PERSONALIZED TOPIC MODELLING OF DOMAIN-SPECIFIC DOCUMENT COLLECTIONS

by

Bhuvaneshwari Basquarane

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
April 2023

Dedicated to parents, friends, roommates and relatives.

Table of Contents

List of Tables

List of Figures

Abstract

Topic modelling refers to the discovery of abstract topics in a document collection. The abstract topics are often described by a statistical model that models the probabilistic relationship between topics, documents and words, typically through identifying the distribution of words within the topic and the distribution of topics in a document. One criticism is that we recognize that there can be several possible sets of topics, so in this study, we propose a personalizable topic modelling algorithm wherein a user guides the method by suggesting edits to the statistical models. In order to do this, we build upon Top2Vec, a recent topic-modelling algorithm that represents documents by their embeddings and then defines topics as soft clusters of documents. In our approach, the users are allowed to provide feedback about the documents, which is then used to define a contrastive loss function for fine-tuning the pre-trained BERT model used to derive embeddings of documents. In this work, we made the following contributions. First, we encapsulate the Top2Vec algorithm within a probabilistic framework—which we call Probabilistic Top2Vec— to represent the topics in terms of the joint probabilities of words, documents, and topics. Finally, we introduce two personalization techniques that allow the user to provide weaker word-level supervision—describing each topic with a few central words—and stronger document-level supervision—wherein the user explicitly places the document in the desired topic cluster—in guiding the topic discovery. We evaluate this model quantitatively with the help of an oracle on labelled datasets: the quantitative evaluations measure how well the model can adapt to user feedback with the help of an oracle simulating the user and help determine the appropriate hyperparameters of the algorithm. Based on our quantitative evaluations, providing even weak feedback to the model can result in topic modelling that better aligns with the user's preferences. These results can be further improved with document-level feedback. More specifically, the results of Top2Vec visualized as probabilities should enable the user to clearly understand the discovered topics and then provide the appropriate feedback to personalize the topic modelling result.

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to the esteemed professors, Dr. Evangelos Milios, Dr. Vlado Kessel, Dr. Sageev Oore, Dr. Fernando Paulovich and Dr. Yannick Marchand, who has tirelessly taught and guided me throughout my master's degree program. I am grateful to all my course instructors, who have been instrumental in equipping me with the knowledge, skills, and confidence necessary to conduct my research and complete my thesis successfully.

In particular, I extend my heartfelt appreciation to my thesis defense committee, Professors Dr. Evangelos Milios, Dr. Axel Soto, Dr. Vlado Kessel, and Dr. Ana Maguitman, for their insightful feedback, constructive criticism, and guidance, which have been invaluable in shaping my research work and refining my ideas. A special thanks goes to Chandramouli Shama Sastry and Mariano Maisonnave for their dedication and mentorship throughout the course of my project. Your expertise, insight, and encouragement have been instrumental in the success of this project, and I am deeply grateful for your support. Your commitment to mentoring has been inspiring, and I have learned so much from you. Thank you for sharing your knowledge and being valuable mentors to me.

Furthermore, I would like to acknowledge and thank my Grandmom Sarasu and beloved parents Basquarane and Ameena for their unconditional love, encouragement, and unwavering support. I am also grateful to my brother Balaji Basquarane, who has always been there for me. I would like to express my gratitude to my dear friends in Canada – Chandramouli, Harpreet, Rajveen, Muthu, Rashmi, Theresa, Harsha, and Aman, – for their understanding, support, and patience during the research process. I am thankful to my friends – Jeniffer, Aishwarya, Juliana, Krithika, Arthi, Padmashree, Sankar Narayana, Ezhilane, Sivakumar, Naveen and Micimeena – for keeping in touch and keeping me informed of the happenings back home, and for never allowing me to feel left out. I am also deeply grateful to my school teacher, Ms. Naina, and Velmurugan, who have been my constant source of support and encouragement throughout this journey. Your unwavering belief in me and your

endless love and support have sustained me through the ups and downs of the research process. To everyone who has played a role, no matter how small, in helping me achieve this milestone in my academic journey, thank you from the bottom of my heart. Your support, encouragement, and belief in me have been invaluable, and I am deeply grateful.

Chapter 1

Introduction

Topic modelling identifies abstract topics in a collection of documents and it is useful for understanding a document collection through distributions such as: a) topics in a document collection and in a given document; b) words in a topic; c) documents in a topic and so on. In the past, the most widely used probabilistic topic modelling algorithm is Latent Dirichlet Allocation (LDA) Blei et al. (2003) where each document consists of different topics and each topic is a distribution of words. LDA uses the Dirichlet prior distribution over document-topic and topic-word distributions.

In recent times, we encountered a new topic modelling algorithm which is Top2Vec (Angelov, 2020). This is a framework for topic-modelling that uses sophisticated neural text embedding models —such as BERT (Devlin et al., 2019) for the discovery and assignment of topics. In other words, Top2Vec generates jointly embedded topics, documents, and word vectors. Broadly, Top2Vec involves the following steps:

1. Embed: Embed the documents into vectors—for example, using BERT or Sentence-BERT (Reimers and Gurevych, 2019). The intuition is that semantically similar documents are embedded closer together while dissimilar documents are embedded away from each other.

2. Project: Project the vectors into a lower dimensional manifold using an algorithm such as UMAP (McInnes et al., 2018) or t-SNE (Rauber et al., 2016). The application of the dimensionality reduction approach is prescribed in order to enable better cluster quality when using HDBSCAN (Campello et al., 2013; McInnes et al., 2017) while still retaining local and global structure of the high-dimensional data.

3. Cluster: Apply a clustering algorithm such as HDBSCAN (Campello et al., 2013; McInnes et al., 2017) to form document clusters.

Finally, the algorithm proposes to derive topic vectors as the mean of the documents constituting each cluster and chooses words embedded closer to the topic vector as the topic words. The centroids are computed by considering the high dimensional space, instead of the reduced embedding space.

Similar to Top2Vec, BERTopic (Grootendorst, 2022) is another topic modelling framework that uses large pretrained language models for guiding the topic discovery process. Similar to Top2Vec, BERTopic also applies the Embed-Project-Cluster steps. Finally, the words characterizing each topic is identified through an adaptation of TF-IDF to work on a cluster or topic level,i.e.,c-TF-IDF, that is applied separately to each cluster.

Analysis of Drawbacks. While these models enable the discovery of semantically-coherent topics, neither of them allow a probabilistic interpretation of the relationship among topics, documents and words. Encoding these relationships through probabilities help answer the following questions which can be crucial in understanding a document collection. In the following, t_i refers to the i-th topic, d refers to some document in the collection of documents $\mathcal{D}$ and w refers to some word in the vocabulary $\mathcal{W}$.

- What is the distribution of words given a specific topic? The euclidean distance of a word embedding to the topic vector is not a normalized quantity (i.e. we do not know what distances are near and what distances are far just by looking at the euclidean distance) and the value itself cannot be as naturally interpreted as $p(words|topics)$.

- What is the distribution of topics given a specific word? This question is perhaps more interesting and relevant to the users than the above question and the answer is contained in the distribution $p(t_i|w)$. Likewise, the model does not explicitly model the conditional distributions $p(d|t_i)$ and $p(t_i|d)$, which are useful in answering questions related to the typicality of documents (d) in topic t_i and the distribution of topics in a document. Finally, we note that these topic models are not personalizable which makes them harder to adapt to different domains and user requirements.

Contributions. The contributions of this study can be summarized as follows:

- We formulate Top2Vec within a probabilistic framework, called Probabilistic Top2Vec.
- We propose personalization via fine tuning the SBERT document embeddings by incorporating user feedback.
- We evaluate the techniques for personalized topic modelling quantitatively by constructing oracles for simulating human interactions.

Problem Statement Given a document collection $\mathcal{D}$ and user feedback δ, fine-tune the SBERT model such that the revised topic allocations follow user feedback. The key research questions are as follows:

- What are the choices for user feedback δ (word level or document level feedback) and the corresponding tradeoffs between them?
- How can the user feedback be incorporated into the SBERT model loss?
- What is a suitable fine-tuning procedure?

We introduce the related work and background in the Background chapter. Next, we describe our methodology contributions – Probabilistic Top2Vec and Personalized topic modelling – in the Methodology chapter. Finally, we describe the evaluation protocol in the Experiments chapter wherein we also describe the construction of Oracles to mimic Human feedback.

