School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

Seed-Guided Topic Modelling

by

Chi Zhang

Thesis submitted as a requirement for the degree of

Bachelor of Engineering in Computer Engineering

Submitted: November 2022

Supervisor: Prof. Raymond Wong Student ID: z5211214

Abstract

Topic modelling is an important field in natural language processing. It helps users to extract keywords from the a large collection of documents. Some models even allow users to provide the words they are interested in and looks up relevant keywords from the documents. However, existing methods are not able to handle the user-provided topics that are not in the document collection. We proposed a method to address this issue by leveraging the extensive vocabulary of pretrained language models.

Acknowledgements

I appreciate my friend, Yiyang Nan and the compute cluster, katana, at UNSW, provided me with extra compute power to train the models

Abbreviations

CaTE Categorical-name-guided Topic Embeddings

BERT Bidirectional Encoder Representations from Transformer

PMI Pointwise Mutual Information

NPMI Normalised Pointwise Mutual Information

Contents

Chapter 1

Introduction

Being able to group words or phrases under a common theme within an extensive collection of documents automatically will help users to analyse them more efficiently and speed up certain down-stream tasks such as document summarisation. Topic models have demonstrated themselves to be powerful tools for this. Most topic models use unsupervised learning to learn the context and group words with similar meanings together. Latent Dirichlet Allocation (LDA) [BNJ03] is one classic approach in the field and has achieved promising results.

This approach has a significant limitation, which is not able to account for user preferences. For instance, when users are analysing reviews on yelp datasett, they may want cuisines names from different countries, (e.g. *Chinese food* or *Japanese food*) but the model may select words based on ingredients. (e.g. *meat* or *vegetable*) as shown in Table 1.1. To overcome this issue, seed-guided topic models are invented, which allow users to provide topics they are interested in as "seeds". These models will discriminate towards these "seeds" during learning and relate with words of close meanings. But some of these methods are not able to consider the neighbours words (i.e. local context). CaTE [Men+20] overcomes this issue by using a sliding window approach when learning the relationships between words.

All previous methods assume the seed words appear in the corpus at least once, so

Dataset	**Seeds**	
SciDoc	cardiovascular disease	**hepatitis a/b/c/e**
	chronic kidney disease	mental disorders
	chronic respiratory disease	musculoskeletal disorders
	diabetes mellitus	neoplasm
	digestive disease	neurological disorder
	hiv/aids	
20NewsGroup	religion	hardware
	space	gun
	medicine	**sakura**
	superhero	**himalayas**
	lyrics	**html**
Yelp	Seafood	burger
	sushi	chinese food
	dessert	**complained**
	vivid	**breast**
	own	**UK**
DBPedia	educational institution	artist
	athlete	company
	plant	village
	insulin	**jeans**
	refrigerator	**bartender**

Table 1.1: 3 Datasets from different areas with seeds selected with categorical seeds. The seeds not appearing in any of the documents are marked in red

they can learn the relationships between the seeds and other words. However, in most cases, this assumption can be too restrictive. For example, in the SciDoc dataset, the phrase “chronic respiratory disease” is one of the provided categories of the dataset, but it appears nowhere in the dataset. In this case, all previously mentioned models cannot find relevant words for that topic.

recent developments in pre-trained language models showed a path to address this issue by their extensive internal vocabulary, which is learnt from large datasets. Furthermore, they can update their internal representations of the words inside their own vocabulary as they encounter new documents, which allows these models to reflect the relationships of words in the dataset correctly.

We propose a new method in this paper called EnsembleTM. It uses an ensemble approach by combining a pre-trained language model, namely BERT, with CaTE. This method takes advantage of BERT's large vocabulary to handle out-of-vocabulary words and derives seeds for CaTE. And also leverages CaTE's ability to better learn the local context as well as phrases.

Chapter 2

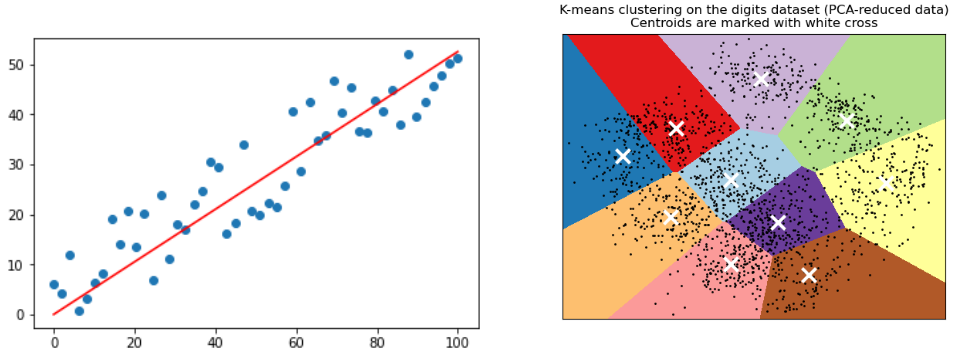
Background

2.1 Types of Machine Learning Methods

Machine learning models can be classified into two major categories by the training data, which are supervised learning, unsupervised[Bar89] learning and semi-supervised learning.

Supervised learning methods can make predictions on unseen input data. To achieve this, these methods must see both input values and corresponding output values or ground truth during the training process. So they can link the inputs with the outputs and learn how they are related. Linear regression is a classical supervised learning method. As shown in Figure 2.1a, The red line is the model learnt from the blue data points, which tries to find the line of best fit from the training data. The x coordinate is the input value, and the y coordinate is the output value. This red line shows the basic 2D linear regression model $y = w_1x + w_2$. In this example, w_1 is approximately 0.5, and w_2 is approximately 0.

Unsupervised learning methods, on the other hand, do not require users to provide ground truth for input data during training. These models will emphasise the connections between the existing data points rather than predicting results for unseen data.



(a) Linear Regression

(b) k-Mean Clustering

Figure 2.1

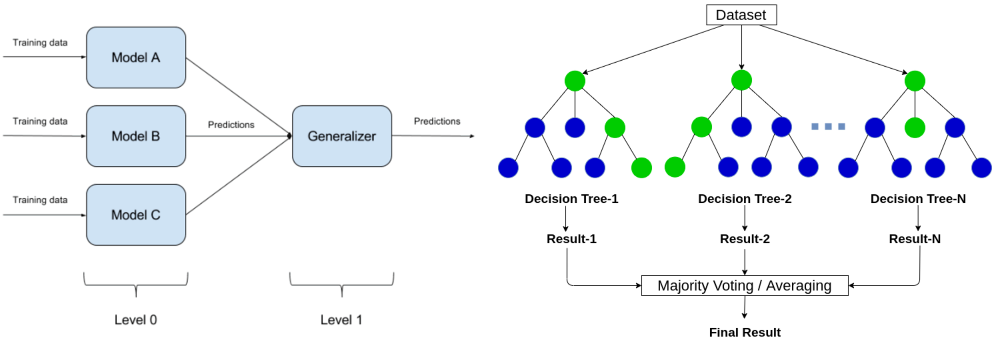
Hence these models are often used for tasks like clustering similar data points or (samples) together. Figure 2.1b illustrates the idea of k-means clustering, a typical example of unsupervised learning. In the figure, the data are clustered into ten groups ($k = 10$). Each group has a mean or "centroid", such that for each data point in this cluster, the distance to the centroid is minimised compared to other points.

In the middle ground of these two methods, there is a new category called semi-supervised learning, taking both partially labelled data as training inputs. During training, the model can utilise the provided labelled data to improve the accuracy of the underlying unsupervised methods. Examples will be explained in later sections.

2.2 Ensemble Learning

Ensemble learning [Die00] is a technique that aggregates the results of multiple models together as the final result, which can be interpreted as "crowd wisdom". Figure 2.2a exemplifies the general idea of an ensemble model that combines 3 models. A more solid example of ensemble models is a random forest consisting of several decision trees. Suppose there are 3 trees in the forest, predicting for the same inputs. Two produced the result of "true", and one predicted "false". So if we aggregate the results simply by majority voting, then the result will be "true".

The biggest advantage of this method is that the accuracy is improved by combining the results from different models.



(a) Ensemble Structure (b) Random Forest

Figure 2.2

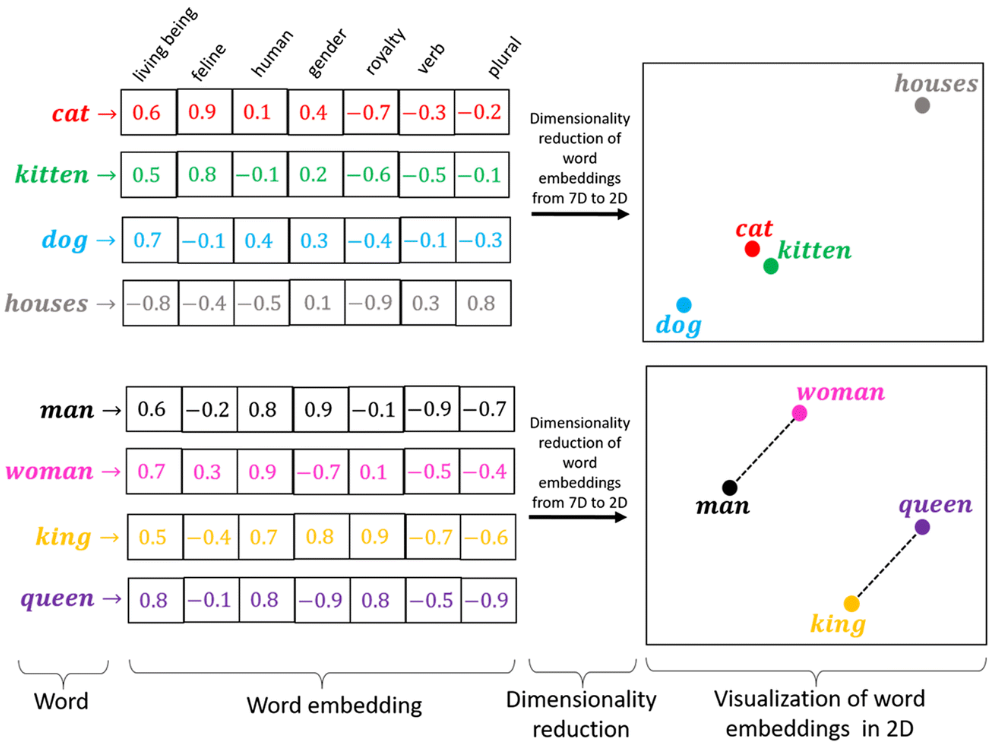
2.3 Word Embeddings

Word embeddings are vectors representing words. They are often derived from the given dataset. The word embedding reflects the meaning of the word in the particular dataset, which implies that for the same word in a different dataset, the embedding might be different. In Figure 2.3, The word “kitten” and “cat” are very close to each other as they have similar means.

