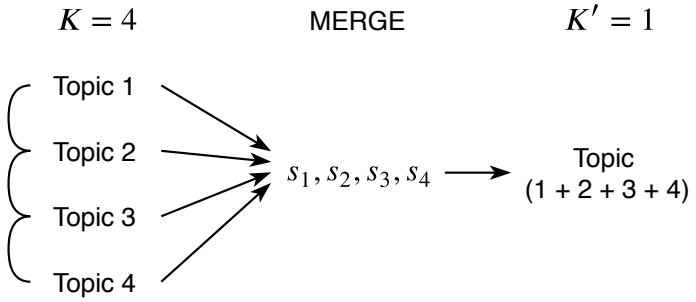
**Data:** SortedTopicPairs, $S$, $K$, $K'$
**Result:** TandemAnchorWords, RemovedAnchors
RemovedAnchors $\longleftarrow$ {}
TandemAnchors $\longleftarrow$ {}
**for** $k_1, k_2 \in$ *SortedTopicPairs* **do**
  **if** $k_1, k_2 \notin$ *RemovedAnchors* **then**
    TandemAnchorWords $\longleftarrow$ TandemAnchorWords $\cup\{(k_1, k_2)\}$
    RemovedAnchors $\longleftarrow$ RemovedAnchors $\cup\{k_1, k_2\}$
  **end**
  **if** $K + |\textit{TandemAnchorWords}| - |\textit{RemovedAnchors}| = K'$ **then**
    break
  **end**
**end**

**Algorithm 3:** `Many` strategy for merging anchors. `UniqueTandems` returns a set of sets containing the unique tandem anchor words of length $> 1$.

**Data:** SortedTopicPairs, $S$, $K$, $K'$
**Result:** TandemAnchorWords, RemovedAnchors
RemovedAnchors $\longleftarrow$ {}
AnchorMap $\longleftarrow$ Map ($k_i \rightarrow \{k_i\}$)
**for** $k_1, k_2 \in$ *SortedTopicPairs* **do**
  **if** *AnchorMap*[$k_1$] $\neq$ *AnchorMap*[$k_2$] **then**
    **for** $k_i \in$ *AnchorMap*[$k_1$] **do**
      AnchorMap[$k_i$] $\longleftarrow$ AnchorMap[$k_1$] $\cup$ AnchorMap[$k_2$]
    **end**
    **for** $k_i \in$ *AnchorMap*[$k_2$] **do**
      AnchorMap[$k_i$] $\longleftarrow$ AnchorMap[$k_1$] $\cup$ AnchorMap[$k_2$]
    **end**
    RemovedAnchors $\longleftarrow$ RemovedAnchors $\cup\{k_1, k_2\}$
  **end**
  **if** $K + |$ `UniqueTandems(AnchorMap)` $| - |\textit{RemovedAnchors}| = K'$ **then**
    break
  **end**
**end**
TandemAnchorWords $\longleftarrow$ `UniqueTandems`(AnchorMap)




(a) The `Unique` strategy could merge the four topics to at least two new topics in a single merge phase.




(b) The `Many` strategy was able to merge the four topics in a single merging phase.

Figure 3.2: Illustration of the potential differences between merging strategies for an initial merge when four topics were all alike. The edges between topics symbolize that they were strongly correlated.

To evaluate the effectiveness of merging anchor words based on topic correlation, a number of topic sequences were generated. A topic sequence is a descending sequence of topic counts from an initial value, $K_0$, to a final value, $K_n$. The sequence is defined by the initial and final value, and a ratio $0.5 \leqslant r < 1$, and constructed such that $K_i = \max(\lceil r \times K_{i-1} \rceil, K_n)$. As an example, the parameters $\{K_0 = 60, K_n = 20, r = 0.5\}$ generated the sequence $\{60, 30, 20\}$. The chains used for evaluation were generated by the following parameters:

$$\underbrace{\{160, 120, \ldots, 60\}}_{K_0} \times \underbrace{\{100, 80, \ldots, 20, 10\}}_{K_n} \times \underbrace{\{0.5, 0.66, 0.75\}}_{r} \tag{3.2}$$

This resulted in a number of chains, which were tested for each merging strategy for each data set. The chains were tested by evaluating the anchor method for each $K_i$ in the sequence. An initial model was calculated for $K_0$ using anchor words obtained through `FastAnchorWords`, while each subsequent model in the sequence used anchor words produced by the merging strategy and the previous model in the sequence. The model quality was evaluated at each step.

## 3.6 Evaluation

All coherence metrics were measured using co-occurrence statistics estimated from a reference corpus or the original corpus. The quality of the metrics depended highly on the size of this reference corpus, therefore the Wikipedia corpus (version 20200120[8]) was used when an external corpus was required. This corpus contained roughly 6 million articles written in formal English, and the choice of Wikipedia as reference corpus was common within the literature [43]. When a reference corpus was used, the co-occurrence statistic was measured using

[8]https://dumps.wikimedia.org/enwiki/20200120/

a sliding window of size 110 [16]. Documents were not merged such that the window could slide across documents since no indication of such a process had been made in the literature.

To determine the vocabulary of the reference corpus and parse each Wikipedia article, the library gensim was once again used. The library filtered articles with less than 50 words, performed tokenization, and lower-cased words. No other normalization or filtering was performed, which resulted in 2,006,500 unique word types. The co-occurrences were saved as a $2006500 \times 2006500$ upper triangular sparse matrix, containing 4,343,028,872 values, resulting in a sparsity of $\approx$ 99.89%. Co-occurrence probabilities were calculated by normalizing by the sum of the matrix. Word occurrences were normalized by the total number of occurrences. The total number of word occurrences in the reference corpus was 2,715,740,865, and the total number of word co-occurrences was 260,629,856,355. The most common word in the corpus was the word "the", which occurred 183,475,338 times, It was also the most common co-occurring word, with a co-occurrence count of 1,418,243,132 with itself. This resulted in the following probabilities:

$$p(\text{the}) = \frac{183,475,338}{2,715,740,865} \approx 0.068$$

$$p(\text{the}, \text{the}) = \frac{1,418,243,132}{260,629,856,355} \approx 0.0054$$

These values were stated to improve reproducibility and aid the understanding of metrics which use a reference corpus for evaluation.

When a word contained in a topic descriptor was out-of-vocabulary it was removed for the descriptor without replacement. The topic cardinality was updated to reflect the removal of a word so as to not affect the metric negatively when containing out-of-vocabulary words.

The coherence metrics also depended on the choice of topic descriptor [39], and its cardinality [48]. In general, the topic descriptor used during evaluation is standard probability ordered descriptor, unless stated otherwise. Both coherence and similarity metrics were averaged over three values of topic cardinality, 5, 10, and 20. Unless stated otherwise the coherence metric used to determine model quality is the $C_{NPMI}$ metric, shown to correlate strongest with human judgement [16].

# 4 Results

In this chapter the results for each of the baseline models, as well as the enhanced models are presented. Model quality was measured by the metrics $C_{NPMI}$ (coherence based on distributional semantics), topic specificity, and uniqueness ($1 - JS$). Coherence and uniqueness were calculated for the probability ordered topic descriptor, and both were averaged over multiple descriptor cardinalities. For all models, model quality is presented as a function of topic count. Model quality may be averaged over multiple hyperparameter settings and if so it is referred to as average model quality.

## 4.1 Baselines

The model quality as a function of topic count is presented for each baseline for comparison against the enhancements presented later. Two standard baselines, LDA and NMF, are presented first, followed by a hyperparameter investigation of the unmodified anchor method. Results for the three baselines is presented in Figure 4.1 using the same scaling for the quality metrics to allow for easy comparison. Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

### LDA

The LDA baseline was evaluated for topic counts ranging from 10 to 200 with increments of 5. Results, graphed as the model quality as a function of topic count, are presented in the first column of Figure 4.1. The Twitter data set resulted in a topic matrix for which specificity was undefined, which is why it is missing in the figure. For the other data sets topic specificity gradually increased with topic count. Uniqueness increased and coherence generally decreased as the number of topics increased. For the Twitter data set, coherence of topic descriptors declined sharply as topic count increased.

Correlation metrics and Arun score, used to guide the selection of the topic count parameter, are presented in the first column of Figure 4.2. The correlation metrics increased periodically due to splitting well separated topics into subtopics with higher correlation. The results from the Arun metric are very erratic but could be used to select an appropriate topic count value in certain regions for all data sets except Twitter.