Chapter 2

Background

In this section, we examine several approaches, such as self-supervised learning, contrastive learning, active learning, and human-in-the-loop, as well as topic modelling techniques. These approaches aim to address the limitations in Top2Vec. The main contribution of our work is to investigate how user feedback can be utilized for a contrastive training objective. We assume that users have domain knowledge and can provide feedback on which groups of documents are similar or dissimilar and which words are relevant to a given topic. By leveraging user feedback, we aim to improve the accuracy and efficiency of our contrastive learning approach and enable users to have greater control over the topic modelling process. This study presents a novel perspective on how human input can enhance the quality of unsupervised learning techniques like contrastive learning.

2.1 Self-supervised learning in NLP

Self-supervised learning is a machine learning technique that predicts one part of the input using another part of the input that is not hidden. This technique trains the model to find the pretext of any supervised task, such as classification or regression. One popular self-supervised learning model is BERT, or Bidirectional Encoder Representations from Transformers (Devlin et al., 2019). BERT has achieved state-of-the-art results in various NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI) (Williams et al., 2018), Semantics similarity, and text summarization. BERT's key technical innovation is bidirectional training of a Transformer to language modelling, which is different from previous efforts that only looked at a text sequence from left to right or used left-to-right and right-to-left training. Sentence-BERT uses Siamese and triplet network structures (Reimers and Gurevych, 2019). This modification allows for deriving semantically

meaningful sentence embeddings that can be compared using cosine-similarity methods to organize the corpus. XLNet is another auto-regressive language model (Yang et al., 2019) that outputs the joint probability of a sequence of tokens based on the transformer architecture with recurrence. Unlike previous models that only consider tokens to the left or right of the target token, XLNet's training objective calculates the probability of a word token conditioned on all permutations of word tokens in a sentence. The contribution of the XLNet model is to predict each word in a sequence using any combination of other words in that sequence. T5, or Text-to-Text Transfer Transformer, is a transformer-based architecture that uses a text-to-text approach (Raffel et al., 2020). This approach casts every task, including translation, question answering, and classification. The Text-to-Text Transfer Transformer model allows for using the same model, loss function, or hyperparameters across diverse tasks. Compared to BERT, T5 adds a causal decoder to the bidirectional architecture and replaces the fill-in-the-blank cloze work with a mix of alternative pre-training tasks. These models are relevant to our work since we use them to generate embeddings.

2.2 Contrastive Learning in NLP

The objective of contrastive learning is to uncover the general features of unlabelled datasets by developing document or word embeddings that highlight similarities and differences among them. When the embeddings are similar, sample pairs are kept close together, whereas dissimilar pairs are pulled apart from similar ones. SimCSE (Gao et al., 2021) is a framework for generating sentence embeddings using unsupervised contrastive learning. It predicts an input sentence using a contrastive objective, with "entailment" and "contradiction" pairs serving as positives. DeCLUTR (Giorgi et al., 2021) is a method for learning universal sentence embeddings with a self-supervised objective that does not rely on labelled training data. By minimizing the distance between embeddings of randomly selected textual segments from the same document, the objective learns universal sentence embeddings. This approach yields models that are competitive with BM25 on many domains or applications, even without supervised training data (Izacard et al., 2022).

2.3 Active Learning/Human-in-the-loop

In the realm of fine-tuning language models, human-in-the-loop and active learning techniques (Xu et al., 2013) play a pivotal role. These approaches enable AI models to leverage human input to enhance their algorithms. One of the critical steps that necessitate human contribution is data labeling. In recent research, Smith et al. (2018) proposed a human-in-the-loop topic modelling technique that empowers users to direct the creation of topic models and enhance model quality without requiring them to be experts in topic modelling algorithms. Similarly, Cai et al. (2018) employed an interactive methodology to guide the topic model curation process by soliciting user feedback and modifying the clustering technique, encompassing tasks such as generating, interpreting, diagnosing, and refining. Moreover, (Sherkat et al., 2020) proposed an interactive document clustering approach based on user feedback that characterizes each cluster with a set of keywords. In contrast, our approach differs in training the neural embedding model to learn from user preferences.

2.4 Sentence-BERT

The paper on Sentence-BERT aims to modify the standard pre-trained BERT network by utilizing siamese and triplet networks to create sentence embeddings for each sentence. These embeddings can be compared using cosine-similarity, making it possible to perform semantic searches for a large number of sentences in just a few seconds of training time Reimers and Gurevych (2019). Unlike the original cross-encoder architecture of BERT, Sentence-BERT uses a siamese architecture, which contains two BERT architectures that are identical and share the same weights. During training, Sentence-BERT processes two sentences as pairs, feeding sentence A to BERT A and sentence B to BERT B, resulting in two embeddings: one for sentence A and one for sentence B. After pooling is applied, mean-pooling is used to generate two embeddings, which are concatenated and trained using a softmax-loss function. During inference, the two embeddings are compared using cosine similarity, which produces a similarity score for the two sentences. In our project, we will use Sentence-BERT along with contrastive-based learning to fine-tune the model and differentiate between similar and dissimilar document pairs.

2.5 Clustering

Clustering aims to find structure in the data and group similar data points together. In contrast, topic modelling aims to discover latent topics that underlie a corpus of text documents, with the output being a set of topics represented by a distribution of words. Soft clustering assigns a degree of membership to each cluster data point, allowing documents to belong to multiple clusters with varying importance; this way, each cluster can be interpreted as a topic and a document can be seen as a mixture of topics. Soft-clustering has been shown to guide the discovery of topics and has been adopted by recent topic-modeling algorithms such as Top2Vec and BERTopic Sia et al. (2020); in this section, we briefly review Gaussian Mixture Models as a candidate soft-clustering algorithm.

Gaussian Mixture Models (GMM) is a probabilistic model used for clustering (Tribbey, 2010) and density estimation, where a dataset is modelled as a mixture of several Gaussian distributions. GMM assumes that each data point is generated from one of these Gaussian distributions with a certain probability. GMM is useful for handling non-spherical or overlapping clusters and provides a probabilistic output for anomaly detection or uncertainty estimation tasks. Hierarchical Gaussian Mixture Models (HGMM) is an extension of GMM that allows for a hierarchical representation (Sun et al., 2004) of the data. Each data point is modeled as a mixture of several Gaussian distributions, with each Gaussian representing a cluster at a certain level of the hierarchy. HGMM can handle datasets with non-spherical or overlapping clusters, capture the hierarchical structure of the data, and provide a probabilistic output. GMM and HGMM are soft clustering techniques that assign each data point a probability of belonging to each cluster rather than a binary assignment to a single cluster. This is useful when data points are ambiguous or when a single data point may belong to multiple clusters to varying degrees.

2.6 Topic modelling

Document clustering and topic modelling are two widely used techniques for analyzing large collections of documents. The former aims to group together documents that share similar content, which is useful for tasks such as document organization,

browsing, summarization, and classification. On the other hand, topic modelling aims to discover the underlying themes or topics that are present in a large collection of documents (Blei et al., 2003). In this project, we have used Top2Vec, a topic modelling algorithm (Angelov, 2020) that leverages joint document and word embeddings to identify topic vectors. Top2Vec does not require any pre-processing steps such as stop-word removal or stemming and automatically determines the number of topics present in the corpus. It also uses c-TF-IDF (Grootendorst, 2022) to identify the most relevant words for each topic. We compared Top2Vec to BERTopic, a probabilistic generative model that uses transformer models to identify topics in text. They found that Top2Vec produced more informative and representative topics compared to BERTopic. This could be due to the fact that Top2Vec's joint document and word embeddings capture more of the semantic relationships between words and documents. Overall, the use of Top2Vec for topic modelling can be very useful in a variety of applications, including information retrieval, text classification, and summarization. Its ability to automatically determine the number of topics and identify the most relevant words for each topic and applies a clustering technique called HDBSCAN (McInnes et al., 2017) makes it a powerful tool for analyzing large collections of documents.

2.7 Interactive Topic modelling Systems

iVisClustering (Lee et al., 2012) proposed a framework that combines Latent Dirichlet Allocation (LDA) topic modelling with interactive visualization techniques to facilitate user exploration and refinement of document clusters. The LDA algorithm is used to identify the underlying topics in the corpus of documents. The resulting topic proportions for each document are used as input to the visualization tool. This framework enables users to interactively explore and cluster the documents based on their similarity in topic space, and to refine the clustering by manually merging or splitting clusters. UTOPIAN (Choo et al., 2013) is a topic modelling framework that uses interactive non-negative matrix factorization (iNMF) to extract and visualize topics from large text datasets. The framework is designed to be user-driven, where users can interactively steer the topic modelling process by providing feedback on

the extracted topics and visualizations. UTOPIAN preprocesses the text data, represents it as a term-document matrix, and then factorizes it into two matrices using iNMF, one representing the topics and the other representing the document-topic distributions. UTOPIAN provides various interactive visualizations, such as word clouds, bar charts, and heatmaps, to help users interpret the extracted topics. One of the key features of UTOPIAN is its ability to handle multiple views of the same dataset, allowing users to create different topic models by selecting different subsets of the text data or applying different preprocessing techniques. Other examples, that use traditional topic modelling algorithms to build interactive topic modelling systems include ConVisIT (Hoque and Carenini, 2015) and ITM (Hu et al., 2011). In contrast to UTOPIAN, we use deep neural text embedding models to drive our topic discovery model. In this work, we focus on model finetuning techniques and leave visualization-related research directions such as those described in UTOPIAN for future work.