Word embeddings are crucial to this project as they enable us to calculate the similarities of the words.

2.4 BERT

Bidirectional Encoder Representations from Transformers (BERT)[Dev+18] is a transformer-based neural network that revolutionises the area of Natural Language Processing (NLP), as the model can be adapted for various tasks and achieves state-of-the-art results in many areas by adding an output layer. As its name suggests, BERT consists of multiple encoders, which can benefit from the significantly improved training speed



	living being	feline	human	gender	royalty	verb	plural
cat →	0.6	0.9	0.1	0.4	−0.7	−0.3	−0.2
kitten →	0.5	0.8	−0.1	0.2	−0.6	−0.5	−0.1
dog →	0.7	−0.1	0.4	0.3	−0.4	−0.1	−0.3
houses →	−0.8	−0.4	−0.5	0.1	−0.9	0.3	0.8
man →	0.6	−0.2	0.8	0.9	−0.1	−0.9	−0.7
woman →	0.7	0.3	0.9	−0.7	0.1	−0.5	−0.4
king →	0.5	−0.4	0.7	0.8	0.9	−0.7	−0.6
queen →	0.8	−0.1	0.8	−0.9	0.8	−0.5	−0.9

Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

Figure 2.3: Word2Vec embeddings

and accuracy of the underlying encoders. The process of training a pre-trained model with new datasets is referred as fine-tuning.

2.4.1 Transformers

Transformers[Vas+17] are the fundamental building blocks of BERT. It consists of two parts, which are an encoder network and a decoder network. The encoder network transforms the input sentence into a sequence of vectors, while each vector is a numerical representation of the corresponding word, known as embeddings. The positional encoding algorithm will then integrates the relative position of the word in the vector. Then the embeddings will be sent to the multi-head attention layer, which works like human brains, allowing the model to focus on what is important by assigning each word a weight to indicate its importance. The feed-forward layer will reshape the output from the attention layer to match the required dimensions and produce contextualised word embeddings for each word in the sequence.

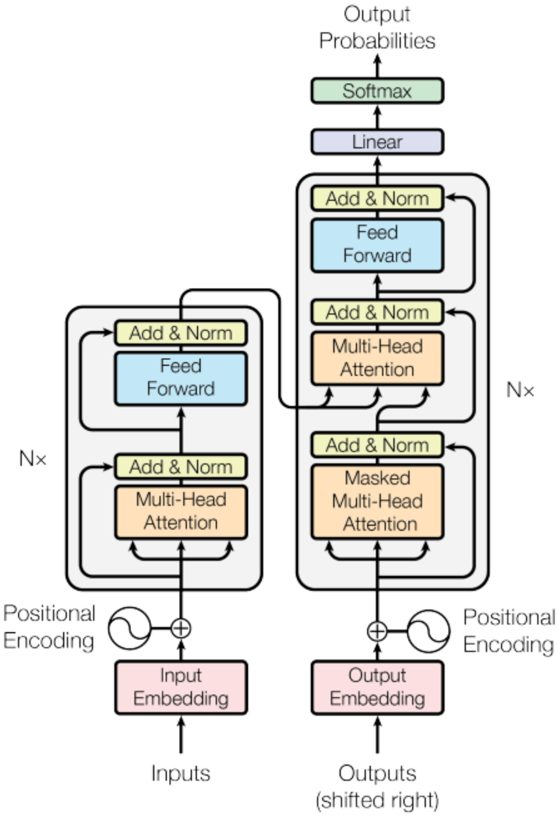


Figure 2.4: Architecture of transformer network. The left part is the encoder network. The right is the decoder network

The decoder network has a very similar architecture to the encoder network, with only one additional attention layer. It learns how to predict the results for the encoded input by taking the ground truth as input.

2.4.2 Tokenizer

The tokenizer is an algorithm that maps words to numbers, allowing BERT to look up the corresponding embeddings and update them during training. If the input is a sentence, it will also break it into a sequence of words. In particular, BERT uses a WordPiece tokenizer, which breaks phrases or complex words into sub-words. For instance, The word '*childcare*', will be tokenized to '[CLS]', '*child*', '*###care*' and '[SEP]'. The '[CLS]' and '[SEP]' represents the beginning and end of a sequence, respectively. The hashes in front of the word ‘care’ indicate it is a suffix of a word or phrase,

Chapter 3

Problem Statement & Assumptions

Given a collection of documents $\mathcal{D} = \left\{d_1, d_2, \ldots, d_{|\mathcal{D}|}\right\}$ and a set of user-provided topics $\mathcal{C} = \left\{c_1, c_2, \ldots, c_{|\mathcal{D}|}\right\}$. For each topic c_i, find a set of word $S_i = \{s_{i1}, s_{i2}, \ldots, s_{iN}\} \subseteq \mathcal{V}_{\mathcal{D}}$, where $\mathcal{V}_{\mathcal{D}}$ is is the vocabulary of documents. For all s in S_i, it should be semantically close to c_i. Additionally, s_i and c_i can be either phrases or single words.

This method assumes all the topics that are not in the document vocabulary $\mathcal{V}_D$ are in the BERT's vocabulary (V_T)

Chapter 4

Methodology

The EnsembleTM combines the output embeddings of two models. As shown in Figure 4.1, The top branch is the global knowledge model, which is based on BERT. The bottom branch is the local knowledge model that is built on top of CaTE.

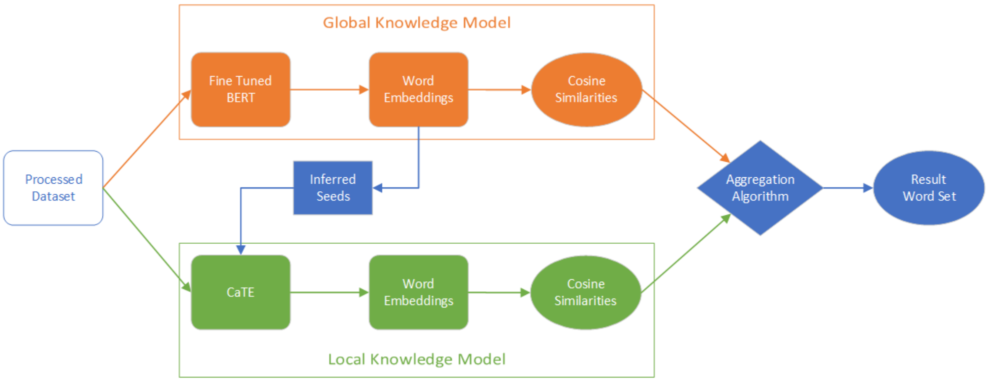


Figure 4.1

4.1 Global Knowledge Model

Global knowledge model is the key to handle out-of-vocabulary seeds by leveraging the power of Pretrained Language Models (PLMs), namely BERT. Its embeddings reflect the global context as it has already learnt the information from external datasets. We will use the embeddings to derive seeds for local knowledge models and compute a similarity score.

4.1.1 The Power of Pretrained Language Models

Recently, the PLMs have dominated the field of NLP as they have demonstrated their superior performance in many tasks, such as predicting the next sentence or filling in the missing word in a sentence. Compared to traditional methods, PLMs have several advantages.

Large-scale Training Dataset. PLMs are trained with large datasets. For example, BERT is trained on BookCorpus (approximately 800 million words) and English Wikipedia with only text passages (approximately 2500 million words).[Dev+18] This allows them to properly learn the meanings of the out-of-vocabulary words and relate them to in-vocabulary words.

Uniform presentation for polysemous words BERT can represent polysemous words (single words have multiple meanings) and synonyms with a single embedding vector, which allows us to handle polysemous words without considering them explicitly. For instance, the word '*order*' can either refer to a 'sequence' or 'buy something. As shown in Table 4.1, the word 'purchase' and 'sequence' have close scores, which implies they are equally close to '*order*', while the word 'technology' is less related.

Word	Score
purchase	0.75036
sequence	0.75749
technology	0.52605

Table 4.1: The cosine similarities between 'order' and other words

Large Vocabulary Space In the assumption, we explicitly state that all the seeds that do not appear in the documents must be in the BERT's vocabulary. In other words, all the seeds must be in $V_T \cup V_D$ as shown in Figure 4.2. The reason is that we can utilise the pre-trained knowledge of BERT to fine-tune BERT, so we can obtain an embedding that reflects the context of the dataset. This assumption is very easy to meet as pre-trained BRET and its tokenizer comes with a vocabulary (V_T) of approximately 30000 words, which is large enough to cover most common words in English.