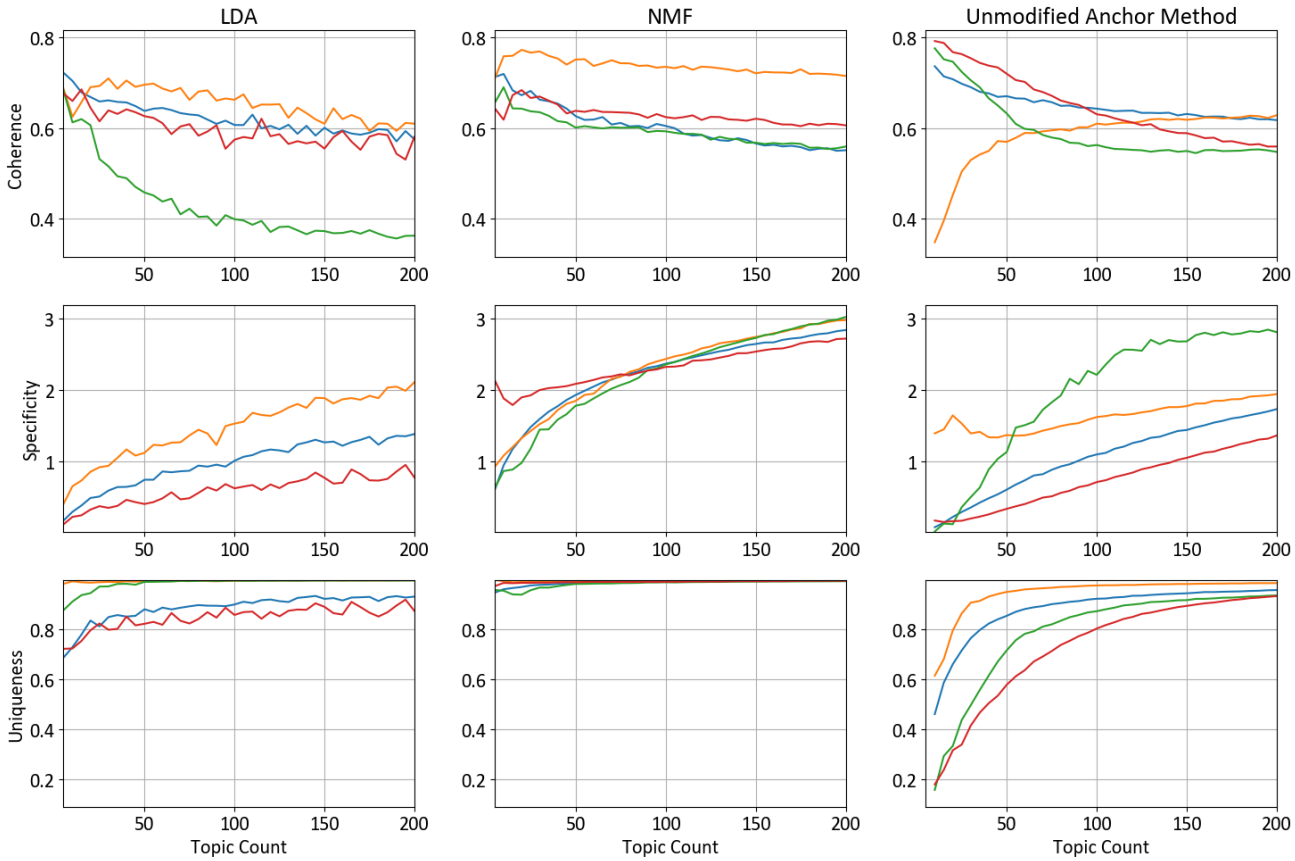



Figure 4.1: Model quality results for the baseline topic models. Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

### NMF

The NMF baseline was evaluated in the same way as the LDA model, and the results are presented in center column of Figure 4.1. Similarly to the LDA model, specificity and uniqueness increased with topic count, while coherence decreased. Topic uniqueness remained high across all topic counts, contrary to the LDA model where it was slightly lower when fewer topics were recovered. The minimum uniqueness value for the NMF baseline was approximately 0.94, while the LDA baseline recorded a minimum of 0.70. Both baselines produced topic descriptors of roughly the same coherence for all data sets except Twitter, where the NMF baseline performed better for higher topic counts.

The results for the metrics used to guide selection of topic count are presented in the center column Figure 4.2. Unlike the LDA baseline, NMF produced topics for which correlation monotonically decreased with topic count. The Arun metric did not produce interpretable results for this baseline, and exhibited sudden spikes for the NIPS and NYT data sets.

### Unmodified Anchor Method

The unmodified anchor method selected anchors using `FastAnchorWords` with the standard BOW design matrix as input. Topics were recovered using the unregularized recovery method. The hyperparameters investigated for this baseline were topic count, anchor threshold, and distortion rate.

The average model quality as a function of topic count is presented in the last column of Figure 4.1. Quality was averaged over all tested anchor threshold values. Across all data sets,

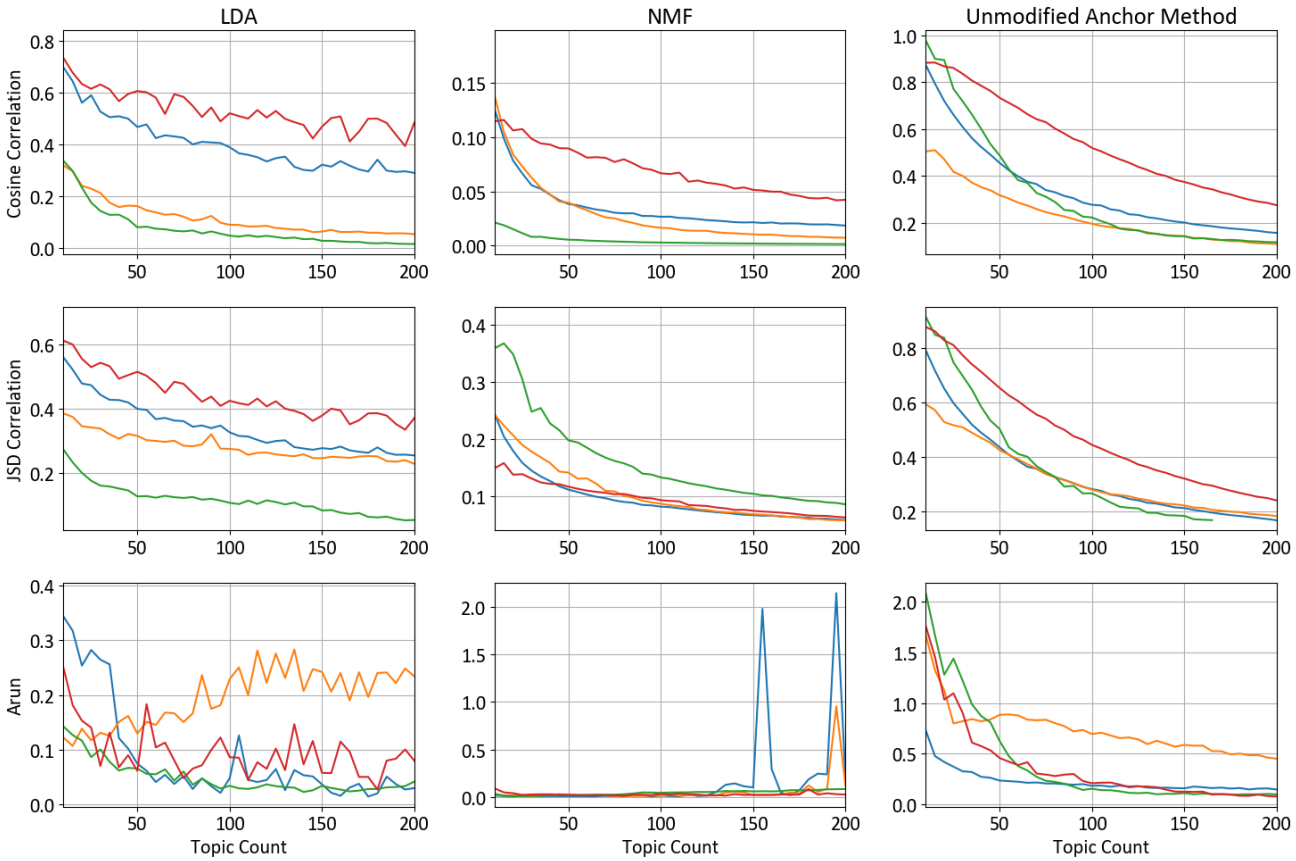



Figure 4.2: Cosine and JSD correlation should converge or reach an optima at the optimal topic count. Arun score should reach its minimum at the natural number of topics. JSD correlation has been changed from a distance to a similarity measure to match cosine correlation. Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

specificity and uniqueness increased with topic count, as was the case for the other baselines. Coherence also decreased with topic count for all data sets except for the NYT data set. The anchor method failed to produce unique topics when topic count was low, and the method exhibited much lower uniqueness scores when in the lower half of the topic count range (10 to 100).

Selecting topic count based on correlation distance or Arun score proved difficult. The results of these metrics as a function of topic count is shown in the last column of Figure 4.2. Arun score reached a minimum only in the case of the Twitter data set, and the correlation metrics all decreased as topics were added.

The average model quality as a function of anchor threshold is presented in Figure 4.3 for different ranges of topic count in each column. In general, results indicate that the method is more sensitive to this hyperparameter when topic count is low. The results show that the NG20 data set was most sensitive to this hyperparameter with regards to coherence and uniqueness. A higher anchor threshold resulted in topic descriptors with higher uniqueness and specificity. Coherence only improved with a higher anchor threshold for some data sets, while it regressed for the NG20 data set. When topic count was low, uniqueness was affected more by anchor threshold; a low anchor threshold in combination with a very low topic count resulted in topics which were practically identical for all data sets but NYT.

Model quality as a function of distortion rate is presented in Figure 4.4 for the data sets NIPS and NYT. The results showed that the distortion rate of the random projection did not