One issue with the current literature is that while topic modelling can help understand the content of a large text corpus, the discovered topics may not always be easily interpreted by human analysts. Additionally, existing frameworks are limited in their ability to answer questions such as the next relevant topic for a given document or the distribution of documents, words, and topics within a collection. Probabilistic and interactive topic modelling addresses these challenges by enabling user feedback and facilitating a more comprehensive understanding of document collections.

Chapter 3

Methodology

In this chapter, we present our two main contributions: the probabilistic Top2Vec algorithm and the Personalized Topic modelling algorithm. The probabilistic Top2Vec algorithm improves upon the Top2Vec algorithm in that all the results are expressed in terms of probabilities: this allows the user to answer questions related to the relationship between topics, documents and words using probabilities. Finally, we describe the personalization framework wherein we detail two feedback strategies: Document-level and word-level feedback. Document-level feedback refers to feedback on the document in which the user analyzes a given document and provides feedback indicating the desired topic assignment. On the other hand, word-level feedback involves providing feedback at the topic level wherein a user describes each topic in terms of its central words and guides the topic discovery and assignment. This section presents the proposed work and formally defines the problem in the focus of this work. An overview is shown in Figure 3.2. We also introduce the primary notation and terms used throughout the document.

3.1 Probabilistic Top2Vec

In this section, we propose and describe the probabilistic Top2Vec algorithm which allows us to model the conditional probability distributions between topics, documents and words. As a first step, we propose to replace the HDBSCAN algorithm with Gaussian Mixture Models (GMMs) because GMMs allows us to compute cluster membership probabilities in closed form, i.e., we represent the relationship between topics (clusters) and documents as a GMM. Subsequently, we model the distribution of words in a document as a categorical distribution and then use this to compute the probabilistic relationship between topics and words. In contrast, the Top2Vec algorithm uses euclidean distances between topic and word vectors to model the relationship between topics and words and is not as easily interpretable as probabilities.

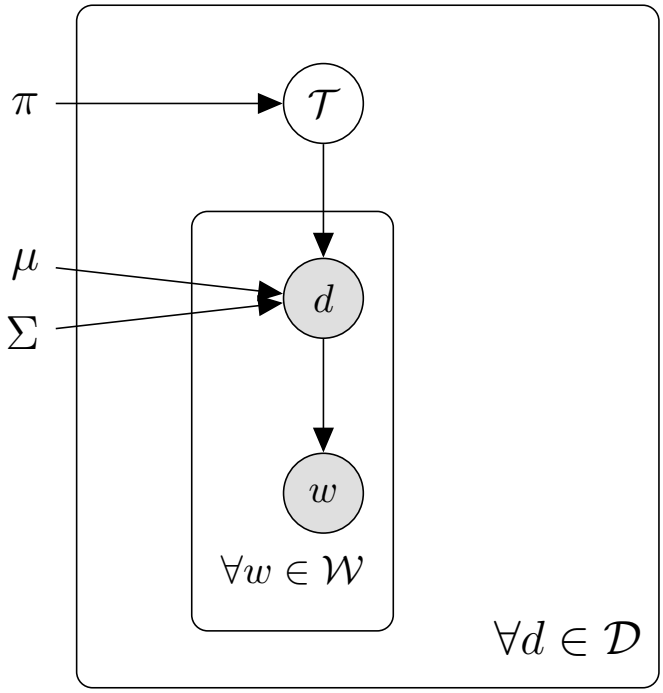


Figure 3.1: **Probabilistic Top2Vec**:We describe the probabilistic Top2Vec in terms of a box-plate notation. The random variables are denoted in circles: the observed variables d (a document from a given document collection $\mathcal{D}$) and w (a word from a given vocabulary $\mathcal{W}$) are shaded. Given a document collection $\mathcal{D}$ and vocabulary $\mathcal{W}$, the latent topic-distribution vector $\mathcal{T}$ is inferred by fitting a Gaussian Mixture Model, whose parameters are π, μ and Σ. $p(w|d)$ is computed as a normalized TF-IDF as described in Algorithm 1. Using this graphical model, we can now compute $p(w|d)$, $p(d|t)$, $p(w|t)$, $p(t|d)$, $p(t|w)$.

Let $\mathcal{D}$ denote the document collection. The documents are processed by neural text models such as the Sentence-BERT to derive document embeddings $\mathrm{E} \in \mathcal{R}^n$. Next, we propose to model the distribution of words in a document as the categorical distribution, whose probabilities are defined as the normalized TF-IDF values: in other words, instead of simply using the word counts, we propose to consider the important words by also factoring in the IDF values. Since we will use the distribution of words in the documents to construct the distribution of words in a topic, simply using word-counts would cause stop-words to get higher probabilities in all topics. Finally, we follow the box-plate diagram in Figure 3.1 for computing the joint probabilities $p(w|d)$, $p(d|t)$, $p(w|t)$, $p(t|d)$, and $p(t|w)$. The detailed algorithm with annotations is described in Algorithm 1.

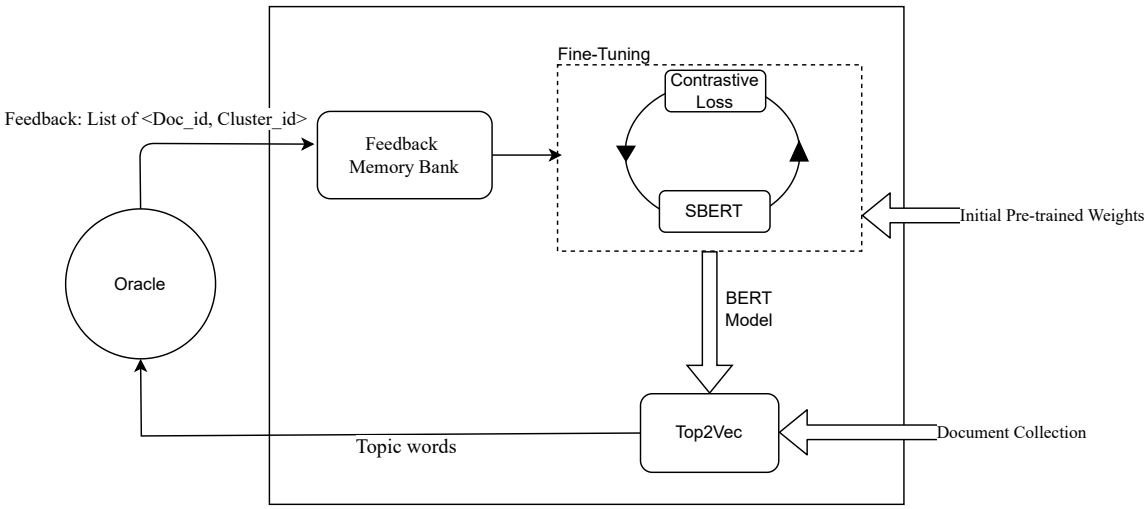


Figure 3.2: **Personalization Framework**: We describe the personalization pipeline. The Human/Oracle feedback is used to define a contrastive loss for fine-tuning the pre-trained SBERT model. In our proposed pipeline, the personalization is an iterative process wherein the Human/oracle analyses the Top2Vec output and provides feedback for personalizing the topic model.

3.2 Personalized Topic Modelling

In this section, we describe two alternative feedback formats for personalizing the topic model: the first one is user feedback on documents, and the latter one is feedback on words. Broadly, we use the user feedback for constructing a contrastive loss that we optimize by updating the underlying neural embedding model—for example, the SBERT model. The detailed algorithm with annotations is described in Algorithm 2.