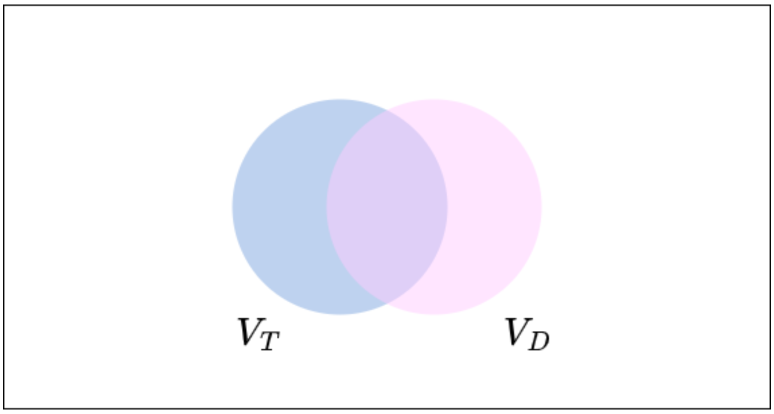


Figure 4.2: Caption

4.1.2 Extending BERT vocabulary to handle phrases

BERT's WordPiece tokenizer will break phrases into words and split words into subwords. This makes us difficult to represent phrases in the user-provided seeds as computing similarities for two words requires their embeddings to have the same dimension. For instance, for the word "Chinese food", the tokenizer will break into "Chinese" and "food", which will have an embedding of length $768 \times 2 = 1536$. However, the embeddings for "dumpling" will only have a length of 768.

Our approach is to force the tokenizer not to split the phrases by adding the phrases into the tokenizer vocabulary and BERT's embedding layer. So when we are fine-tuning BERT with our datasets, the model can learn the embedding for that phrase.

Comparing with training a BERT and a tokenizer from scratch to learn the embeddings. This approach retains the knowledge already learnt and enables to model to relate new embeddings with existing ones.

4.1.3 Fine Tuning BERT

Fine-tuning is a technique that trains a pre-trained model for specific tasks on new datasets. For most tasks, fine-tuning will also extend PLMs with additional output layers, which is very similar to the decoder network. This enables PLMs to leverage the existing knowledge to learn the information in the new dataset and adapt to new tasks.

The global knowledge model is a fine-tuned BERT model for Masked Language Modelling (MLM). As the name suggests, during training, A word of random choice will be hidden. For example, the sentence "The way to get started is to quit talking and begin doing", will become "The way to get started is to quit talking and [MASK] doing" and the model needs to decide what word the missing word '[MASK]' is. This mimics the behaviours of human brains, allowing the model to fuse the left and the right context [Dev+18]

One limitation of CaTE, as mentioned above, is that it cannot handle out-of-vocabulary words. So embeddings from Fine-tuned BERT will be used to derive seeds. We first compute the cosine similarities for all the in-vocabulary words against the topic c_i. Then we use the word with the highest score as the derived seeds for CaTE.

Algorithm 1 Deriving seeds for out-of-vocabulary words using BERT

```
for all $c_i \in \mathcal{C} \mid c_i \notin E_{\mathcal{D}}$ do
    $S_i \leftarrow$ similarities($c_i, \mathcal{V}_{\mathcal{D}}$)        ▷ $S_i$ is an array of cosine similarities
    $c'_i \leftarrow$ sort_descending($S_i$)[0]
    $\mathcal{C}[i] \leftarrow c'_i$
end for
```

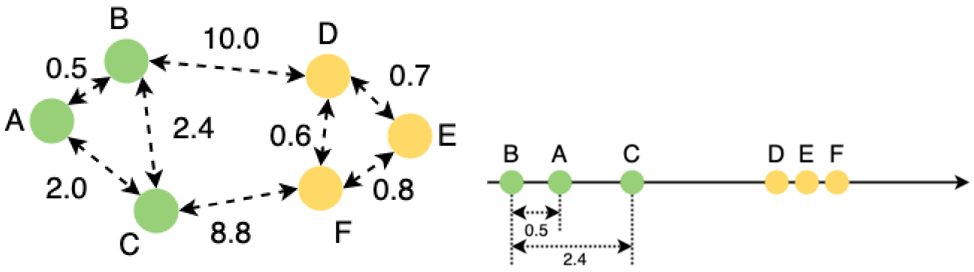
4.1.4 Rank Reduction

Rank reduction is a technique that maps high-dimension vectors or matrices to low-dimension vectors/matrices. This is crucial to our method as rank reduction helps to reduce noise. This is especially useful because BERT's embeddings have a dimension size of 768, which contains a lot of redundant information.

Though BERT can consider the local context in the document. However, the training process is unsupervised, which means it will not bias towards any seeds during training. We use Uniform Manifold Approximation and Projection (UMAP) [MHM18] to emphasise the existence of the seeds, not only for its performance but also because it also supports semi-supervised learning, which allows us to put more emphasis on seeds for BERT embeddings.

The semi-supervised UMAP is a rank reduction tool allowing us to provide partially labelled data as guidance. If one embedding is closely related to a specific seed word (i.e. above the cosine similarity threshold) we will assign the seed topic to the embedding. We then pass the partially labelled embeddings into UMAP to perform rank reduction, allowing the embeddings to bias towards the seeds.

The rank reduction process of UMAP can be considered as two parts. The first part is to initialise a low-dimensional graph. It will first compute distances between points and plot them to that low-dimensional graph based on the distances computed, as shown in Figure 4.3.



(a) Distance between points

(b) Initial low-dimensional Mapping

Figure 4.3: Example of UMAP (step1) mapping 2D points to 1D number line

The second part is to move points in the low-dimensional space to reflect their original positions relative to their neighbours. To do this, UMAP requires users to specify the number of neighbours($|N|$) and then calculate the similarity scores for each pair of points in the cluster, such that for any point p_i in the cluster,

$$\sum_{p_j \in N, i \neq j} \text{similarity}(p_i, p_j) = \log(|N|)$$

However, the similarity score is asymmetric, which means similarity$(p_i, p_j) \neq$ similarity(p_j, p_i). The algorithm will then convert them to symmetric similarity scores. After that, it will pick a pair of points based on similarities and a direction to move. (e.g. A → B). The pairs with higher similarities are more likely to be chosen. It will also pick a point in other clusters as the point to move away from (e.g. D). Once the direction is decided, UMAP will compute the magnitude of the movement by maximising similarity'(A, B) and minimising similarity'(D, B).

4.2 Local Knowledge Model

The local knowledge model has only a single responsibility, which is learning the embeddings of corpus vocabulary from training data. To accomplish this, we use a method called Category-name-guided Text Embedding (CaTE).

4.2.1 Category-name-guided Text Embedding (CaTE)

CaTE is a semi-supervised learning method that can learn word embeddings with consideration of the word's surrounding context and the word. The model contains two parts. One is the "category-name guided embeddings module". The other one is a "category representative word retrieval module".

The "category-name guided embedding module" uses a generative model to learn the word embeddings as shown in Equation 4.1. The model contains 3 parts, $p1$ models how each document relates to the seed words, which assumes each document associates only with one topic. The second part $p2$ reflects the word distribution within the document (i.e. the global context). The third part model describes the relationships between the word w_i and its surrounding words (i.e. the local context).

$$P(\mathcal{D} \mid \mathcal{C}) = \underbrace{\prod_{d\in\mathcal{D}} p(d \mid c_d)}_{p1} \underbrace{\prod_{w_i\in d} p(w_i \mid d)}_{p2} \overbrace{\prod_{\substack{w_{i+j}\in d\\ -h\leq j\leq h, j\neq 0}} p\left(w_{i+j} \mid w_i\right)}^{p3} \tag{4.1}$$

The second part, “category representative word retrieval module”, selects words based on the embeddings trained from the previous iteration and “distributional specificities”. It is adapted from distributional generality [WWM04], and distributional inclusion hypothesis [GD05], which measures how specific the meaning of a word is. For instance, as shown in Figure 5.2b, the word “seafood” has a higher distributional specificity score than “food” as seafood is a kind of food. The distributional specificity acts as a constraint on the results such that chosen words should be more specific than the seed words. Finally, for all the words that meet such constraint, it will prefer the words with **1) higher cosine similarity score** indicating the word is more related to the seed word and **2) lower specificity**, which means the word has wider semantic coverage.