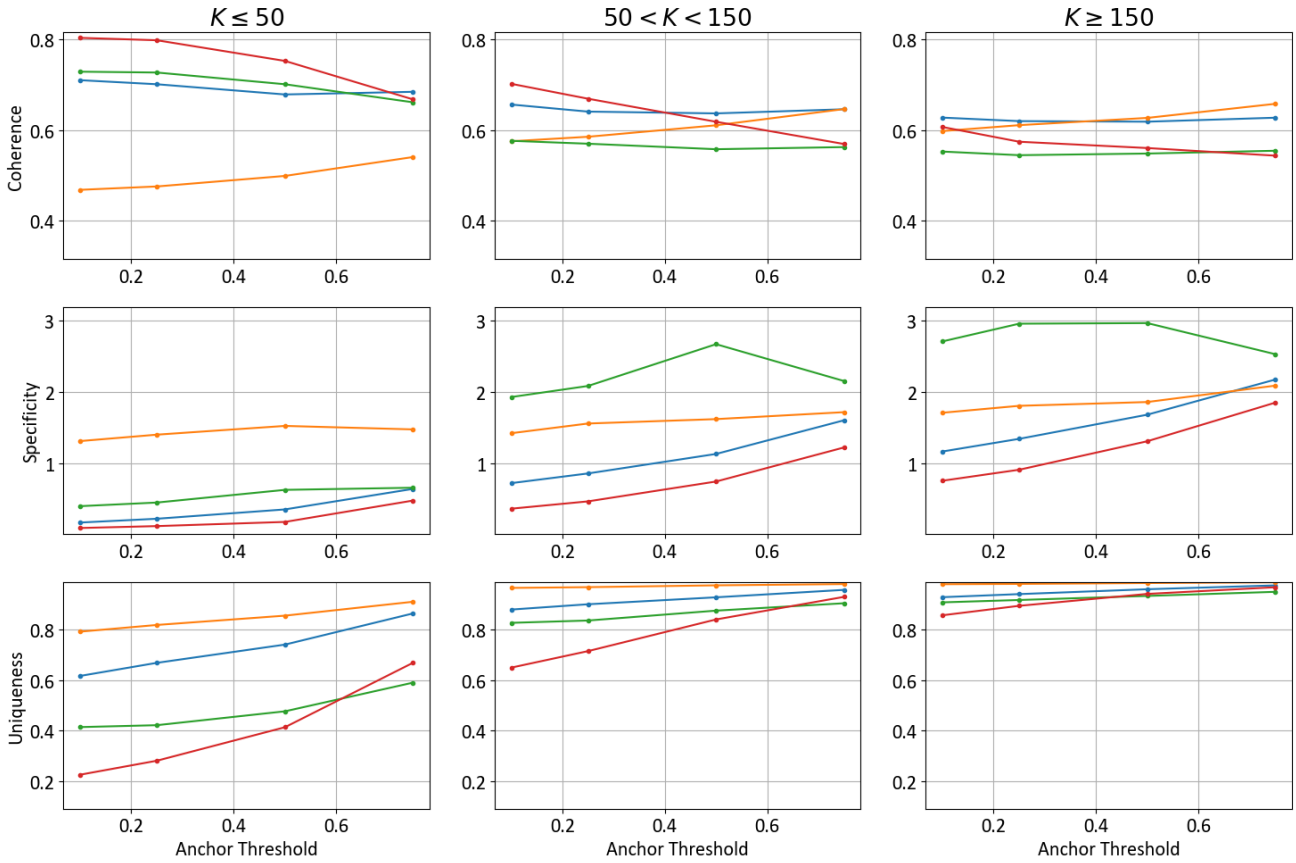



Figure 4.3: Average model quality of the standard anchor method as a function of anchor threshold. The columns show model quality for different ranges of topic count ($K$). Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

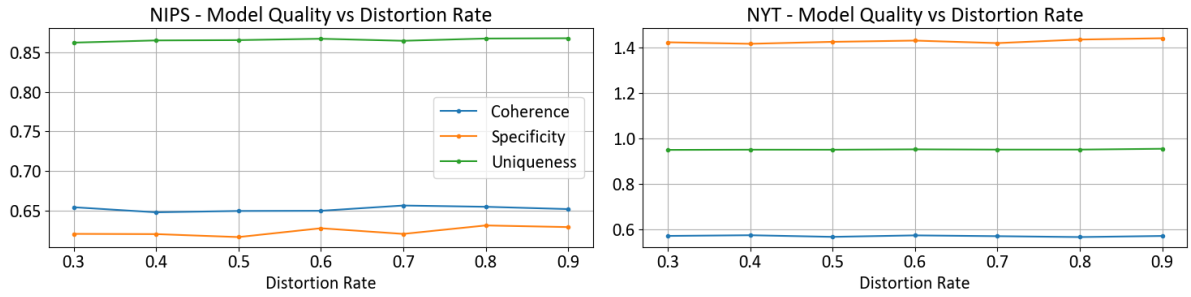



Figure 4.4: Average model quality of the standard anchor method as a function of distortion rate for the NIPS and NYT data sets.

affect model quality in the range tested. Therefore distortion rate was set high (0.8) for all following models which used `FastAnchorWords` to recover anchors.

## 4.2 Word Embeddings

Word embeddings were used to enhance the NMF baseline and anchor method by increasing the density of the design matrix through a CluWord dictionary. The enhanced methods are presented as CluWord baseline and CluWord anchor method respectively. The CluWord baseline used the $C_{\text{TF-IDF}}$ design matrix as input to the NMF method, while the anchor method

used the $C_{TF}$ design matrix. Results for the the two enhanced methods, as well as the unmodified anchor method with anchor threshold set to 0.5, is presented in Figure 4.5.

### CluWords Baseline

Model quality as a function topic count for the CluWord baseline is presented in the center column of Figure 4.5. Tests were run for topic counts in the range from 10 to 100, with an increment of 10. The increment was selected as 10, as opposed to 5 as in other tests, due to the increased estimation time of the higher density design matrices. For all data sets except NYT, the cosine threshold was set to 0.5, as explained in the previous chapter. Due to memory limitations however, the tests run on the NYT data set used a cosine threshold of 0.6.

The CluWord baseline produced models of significantly higher quality across all topic counts and quality metrics for all data sets, when compared to the NMF baseline. Additionally, the CluWord baseline exhibited a significantly higher specificity score for low topic counts. Uniqueness, as in the NMF baseline, was high across all topic counts and even higher than the aforementioned model. As topic count increased, so did specificity, while coherence decreased. However, the lowest measured coherence score for each data set was still about as high as the highest score measured by the other baselines.

The increased density of the design matrix resulted in significantly higher estimation times, generally two to five times as long. This may have been exacerbated by the high variance of estimation time exhibited by the scikit-learn NMF solver, due to random initialization.

### Anchor Method with CluWord Vocabulary

The standard anchor method with the CluWord TF matrix as input was evaluated with anchor threshold set to 0.5 and distortion rate set to 0.8. Results are presented in the last column of Figure 4.5. Note that the experiments were run using a higher cosine threshold (0.6) than the CluWords baseline, for reasons described in the previous chapter.

Word embeddings had a similar effect on the results as in the CluWord baseline. In particular, the CluWord anchor method exhibited significantly higher uniqueness scores when topic count was low. E.g. for a topic count of 20, the unmodified method had a uniqueness score of 0.20 for the NG20 data set, while the CluWord enhanced anchor method had a score of 0.87. The enhanced method produced less coherent topics for all but the NYT data set when topic count was low, as compared to the anchor method baseline. However, the unmodified anchor method produced topics with extremely low uniqueness in these ranges, i.e. it repeated the same coherent topic many times.

Using word embeddings to increase the density of the design matrix has a large performance impact on the co-occurrence estimation. The approximate impact on estimation time is presented in Table 4.1. The NYT data set was measured using a higher cosine threshold since setting it to 0.6 caused memory thrashing during co-occurrence calculation. Note that the co-occurrence statistic is calculated once per data set, not once per model estimation. Therefore subsequent model estimations were as fast as the unmodified anchor method.

## 4.3 Regularized Objective with t-SNE Anchors

The regularized model with t-SNE anchors was evaluated for a range of regularization coefficients. The results for each data set, including results with no regularization (coefficient set to 0), is presented in Figure 4.6. Also presented, as dashed lines, is the comparative performance of the anchor method at the closest topic count value with anchor threshold set to 0.5. Anchor words obtained from the convex hull of the t-SNE embedding for different anchor thresholds, resulted in average values of topic count presented in Table 4.2.

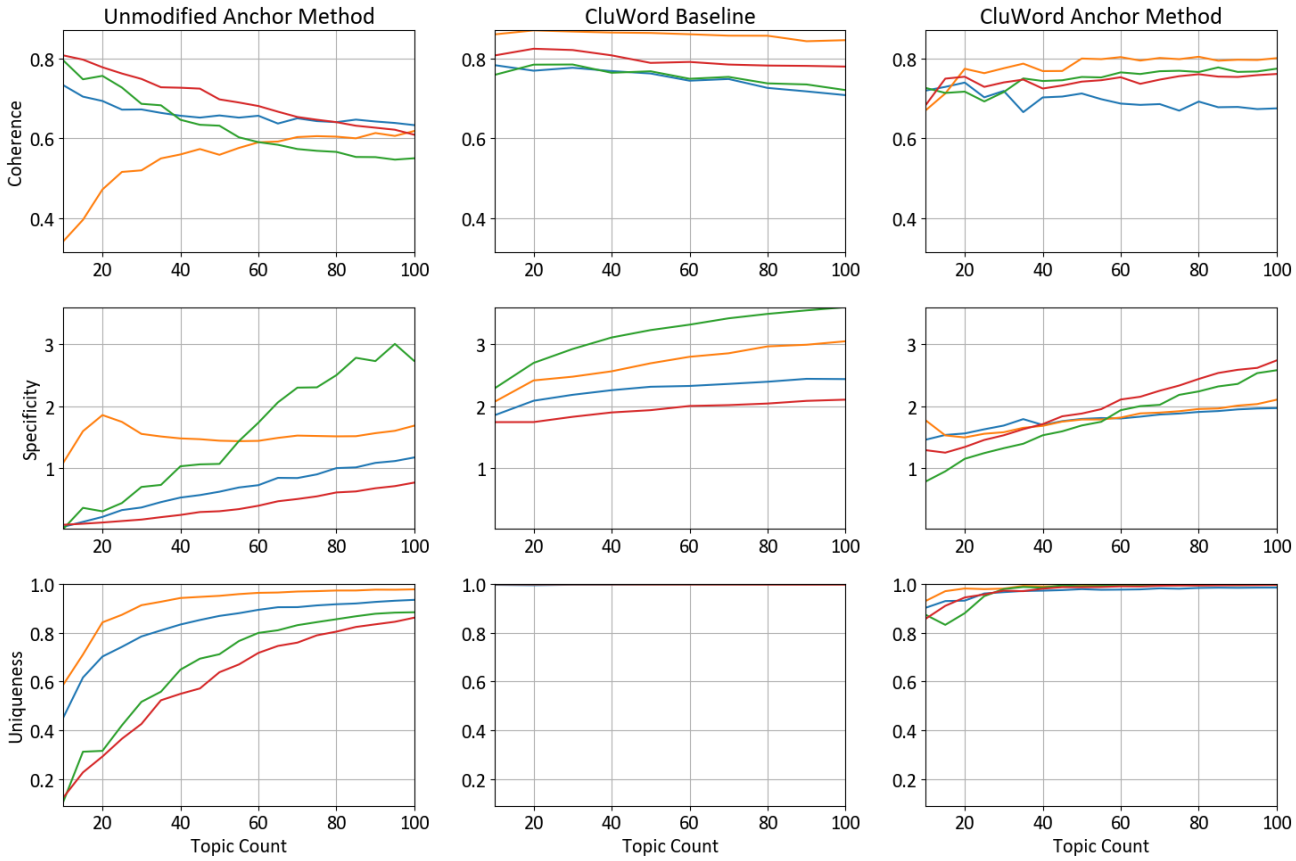



Figure 4.5: Model quality as a function of topic count for the models enhanced with word embeddings. The first column shows the results of the unmodified anchor method with anchor threshold set to 0.5 for fair comparison. For the CluWord baseline the cosine threshold was set to 0.5 for all data sets, except NYT for which it was set to 0.6. For the CluWord anchor method the cosine threshold was set to 0.6. The uniqueness score for the CluWord baseline was very close to 1 for all measured topic counts. Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