3.2.1 Incorporating Document-Level Feedback

In this section, the user provides document-level feedback indicating the preferred target clusters for a given document (or sets of documents)—these target clusters could optionally be -1 to denote that the user is not aware of the precise target cluster. If the user prefers a document to be placed in a target cluster $t \neq -1$ instead of its current cluster s, we incorporate this feedback into the model by minimizing the document's embedding distance from the documents identified as topic t and maximizing the embedding distance from the documents identified as topic s. On the other hand, if the target cluster $t = -1$, we only maximize the document's

embedding distance from the documents identified as topic s. Formally, let us denote the list of documents identified as topic i by $\mathcal{D}(i)$. Let the user feedback be denoted as an ordered set of document-label pairs δ with δ_d and δ_t referring to the ordered set of documents and target clusters respectively. If the k-th document-label pair can be denoted as δ^k and the remainder set $\overline{\mathcal{D}(i)} = \mathcal{D}(i) - \delta_d$, the contrastive-loss for the i-th topic cluster can then be defined as:

$$\mathcal{L}_{CL}(t_i) = \sum_{\substack{1 \leq k \leq |\delta| \\ d \in \overline{\mathcal{D}(t_i)}}} \mathbb{I}\left[t_i == \delta_t^k\right] \left|\left|f(d) - f(\delta_d^k)\right|\right| - \mathbb{I}\left[\delta_d^k \in \mathcal{D}(i)\right] \left|\left|f(d) - f(\delta_d^k)\right|\right| \quad (3.1)$$

where, f is a text-embedding network, $||\cdot||$ denotes any valid distance metric and $\mathbb{I}$ is a binary indicator function. The two parts of the loss can be understood as follows: if the k-th feedback requires document δ_d^k to be similar to documents labeled as topic t_i, then the embedding distances are to be reduced; on the other hand, if the document δ_d^k belongs to $\mathcal{D}(i)$, we maximize the distance of this document from the remainder set $\overline{\mathcal{D}(i)}$. Observe that we take care to loop over the remainder set in order to avoid using documents that require to be moved in computing the losses; also observe that the second indicator function makes use of the original set.

3.2.2 Incorporating Word-Level Feedback

In this setting, we receive user suggestions in the form of a list of *central* words for each topic. Let's denote the list of words for i-th topic by $\mathcal{W}(t_i)$. For each topic t_i, we use BM25 ranking to retrieve K documents from the document collection using $\mathcal{W}(t_i)$ as the search query: each of these documents can be *weakly* labeled as belonging to topic t_i. Our goal now is to construct a contrastive loss term such that the distances between documents belonging to the same topic are minimized and those belonging to different topics are maximized. If we use $\mathcal{D}(t_i)^j$ where $1 \leq j \leq K$ to denote the j-th document in the list of retrieved K documents for topic t_i, the contrastive loss can be defined for topic t_i as:

$$\mathcal{L}_{CL}(t_i) = \sum_{1 \,\leq\, p,\, q \,\leq\, K} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{D}(t_i)^q)|\,| - \sum_{\substack{1 \,\leq\, p,\, q \,\leq\, K \\ j \neq i}} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{D}(t_j)^q)|\,| \quad (3.2)$$

where f is a text-embedding network and $||\cdot||$ denotes any valid distance metric such as the euclidean distance or cosine similarity. The two pieces of the loss can be understood as follows: in the first piece, the embedding distance between all pairs of documents *weakly* labelled as belonging to the same topic t_i is minimized, whereas, in the second piece, the embedding distance between all pairs of documents *weakly* labelled as belonging to different topics t_i and t_j is maximized. The total loss can now be written as the sum of the contrastive losses for each topic t_i taking care to consider each topic pair (t_i, t_j) only once:

$$\mathcal{L}_{CL} = \sum_{\substack{1 \leq p,\ q \leq K \\ 1 \leq i \leq \mathcal{N}_{\mathcal{T}}}} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{D}(t_i)^q)||\,| - \sum_{\substack{1 \leq p,\ q \leq K \\ i \leq j}} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{D}(t_j)^q)||\,| \tag{3.3}$$

In practice, optimizing contrastive losses is difficult and requires large batch sizes to enable fast training. In this particular application, however, we can modify the above contrastive loss by introducing a *pseudo-document* $\mathcal{PD}(t_i)$ that simply consists of all words $\mathcal{W}(t_i)$ for each topic t_i and rewrite Eq. 3.3 as follows:

$$\mathcal{L}_{CL} = \sum_{\substack{1 \leq p \leq K \\ 1 \leq i \leq \mathcal{N}_{\mathcal{T}}}} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{PD}(t_i))||\,| - \sum_{\substack{1 \leq p \leq K \\ i \neq j}} ||f(\mathcal{D}(t_i)^p) - f(\mathcal{PD}(t_j))||\,| \tag{3.4}$$

It is straightforward to show that Eq. 3.4 is identical to Eq. 3.3 in terms of the optimization goal. We also note that this alternative objective also helps eliminate the quadratic complexity. Finally, we point out that the pseudo-document that contains the core topic words helps inform the embedding network f about *how* the embedding distances are to be measured. In generic applications of the contrastive loss, we cannot usually know beforehand what embedding each document is supposed to move close to or away from—in other words, while we know that all semantically similar documents should have similar embeddings, we may not know what those embeddings should be. One could also briefly consider designating a randomly chosen document from each document as the pseudo-document, but that would lead to undesirable bias in the training objective. We also observe that Eq. 3.4 resembles the standard cross-entropy loss in the sense that distance is reduced with respect

to one topic's pseudovector and maximized with respect to everything else. We observed that rewriting Eq. 3.4 as a cross-entropy loss by using the pseudo-document embeddings as class normals (i.e., softmax parameters) would strongly penalize the network that leads to catastrophic forgetting and overfitting to the small weakly-labelled dataset (see paragraph in Section 4.3.1 for more details)—likewise, we also found in our experiments that directly reframing this as a classification task with a randomly initialized softmax head also does not yield the desired topic-word distributions.

Finally, we also consider the unlabeled documents and use the closest topic as the pseudo label and extend Eq. 3.4 for unlabeled documents as:

$$\mathcal{L}_{CL}^{u} = \sum_{d \in \mathcal{D} - \bigcup_i \mathcal{D}(t_i)} \min_i ||f(d) - f(\mathcal{PD}(t_i))||_2 \tag{3.5}$$

We combine the two losses using a weighting factor λ as:

$$\mathcal{L} = \mathcal{L}_{CL} + \lambda \mathcal{L}_{CL}^{u} \tag{3.6}$$

In our experiments, we use $\lambda = 0.1$.

Algorithm 1 Top2Vec with Probabilities. (Section 3.1)

Input:

$\mathcal{D}$: Collection of all documents.
$\mathcal{N}_\mathcal{T}$: Number of topics.
u: The UMAP dimensionality reduction parameter.
SBERT: A pretrained (and, possibly finetuned) sentence BERT model.

Output:

$\mathcal{W}$: set of words
$p(w|d)$: matrix of shape $|\mathcal{W}| \times |\mathcal{D}|$
$p(d|t)$: matrix of shape $|\mathcal{D}| \times \mathcal{N}_\mathcal{T}$
$p(w|t)$: matrix of shape $|\mathcal{W}| \times \mathcal{N}_\mathcal{T}$
$p(t|d)$: matrix of shape $\mathcal{N}_\mathcal{T} \times |\mathcal{D}|$
$p(t|w)$: matrix of shape $\mathcal{N}_\mathcal{T} \times |\mathcal{W}|$

E $\leftarrow$ SBERT($\mathcal{D}$) ▷ Obtain the document embeddings E with SBERT applied to document collection $\mathcal{D}$
$\mathrm{E}_p \leftarrow$ UMAP(E, u) ▷ Project $\mathrm{E} \rightarrow \mathrm{E}_p \in \mathcal{R}^u$ with UMAP
$\pi, \mu, \Sigma \leftarrow$ GMM($\mathrm{E}_p, \mathcal{N}_\mathcal{T}$) ▷ Cluster the embeddings E_p using GMM with $\mathcal{N}_\mathcal{T}$ number of clusters

- $\pi \in \mathcal{R}_+^{\mathcal{N}_\mathcal{T}}$ contains the mixture probabilities and sums to 1.0
- $\mu \in \mathcal{R}^{\mathcal{N}_\mathcal{T} \times u}$ contains the component-wise means
- $\Sigma \in \mathcal{R}_+^{\mathcal{N}_\mathcal{T} \times u \times u}$ contains the component-wise covariances. The $\mathcal{N}_\mathcal{T}$ square matrices are positive semi-definite.

DT, $\mathcal{W} \leftarrow$ TFIDF($\mathcal{D}$) ▷ Returns vocabulary $\mathcal{W}$ and a Document-term matrix of shape $|\mathcal{W}| \times |\mathcal{D}|$ by applying TFIDF.
$p(w|d) \leftarrow$ DT.normalize(axis=0) ▷ Create the $p(w|d)$ matrix by normalizing DT such that $\sum_{w \in \mathcal{W}} p(w|d) = 1.0$.