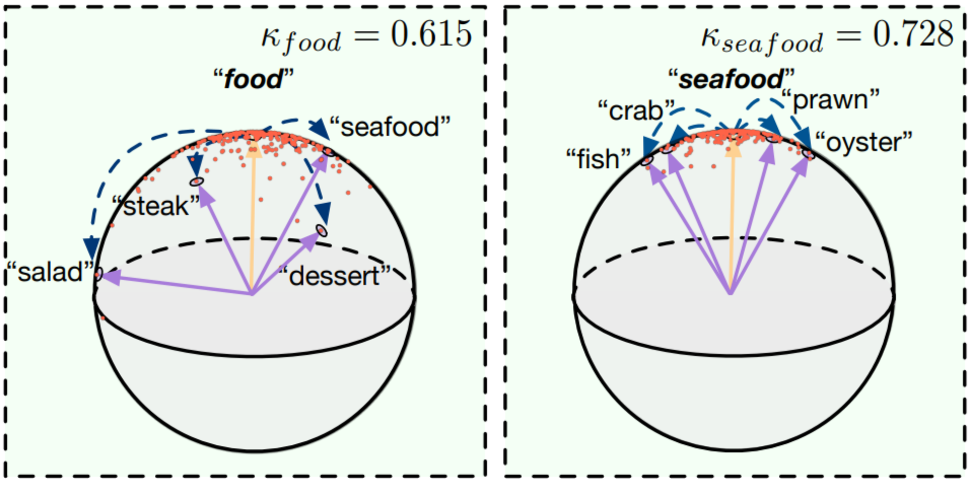


Figure 4.4

Our method only uses the first part as we need only embeddings, while the second part picks the representative words for each topic. We choose the model to reflect local knowledge for several reasons. **1) It has no prior knowledge**. CaTE is trained only with the given dataset, which means the embeddings will only reflect the local context without compensating for information from other sources. **2) It can handle phrases**

better. CaTE does not apply any tokenization before training, which implies it will treat the phrase as a single word by default.

4.3 Result Aggregation

This part combines the cosine similarities derived from both models with the following formula

$$\operatorname{score}\left(w \mid \mathcal{S}_i\right)=\left(\left(W_G \frac{1}{\operatorname{score}_G(w)}\right)^\rho+\left(W_L \frac{1}{\operatorname{score}_L(w)}\right)^\rho\right)^{1 / \rho} \tag{4.2}$$

W_L, W_G and ρ are user-controlled variables. W_G and W_L are the weights that control how much the global and local knowledge models will impact the final results. while ρ defines how the results of two models will be combined. If $\rho = 1$, then the weighted scores will just be added together. However, as $\rho \to 0$, the product of scores will become the square root of the reciprocal product.

4.4 Implementation Limitations

First, some words are in the training dataset but not in the BERT's embeddings, such as some professional terms in some scientific works. We are not adding them in as they may potentially pollute the tokenizer's vocabulary, as the tokenizer may potentially break the existing words into sub-words. For example, it will break apple into 'a', '###ppl', '###e'.

Secondly, the semi-supervised UMAP assumes a word can only be associated with one topic, which is not true in real life. Some words may relate to multiple topics. For example, the word apple can be either related to "pear" as a fruit or "hardware" as a tech company.

Chapter 5

Evaluation

We applied our method to 4 different datasets and compared them against a few existing methods.

5.1 Experiment Setup

5.1.1 Seed Selection

All the datasets we use come with some labels. However, for some datasets, all the provided labels are in-vocabulary, which means we need to find the out-of-vocabulary seeds from the tokenizer vocabulary.

To do this, We first do an automatic sampling by extracting the embeddings E_O in the BERT's tokenizer vocabulary but not in the dataset's vocabulary. Then we performed K-means clustering on the extracted embeddings and randomly pick one word from each cluster. This process ensures the minimum difference between seeds. However, the algorithm is still likely to pick similar words as seeds, so we will then manually examine the results and pick from the selected seeds to guarantee each seed are distinct enough.

5.1.2 Dataset

The statistics of each dataset are shown in Table 5.1, and the seeds are in table 1.1

- **SciDoc**[Coh+20]: A large collection of scientific papers for various tasks. We are using one of the subsets in the dataset, MeSH, which contains approximately 23K medical papers, each belonging to one of the 11 research areas. We concatenate the title and abstract of each paper as one entry. And the 11 research areas are used as seeds. All the seeds are phrases, and the out-of-vocabulary seeds are not in BERT's vocabulary. So we want to use this dataset to demonstrate the limitation of our method.

- **Yelp**[] Collection of user reviews of different restaurants worldwide. We sampled a subset of English reviews (29K) as the dataset. Since all the provided topics are in the vocabulary, we sampled 6 in-vocabulary seeds and 4 out-of-vocabulary seeds using the process described in section 5.1.1.

- **20NewsGroup**[Mit]: This dataset contains a collection of email communications (18K entries), each falling under one of the 20 provided topics. However, similar to the Yelp dataset, the given topics are all in vocabulary. So we use the same sample algorithm as the Yelp dataset to pick 6 vocabulary seeds, and 4 out-of-vocabulary with the same algorithm we used for Yelp dataset

- **DBPedia**[] The dataset selects 14 non-overlapping topics from the DBPedia 2014 dataset. We sampled 2500 entries from each category as the original dataset is too large. Also, since we cannot find any out-of-vocabulary seeds that meet the requirement from the provided seeds, we used the seed selection algorithm described in section 5.1.1 to sample 4 out-of-vocabulary seeds. Combining with the 6 in-vocabulary seeds we used, we selected 10 seeds from the dataset.

	# Words	# Phrases	Total
SciDoc	2,976,386	252,969	3,229,355
Yelp	1,496,068	31,549	1,527,617
20NewsGroup	2,600,061	2,741	2,602,802
DBPedia	745,128	72,173	817,301

Table 5.1: Statistics of each dataset

5.1.3 Evaluation Metrics

- **Pointwise Mutual Information (PMI)** PMI [New+10] measures how closely each word is related to other words in the same group using the co-occurrence of two different words.

$$\text{PMI} = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \sum_{w_j, w_k \in \mathcal{S}_i} \log \frac{P\left(w_j, w_k\right)}{P\left(w_j\right) P\left(w_k\right)}$$

  where $P(w_j, w_k)$ is the likelihood that w_j and w_k appear in the same document, which is the number of times both appear in the same document divided by the number of words. $P(w_i)$ is the likelihood w_i appear in the dataset, computed by the counts of w_i divided by the total number of words in the dataset.

$$P(w_j, w_k) = \frac{\#\,\text{co-occurence}(w_j, w_k)}{\#\text{words in dataset}}$$
$$P(w_i) = \frac{\#\text{occurrence of } w_i}{\#\text{words in dataset}}$$

- **Normalised Pointwise Mutual Information (NPMI)** NMPI stands for normalised PMI, which addresses the issue that PMI does not impose an upper bound[Bou09]

$$\text{NPMI} = \frac{1}{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{C}|} \sum_{w_j, w_k \in \mathcal{S}_i} \frac{\log \frac{P(w_j, w_k)}{P(w_j) P(w_k)}}{-\log P\left(w_j, w_k\right)}$$

- **Distinctiveness** Distinctiveness measures how the words in each word set differ from the others by taking the length of the union of all the words in the word set and dividing it by the sum of the length of all the words.

$$\text{Distinctiveness } = \frac{\left|\bigcup_{i=1}^{|\mathcal{C}|} \mathcal{S}_i\right|}{\sum_{i=1}^{|\mathcal{C}|} |\mathcal{S}_i|}$$

5.1.4 Hyper-parameter Settings

Hyper-parameters are user-controlled variables for configuring the model. We use the following settings when evaluating our method.

- **BERT** We fine-tuned the `bert-base-uncased` model implementation in the python package `transformers` with its default configuration. That is, embedding dimensionality (`hidden_size`) = 768, number of hidden layers = 12, number of attention heads = 12, feed-forward layer dimensionality = 3072. We extended the vocabulary of BERT to learn phrases, as described in Section 4.1.2, so the vocabulary is larger than 30522. WE fine-tuned BERT for 10 epochs. (i.e. feed the entire datasets into the model for 10 iterations)

- **UMAP** The two key hyper-parameter for UMAP are the number of neighbours (`num_neighbours`) and reduced dimensionality (`n_compnents`). By default, we set `num_neighbous` = 15 and `n_compnents` = 80.

- **CaTE** For CaTE, we use the original C implementation provided by the paper authors. We followed the default hyper-parameter settings in the provided runner script, except we set the number of pre-training iteration (`-pretrain`) to 3

5.1.5 Methods Compared

- **SeededLDA**: Improved version of LDA, it integrates seeds by discriminating topics towards seeds and makes the documents favour the users provided topics.

- **CaTE**: A seed guided topic model explained in Section 4.2.1

- **pretrained BERT**: A neural network explained in section 4.1.1. Since it cannot handle phrases, we will not evaluate with SciDoc.

- **Finetuned BERT**: BERT continued training on the existing dataset.

5.2 Analysis of Results

Methods	20NewsGroup		Yelp		SciDoc		DBPedia	
	PMI	NPMI	PMI	NPMI	PMI	NPMI	PMI	NPMI
SeededLDA	261.34	52.05	196.53	38.77	228.27	34.50	166.50	22.49
CaTE	194.44	17.77	103.50	8.85	55.10	4.52	56.86	4.75
Pretrained BERT	195.67	17.27	70.59	6.60	-	-	59.40	5.08
Finetuned BERT	126.73	12.80	98.20	9.05	41.88	3.31	71.84	6.59
EnsembleTM	198.02	29.68	128.70	12.26	94.85	7.69	93.82	8.51
EnsembleTM (With Reduction)	179.31	18.34	123.81	11.63	96.03	7.74	87.91	7.99

Table 5.2: Evaluation results of different methods, higher is better

Table A.1 shows the words selected by our method while table 5.2 quantitatively measures the performance of our methods and other comparing methods. SeededLDA is considered an outlier as PMI only reflects the coherence between words selected for each topic but does not measure the relevance between the chosen words and the topic as the topic may not be in the vocabulary. As shown in table 5.4 SeededLDA tends to choose the words frequently , which will yield a high probability of co-occurrence (i.e. high $P(w_j, w_k)$) and hence higher PMI and NPMI. Additionally, SeededLDA tend to repeat the same words for different topics, which will result in a low distinctiveness score, as shown in Table 5.3.