Table 4.1: Approximate impact of design matrix density on co-occurrence estimation time for the unodified anchor method (UAM) and CluWord anchor method (CAM). Times were measured as wall clock time on an Intel Xeon Processor E3-1245 v5.

| | | Matrix Density | | Estimation Time | |
|---|---|---|---|---|---|
| Data set | Cosine Threshold | UAM | CAM | UAM | CAM |
| NIPS | 0.6 | 4.1% | 26.0% | 7s | 50s |
| NYT | 0.7 | 1.2% | 3.2% | 100s | 400s |
| Twitter | 0.6 | 0.1% | 1.6% | 3s | 50s |
| NG20 | 0.6 | 0.6% | 6.4% | 4s | 30s |

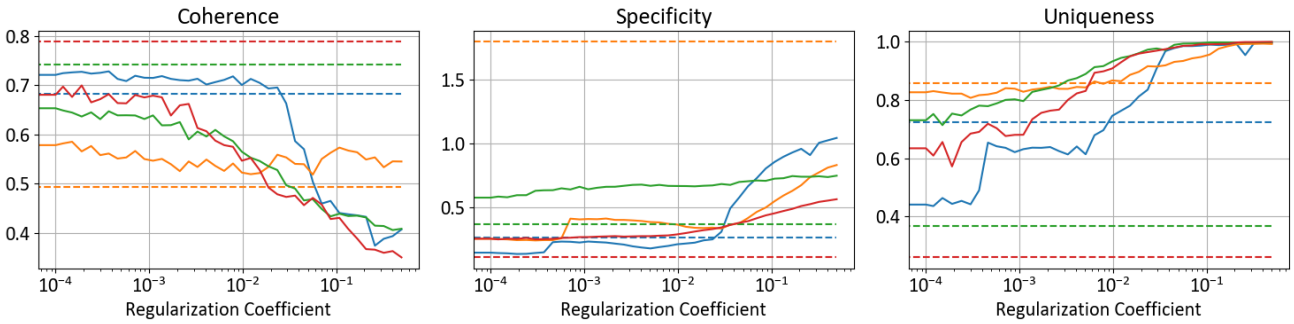



Figure 4.6: Average model quality as a function of regularization coefficient for a single t-SNE estimation with anchor threshold set to 0.5. Dotted lines shows the performance of the unmodified anchor method at the closest topic count value with anchor threshold set to 0.5. Data sets are colored as: **NIPS**, **NYT**, **Twitter**, and **NG20**.

Table 4.2: Average topic count produced by the t-SNE anchor word recovery method. t-SNE embedding dimension was set to 2 for all data sets.

| Data set | Anchor Threshold | Average Topic Count |
|---|---|---|
| NIPS | 0.75 | 21.3 |
| | 0.50 | 26.1 |
| | 0.25 | 26.5 |
| NYT | 0.75 | 17.3 |
| | 0.50 | 21.7 |
| | 0.25 | 20.7 |
| Twitter | 0.75 | 19.1 |
| | 0.50 | 20.2 |
| | 0.25 | 23.2 |
| NG20 | 0.75 | 16.4 |
| | 0.50 | 21.2 |
| | 0.25 | 26.2 |

Results indicated that Beta regularization, when applied in combination with t-SNE anchors, did not improve the coherence of topics. Beta regularization did however increase topic uniqueness significantly, and increased topic specificity. The t-SNE anchors, without regularization applied, sometimes resulted in topics that were more unique when compared to the unmodified anchor method baseline for the same topic count. The number of anchor words in the convex hull of the t-SNE embedding changed slightly with anchor threshold.

## 4.4 Automatic Anchor Merging

The topic sequences, described in the previous chapter, were tested with anchor threshold set to 0.5 and distortion rate set to 0.8. A selection of results, graphed as the progression of model quality as more topics are merged, are shown in Figure 4.7. The first three results, for data sets NIPS, NYT, and Twitter, show positive results for the `Unique` strategy. The final result, for the data set NG20, shows an example where model quality drops significantly in terms of coherence.

Results showed that the different strategies have very different characteristics. The `Many` strategy often resulted in sequences where topic uniqueness decreased significantly. A trade-off between uniqueness and coherence frequently occurred, resulting in the `Many` strategy often producing more coherent topics, but with significantly less uniqueness. This can be

observed clearly in Figure 4.7, where a decrease in uniqueness is often mirrored by an increase in coherence.

In general, the `Unique` strategy produced more predictable results, often keeping uniqueness stable while sacrificing coherence. When the drop in coherence was small, the strategy resulted in models that improved upon the anchor method baseline. E.g. the sequence for the Twitter data set, produced a model at topic count 20 with a uniqueness score of approximately 0.9, as compared to the anchor method baseline which scored less than 0.5.

The poor performance of the `Many` strategy may be attributed to the size of tandem anchors produced. The mean and max tandem anchor size for the same chains presented in Figure 4.7, is shown in Figure 4.8. These results show the tendency of the `Many` strategy to produce one or two large topics in the first merging phase. In the worst case, shown for the NYT data set, the initial merging phase goes from 160 topics to 105 by merging 55 topics into one. This means that one topic is anchored by the harmonic mean of 55 words. The initial merging phase resulted in a single massive topic in approximately one out of three sequences. In almost three out of four sequences the strategy produced a topic, during the initial merging phase, at least 80% as large as the difference between the starting and target topic count.

The difference in tandem anchor size between the strategies is presented as a function of merge step in Figure 4.9. To compare across topic sequences, the mean and maximum are normalized by the topic count of the initial model. Both the mean and maximum tandem anchor size of the `Many` strategy was consistently higher than the `Unique` strategy. However, the tandem anchors did not grow as significantly for the `Many` strategy after the initial merge. In fact, the mean tandem anchor size generally decreased after the initial merge of the `Many` strategy.

## 4.5 Overall Model Estimation Comparison

To illustrate the intrinsic properties of the different models, an example of the resulting topic descriptors, and quality metrics are presented for each model. This section is mainly included for discussion purposes and to provide actual examples of the results which the models produce. The data set used is NYT, and the topic count is selected to match the number of t-SNE anchors, in this case 22. Regularization coefficient is set to $10^{-3}$, and cosine threshold is set to 0.6 and 0.7 for the CluWord baseline and the CluWord anchor method respectively.

The quality metrics, as well as estimation times relative to LDA, are presented in Table 4.3. Note that estimation times are not linearly related across number of topics, corpus size, or vocabulary size, the times are presented to give the reader an idea of the differences in estimation time. The estimation times were measured on a machine with 4 CPU cores which benefits the parallelized methods, which includes at least the LDA model and anchor method (the NMF solver may also be partially parallelized). Times are presented as the time it takes for the initial estimation (I) and the time for a re-estimation (R), this is to illustrate the speed of the anchor method post co-occurrence calculation.

Topic descriptors for the same topic produced by each model is presented in Table 4.4. The topic was matched across models using Jaccard similarity of the top 10 words and seems to reflect a topic about politics. The topic descriptor presented is the standard probability based version and only the top 5 words are presented. Despite the CluWord enhanced methods producing descriptors with high coherence, they do not reflect the underlying topic well. Instead, they seem to mostly contain words synonymous with the words “asked” or “suggested”. All other models seem to produce descriptors which can easily be interpreted as political. Note that only the anchor method which utilizes merging managed to produce an anchor somewhat related to politics, “pardon”. This visualizes the unintuitive nature of automatically found anchor words.

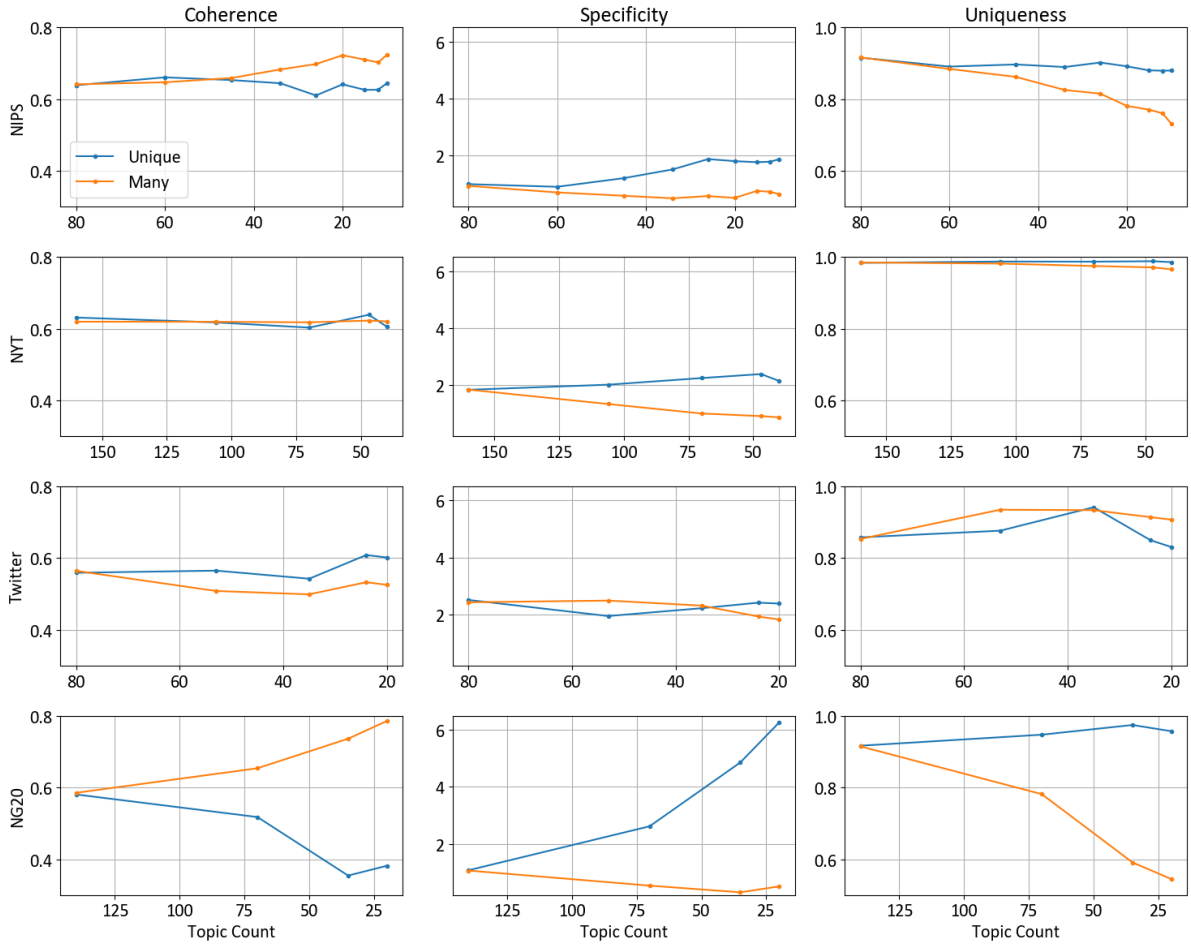



Figure 4.7: Model quality as a function of topic count for a select number of topic sequences. The first three rows show positive results while the last row shows an example of model quality regression during final the merges. Note that the x-axis is reversed since topic count is iteratively reduced through merging. The sequences measured indicated visible in the plot titles.