- While the usual policy may be to compute $p(w|d)$ using just the term-frequencies, we anticipate that the probabilities of the stop-words may get accentuated and thus use the TFIDF values instead of the term-frequencies alone.

$m \leftarrow$ new Matrix($|\mathcal{D}|, \mathcal{N}_\mathcal{T}$) ▷ Initialize a matrix of zeros.
for i in $1 \ldots |\mathcal{D}|$ do
  for j in $1 \ldots \mathcal{N}_\mathcal{T}$ do
    m[i,j] $\leftarrow N(\mathrm{E}_p[i]; \mu[j], \Sigma[j])$

- Computes the density of the projected embedding of the i-th document $\mathrm{E}_p[i]$.
- The mean and covariance matrix of the j-th cluster are given by $\mu[j]$ and $\Sigma[j]$.

$p(d|t) \leftarrow m$
$p(w|t) \leftarrow p(w|d)p(d|t)$ ▷ This is a matrix multiplication.
$p(t|d) \leftarrow [p(d|t) * \pi$.reshape(-1,1)$]^\top$.normalize($axis = 0$)
$p(t|w) \leftarrow [p(w|t) * \pi$.reshape(-1,1)$]^\top$.normalize($axis = 0$)

- $*$ denotes elementwise multiplication and $(.)^\top$ denotes the matrix transpose.
- As the set of documents $\mathcal{D}$ is finite, $\sum_{d \in \mathcal{D}} p(d|t) < 1.0$ and $p(d|t)$ is not normalized. Likewise, $p(w|t)$ is not normalized.

return $\mathcal{W}$, $p(w|d)$, $p(d|t)$, $p(w|t)$, $p(t|d)$, $p(t|w)$

Algorithm 2 Personalization Algorithm For Document-Level Feedback. (Section 3.2)

Input:

Oracle: Oracle.
$\mathcal{D}$: Collection of all documents.
$\mathcal{N}_{\mathcal{T}}$: Number of topics.
u: The UMAP dimensionality reduction parameter.
SBERT: A pretrained (and, possibly finetuned) sentence BERT model.
labels: ground truth labels
Δ : Feedback memory bank

Top2vec Output:

$\mathcal{W}$: set of words
DT: Document-term matrix of shape $|\mathcal{W}| \times |\mathcal{D}|$
$p(w|d)$: matrix of shape $|\mathcal{W}| \times |\mathcal{D}|$
$p(d|t)$: matrix of shape $|\mathcal{D}| \times \mathcal{N}_{\mathcal{T}}$
$p(w|t)$: matrix of shape $|\mathcal{W}| \times \mathcal{N}_{\mathcal{T}}$
$p(t|d)$: matrix of shape $\mathcal{N}_{\mathcal{T}} \times |\mathcal{D}|$
$p(t|w)$: matrix of shape $\mathcal{N}_{\mathcal{T}} \times |\mathcal{W}|$

1: $\mathcal{W}$, $p(w|d)$, $p(d|t)$, $p(w|t)$, $p(t|d)$, $p(t|w)$ $\leftarrow$ `Top2Vec(`$\mathcal{D}$`,` $\mathcal{N}_{\mathcal{T}}$`,` u`, SBERT)` ▷ `Apply top2vec algorithm with pre-trained BERT weights and obtain the vocabulary` $\mathcal{W}$ `and a Document-term matrix of shape` $|\mathcal{W}| \times |\mathcal{D}|$
2: δ`[<doc#, cluster#>]` $\leftarrow$ `Oracle(`$\mathcal{D}$`, labels, SBERT)` ▷ `Feedback` δ `as a list of Document IDs and suggested target cluster`
3: `Update` Δ `with` δ ▷ `The feedback as a list of Document IDs and Cluster IDs are stored in the feedback memory bank`
4: `Fine-tune SBERT using` Δ
5: `Go to Step 1`

Chapter 4

Experiments

In this chapter, we describe a framework for evaluating the proposed Personalized Top2Vec and report the results obtained with it. In order to evaluate our model on the personalization task, we make use of labeled classification datasets and construct an oracle to interact with our system. The end goal of the oracle is to provide feedback using the ground truth labels to align the Top2Vec topic's assignments with the ground truth labels. In the following, we describe the evaluation metric, the design and rationale behind the construction of Oracles and the final experimental results.

Experiment Setup. Our decision to make use of labeled classification datasets for evaluation of the topic modelling results follows previous work (e.g., (Angelov, 2020)) and is not intended to suggest that text classification is same as topic modelling. The number of classes and their semantics are fixed in a text classification system whereas the number and semantics of topics are not fixed or predefined in advance. In fact, our model is designed to accept user feedback about the preferred semantics per topic (e.g., central words); more crucially, text classification usually assigns a single label per document, whereas a document can discuss more than one topic. Constructing an oracle that simulates a real human user in terms of feedback quality can be quite hard: for example, a human user may be more accurate if they analyze one cluster at a time whereas analyzing several clusters would not be hard for a software bot. Likewise, the errors and noise introduced in human feedback cannot always be accurately modeled by simple programs and the true resilience of the system may only be evaluated with a user study. We acknowledge the above-mentioned drawbacks with an oracle-based evaluation and yet, propose to evaluate the performance of the system using an Oracle to obtain *cheap* evaluations of the topic-modelling performance that will ultimately help us in fixing the appropriate hyperparameters of the algorithm in

advance of the real user-based evaluation.

In order to evaluate our model, we use cluster purity as the metric and track how the purity improves with feedback from the oracle. We consider the following definition of cluster purity P:

$$P = \frac{1}{\mathcal{N}_{\mathcal{T}}} \sum_{i} \frac{\max_j |\mathcal{D}(i) \cap j|}{|\mathcal{D}(i)|} \tag{4.1}$$

where, $\mathcal{D}(i)$ denotes the set of documents clustered into topic i, $|\mathcal{D}(i) \cap j|$ denotes the number of documents in cluster i having ground truth label j and $\mathcal{N}_{\mathcal{T}}$ is the number of topics and is assumed to be equal to the number of possible ground truth labels. We propose two oracles: a) Document-level Oracle: the document-level oracle is designed to analyse a single cluster in complete detail and give feedback for some of the documents in that cluster; b) Word-level Oracle: the word-level Oracle does not analyse the results of the topic modelling algorithm but instead suggests what words constitute a topic. The word-level oracle is closer to what can be expected from a human user of our topic modelling system (Sharma et al., 2015; Sherkat et al., 2020).

Homogeneity measures the extent to which instances in a cluster belong to the same class, while completeness measures the extent to which instances of a class are assigned to the same cluster. In general, Cluster Purity measures homogeneity, whereas Normalized Mutual Information measures both homogeneity and completeness. When we have a higher number of clusters, cluster purity can reach 1.0 because each cluster can contain only one point. In our evaluations, we use balanced datasets, and the number of clusters is fixed to the number of topics in this dataset therefore only use cluster purity as the evaluation metric. Furthermore, although topic modelling is a soft-clustering problem, we assume that the document belongs to the topic with the highest membership probability for the purposes of evaluation.

If the evaluation was conducted by interacting with actual users rather than using a labeled dataset as ground truth, it would change the evaluation setting significantly. In such cases, other metrics like user satisfaction, task completion time, and accuracy of user labeling would become important. The evaluation would also require more subjective judgments on the quality of the clustering based on how well it meets

the user's needs and expectations. Additionally, the evaluation would need to take into account the user's domain knowledge, preferences, and biases, which may not be reflected in the ground truth labels. We leave this exploration for future work.

Datasets We use the following datasets:

1. **20Newsgroups**: We use all 20 topics (Pedregosa et al., 2011) which contains 18,831 of labelled classes based on Newsgroups categories.

2. **WvsH**: This dataset (Sharma et al., 2015) is derived from 20Newsgroups and consists of separating documents related to the following two topics – comp.os.ms-windows.misc and comp.sys.ibm.pc.hardware – and contains 1176 documents.

3. **Nova**: This is derived from the 20-newsgroups dataset and contains 12k documents: it consists of separating talk/atheism/religion related articles from the remaining Guyon et al. (2011).

4. **SimVsReal**: This consists of 48K documents from the SRAA dataset (Sharma et al., 2015) which separates documents related to simulated driving/flying vs real driving/flying.

5. **AutoVsAviation**: This consists of 48K documents from the SRAA dataset (Sharma et al., 2015) which separates documents on automobiles vs aviations.

The WvsH, Nova and SRAA are all binary datasets whereas 20 Newsgroups is a dataset of 20 classes.