Methods DBPedia Distinctiveness	20NewsGroup	Yelp	SciDoc
	Distinctiveness	Distinctiveness	Distinctiveness
SeededLDA 0.88	0.56	0.44	0.71
CaTE 1.0	1.00	1.00	1.00
Pretrained BERT 0.98	0.96	0.98	-
Finetuned BERT 0.90	0.96	0.95	0.97
EnsembleTM 1.0	1.00	1.00	1.00

Table 5.3: Distinctiveness of different methods

educational institution	album, release, film, first, bear
artist	specie, family, genus, find, plant
athlete	bear, school, play, high school, university
company	company, build, service, class, name
plant	historic, locate, building, build, house
village	village, river, district, also, population

Table 5.4: Results of Seeded LDA

Our EnsembleTM actually has the best performance among all these methods thanks to the ensemble learning approach, which allows it to utilise not only the dataset-specific context provided by CaTE but also the background knowledge provided by BERT as complements to produce final results.

5.3 In-vocabulary Results vs. Out-of-vocabulary Results

Figure 5.1 breaks down overall PMI into an in-vocabulary part and an out-of-vocabulary part. We can see there is a significant improvement in in-vocabulary results. Additionally, we added phrases such as unique tokens to BERT's vocabulary. It can handle phrases properly, as shown in table A.1.

However, the results for out-of-vocabulary seeds are not ideal. It is worth mentioning that the out-of-vocabulary seeds for SciDoc break the assumption as they are not in BERT's vocabulary, which means BERT does not have any prior knowledge of this word and has no way to learn the embeddings for those seed words. Hence BERT cannot relate them with in-vocabulary words, leading to generating wrong seeds for CaTE. For the rest of the datasets, the out-of-vocabulary PMI shows our methods can get

Seeds	Selected Words				
Complained (Markus)	Markus	Der	Bahn	Karl	Nello
Vivid (intricate)	richly	illusion	weave	coloured	layered
Own (Stevens)	Stevens	Tucker	Leary	Fulton	Anderson
UK (British)	British	Australia	Australian	Malaya	pub

Table 5.5: Selected Words for out-of-vocabulary seeds in the Yelp dataset, the words in brackets are the words inferred seeds

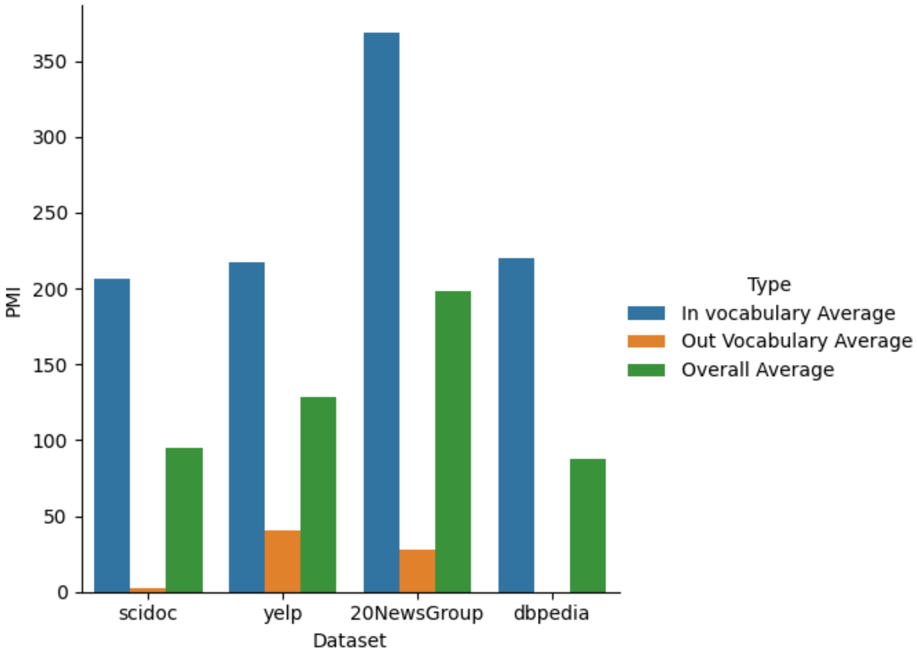


Figure 5.1: In-vocabulary PMI Average vs Out-of-vocabulary PMI Average

some of the seeds correctly for most of datasets. The key issue here is that fine-tuned BERT cannot relate out-of-vocabulary seed and in-vocabulary words correctly. Taking the results on the Yelp dataset as an example, In table 5.5, we can see that BERT got two out-of-vocabulary seeds correctly (vivid and UK) but related the other two seeds with names. The primary cause is that neural networks like BERT will struggle to understand rare words, as demonstrated by Schick et al.[SS20].

For pre-trained BERT, the word “complaint” is more frequent in its training dataset and hence more likely to relate with similar words as shown in Table 5.6. However, for the fine-tuned BERT, “complained” is nowhere in the dataset, which made fine-tuned BERT place it in the category of infrequent words. For instance, the word “Markus” only appears 3 times in the dataset and “thereof” shows up 6 times.

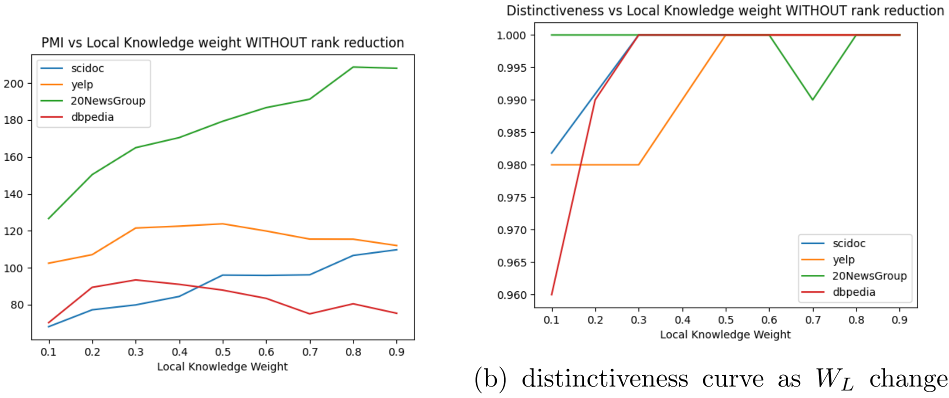
Additionally, our method did not get the out-of-vocabulary words at all. Despite the above-mentioned issue, the seeds are irrelevant to our datasets.

Selected Words(Score)	
Pretrained BERT	**Fine-tuned BERT**
complaint(0.641)	Markus(0.990)
notification(0.607)	thereof(0.990)
commet(0.589)	bodied(0.989)
reply(0.584)	fractured(0.989)
complainer(0.577)	Stevens(0.989)

Table 5.6: Top-5 similar words for the out-of-vocabulary seed "complained"

5.4 Parametric Studies

the 3 user-controlled parameters W_L, W_G, and ρ in the similarity aggregation formula will have a direct impact on the final result. W_L and W_G represent the weights for local knowledge model (CaTE) and global knowledge model (BERT) respectively. ρ controls the behaviour of the aggregation formula.

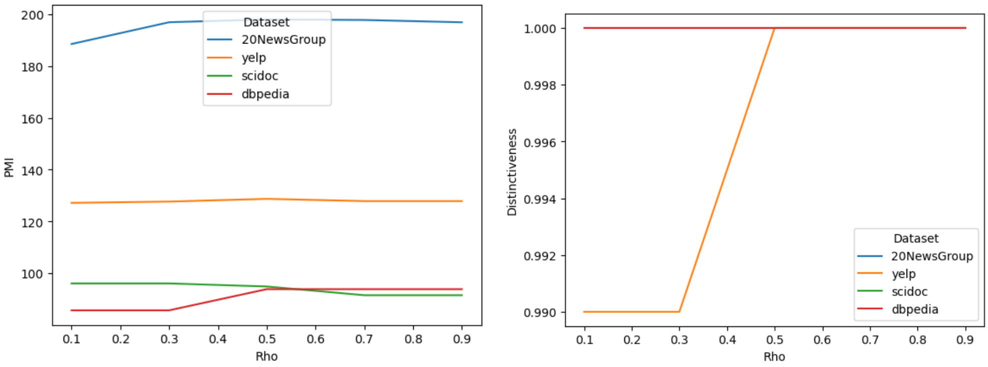


(a) PMI curve as W_L changes when $\rho = 0.5$

(b) distinctiveness curve as W_L changes, $\rho = 0.5$

Figure 5.2

From the figures, we can see that there is an optimal combination for W_L and W_G that achieves the highest PMI score for each dataset. If the W_L is too low, there is enough emphasis on the local knowledge. However, if W_L is too high, the model will overfit as we put too much stress on the local knowledge model. The only exception here is the SciDoc dataset. The PMI keeps increasing as W_L grows. This is expected as the out-of-vocabulary seeds broke our assumption. They are neither in the dataset nor in the BERT's vocabulary. This made the score only relies on CaTE's performance, and shown in table 5.2, CaTE outperforms fine-tuned BERT, so as the weight grows, the model keeps improving.



(a) PMI curve as ρ changes

(b) distinctiveness curve as ρ changes

Figure 5.3: Changes in performance as ρ changes, when $W_L = W_G = 0.5$

5.5 Result of Rank Reduction

Rank reduction is used to emphasise the existence of user-provided seeds, especially for BERT. From our experiment results, the rank reduction did not improve our model's performance and may cause a performance loss in some cases. We noticed significant PMI drops for 20NewsGroups and DBPedia. One potential cause for this is that rank reduction is too concentrated. The remaining dimensions are not enough to hold the contextual information. For BERT, we are mapping the embeddings from a dimension of 768 to 80, which means it uses 1/10 of the original space to hold as much contextual data as possible. This may cause a loss of contextual information, especially for datasets like 20NewsGroup containing a large number of topics.