Table 4.3: Example of quality metrics and estimation time relative to LDA for the NYT data set for each model. Topic count was set 22 to match the convex hull of t-SNE embedding. Anchor threshold was set 0.5, and distrotion rate to 0.7 for the anchor methods. Cosine threshold was set to 0.6 and 0.7 for the CluWord baseline and CluWord anchor method (CAM) respectively. The topic count sequence used by the merge strategy was $\{60, 50, 40, 30, 22\}$.

| Model | Coherence ($M = 10$) | Specificity | Uniqueness ($M = 10$) | Estimation Time | |
|---|---|---|---|---|---|
| | | | | I | R |
| LDA | 0.68 | 0.94 | 0.99 | 1.00 | 1.00 |
| NMF | 0.76 | 1.36 | 0.99 | 0.72 | 0.72 |
| UAM | 0.52 | 1.89 | 0.83 | 0.18 | 0.02 |
| CluWord | 0.84 | 2.39 | 1.00 | 2.86 | 2.86 |
| CAM | 0.58 | 1.78 | 0.94 | 0.69 | 0.02 |
| Regularized | 0.57 | 0.28 | 0.80 | 0.78 | 0.63 |
| Merge | 0.60 | 1.33 | 0.98 | 0.24 | 0.05 |

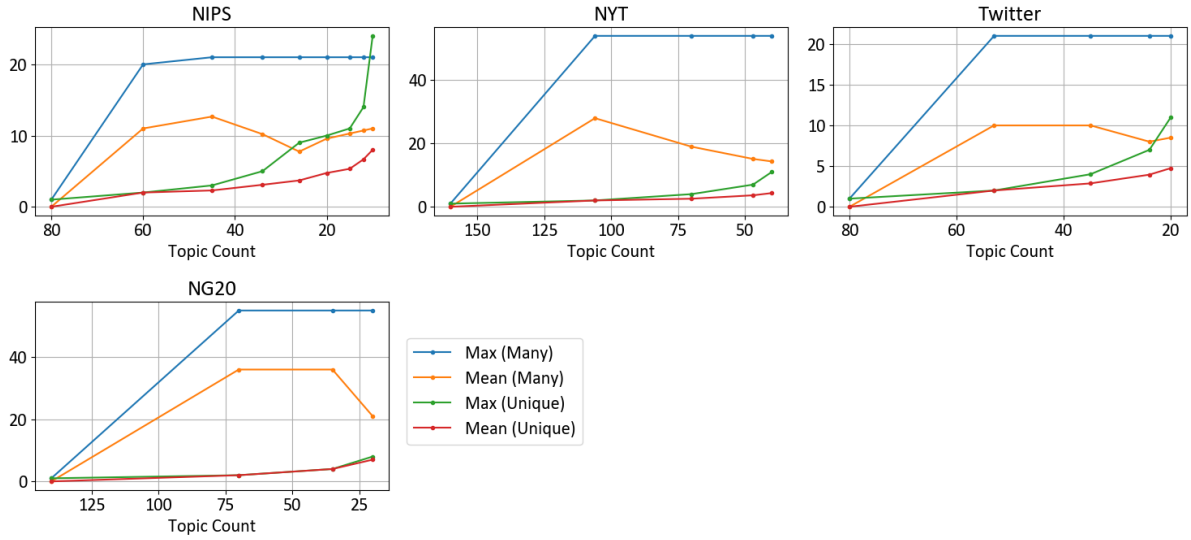



Figure 4.8: Mean and maximum tandem anchor size for the topic sequences presented in Figure 4.7.

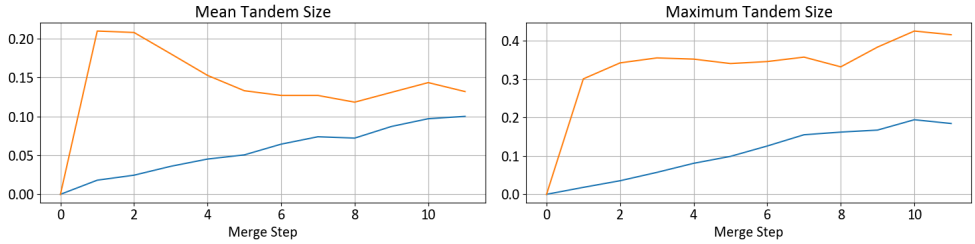



Figure 4.9: Mean and maximum tandem anchor size as a function of merge step normalized by initial topic count. The merge step is the position within the topic sequence. The strategies are colored as: **Unique**, and **Many**.

Table 4.4: Example of the top 5 words in a topic descriptor for each of the models. The topics were matched using Jaccard similarity of the top 10 words. The first row shows the anchor word(s) selected by the appropriate models. The final row shows the $C_{NPMI}$ coherence score of the top 5 words.

| LDA | NMF | UAM | CluWord | CAM | Regularized | Merge |
|---|---|---|---|---|---|---|
| - | - | mandatory | - | absentee | absentee | pardon file tasted |
| campaign | campaign | official | suggested | asked | ballot | campaign |
| tax | election | million | believed | urged | official | president |
| president | president | president | asked | million | percent | political |
| money | vote | right | told | advised | president | money |
| republican | political | government | decided | requested | election | case |
| 0.64 | 0.92 | 0.52 | 0.92 | 0.77 | 0.69 | 0.63 |

# 5 Discussion

This chapter aims to discuss the results and their implications, as well as to critically evaluate the method used to obtain the results. The chapter concludes with the possible impact of the work in a wider context.

## 5.1 Results

Chapter 4 presented the results of the model evaluations as is. This section aims to expand on the implications of the results.

### Baseline Performance

The performance of the baselines were somewhat unexpected due to the NMF model outperforming LDA, and the anchor method performing (especially) poorly when topic count was low. Since NMF is a general matrix factorization algorithm it was unexpected that it outperformed the LDA model on metrics created to evaluate topic models, such as coherence and specificity. The poor performance of the anchor method, even when topic count was as high as 50, was also surprising. This result has been noted before and rectification of the co-occurrence matrix has been proposed as a solution to improving the stability of the method [64]. Co-occurrence rectification was not evaluated in this thesis due to it not being discovered during the pre-study phase. Most papers which have evaluated the anchor method have focused on metrics such as coherence and heldout-likelihood, two metrics which may be high despite the model producing mostly similar topics. However, if the topic model is going to be directly observed through the use of topic descriptors, then the unmodified anchor method does not produce sufficiently unique topics, even when topic count is moderately high. In a real word scenario, topic count would probably be selected low such that the results are easier to interpret, and in such a situation the unmodified anchor method fails. The estimation time of the baseline models for the NYT corpus, as presented in Table 4.3, show the big improvements of the anchor method over LDA and NMF when considering re-estimation time. This would make the unmodified model suitable for interactive topic modeling where the user can modify the anchor words directly, as has previously been investigated [9].

A surprising result was that the distortion rate of the random projection used in the `FastAnchorWords` algorithm did not affect model quality in the range tested. In the case of the NYT data set, a distortion rate of 0.9 resulted in a projection from 17,047 to 240. This reduces the time required to recover anchor words. However, the overall impact of this speed increase is diminished as long as the projected dimensionality is not very large, which may explain why earlier work did not project to lower dimensions.

Correlation metrics aimed to guide the selection of the topic count parameter, presented in Figure 4.2, did not prove unambiguous for the NMF and anchor method baseline. The correlations metrics, and Arun score, did however behave closer to expectations for the LDA model. For the correlation metrics this was expected since they are motivated by properties of an uncorrelated topic model [49]. That is, they reflect a ”bad“ topic split (forced by the correlation assumption) as a sudden increase correlation. The Arun score did not reach a clear minimum for any data set for the NMF or anchor method baselines, indicating that the method may not be appropriate for correlated topic models.

### Word Embeddings

Using word embeddings to increase the density of the design matrix was expected to improve results, especially for the Twitter data set. The results for the CluWord baseline showed significant improvements across all metrics for all data sets over the NMF baseline (which already had relatively good results). Even for data sets which were not short text, such as NIPS and NYT, the CluWord baseline showed big improvements. An improvement in coherence is not entirely unexpected since the word embeddings and the evaluation metric are both based on word co-occurrence in roughly the same corpora. The improvement in specificity was expected since the underlying word distribution of the enhanced design matrix is different from the original word distribution, which the metric compared against.

The time to estimate the NMF of a matrix was found to highly depend on the density of the matrix. For short-text corpora the density of the design matrix is low and the increase in density as a result of the CluWord vocabulary is less likely cause problems. E.g. the density of the design matrix of the Twitter data set increased from less than 1% to 22%, which at 20 topics resulted in twice as long estimation times. For corpora with longer texts the density is already much higher and the CluWord vocabulary increases it even further, which may pose issues. E.g. the density of the NIPS corpora increased from 4% to 57%, resulting in 10 times the estimation time. Therefore, the improvement in results need to weighed against the increase in estimation time. An increase in density also results in higher memory requirements, which may result in the corpora not being able to be kept in memory.