4.1 Document-Level Oracle Definition: Providing Document-Level Feedback

In this section, we introduce the Document-Level Feedback Oracle for simulating a real user and providing user-feedback δ as described above. In each iteration, we apply the Top2Vec algorithm over document collection $\mathcal{D}$ with $\mathcal{N}_{\mathcal{T}}$ number of topics and present the results to the oracle, which evaluates the topic-modelling results and provides feedback. In the first iteration, we use a pre-trained text-embedding model

such as Sentence-BERT; in subsequent iterations, we use the fine-tuned model, which is fine-tuned using oracle provided user-feedback following the description in Section 3.2.1.

In order to construct the user feedback, we propose the following policy: a) Identify the most impure cluster t_{impure}; b) Find the ground truth label c having the highest representation amongst the documents clustered into t_{impure}; c) Construct a user-feedback using the documents having ground truth label c and clustered into t_{impure} with the target label set to -1. While this oracle analyses all the clusters to identify the most impure cluster, the feedback is targeted towards a single class and does not indicate the target cluster; we also note that the number of documents having the highest class representation is the smallest in the most impure cluster t_{impure}. Nevertheless, the amount of human effort to replicate this oracle on poorly clustered documents is not practical as it requires a detailed inspection of the documents before providing feedback; we include this oracle for reference and development purposes.

4.2 Word-Level Oracle Definition: Providing Word-Level Feedback

We describe how Word-Level Oracle provides feedback through a list of topics and their central words as feedback for guiding personalization. Contrary to the Document-level Oracle, the Word-Level Oracle does not construct feedback based on the results of the topic modelling algorithm and instead relies upon the user's domain expertise; in practice, this setting is more practical than the one described above. In order to extract the central words for each topic, we apply the following steps: a) Train a logistic regression classifier on the TF-IDF representations of the documents; b) From the learned weight matrix, choose the top 10 words which have the highest absolute weight for each class. We use the described methodology to select the central words for each topic that the oracle will provide as feedback.

4.3 Results

In this section, we describe and analyze the results obtained by evaluating our model on five datasets using Document-Level and Word-Level Oracles and their combination. All models were fine-tuned with a batch size of 200 using the AdamW optimizer

Class	Class Words
alt.atheism	atheism, god, religion, islam, atheists, islamic, bobby, bible, deletion, motto
comp.graphics	graphics, image, 3d, images, files, format, tiff, pov, cview, polygon
comp.os.ms-windows.misc	windows, file, ax, cica, driver, files, drivers, dos, problem, fonts
comp.sys.ibm.pc.hardware	drive, scsi, card, pc, controller, bus, monitor, ide, 486, port
comp.sys.mac.hardware	mac, apple, quadra, drive, centris, se, simms, lc, duo, powerbook
comp.windows.x	window, server, motif, widget, xterm, x11r5, mit, application, sun, widgets
misc.forsale	sale, offer, shipping, sell, condition, 00, new, email, interested, asking
rec.autos	car, cars, engine, ford, oil, dealer, toyota, auto, vw, gt
rec.motorcycles	bike, dod, bikes, ride, motorcycle, helmet, riding, bmw, dog, motorcycles
rec.sport.baseball	baseball, year, team, game, braves, runs, players, games, stadium, cubs
rec.sport.hockey	hockey, team, game, nhl, season, play, players, games, playoffs, leafs
sci.crypt	key, clipper, encryption, nsa, government, keys, chip, security, crypto, clinton
sci.electronics	circuit, electronics, power, voltage, ground, radio, output, amp, current, tv
sci.med	msg, doctor, disease, medical, pain, patients, treatment, food, cancer, health
sci.space	space, nasa, orbit, launch, moon, spacecraft, shuttle, earth, lunar, solar
soc.religion.christian	god, church, christians, jesus, christ, christian, christianity, faith, bible, sin
talk.politics.guns	gun, guns, weapons, fbi, firearms, weapon, batf, law, government, nra
talk.politics.mideast	israel, israeli, jews, armenians, turkish, arab, armenian, turkey, arabs, jewish
talk.politics.misc	government, tax, clinton, people, drugs, president, men, trial, gay, state
talk.religion.misc	god, jesus, christian, koresh, objective, kent, christians, bible, morality, rosicrucian

Table 4.1: Class words retrieved for 20Newsgroups dataset following Section 4.2.

with a learning rate set to 1e-3.

4.3.1 Preliminary Experiments

In our preliminary experiments, we identified the following:

1. **UMAP**. We observed that applying UMAP (McInnes et al., 2018) before the clustering step in the Top2Vec algorithm would limit the effect of the feedback and do not apply it for generating the following results.

2. **Loss Function**. Direct application of the contrastive loss described in Eq. 3.2 resulted in catastrophic forgetting and led to dips in performance relative to the original purity despite additional feedback. Likewise, applying Sentence-BERT-type loss, wherein one applies a binary classifier to the output embeddings for fine-tuning, did not give us embeddings that formed good clusters. We apply the online contrastive loss: in this setting, the model selects negative pairs which are closer than the closest positive pairs. Similarly, the model selects the positive pairs which are farther than the closest negative pair for optimization. Using cosine distance or Euclidean distance did not produce any changes to the results, and we attribute this to the presence of LayerNorms in the transformer network.

3. **Clustering**. We found that Gaussian mixture models perform better than KMeans, especially if we restrict the covariance matrices to be diagonal.

Dataset	Before Feedback	Word-Level	Epochs	After Document-Wise Feedback	
				Without Word-Level Feedback	With Word-Level Feedback
20NewsGroups	0.57	0.75	1	0.63	0.79
			5	0.67	0.87
			10	0.61	0.88
WvsH	0.8	0.84	1	0.98	0.90
			5	0.97	0.97
			10	0.98	0.99
Nova	0.76	0.92	1	1.00	0.98
			5	1.00	0.99
			10	1.00	1.00
SimVsReal	0.8	0.96	1	1.00	1.00
			5	1.00	1.00
			10	1.00	1.00
AutoVsAviation	0.97	1.00	1	0.97	1.00
			5	0.97	1.00
			10	0.97	1.00

Table 4.2: Cluster purity captured for the five datasets in two cases: 1) before applying user feedback, and 2) after applying user feedback. Cluster purity is the mean cluster purity over all clusters. We use GMM as the clustering algorithm. The model was fine-tuned with word-level feedback for 10 epochs; in general, this can be trained until convergence and we choose the checkpoint at 10 epochs to report the results. The columns labelled "after document-wise feedback" show the results after 10 iterations of user feedback; one run of top2vec followed by one round of user feedback (which can consist of several documents) counts as one iteration. Epochs 1, 5, or 10 refer to the number of times the entire dataset has been used for the model fine tuning. One epoch means that each data point has been seen once by the model.

In order to evaluate our model, we use cluster purity as the metric and track the average cluster purity as the embedding model is updated with feedback from the oracle. We tabulate the results in Table 4.2 for different choices of epochs per feedback for all three feedback choices: word-level feedback, document-wise feedback, and their combination. We report the results at the end of 10 feedbacks and consider three choices for epochs per feedback: 1, 3 and 5; this indicates the number of passes over the dataset for each document-level feedback from the oracle. Note that the number of epochs per feedback is not used when incorporating word-level feedback. Instead, we fine-tune the network with word-level feedback using Eq. 3.6 for 10 epochs and use the last checkpoint. The word-level feedback for 20Newsgroups generated following the procedure described in Section 4.2 is shown in Table 4.1. In the following, we discuss some key observations.