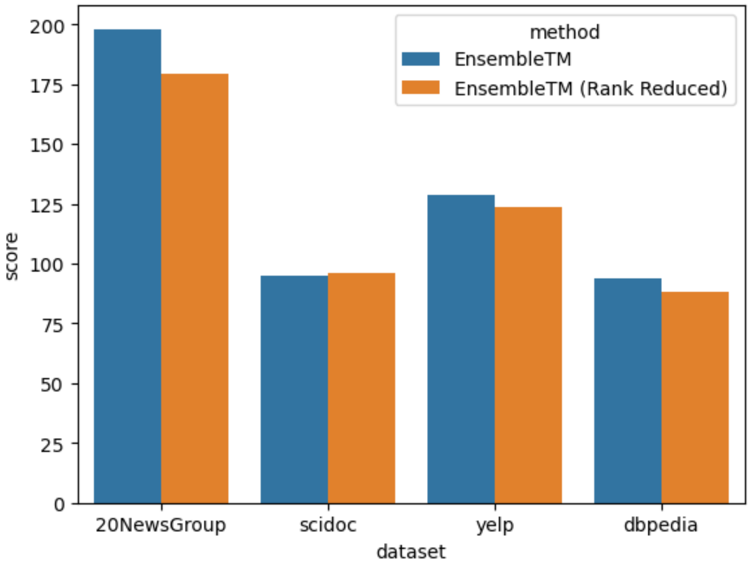


Figure 5.4: Comparison of PMI with rank reduction

Chapter 6

Related Works

6.1 Latent Semantic Analysis

Latent Semantic Analysis was proposed by Deerwester [Dee+90], which is a purely algebraic method. The method follows a 2-step process.

1. Obtain the occurrence matrix from the documents
2. Perform rank reduction with singular value decomposition (SVD) [For77]

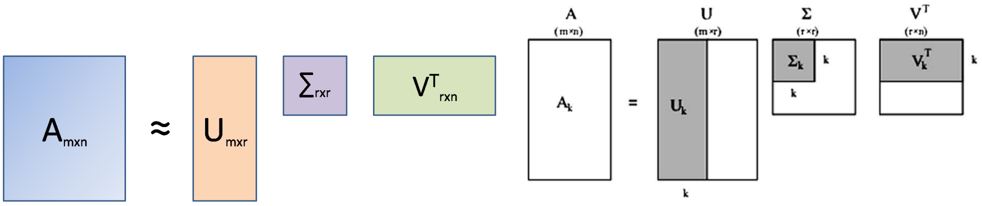
Obtain the occurrence matrix A from the documents. An occurrence matrix or document-term matrix is a matrix that counts the number of times a term appears in a document. Each row represents a word, and each column represents a document, as shown in Table 6.1. Suppose there are n documents and m terms, then A has a size of $m \times n$

Rank Reduction with SVD. Once the occurrence matrix A is computed, SVD is performed on A, It will produce three matrices as shown in Figure 6.1a. Matrix u is the relationship between word and contexts or topics (word-context matrix). $\sum$ is an orthogonal matrix representing the contexts and V^T is the document-context matrix. Since the topics in $\sum$ are sorted by scores, noises are reduced by only picking the first

	D_1	D_2
like	1	0
hate	0	1
apple	1	0
banana	0	1
i	1	1

Table 6.1: Occurrence matrix for two hypothetical documents D_1-"I like apple" D_2="I hate banana"

k topics [MB07], and the "reduced" matrices are multiplied back to "condense" the underlying information, as shown by the shaded areas of matrices in Figure 6.1b.



(a) Sigular Value decomposition

(b) Rank Reduction [MB07]

Figure 6.1

LSA is designed to handle synonyms by taking advantage of latent topics.[Dee+90]. However, it is not able to handle polysemies (multiple meanings for one word) well, as it just "averages" the meaning [Dee+90].

6.2 Probabilistic Latent Semantics Analysis - pLSA

Probabilistic Latent Semantics Analysis (pLSA)[Hof99] was proposed by T. Hoffmann in 1999. It re-models LSA with statistical representations by introducing latent topics into the model, which provides a firmer mathematical foundation. The model expresses the joint probability of words and documents by a mixture of conditional probabilities.

$$P(d, w) = \sum_{z \in \mathcal{Z}} P(z) P(d \mid z) P(w \mid z) \tag{6.1}$$

In equation 6.1, z is a hidden variable that represents topics. $P(d, w)$ represents the probability distribution of a word w appears in document d, which is similar to the document-term matrix A in LSA. $P(z)$ is the probability of topic z appearing in the corpus, which corresponds to LSA's orthogonal matrix $\sum$. $P(d|z)$ represents probability of document d belongs to a given topic z, corresponding to the matrix V^T in Figure 6.1a. Similarly, $P(w|z)$ is the probability of a word w belongs to a given topic z, which is like the matrix U in LSA.

Compared to LSA, the most significant improvement of pLSA is that it can handle polysemies as they are appropriately expressed with probabilities.

6.3 Latent Dirichlet Allocation - LDA

Latent Dirichlet Allocation (LDA) was proposed by David M. Blei in 2003 [BNJ03]. LDA is a generative model that produces documents based on several probability distributions learnt from the training data. For simplicity of discussion, each document in the corpus is assumed to have a fixed length of N.

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p\left(z_n \mid \theta\right) p\left(w_n \mid z_n, \beta\right) \tag{6.2}$$

The left side of the equation is the probability representing a document to be generated parameterised with words **w** and topics **z** and some other parameters that will be learnt from the training dataset. The right-hand side of the model represents the document generation process that can be decomposed as follows [BNJ03]:

1. Pick $\theta \sim \text{Dir}(\alpha)$
2. For each word w_i
    (a) Pick $\mathbf{z_i} \sim \text{Multinomial}(\theta)$
    (b) Pick word w_i from $p(w_i \mid z_i, \beta)$

In step 1, the topic distribution or document-topic model is modelled with a Dirichlet distribution. A k-topic Dirichlet distribution can represent a $(k-1)$-dimension simplex.

Its parameter α describes how the observations are distributed inside the simplex. In this case, the observations are the documents, while the vertices represent the topics. The distances between the vertices reflect the weights of each topic inside a document. The shorter the distance, the greater the weight. α is the corpus-level variable that describes how the documents are distributed inside the simplex that will be learnt from the training dataset. Figure 6.2 is an example that shows how different α values affect the distribution of the points [Bog].

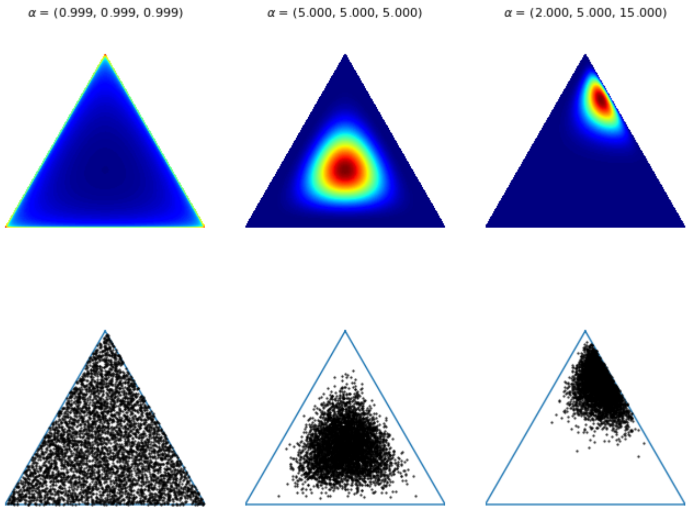


Figure 6.2

A vector θ of size k is picked from the Dirichlet distribution, which represents the topic distribution inside a generated document. Note that θ is a document-level variable that will change between documents.

In step 2a, the topics inside the document are modelled with a multinomial distribution parameterised by θ, a vector including the weights of each topic inside the multinomial distribution, which reflects the probability that each topic z will be chosen. In step 2b, the multinomial distribution $p(w_i \mid z_i, \beta)$ describes the distribution of words inside a document given a topic z_i. The parameter β is a matrix where β_{ij} represents a single probability for a word w_i belongs to the topic z_j (i.e., $p(w_i = 1 \mid z_j = 1)$).

Overall, the model can be considered a machine that generates documents with two dials. One dial is the Dirichlet distribution that generates topics. The other one is the multinomial distribution that produces words. Once the dials are tweaked, the machine can produce the documents the user wants.

Compared to pLSA, LDA is less likely to overfit since it is parameterised by two variables with a total of $k + kV$ parameters, which is significantly less than the number of parameters in LSA. Also, the model can handle unseen documents as the documents are treated as hidden random variables [BNJ03].

However, the topics generated by LDA cannot reflect user preference. For instance, the users may want words for “statistics”, but the model may pick words for “trigonometry”. Additionally, the model is not able to relate out-of-vocabulary with the word in the document.

6.4 Seeded Latent Dirichlet Allocation - SeededLDA

SeededLDA is a variant of LDA proposed by J. Jagarlamudi et al. in 2012 [JDU12]. As the name suggests, which can consider the user-provided seeds as shown in Table 6.2. The model still uses the same topic-document model and the word-document model but with the ability to bias towards seeds.