The same logic applies to the anchor method, but the increase in estimation time only applies to the initial model evaluation. As seen in Table 4.3, where initial estimation time is almost as slow as LDA, while re-estimation time is orders of magnitude faster. This is explained by the increased density of the design matrix resulting in much longer estimation times for the co-occurrence matrix, which does not affect the subsequent steps required for re-estimation. In other words, the estimation time of the anchor method enhanced with word embeddings is independent of the design matrix density once the co-occurrence statistic has been calculated. The cosine threshold parameter greatly affects the resulting density of the design matrix, and since the co-occurrence estimation scales quadratically with density it is important to not set this parameter too low.

Results of the anchor method with the enhanced design matrix as input showed massive improvements over the unmodified baseline. In particular, the models estimated produced topic descriptors with much higher uniqueness scores, comparable to the LDA baseline. This means that if multiple topic models are going to be fit to the same corpora, then the increase in initial co-occurrence estimation may be fine since subsequent models will be of much higher quality. Results also showed an increase in coherence and specificity, which was to be expected due to the same reasons mentioned for the CluWord baseline. These results were

obtained despite increasing the cosine threshold by 20% compared to the CluWord baseline, indicating that even a high cosine threshold may improve the model significantly without big sacrifices in performance.

The major problem of the anchor method is that the co-occurrence matrix is only a crude approximation of the true co-occurrence statistic, a property which is partially observed through the density of the matrix. As an example, if the co-occurrence matrix is only 50% dense then this implies that half of the words have a 0% probability to co-occur. Increasing the density of the design matrix also increases the density of the co-occurrence matrix, simulating a statistic with potentially less noise. This assumes that the distributional patterns in the reference corpus observed by the word embeddings also apply to the original corpus, which may not be the case. The drawback of this method is that commonly co-occurring words, which may not be indicative of the topic, are promoted in the topic descriptors generated. To alleviate this problem a different topic descriptor may be used which down weighs words uncommon in the original corpus but common in the reference corpus. Neither the relevance based descriptor or the TF-IDF descriptor solved this issue.

### t-SNE Anchors and Regularized Objective

The results of combining already investigated improvements of the anchor method were disappointing. Tthe t-SNE anchors improve uniqueness scores of the produced topic descriptors for some of the data sets. However, this improvement came at massive costs in computation time and the loss of the topic count hyperparameter. For corpora with many documents the estimation time of the t-SNE embedding is orders of magnitude slower than the `FastAnchorWords` method. This can be compared to the results of the CluWord enhanced anchor method, which performed even better without sacrificing performance or flexibility.

Results indicated that Beta regularization did not improve coherence, even in the case when the model was estimated on the same underlying corpus as in the original paper. This could be due to a multitude of reasons such as: different corpus pre-processing, the use of t-SNE anchors, regularizing a different objective function, or problems with the implementation used in this thesis. Using the Sequential Least Squares Programming (SLSQP) minimizer from scikit-learn also proved orders magnitude slower than the exponentiated gradient based recovery method. Unfortunately, no publicly available implementation of this method was available to our knowledge. It is possible that the regularization term overpowered the the L2 objective (indicated by the much lower value of regularization coefficient necessary in this thesis as compared to the original paper [11]) which in turn may have negatively impacted the minimizer. Selecting the $B$ matrix randomly from a Dirichlet distribution resulted in the minimizer not being able to minimize the objective, probably due numerical precision problems. To solve this, each row of the $B$ matrix was smoothed by a small constant ($10^{-3}$), which may also have affected the results.

### Automatic Topic Merging

Merging similar topics by combining their anchor words into tandem anchors showed promising results, especially with regards to increasing uniqueness scores for low topic counts. Unfortunately, the performance of neither merging strategy turned out to be reliable across all topic sequences.

The `Many` strategy turned out to be especially unstable, often resulting in very big tandem anchors. This could perhaps be due to the pair-wise correlations being calculated once before every strategy run, instead of re-calculating the pair-wise similarities between the remaining topics and the merged topics. A strategy which finds a way to re-estimate such a correlation may remedy the problem of big tandem anchors. Using a different correlation or similarity metric, or putting restrictions on how many topics can be merged into the same topic may improve the stability, but was not tested.

The `Unique` strategy was more stable and for many topic sequences managed to maintain or improve the uniqueness and specificity scores without sacrificing much in coherence. However, certain sequences (such as the final one showed in the results) exhibited massive drops in performance due to a single or very few merges. This could have perhaps been due to being forced to merge topics which were not similar enough, due to all remaining unmerged topics being sufficiently different. A remedy for such a situation would be to stop the sequence early once all topics have a sufficiently low pair-wise correlation. Another reason could be a merge breaking the anchor word assumption to a degree extreme enough to break the recovery process. Unlike the `Many` strategy, this strategy did not suffer from very big tandem anchors (by design). This restriction came at the cost of sometimes being forced to merge topics which were not very similar.

### Overall Model Comparison

Chapter 4 concluded by showing results of a model comparison at a set topic count value for the NYT data set. This was included as an example of what the models actually produce. The topic descriptors for the topic selected showed one of the drawbacks of the methods enhanced with word embeddings, the inclusion of multiple synonyms. This could be seen in other topics as well, e.g. one topic for the CluWord baseline contained words such as: American, European, British, Russian, German, Japanese, Chinese, Indian etc. The closest matching topic in the NMF model is about war and politics, and the words American and European also appear among the top words. However, the words relating to the other nationalities do not show up despite increasing the size of the topic descriptor considerably. In this case, including similar words to American indicates that the topic includes many other nationalities, when in fact it may not. The closeness of these words in the vector space of the word embedding is a success, but in the case of topic models it may obfuscate or even mislead the actual content of the topic.

Word embeddings are based on the distributional hypothesis of words: words which occur in similar contexts have similar meanings. This allows word embeddings to capture similarity between words which which are not strictly synonymous, such as morphological or syntactic relationships. Using word embeddings to synthetically increase the density of the design matrix, effectively adding pseudo occurrence counts for words which may not be synonymous with the original word, may be inappropriate in the context of topic models. However, words which occur in the same contexts do not necessarily co-occur with each other, which is what the coherence metrics measure. Despite this, the coherence measures do increase when utilizing word embeddings.

## 5.2 Method

This section takes a critical view of the method used to produce the results of the thesis, and discusses the reasoning for why a particular method was chosen. This includes the corpus selection and the following pre-processing, the implementations of the models, the choice of word embeddings, the choice of merging strategies, and finally the metrics used for evaluation.

### Corpus Selection and Pre-processing

The corpus selection was made to include data sets which are common within the literature and of varying types. The only uncommon data set used is the one collected specifically for this thesis via Twitter. Observations made during testing, and partially confirmed by the quality metrics, indicated that the topics recovered from the Twitter data set were of poor quality. An alternative would have been to use a public Twitter data set, previously used

within the literature, collected with topic modeling in mind. Examples of this would be "Health News in Twitter" [65] (available via [55]).

The Twitter data set was also problematic when combined with minimal pre-processing. Informal text published on social media often contain abbreviations, misspellings, noise from spam etc. Since no pre-processing was done to correct for these issues the Twitter data set had to use a lower document frequency cut-off during vocabulary creation. Publicly available tools created to pre-process social media texts exists [66] but were not discovered in time for the thesis experiments. It is therefore unclear if such pre-processing step would have improved the results. The data set, while problematic, was however a realistic corpus obtained by simply recording tweets. As indicated by the results of this data set, topic modeling may not be an effective method for corpora which lack sufficient pre-processing.

The pre-processing steps were selected to be as general as possible, which seemed to work well for three out of the four data sets. Restricting the size of the vocabulary was important, both to avoid noise and since the anchor method scales quadratically with vocabulary size. A document frequency cut-off seemed to work well for all data sets but Twitter. An alternative approach would have been to simply set a maximum vocabulary size across all data sets.

Pre-processing textual data is an important step in pretty much every task within natural language processing. The aim of this thesis was not to evaluate pre-processing, but for practical purposes the pre-processing steps should be carefully selected. These steps include filtering based on document frequency to limit vocabulary size, mitigating noise in the text corpus (such as misspellings), and removing words such as stop words.

### Model Implementations

The model implementations used in this thesis were a mix of publicly available ones and ones implemented for this particular thesis. Both the LDA and NMF models used implementations which were popular within the Python ecosystem and whose libraries has been featured as part of earlier scientific publications. The results presented in this thesis did not indicate any problems with these implementations either. The unmodified anchor method has no common, established implementation in Python. Fortunately, the anchor method has a very simple formulation and is not hard to implement. The implementation used in this thesis utilized the Numba [67] compiler to parallelize and speed up both the anchor recovery process and the exponentiated gradient algorithm. A non compiled Python implementation would be significantly slower. It is unclear whether or not the other baselines would benefit from a compilation step or if they already do.

The implementation of the regularized anchor method used the SLSQP minimizer available in the Scipy library. To make use of this implementation the constraints, bounds, and gradient were supplied. Unfortunately, this implementation did not match that of the one used in the original paper [11], which used a minimizer based on L-BFGS. It is unclear how the constraints were imposed using this method, since the minimizer is not a constrained solver, unlike SLSQP. However, it is not clear if the minimizer actually affected the results negatively, but the reliability of the results may be of issue. As described in the discussion of the results, this thesis uses a different anchor recovery method and an objective function based on L2 distance as opposed to KL divergence. While the paper introducing the Beta regularization has been cited many times, no follow-up paper which actually utilizes the method was found during the literary study, and no public implementation was found.

The models which utilizes word embeddings, by constructing a CluWord vocabulary, was implemented according to the original paper [14]. Results obtained in this thesis do not suggest any problems with the implementation. Unlike the original paper, the cosine threshold was found to be higher to select the appropriate proportion of similar words despite both using publicly available fastText vectors. It is unclear what this discrepancy is due to, perhaps the vectors have been updated since the original paper. This difference did not seem to affect

results of the baseline, and a yet higher threshold had to be selected for the anchor method anyway due to memory and time limitations.