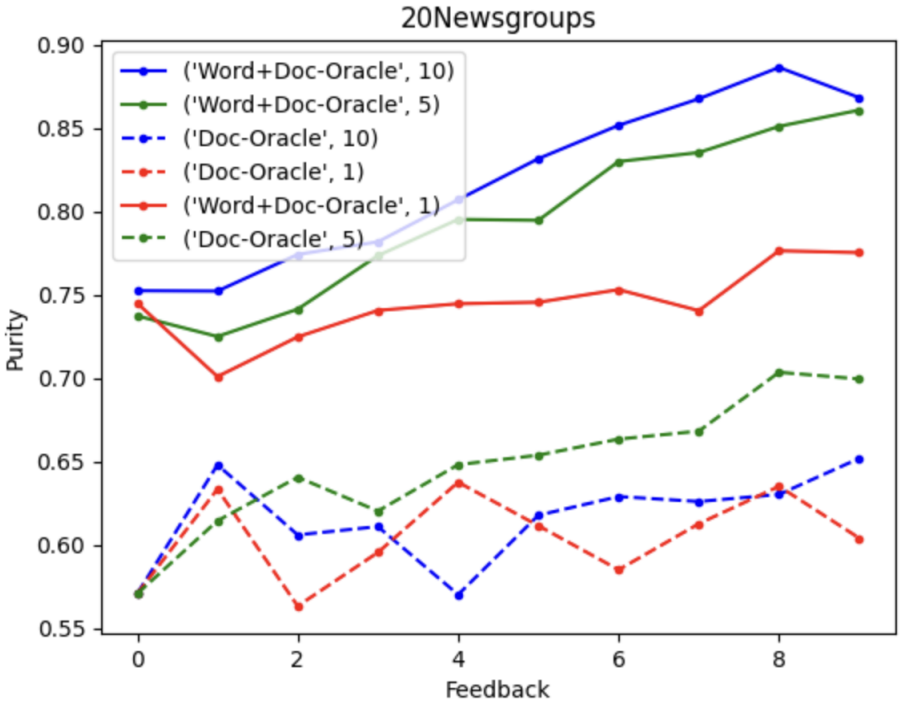


Figure 4.1: 20Newsgroups results obtained after applying a personalized topic modelling algorithm where the x-axis is feedback and the y-axis is cluster purity percentage. The dotted lines correspond to feedback by the Doc-level Oracle whereas the solid lines correspond to the combination Word-Level Oracle and Doc-Level Oracle. The color encodes the number of epochs per feedback. We observe that weak supervision helps warm-start the document-wise refinement. The dips in cluster purities are natural in the course of stochastic gradient descent training and we observe that the training stabilizes with more number of epochs especially when warm-started with word-level feedback.

4.3.2 Weak supervision helps warm-start the document-wise refinement.

For the 20Newsgroups dataset, we find that word-level feedback—i.e., weak guidance by describing central words per topic —yields a significantly higher cluster purity of 0.75 as compared to the cluster-purities obtained by document-wise feedback. Furthermore, we also see that bootstrapping document-wise feedback with word-level feedback results in higher cluster purities. For example, applying document-wise refinement at ten epochs per feedback gives us a cluster purity of 0.87 when bootstrapping with word-level feedback. On the other hand, we only get a cluster purity of 0.67 when directly fine-tuning starting with a pretrained Sentence-BERT checkpoint.

4.3.3 Analyzing Labelling Effort.

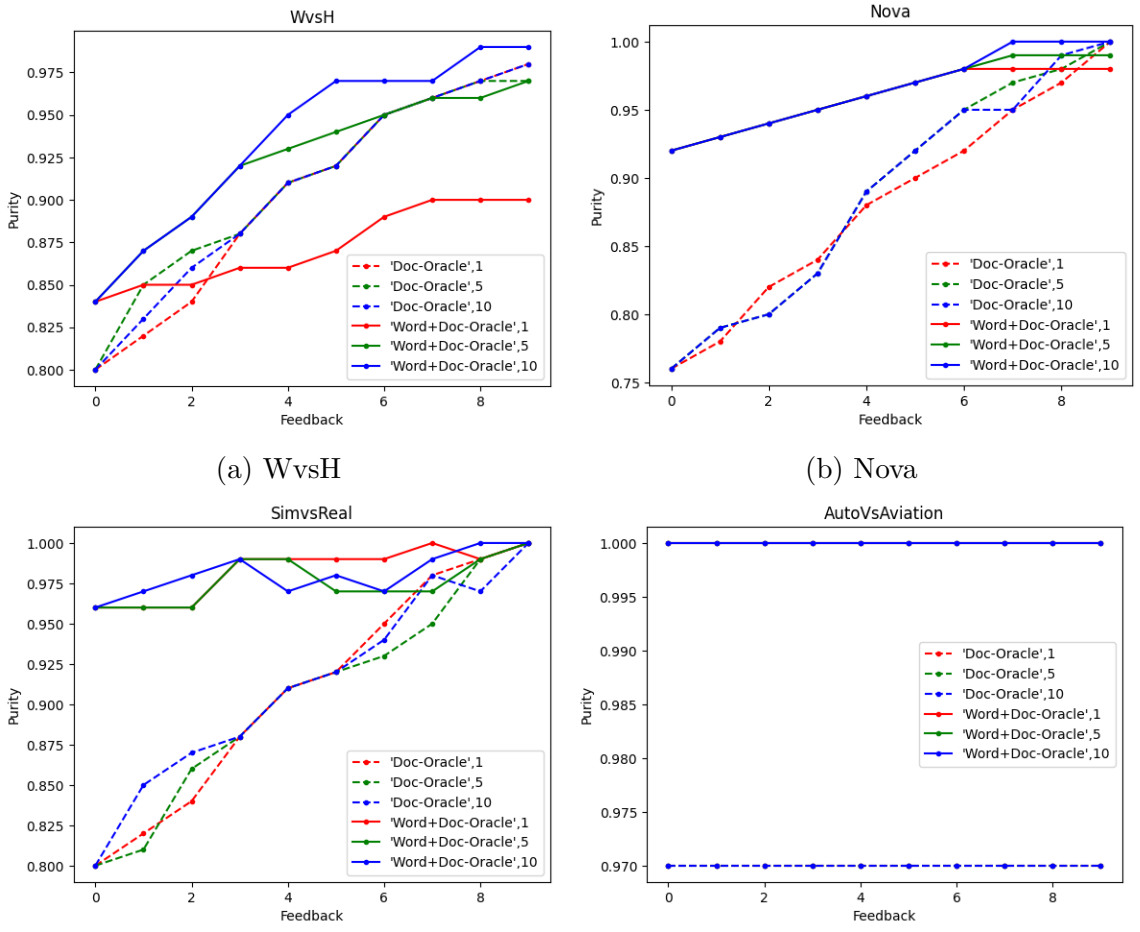
The labelling effort is generally budgeted in terms of the number of documents reviewed by the user. For word-level feedback, the human user is expected to provide a list of central words for each topic which is a much lower cost as compared to document-level feedback. In our design of the document-level oracle (Section 4.1), we do not consider the labeling budget when providing feedback to refine the most impure cluster and the labeling effort is not directly reflected in the results. In the case of the 20newsgroups dataset, the document-wise oracle provided feedback on about 5k documents over 10 feedback iterations and yet resulted in a cluster purity of just 0.67 with 10 epochs/feedback. In contrast, when bootstrapping document-wise feedback with word-level feedback, we can get a cluster purity of about 0.88 with just 1.5k documents. For the binary datasets (e.g., WvsH, Nova, SimVsReal and AutovsAviation), document-wise feedback gets near-perfect cluster-purity scores as the document level oracle indirectly labels all the documents in the most impure cluster even though the feedback is only provided for a subset of this cluster. In contrast, we see that word-level feedback produces results that are closely comparable with that of document-wise feedback with much lesser labelling effort. For the AutoVsAviation, we see that document-wise feedback does not yield any improvement as compared to the initial clustering result: as the document-wise oracle provides feedback only to some of the documents belonging to the most impure cluster, the model fine-tuning continually introduces different impurities similar to the whac-a-mole cycle. On the other hand, word-level oracle targets feedback towards all clusters, albeit at a weaker level, instead of one cluster at a time.

4.3.4 Selecting the hyperparameters of word-level feedback

We study the effects of varying number of words selected by word-level oracle (Section 4.2) and number of BM25 retrieved documents for incorporating word-level feedback (Section 3.2.2) in Table 4.3. We observe that the algorithm is fairly robust to these choices but choosing higher number of documents can introduce noise and result in lower cluster purities.

#Words \ #Documents	50	100	200
5	0.72	0.74	0.71
10	0.76	0.75	0.70

Table 4.3: We show the final cluster-purities with Word-Level Oracle for different choices of #Words and #Documents. #Words are the number of important words selected by the Word-Level Oracle in Section 4.2. #Documents are the number of documents retrieved by BM25 for incorporating word level feedback as described in Section 3.2.2.



(a) WvsH

(b) Nova

(c) SimVsReal

(d) AutoVsAviation

Figure 4.2: Results obtained after applying a personalized topic modelling algorithm where the x-axis is feedback and the y-axis is cluster purity percentage. The dotted lines correspond to feedback by the Doc-level Oracle whereas the solid lines correspond to the combination Word-Level Oracle and Doc-Level Oracle. The color encodes the number of epochs per feedback. We observe that weak supervision helps warm-start the document-wise refinement.

Chapter 5

Future Work and Conclusions

5.1 Conclusions

In conclusion, we propose a Personalized Topic Modeling framework by extending Top2Vec. We also present an algorithm for encapsulating Top2Vec results in a probabilistic setting. We evaluate the proposed Personalized Top2Vec model using labelled classification datasets and constructing an oracle to interact with the system. Our evaluation metric is cluster purity, and we proposed two types of oracles: a document-level oracle and a word-level oracle. We used five different datasets for evaluation purposes, including 20 Newsgroups, WvsH, Nova, SimVsReal, and AutoVsAviation.