1 (tech)	Apple, Google, Facebook, iPad, iPhone
2 (agriculture)	grain, wheat, corn, harvest
3 (energy)	oil, petrol, gas, fuel

Table 6.2: Example of seed words

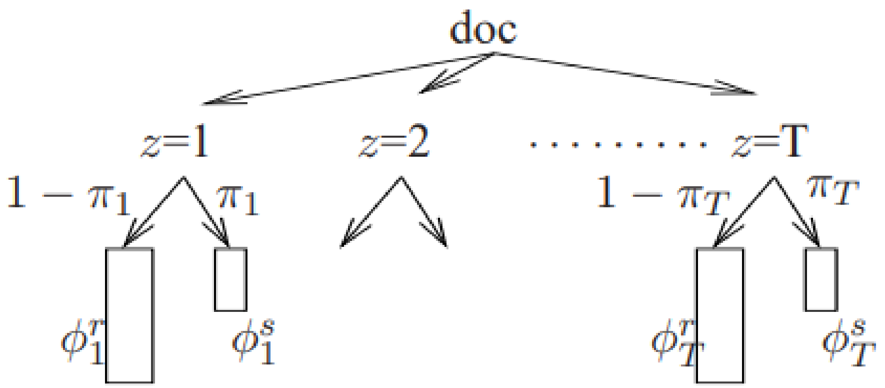


Figure 6.3: Topic mixture in a document as a tree [JDU12]

6.4.1 Topic Selection for Document-Topic Model

As shown in Figure 6.3, each topic z_i in a document contains a regular topic ϕ_i^r and a seed topic ϕ_i^s. The document-topic model will pick one topic by "flipping a coin", which is reflected in the following process.

1. For each topic z_i in the T topics:
   (a) Choose a $\pi_{z_i} \sim \text{Beta}(1, 1)$
2. For each document d, choose $\theta_d \sim \text{Dir}(\alpha)$
3. For each word in the document d
   (a) Choose a topic $z_i \sim \text{Multinomial}(\theta_d)$
   (b) Choose $x_i \sim \text{Bern}(\pi_{z_i})$
   (c) if $x_i = 0$, choose a regular seed word $w_i \sim p(w_i|z_i, \beta_r)$
   (d) if $x_i = 1$, choose a user-provided seed word $w_i \sim p(w_i|z_i, \beta_s)$

The process extends the original topic-document model with two extra steps. Step 1 forges a biased coin for π_{z_i} for each topic z_i and the coin is tossed in Step 3b. Then based on the result, it will pick either the seed topic or the regular topic.

6.4.2 Document Generation from Document-Topic Distribution

For simplicity, the seeds for each topic will be referred as “group”. The document-topic distribution generates words with the following process.

1. Choose topic weights $\psi_s \sim \text{Dir}(\alpha)$ for each group s in the S groups.
2. For each document d,
   (a) Compute a binary vector $\mathbf{b}$ of length S
   (b) Choose a group weights $\zeta_d \sim \text{Dir}(\tau\mathbf{b})$
   (c) Choose a group label $g \sim \text{Multinomial}(\zeta_d)$
   (d) Choose topic weights for the document $\theta_d \sim Dir(\psi_g)$
3. For each word in the document d
   (a) Choose a topic $z_i \sim \text{Multinomial}(\theta_d)$
   (b) Choose a word $w_i \sim p(w_i|z_i, \beta_r)$

The model first treats each seed set as documents and selects a topic weights for each set (step 1). The purpose of step 2 is to make the model biased toward the seed topics. For each document, The model first computes the presence of seed topics in the document (step 2a) represented by the vector $\mathbf{b}$. Consider the example ”A staff in an oil company bought an iPad”. Its $\mathbf{b} = \langle 1, 0, 1\rangle$, which indicates the presence of seed topics 1 and 3. Secondly, the model computes the weights for each seed topic in the document based on a Dirichlet distribution parameterised by $\mathbf{b}$. (step 2b) τ is a hyper-parameter set by users that reflect the confidence of the seeds belonging to the seed topic. Finally, a seed topic g is chosen based on the weights using a multinomial distribution (step 2c). And the regular topic weights will be generated based on the weights computed in step 1. (step 2d). Step 3 is the same as the original LDA model, so there is no need to discuss it again.

6.4.3 Combining previous steps

The final SeededLDA combines the two models discussed above. The improvement is noticeable. It allows the users to guide the model to choose the topics they want by providing seed words. However, it can still not handle the topics not in the vocabulary because the words are generated by a multinomial distribution based on the weights of words in the vocabulary. More importantly, the model assumes that each word is generated independently [Men+20]. Hence the contextual information between words is lost during the text generation process.

6.5 BERTopic

BERTopic[Gro22] is proposed by Maarten G in 2022. It is a clustering-based method that leverages the embeddings from pre-trained language models. It follows 2 steps, which are “clustering embeddings” and “representing topics”

6.5.1 Clustering embeddings

Before running the actual clustering method, the method will run a rank reduction algorithm, namely UMAP, to reduce the dimensionalities of the embeddings, but unlike what we did, they use an unsupervised rank reduction by feeding only the embeddings, as in higher dimensions, the distances between points are more indifferent[Bey+99; AHK01]. The rank reduction will make the distances more distinct and will improve the performance of clustering.

Once the rank reduction is done, the model will group the documents with HDBSCAN[MHA17]. HDBSCAN does the same thing as K-means clustering. The key difference is that HDBSCAN uses “soft clustering,” allowing noise in the data to be considered outliers.

6.5.2 Representing topics

Topics are modelled based on the document clusters generated in the previous step. A topic will be chosen for each cluster. the topic acts like a latent variable derived from the cluster rather than a "human-readable" word. The model will then choose words to represent the topic with 'cTF-IDF' score. the "cTF-IDF" took the idea of the original 'DF-IDF' score, which measures how important a word is for the document.

$$W_{t,d} = tf_{t,d} \times \log(\frac{N}{df_t}) \tag{6.3}$$

the formula of the original "TF-IDF" is shown in Equation 6.3 measures the importance of word t in document d. It takes into account the term frequency $tf_{t,d}$ (how many times the word appears in the document) and inverse document frequency (how much information it provides), which is the logarithm of the total number of documents N divided by the number of documents containing t

$$W_{t,c} = tf_{t,c} \times \log(1 + \frac{A}{tf_t}) \tag{6.4}$$

Similarly, the "cTF-IDF" measures how important the word t is to the cluster c. Its formula is shown in Equation 6.4 The cluster c is considered as a single document made by concatenating the documents in that cluster. The only difference is that the inverse document frequency is replaced by the "inverse topic frequency". A is the average number of words per cluster, tf_t is the sum of frequencies of term t in all clusters. To make the score always positive, 1 is added inside the logarithm.

6.5.3 Incorporating Seeds

Seeds are incorporated into the model with two steps. Firstly, the model will obtain the embeddings and compare seed topics with existing embeddings with cosine similarities. For each document, the algorithm will pick the most similar topic and assign it to the document. However, if the document is most similar to the average document embedding. It will remain unlabelled, then the partially labelled embeddings will be sent to UMAP for rank reduction, similar to what we did in EnsembleTM.

Then, for each topic, it will be assigned a factor f_t greater than 1, which will be used when computing ‘cTF-IDF‘. By introducing the factor, the formula becomes what is shown in Equation 6.5, which will stress

$$W_{t,c} = tf_{t,c} \times f_t \times \log(1 + \frac{A}{tf_t}) \tag{6.5}$$

the user-provided seeds more when choosing the representation words.

Finally, user is able to look up clusters based on seeds as they are sorted based on the cosine similarities.

Chapter 7

Conclusion

7.1 Our contribution

In this thesis, we evaluated our newly proposed method, EnsembleTM, designed to address the problem of out-of-vocabulary seeds. Our main contributions include the following:

- We have analysed the performance of our proposed method by testing it on different datasets and compared it against existing methods. The evaluation results show that our methods outperformed existing methods.
- We have tested the effect of rank reduction in our method. The results reflected that rank reduction is not necessary for our methods and may even negatively impact our method's performance in some cases.
- We examined our attempts on inferring seeds for CaTE to make it able to handle out-of-vocabulary seeds. The evaluation outcome indicates our attempt has achieved limited success for this, as we fine-tuned BERT to make it learn phrases, but it comes with a cost of losing correct embeddings for out-of-vocabulary words when compared against pre-trained BERT.

Chapter 8

Future Works

We noticed several issues and limitations in our method. Some of them are the design trade-offs we made during the implementation phase. The others are identified in the experiments as discussed above.

- **Addressing out-of-vocabulary seeds** Our attempt to infer out-of-vocabulary seeds did not achieve the result we expected because fine-tuned BERT loses the correct embeddings for out-of-vocabulary words. This is not the desired behaviour. We expect research on either new language models to solve this problem or on some data augmentation strategy to help existing language models reinforce learning outcomes.

- **Improve on learning phrases** Our proposed method learns the embeddings for phrases by overriding the rules in the tokenizer. This allows the tokenizer to treat the phrase as a new word and learn its embedding from the dataset, but discards the benefit of tokenization as an out-of-vocabulary phrase is very likely to contain an in-vocabulary word. We would like to see future works to find a way to combine embeddings of words in phrases.

- **Incorporating Seeds into BERT Embeddings** The fine-tuned BERT in the proposed method is completely unsupervised, which means the user-provided seeds are disregarded during training. We tried to emphasise those seeds through a semi-supervised rank reduction. But there is no positive effect on the results. In future works, we hope to seed methods to incorporate the seed into BERT.