### Choice of Word Embeddings

For this thesis the publicly available fastText vectors were used to enhance the NMF and anchor method baselines. These vectors were trained on the 2017 Wikipedia corpus which should be fairly similar to the 2020 Wikipedia corpus used for evaluation. Using the same underlying corpus may have artificially improved the coherence scores of the enhanced methods. However, since the Wikipedia corpus have been shown to correlate best with human interpretation [16] it was also selected for this thesis. As described in the discussion of the model comparison results, it is not obvious that words which are close in the vector space of the word embedding, necessarily co-occur within a sliding window in the reference corpus.

Another problem with using word embeddings is if the co-occurrence patterns observed in the reference corpus are very different from the ones in the original corpus. This may results in misleading topics for the estimated topic model. Words which are often found in the same context in the original corpus, may not be found as often in the reference corpus, and therefore will not be deemed as similar. Therefore, it may be more valuable to use a word embedding trained on a corpus similar to the original corpus. This should reflect the original corpus more truthfully in the final topic model.

### Merging Strategies

Only two relatively simple merging strategies were tested for this thesis, `Unique` and `Many`, with the former strategy producing promising results. Topic modeling can also be seen as a form of soft clustering, where documents are clustered around topics. Therefore, the merging strategies can be seen as a form of hierarchical clustering, where topic hierarchies are built using subtopics. No knowledge from the field of hierarchical clustering was used when designing the merging strategies, an obvious drawback which could have lead to better results. The main point of research question 2 was to investigate if merging tandem anchors of similar topics could lead to better results, which the results indicate despite the aforementioned lack of knowledge on hierarchical clustering. Using tandem anchors as a vehicle for evaluating bottom-up merging strategies within topic models could be of interest in future work.

Merging anchors into tandem anchors, especially when the merged anchors grow large, may worsen the theoretical guarantees of the anchor method. It is unclear what the consequences of this would be, and if they are worse than the ones associated with noisy anchors (all anchors recovered using the noisy co-occurrence statistic are noisy and referred to as almost-anchor words in the original article [7]). No proofs of theoretical guarantees were attempted for this thesis, it was assumed that the same guarantees hold for automatically merged tandem anchors as in the original paper [9]. However, the results for the `Many` strategy suggest that limiting the size of tandem anchors is likely necessary.

### Evaluation

Evaluating model quality of unsupervised learning models is a difficult task, even more so when the model quality have to correlate with human judgement. Only the coherence metrics used in this paper had been shown to correlate with perceived quality by humans, all of which were based on word co-occurrence, and most of which use a reference corpus. Evaluating multiple coherence measures on multiple topic descriptors did not produce interesting results. During initial testing, the coherence metrics based on co-occurrence of a reference corpus mostly followed each other. Therefore, only the $C_{NPMI}$ coherence metric was reported, since it had been shown to correlate best with human judgement [16].

Using specificity and uniqueness as additional model quality metrics revealed some of the big issues with the unmodified anchor method not apparent in a lot of the literature. The results of the unmodified anchor method were presented in Figure 4.1. At even a moderate number of topics the unmodified anchor method produced topics which, despite being coherent, were mostly the same and similar to the underlying word distribution. Coherence did not reflect this issue since it did not penalize repeating the same words across many topics.

Another issue of coherence metrics could be seen when qualitatively judging the topic descriptors of the models enhanced with word embeddings. These models achieved great average coherence scores, but the topic descriptors produced had a tendency of containing many synonymous words. This may in some cases reinforce the subject of the topic, but as shown in Table 4.4 may also obfuscate the topic by repeating a non-informative word and its synonyms. As discussed earlier, it is not obvious that synonymous words (or words with other syntactic relationships captured by the word embedding) co-occur with each other, but the results do seem to indicate that they are at least more likely to do so than not.

Using correlation metrics or Arun score to select topic count a value would have allowed for comparing the quality of each estimation method's optimal model. Unfortunately, these methods did not produce unambiguous answers for the correlated topic models. Instead, model quality had to be measured across a range of values for topic count. No other model selection method, based on the intrinsic properties of the topic model, was studied. Selecting an optimal topic model for human consumption is most likely a flawed concept since humans perceive interactively changing the model as an improvement [68].

### Source Criticism

The vast majority of sources used for this thesis were either published in scientific journals or as conference proceedings. Two authors, Mimno and Boyd-Graber, show up as co-authors of more than 10% of the papers cited due to their long standing involvement in topic modeling and involvement in the anchor method or its extensions. Three sources were pre-prints [28, 31, 61], but only one of these was somewhat important to the theoretical foundation of the thesis (the word2vec paper) and it had over 15,000 citations according to Google Scholar and was a seminal work in the field of word embeddings.

## 5.3 The work in a wider context

Using topic models to explore a corpus may be detrimental if the model is unable to discover important topics or produces a topic representation which obfuscates the actual topic. If topic models are to be used directly through observation of topic descriptors and topic assignment of documents, it is important to allow the user to explore the underlying text documents. This allows the user to verify if their interpretation of a topic actually matches that of the document to which it is assigned. It would also be important to inform the user that topic modeling is not perfect and may not even produce coherent results for certain corpora. Topic models are also very sensitive to the selected hyperparameters, especially topic count, so it would be important to allow the user to tweak this value if the topic model is deemed low quality. The anchor method, and the extensions studied in this thesis, are interpretable and fast to re-estimate, making them especially well suited for this task.

The consequences of presenting topic models as a perfect solution for summarising or gaining an overview of a corpus without the ability to prod at the underlying model may have dire consequences. This would include giving the observer a false sense security about what the corpus contains and what its most important themes are. Making decisions based on such biased or incomplete information could be problematic.

# 6 Conclusion

This thesis has investigated topics models with a focus on the anchor method combined with certain enhancements aimed at increasing the interpretability of the produced model. Evaluation was done across a range of hyperparameters for four data sets of different characteristics. The focus of the thesis was on three enhancements: anchor words recovered from a t-SNE embedding in combination with a Beta regularized objective, automatically merging similar topics by combining their anchor words, and enhancing the design matrix using word embeddings. The first enhancement did not produce promising results, which may have been due to problems related to its implementation in this thesis. However, the use of t-SNE in the enhancement significantly impacts the efficiency of the estimation and re-estimation of the anchor method, which is one of its main properties. Automatically merging similar topics through the use of tandem anchors showed promising but somewhat unpredictable results, especially when tandem anchors grew very large. Using word embeddings to enhance the design matrix improved model quality significantly, with the main drawback being a higher initial estimation time. This drawback could be mitigated by tweaking the cosine threshold used to construct the CluWord vocabulary used to enhance the design matrix.

## 6.1 An Efficient Model with Human Interpretable Results

The purpose of this thesis was to investigate efficient topic models with human interpretable results, which could be used in a data visualization application to improve its ability to visualize and gain insights from textual data. The results of this thesis indicate that for English text a possible solution would be to calculate the co-occurrence statistic automatically, preferably enhanced with word embeddings, and let the user tweak the model through editing topic count and the anchor words. If no word embeddings can be used then an alternative initial model could be estimated through bottom-up merging of similar topics. Both solutions mitigate the problem of the anchor method producing low quality topics at a limited topic count, while at the same time keeping the efficiency of the anchor method and allowing the user to easily modify the model. For use in the application, it would be very important to consider the size of the generated vocabulary, as well as the selected cosine threshold used during the CluWord vocabulary creation. Both of these parameters greatly affect the memory consumption and speed of the model estimation. Combined with visualization techniques designed for topic models, such a feature could be useful in the task of corpus exploration.

## 6.2 Research Questions

In this section the research questions presented in Chapter 1 will be answered as clearly as possible.

### How does the combination of existing extensions to the anchor method affect model quality?

Our results indicate that combining t-SNE anchors with a beta regularized objective does not affect model quality in an unambiguously positive manner. The t-SNE anchors sometimes improve model quality but come at a big cost in model estimation and re-estimation time. The beta regularization, combined with the L2 objective and minimized through `SLSQP`, improved specificity and uniqueness metrics, but came at the cost of topic descriptors with lower coherence. This is unlike previous results, in which Beta regularization improved coherence scores.

### How does combining anchor words, whose resulting topic distributions are alike, affect model quality?

Automatically combining anchor words into tandem anchors can have a big positive impact on model quality. The merging strategies explored in this thesis could be unpredictable but produced promising results when the size of the combined anchor was restricted to not grow large. More sophisticated merging strategies using tandem anchors as a vehicle for merging topics should be able to produce human interpretable models even at low topic counts, unlike the unmodified anchor method. The merging strategies had no significant runtime impact, and since the re-estimation time of the anchor method is fast, so was the merging based method.

### How is model quality affected when incorporating word embeddings into the design matrix of the anchor method?

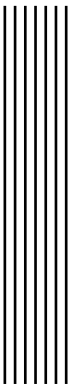
Incorporating word embeddings into the design matrix through the use of a CluWord vocabulary improved model quality across all metrics significantly. The standard anchor method was not able to produce topics of sufficient quality even for a moderately high topic count, while the anchor method enhanced with word embeddings produced topics of high quality across all values of topic count tested. This improvement is a result of the increased density of the design matrix, which impacts the initial estimation time of the anchor method. The increased density did not affect the re-estimation time after the co-occurrence statistic had been calculated. Using word embeddings may obfuscate the true subject of a topic promoting many similar words to the topic descriptor, indicating that perhaps a different descriptor may be of interest.

## 6.3 Future Work

This thesis has revealed two interesting avenues for future work: sophisticated automatic merging strategies based on tandem anchors, and topic descriptors designed for topic models which utilize word embeddings.

Since topic modeling can be seen as a clustering technique, performing automatic merging of topics into fewer topics can be seen as a form of hierarchical clustering. Investigating strategies used within this field, or developing strategies more sophisticated than the ones used in this thesis, may be of interest. The results found in this thesis indicate that such strategies should limit the size of combined anchors.

Qualitative observations made during the experiments of this thesis indicated that topic models enhanced with word embeddings had a tendency of promoting similar words in topic descriptors. This improved the coherence of said topics but seemed to obfuscate the true subject of the underlying topic. Topic descriptors which attempt to mitigate this issue may be of interest since the results of this thesis indicate that these enhanced topic models produce results of much higher quality for the anchor method.