We acknowledged the limitations of using an Oracle-based evaluation but proposed it to obtain inexpensive evaluations of the topic modelling performance to help us fix the algorithm's appropriate hyperparameters before a user-based evaluation.

We introduced the document-level feedback oracle to simulate a real user and provide user feedback, which can be used to fine-tune the model in subsequent iterations. We proposed an oracle/user policy, which consists of the oracle/user identifying the most impure cluster and suggesting feedback at the document level. We also introduced the word-level feedback oracle, which provides feedback on the preferred words constituting a topic.

Our experiments showed that Personalized Top2Vec achieved significantly higher cluster purity than non-personalized Top2Vec models. The results demonstrated that the proposed framework is a promising approach for introducing personalization to topic modelling.

Future Directions Our future work will integrate Probabilistic Top2Vec and personalization techniques into a visual analytics topic modelling dashboard. Our proposed method may be evaluated regarding statistical significance using k-fold validation techniques. In a user study, we will invite users to apply our model to their

document collections and perform several personalization iterations using the dashboard. The user study will support the evaluation of the effectiveness of the visual analytics dashboard and the model's fine-tuning capabilities.

Bibliography

Dimo Angelov. 2020. Top2Vec: Distributed Representations of Topics. *arXiv:2008.09470 [cs, stat]* (Aug. 2020). http://arxiv.org/abs/2008.09470 arXiv: 2008.09470.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Guoray Cai, Feng Sun, and Yongzhong Sha. 2018. Interactive Visualization for Topic Model Curation. In *Joint Proceedings of the ACM IUI 2018 Workshops co-located with the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018), Tokyo, Japan, March 11, 2018 (CEUR Workshop Proceedings)*, Alan Said and Takanori Komatsu (Eds.), Vol. 2068. CEUR-WS.org. https://ceur-ws.org/Vol-2068/esida5.pdf

Ricardo Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates, Vol. 7819. 160–172. https://doi.org/10.1007/978-3-642-37456-2_14

Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 1992–2001. https://doi.org/10.1109/TVCG.2013.212

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. https://doi.org/10.48550/arXiv.2006.03659 arXiv:2006.03659 [cs].

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. https://doi.org/10.48550/arXiv.2203.05794 arXiv:2203.05794 [cs].

Isabelle Guyon, Gavin C. Cawley, Gideon Dror, and Vincent Lemaire. 2011. Results of the Active Learning Challenge. In *Active Learning and Experimental Design workshop, In conjunction with the International Conference on Artificial Intelligence and Statistics (AISTATS)*. Sardinia, Italy, 19–45. http://jmlr.org/proceedings/papers/v16/guyon11a/guyon11a.pdf

Enamul Hoque and Giuseppe Carenini. 2015. ConVisIT: Interactive Topic Modeling for Exploring Asynchronous Online Conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces, IUI 2015, Atlanta, GA, USA, March 29 - April 01, 2015*, Oliver Brdiczka, Polo Chau, Giuseppe Carenini, Shimei Pan, and Per Ola Kristensson (Eds.). ACM, 169–180. https://doi.org/10.1145/2678025.2701370

Yuening Hu, Jordan L. Boyd-Graber, and Brianna Satinoff. 2011. Interactive Topic Modeling. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (Eds.). The Association for Computer Linguistics, 248–257. https://aclanthology.org/P11-1026/

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. https://doi.org/10.48550/arXiv.2112.09118 arXiv:2112.09118 [cs].

Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John T. Stasko, and Haesun Park. 2012. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Comput. Graph. Forum* 31, 3pt3 (2012), 1155–1164. https://doi.org/10.1111/j.1467-8659.2012.03108.x

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software* 2 (03 2017). https://doi.org/10.21105/joss.00205

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. https://doi.org/10.48550/ARXIV.1802.03426

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12, null (nov 2011), 2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. http://jmlr.org/papers/v21/20-074.html

Paulo E. Rauber, Alexandre X. Falcão, and Alexandru C. Telea. 2016. Visualizing Time-Dependent Data Using Dynamic t-SNE. In *18th Eurographics Conference on Visualization, EuroVis 2016*. Eurographics Association, 73–77. https://doi.org/10.2312/eurovisshort.20161164

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

Manali Sharma, Di Zhuang, and Mustafa Bilgic. 2015. Active Learning with Rationales for Text Classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, 441–451. https://doi.org/10.3115/v1/N15-1047

Ehsan Sherkat, Evangelos E. Milios, and Rosane Minghim. 2020. A Visual Analytics Approach for Interactive Document Clustering. *ACM Trans. Interact. Intell. Syst.* 10, 1 (2020), 6:1–6:33. https://doi.org/10.1145/3241380

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. 2020. Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 1728–1736. https://doi.org/10.18653/v1/2020.emnlp-main.135

Alison Smith, Varun Kumar, Jordan Boyd-Graber, Kevin Seppi, and Leah Findlater. 2018. Closing the Loop: User-Centered Design and Evaluation of a Human-in-the-Loop Topic Modeling System. In *23rd International Conference on Intelligent User Interfaces*. ACM, Tokyo Japan, 293–304. https://doi.org/10.1145/3172944.3172965 21 citations (Crossref) [2022-01-27].

Yunda Sun, Baozong Yuan, Zhenjiang Miao, and Wei Wu. 2004. From GMM to HGMM: An Approach In Moving Object Detection. *Comput. Artif. Intell.* 23, 3 (2004), 215–237. http://www.cai.sk/ojs/index.php/cai/article/view/427

Will Tribbey. 2010. Numerical Recipes: The Art of Scientific Computing (3rd Edition) is written by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, and published by Cambridge University Press, (c) 2007, hardback, ISBN 978-0-521-88068-8, 1235 pp. *ACM SIGSOFT Softw. Eng. Notes* 35, 6 (2010), 30–31. https://doi.org/10.1145/1874391.187410

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 1112–1122. http://aclweb.org/anthology/N18-1101

Yan Xu, Fuming Sun, and Xue Zhang. 2013. Literature survey of active learning in multimedia annotation and retrieval. In *International Conference on Internet Multimedia Computing and Service, ICIMCS '13, Huangshan, China - August 17 - 19, 2013*, Ke Lu, Tao Mei, and Xindong Wu (Eds.). ACM, 237–242. https://doi.org/10.1145/2499788.2499794

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5754–5764. https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html

Appendix A

Developer's Guide: Understanding the Codebase

Overview: This project proposes a personalized topic modeling algorithm where users can guide the method by suggesting edits to the statistical models. The algorithm builds upon Top2Vec and allows users to provide feedback about the documents, which is used to define a contrastive loss function for fine-tuning the pre-trained BERT model used to derive embeddings of documents. The proposed algorithm encapsulates Top2Vec within a probabilistic framework, and introduces two personalization techniques for weaker word-level supervision and stronger document-level supervision in guiding the topic discovery. The model is evaluated quantitatively on labelled datasets and results show that providing even weak feedback to the model can result in topic modeling that better aligns with the user's preferences, and document-level feedback further improves the results. The results of Top2Vec visualized as probabilities can enable the user to clearly understand the discovered topics and provide appropriate feedback to personalize the topic modeling result.

Installation To install the project, follow these steps:

- Clone the repository: git clone https://github.com/bhuvaneshwaribasquarane/personalized_topic_modelling.git

- Install the dependencies: pip install -r requirements.txt

Usage To use the project, do the following:

- Run the following Python scripts:

- config_script_gen.py: generates three folders: config, script and logs. The config folder contains configuration files related to num_iters, model_path, clus_method, epoch, loss, umap, workdir, oracle, and dataset. The Script folder contains scripts that specify the resources required, such as the number of GPUs and CPUs, and initiate quant_eval.py. The log folder has the logs maintained for each script execution.
- quant_eval.py: This python file takes the input from config files like num_iters, model_path, clus_method, epoch, loss, umap, workdir, oracle, and dataset to execute the model.py and oracle.py based on the version mentioned in config file. Finally, the output contains the average cluster purity and standard deviation.
- model.py: This file contains the code implementation of SBERT, UMAP, GMM, which will be invoked as part of quant_eval.py input parameters.
- oracle.py: This file does the document_level feedback implementations.
- oraclev2.py: This file does the word_level feedback implementations.
- table_script.py: Generates the average cluster purity and standard deviation results of all the datasets based on different epochs for both before and after feedback.