- **Improving the ensemble aggregation method** Our aggregation formula combines weighted cosine similarity scores. As we mentioned before, the cosine similarities are not ideal for handling phrases as their length are not uniform. We are willing to see research works that directly combine the embeddings and then compute scores for ranking.

Bibliography

[] *DBPedia Dataset.* URL: http://wikidata.dbpedia.org/develop/datasets/latest-core-dataset-releases.

[] *Yelp Dataset.* URL: https://www.yelp.com/dataset/.

[AHK01] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. "On the surprising behavior of distance metrics in high dimensional space". In: *International conference on database theory.* Springer, 2001, pp. 420–434.

[Bar89] Horace B Barlow. "Unsupervised learning". In: *Neural computation* 1.3 (1989). Publisher: MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ..., pp. 295–311.

[Bey+99] Kevin Beyer et al. "When is "nearest neighbor" meaningful?" In: *International conference on database theory.* Springer, 1999, pp. 217–235.

[BNJ03] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3 (Jan 2003), pp. 993–1022.

[Bog] Thomas Boggs. *A script to generate contour plots of Dirichlet distributions.* URL: https://gist.github.com/tboggs/8778945 (visited on 04/22/2022).

[Bou09] Gerlof Bouma. "Normalized (pointwise) mutual information in collocation extraction". In: (2009).

[Coh+20] Arman Cohan et al. "SPECTER: Document-level Representation Learning using Citation-informed Transformers". In: *ACL.* 2020.

[Dee+90] Scott Deerwester et al. "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6 (1990). Publisher: Wiley Online Library, pp. 391–407. ISSN: 0002-8231.

[Dev+18] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[Die00] Thomas G Dietterich. "Ensemble methods in machine learning". In: *International workshop on multiple classifier systems.* Springer, 2000, pp. 1–15.

[For77] George Elmer Forsythe. “Computer methods for mathematical computations.” In: *Prentice-Hall series in automatic computation* 259 (1977). Publisher: Prentice-Hall, Inc.

[GD05] Maayan Geffet and Ido Dagan. “The distributional inclusion hypotheses and lexical entailment”. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. 2005, pp. 107–114.

[Gro22] Maarten Grootendorst. “BERTopic: Neural topic modeling with a class-based TF-IDF procedure”. In: *arXiv preprint arXiv:2203.05794* (2022).

[Hof99] Thomas Hofmann. “Probabilistic latent semantic indexing”. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999, pp. 50–57.

[JDU12] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. “Incorporating lexical priors into topic models”. In: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 2012, pp. 204–213.

[MB07] Dian I Martin and Michael W Berry. “Mathematical foundations behind latent semantic analysis”. In: *Handbook of latent semantic analysis* (2007). Publisher: Mahwah, NJ: Lawrence Erlbaum Associates, pp. 35–56.

[Men+20] Yu Meng et al. “Discriminative topic mining via category-name guided text embedding”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 2121–2132.

[MHA17] Leland McInnes, John Healy, and Steve Astels. “hdbscan: Hierarchical density based clustering.” In: *J. Open Source Softw.* 2.11 (2017), p. 205.

[MHM18] Leland McInnes, John Healy, and James Melville. “Umap: Uniform manifold approximation and projection for dimension reduction”. In: *arXiv preprint arXiv:1802.03426* (2018).

[Mit] Tom Mitchell. *20Newsgroups*. URL: `https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html`.

[New+10] David Newman et al. “Automatic evaluation of topic coherence”. In: *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. 2010, pp. 100–108.

[SS20] Timo Schick and Hinrich Schütze. “Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking”. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. Issue: 05. 2020, pp. 8766–8774. ISBN: 2374-3468.

[Vas+17] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[WWM04] Julie Weeds, David Weir, and Diana McCarthy. “Characterising measures of lexical distributional similarity”. In: *COLING 2004: Proceedings of the 20th international conference on Computational Linguistics*. 2004, pp. 1015–1021.

Appendix

A.0.1 Results of EnsembleTM

20NewsGroup	
religion	religion, christianity, religious, belief, denomination, religionist, religiously, christians, judaism, islam
hardware	hardware, pclk, implementation, software, compatibility, processor, builtin, thrue, pcs, ethernet
space	space, spacehab, spacelink, spaceflight, spacelab, shuttle, orbit, orbital, spacewalk, spacecraft
gun	gun, weapon, firearm, handgun, pistol, shotgun, knife, rifle, gunfight, revolver
medicine	medicine, medical, physician, doctor, healthcare, health, clinical, medication, diagnosis, therapy
sakura	sanskrit, vedic, sufi, asha, aramaic, hindus, mandir, syriac, krishna, sakura
superhero	mythical, gods, humanoid, horrific, monstrous, demonic, telling, enthusiastically, daring, superhero
himalayas	rbis, innings, rushing, batting, catching, padres, marlins, postseason, royals, himalayas
lyrics	forested, expanse, mountainous, canopy, southward, countryside, arid, northernmost, flowing, lyrics
html	cmap compact, prompting, current dir, w id, uubuild mode,vinfo list, max x, window title, n control, html
Yelp	
seafood	seafood, oyster, shrimp, crab, lobster, fisherman, fish, clam chowder, king crab, octopus
burger	burger, burgerfi, burgers, hamburger, fry, cheeseburger, patty, burger joint, bun, steakburger
sushi	sushi, sushimon, sushiholic, sushiya, ayce sushi, miso soup, roll, bento box, spicy tuna, spicy tuna roll
chinese food	chinese food, chinese, dim sum, guangzhou, noodle soup, roast pork, sam woo, orange chicken, china, fried rice
dessert	dessert, chocolate, creme brulee, ice cream, dark chocolate, vanilla, chocolately, strawberry, whipped cream, cake

complained	markus, der, bahn, karl, nello, bei, yvonne, andreas, bruno, complained
vivid	intricate, richly, illusion, weave, coloured, layered, thoughtfully, colourful, fabric, vivid
breast	belly, breast, pork, roasted, shoulder, skin, braised, tender, dried, cheek
own	stevens, tucker, leary, fulton, anderson, horton, bryce, bret, bernie, own
uk	british, australia, australian, malaya, pub, wellington, london, england, yukon, uk
SciDoc	
cardiovascular disease	cardiovascular disease, cardiovascular risk, cardiovascular event, coronary heart disease, cardiovascular risk factor, cardiovascular disease cvd, vascular disease, coronary artery disease, physical inactivity, cardiovascular morbidity mortality
chronic kidney disease	chronic kidney disease, end stage renal disease, kidney disease, renal function, renal disease, renal failure, estimate glomerular filtration rate, estimate glomerular filtration rate egfr, glomerular filtration rate, modification diet renal disease
mental disorder	mental disorder, mental illness, psychiatric, psychiatric disorder, mental health, psychotic symptom, social exclusion, major depression, self harm, illness
musculoskeletal disorder	musculoskeletal disorder, icf category, physical health, rotator cuff, physical psychological, degenerative disease, spinal cord independence measure scim, scale factor, chronic fatigue syndrome, leg length inequality
neoplasm	neoplasm, malignant tumor, squamous cell carcinoma, tumor, urothelial carcinoma, differentiate thyroid carcinoma, kidney cancer, tumor suppressor gene, immunohistochemical staining, benign lesion
neurological disorder	neurological disorder, movement disorder, king college london, princess margaret, computational biology, bethesda maryland, houston texas, genetics, gruppo italiano, nagasaki university
chronic respiratory disease	nmda receptor antagonist, aminobutyric acid gaba, spread depolarization, voltage dependent calcium channel, sickle cell, power density, fluorescence spectroscopy, premenstrual syndrome, dorsal root ganglion, chronic respiratory disease
diabetes mellitus	aspartate transaminase, iron deficiency anemia, gaussian distribution, rectus abdominis, thoracic aorta, transverse colon, locoregional recurrence, low mass, aortic insufficiency, diabetes mellitus
digestive disease	eastern cape province, capital city, private school, southeast michigan, public housing, electoral roll, rakai district, pune india, cape town south africa, digestive disease
hiv aids	mesor, propagation delay, refresh rate, clustered, averaging, ic, supper, rectangular, drinking water, hiv aids

hepatitis a b c e	blunt trauma, vena cava, digital subtraction angiography dsa, conscious sedation, holter monitor, stent assist coil, basi cfe, peri procedural, osteoporotic tumorous, hepatitis a b c e
DBPedia	
educational institution	educational institution, public university, american university, college, non sectarian, private university, anna university, management science, technical education, medical school
artist	artist, painter, sculptor, illustrator, painting, artistic, sculpture, art, writer, muralist
athlete	athlete, sprinter, swimmer, gymnast, compete summer olympics, teammate, jumper, cyclist, olympic, olympian
company	company, corporation, firm, inc, manufacturer, subsidiary, brand, ltd, multinational corporation, retailer
plant	plant, plantaginaceae, plantain, flower, palm tree, endemic, medicinal plant, herbaceous perennial, annual herb, herbaceous
village	village, villager, population, town, municipality, sydalen, settlement, lie, rural district, hegra
insulin	retro, themed, hearted, gangsta, glam, amg, grunge, undertone, offbeat, insulin
jeans	aforementioned, kottonmouth kings, zac brown band, space needle, split album, chronological, supplemental, glam metal, as, jeans
refrigerator	evelyn, brodie, priscilla, draper, purdy, drayton, hough, pusey, keats, refrigerator
bartender	waitress, prostitute, butcher, drunken, bodyguard, nightclub, eatery, addict, maid, bartender

Table A.1: Selected words of EnsembleTM