# Bibliography


[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *Journal of machine Learning research* 3 (2003).

[2] Thomas Hofmann. "Probabilistic latent semantic indexing". In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. Association for Computing Machinery, 1999, pp. 50–57.

[3] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.

[4] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. "Reading Tea Leaves: How Humans Interpret Topic Models". In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Curran Associates, Inc., 2009, pp. 288–296.

[5] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. "Optimizing semantic coherence in topic models". In: *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 262–272.

[6] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. "Interactive topic modeling". In: *Machine learning* 95.3 (2014), pp. 423–469.

[7] Sanjeev Arora, Rong Ge, and Ankur Moitra. "Learning topic models–going beyond SVD". In: *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 2012, pp. 1–10.

[8] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. "A Practical Algorithm for Topic Modeling with Provable Guarantees". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. JMLR.org, 2013, II–280–II–288.

[9] Jeffrey Lund, Connor Cook, Kevin Seppi, and Jordan Boyd-Graber. "Tandem anchoring: A multiword anchor approach for interactive topic modeling". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2017, pp. 896–905.

[10] David Mimno and Moontae Lee. "Low-dimensional embeddings for interpretable anchor-based topic inference". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1319–1328.

[11] Thang Nguyen, Yuening Hu, and Jordan Boyd-Graber. "Anchors regularized: Adding robustness and extensibility to scalable topic-modeling algorithms". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 2014, pp. 359–369.

[12] Thang Nguyen, Jordan Boyd-Graber, Jeffrey Lund, Kevin Seppi, and Eric Ringger. "Is your anchor going up or down? Fast and accurate supervised topic models". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015, pp. 746–755.

[13] Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K Reddy. "Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations". In: *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 1105–1114.

[14] Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. "CluWords: exploiting semantic word clustering representation for enhanced topic modeling". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2019, pp. 753–761.

[15] Loulwah AlSumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. "Topic significance ranking of LDA generative models". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 67–82.

[16] Michael Röder, Andreas Both, and Alexander Hinneburg. "Exploring the space of topic coherence measures". In: *Proceedings of the eighth ACM international conference on Web search and data mining*. Association for Computing Machinery, 2015, pp. 399–408.

[17] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. "Indexing by latent semantic analysis". In: *Journal of the American Society for Information Science* 41.6 (1990), pp. 391–407.

[18] David M. Blei and John D. Lafferty. "Correlated Topic Models". In: *Proceedings of the 18th International Conference on Neural Information Processing Systems*. NIPS'05. Vancouver, British Columbia, Canada: MIT Press, 2005, pp. 147–154.

[19] David Hall, Daniel Jurafsky, and Christopher D. Manning. "Studying the History of Ideas Using Topic Models". In: *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, 2008, pp. 363–371.

[20] Edmund M Talley, David Newman, David Mimno, Bruce W Herr II, Hanna M Wallach, Gully APC Burns, AG Miriam Leenders, and Andrew McCallum. "Database of NIH grants using machine-learned categories and graphical clustering". In: *Nature Methods* 8.6 (2011), p. 443.

[21] Georgios Balikas, Hesam Amoualian, Marianne Clausel, Eric Gaussier, and Massih R. Amini. "Modeling topic dependencies in semantically coherent text spans with copulas". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 1767–1776.

[22] Eric Gaussier and Cyril Goutte. "Relation between PLSA and NMF and Implications". In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '05. Salvador, Brazil: Association for Computing Machinery, 2005, pp. 601–602. ISBN: 1595930345.

[23] Stephen A. Vavasis. "On the Complexity of Nonnegative Matrix Factorization". In: *SIAM J. on Optimization* 20.3 (2009), pp. 1364–1377. ISSN: 1052-6234.

[24] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. "Supplemental Material for "A Practical Algorithm for Topic Modeling with Provable Guarantees". In: ().

[25] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. "Termite: Visualization Techniques for Assessing Textual Topic Models". In: *Proceedings of the International Working Conference on Advanced Visual Interfaces*. AVI '12. Capri Island, Italy: Association for Computing Machinery, 2012, pp. 74–77. ISBN: 9781450312875.

[26] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[27] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. "The Quickhull Algorithm for Convex Hulls". In: *ACM Trans. Math. Softw.* 22.4 (1996), pp. 469–483. ISSN: 0098-3500.

[28] George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. "Efficient algorithms for t-distributed stochastic neighborhood embedding". In: *arXiv preprint arXiv:1712.09005* (2017).

[29] Thang Dai Nguyen. "Rich and Scalable Models for Text". PhD thesis. 2019.

[30] Gerard Salton. "Some Experiments in the Generation of Word and Document Associations". In: *Proceedings of the December 4-6, 1962, Fall Joint Computer Conference*. AFIPS '62 (Fall). Philadelphia, Pennsylvania: Association for Computing Machinery, 1962, pp. 234–250. ISBN: 9781450378796.

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781* (2013).

[32] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pp. 3111–3119.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543.

[34] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018.

[35] Omer Levy and Yoav Goldberg. "Neural Word Embedding as Implicit Matrix Factorization". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2177–2185.

[36] Carson Sievert and Kenneth Shirley. "LDAvis: A method for visualizing and interpreting topics". In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014, pp. 63–70.

[37] Matt Taddy. "On estimation and selection for topic models". In: *Artificial Intelligence and Statistics*. JMLR.org, 2012, pp. 1184–1193.

[38] David M Blei and John D Lafferty. "Topic models". In: *Text mining*. Chapman and Hall/CRC, 2009, pp. 101–124.

[39] Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. "An analysis of the coherence of descriptors in topic modeling". In: *Expert Systems with Applications* 42.13 (2015), pp. 5645–5657. ISSN: 0957-4174.

[40] Xing Wei and W. Bruce Croft. "LDA-Based Document Models for Ad-Hoc Retrieval". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '06. Seattle, Washington, USA: Association for Computing Machinery, 2006, pp. 178–185. ISBN: 1595933697.

[41] Kalyanasundaram Somasundaram and Gail C. Murphy. "Automatic Categorization of Bug Reports Using Latent Dirichlet Allocation". In: *Proceedings of the 5th India Software Engineering Conference*. ISEC '12. Kanpur, India: Association for Computing Machinery, 2012, pp. 125–130. ISBN: 9781450311427.

[42] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. "Evaluation Methods for Topic Models". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 1105–1112. ISBN: 9781605585161.

[43] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. "Automatic evaluation of topic coherence". In: *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 100–108.

[44] Gerlof Bouma. "Normalized (pointwise) mutual information in collocation extraction". In: *Proceedings of GSCL* (2009), pp. 31–40.

[45] Nikolaos Aletras and Mark Stevenson. "Evaluating Topic Coherence Using Distributional Semantics". In: *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*. Potsdam, Germany: Association for Computational Linguistics, 2013, pp. 13–22.

[46] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. "Exploring Topic Coherence over Many Models and Many Topics". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 952–961.

[47] Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. "Using Word Embedding to Evaluate the Coherence of Topics from Twitter Data". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '16. Pisa, Italy: Association for Computing Machinery, 2016, pp. 1057–1060. ISBN: 9781450340694.

[48] Jey Han Lau and Timothy Baldwin. "The Sensitivity of Topic Coherence Evaluation to Topic Cardinality". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016, pp. 483–487.

[49] Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. "A density-based method for adaptive LDA model selection". In: *Neurocomputing* 72.7 (2009). Advances in Machine Learning and Computational Intelligence, pp. 1775–1781. ISSN: 0925-2312.

[50] Romain Deveaud, Eric SanJuan, and Patrice Bellot. "Accurate and effective latent concept modeling for ad hoc information retrieval". In: *Document numérique* 17.1 (2014), pp. 61–84.

[51] Rajkumar Arun, Venkatasubramaniyan Suresh, CE Veni Madhavan, and MN Narasimha Murthy. "On finding the natural number of topics with latent dirichlet allocation: Some observations". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2010, pp. 391–402.

[52] Elias Zavitsanos, Sergios Petridis, Georgios Paliouras, and George A. Vouros. "Determining Automatically the Size of Learned Ontologies". In: *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*. NLD: IOS Press, 2008, pp. 775–776. ISBN: 9781586038915.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[54] Joel Nothman, Hanmin Qin, and Roman Yurchak. "Stop Word Lists in Free Open-source Software Packages". In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 7–12.

[55] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[56] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010, pp. 45–50.

[57] Matthew Hoffman, Francis R. Bach, and David M. Blei. "Online Learning for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta. Curran Associates, Inc., 2010, pp. 856–864.

[58] Andrzej Cichocki and Anh-Huy Phan. "Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations". In: *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* 92.3 (2009), pp. 708–721.

[59] Cédric Févotte and Jérôme Idier. "Algorithms for Nonnegative Matrix Factorization with the $\beta$-Divergence". In: *Neural Comput.* 23.9 (2011), pp. 2421–2456. ISSN: 0899-7667.

[60] Jeffrey Lund, Piper Armstrong, Wilson Fearn, Stephen Cowley, Courtni Byun, Jordan Boyd-Graber, and Kevin Seppi. "Automatic Evaluation of Local Topic Quality". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 788–796.

[61] Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. "openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding". In: *bioRxiv* (2019).

[62] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272.

[63] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. "Learning Topic Models – Provably and Efficiently". In: *Commun. ACM* 61.4 (2018), pp. 85–93. ISSN: 0001-0782.

[64] Moontae Lee, David Bindel, and David Mimno. “Robust Spectral Inference for Joint Stochastic Matrix Factorization”. In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 2710–2718.

[65] Amir Karami, Aryya Gangopadhyay, Bin Zhou, and Hadi H K Kharrazi. “Fuzzy Approach Topic Discovery in Health and Medical Corpora”. In: *International Journal of Fuzzy Systems* 20.4 (2018), pp. 1334–1345. ISSN: 1562-2479.

[66] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. “DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 747–754.

[67] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. “Numba: A LLVM-Based Python JIT Compiler”. In: *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. LLVM ’15. Austin, Texas: Association for Computing Machinery, 2015. ISBN: 9781450340052.

[68] Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. “The human touch: How non-expert users perceive, interpret, and fix topic models”. In: *International Journal of Human-Computer Studies* 105 (2017), pp. 28–42. ISSN: 1071-5819.