INTERACTION OF TOPIC AND CONTEXT INFORMATION FOR TEXT REPRESENTATION

DANG TRAN BINH

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

INTERACTION OF TOPIC AND CONTEXT INFORMATION FOR TEXT REPRESENTATION

DANG TRAN BINH

Supervisor : NGUYEN LE MINH

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science
December 2022

Abstract

Modern living is becoming more and more convenient with machines thanks to the rapid advancement of science and technology. Mainly, computers and the internet are the key elements that enable people to communicate with one another by storing, exchanging, and looking for knowledge in any field. Recently, machine learning and deep learning have developed incredibly quickly, especially in the field of NLP. With their capacity to calculate words and text, semantic tasks continuously advance by enormous leaps and bounds. But human language is highly flexible, inconsistent, and complex. It poses significant difficulties, such as semantic ambiguity, synonyms, contextual words and phrases, and homonyms, which have not yet been fully resolved. So, this study explores semantic challenges in Natural Language Processing (NLP) that improve the efficiency of task-solving techniques.

In this dissertation, we propose effective knowledge injection techniques for NLP semantic problems. We concentrate on using the Transformer architecture, the pre-trained language model, and the topic knowledge from the topic model to complete these tasks in light of the most recent state-of-the-art (SOTA) results. Semantic textual similarity and summarization are two specific tasks where the usefulness of our methods is demonstrated. In order to do this, we suggested a technique to enhance topic information coherence and took into account how it impacts the injection of subject and context information.

The first challenge is the semantic textual similarity task. In most applications, text understanding and representation are essential, especially in automatic processing. Together with the surface features of words, topic information is significant and necessary to provide the context meaning in the text representation. Recently, the integration of linguistic features and topic information has not received close critical attention. To take advantage of topic information, we propose a novel approach to integrate the topic features into the most popular language models called the Sub-word Latent Topic and Sentence Transformer (SubTST). Inspired by Sentence-BERT and tBERT, our proposed architecture has a significant chance to learn and incorporate topic information with linguistic features. The strength of our proposed approach comes from the delicate combination between latent topic information and linguistic features of language models instead of only utilizing topic information in the previous works. The comparison in experiments

and ablation studies against competitive baselines proves the strength of our proposed approach in most benchmark datasets.

The topic information has helped to direct semantics in text summarization, which is the second issue we consider. As a result, we offer research on the tBART, an innovative and effective way for incorporating topic information with the BART model for abstractive summarization. The suggested model incorporates the benefits of the BART, learns latent topics, and uses an align function to translate the token topic vector into context space. The experimental results demonstrate the potency of our suggested approach, which significantly outperforms existing methods on two benchmark datasets.

Finally, we focus on improving topic coherence. Topic coherence is the primary measure of topic modeling. The more accurately the latent topic is exploited, the higher the topic coherence value. In this study, we proposed a novel method for latent topic refinement called Support Learning for Topic Model (SupLeT). The method is based on non-negative matrix factorization and combined with distance metric learning to increase the quality of topic modeling. We used the learned latent topics during the training process as the "soft label" for the teaching of distance metric learning (DML). The idea of using this learning is that it brings the same topic words closer and tries to keep others as far away as possible. With the learning distance metric process iteratively, we can refine the word-document and word-word relations in each step of the training process. Our experiments show that the SupLeT outperforms baseline Latent Dirichlet Allocation and the base models (Non-negative Matrix Factorization and Semantics-assisted Non-negative Matrix Factorization) on the topic coherence metric and accuracy on topic-based document classification, and semantic similarity detection tasks on benchmark datasets.

To summarize, the focus of our research is on solving fundamental issues relating to the interaction between topic information and context information. The efficiency of the suggested methodologies and their potential for domain adaptation was demonstrated by the experimental findings and thorough analysis. The presented models and solution ideas have the potential to be widely applicable to different types of semantic representations of numerous NLP tasks in further studies.

Keywords: Knowledge injection, topic model, transformer, bi-encoder, BART, distance metric learning, semantic similarity detection, summarization.

Acknowledgments

First and foremost, I would like to express my best sincerest gratitude to my supervisor during period of my research, Professor Nguyen Le Minh of Japan Advanced Institute of Science and Technology (JAIST). He gave me the motivation to choose the right research topic and gave me advice on how to handle academic difficulties.

I am deeply to thank the committee members: my second supervisor Associate Professor Kiyoaki Shirai, Professor Satoshi Tojo, Associate Professor Hasegawa Shinobu, and Professor Ken Satoh at the National Institute of Informatics (NII) for effective reviews and discussions on this dissertation. Through discussions, they help me recognize the limited points of my research as well as provided useful suggestions for improving the thesis.

I would like to thank JAIST and Nguyen's lab for providing me with a Doctoral Research Fellow scholarship when I was pursuing my Ph.D. I also want to express my gratitude to the "JAIST Research Grant for Students" for helping me attend and present my research at conferences.

I also like to thank the Badminton clubs and the JAIST personnel for creating such a beautiful environment for both study and daily life. I want to express my sincere gratitude and appreciation to each and every member of Nguyen's lab. It was a beautiful moment in my scientific career while I studied and resided at JAIST.

Finally, I want to say that I sincerely thank my parents and my love Nguyen Thi Van Anh for lovingly and patiently standing beside me. Without their support, I might never complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The problems

Currently, computers and the internet are one of the most critical factors in human life. Data is digitized in most fields and professions of life. In the digital age, more and more human-generated text data is created over time, such as articles, blogs, advertisements, etc. So, understanding, analyzing, and choosing are necessary for real-life.

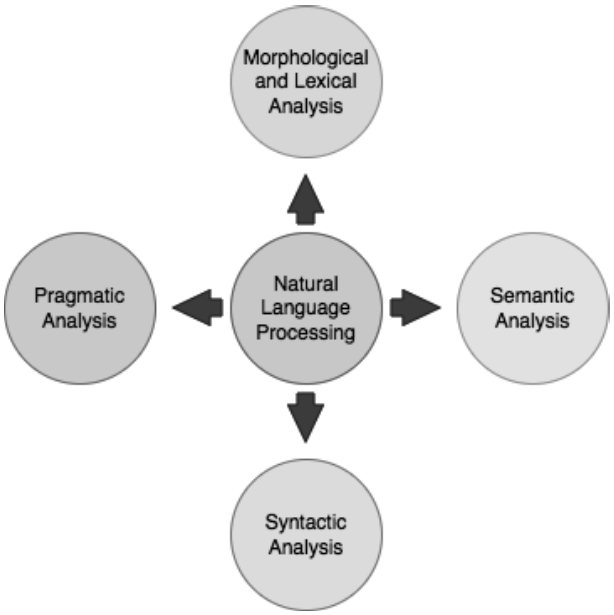


Figure 1.1: Components of NLP

Natural Language Processing (NLP) tremendously contributes to real-world products in information extraction, language comprehension, and language generation. It fuses the potential of linguistics and computer science to study the grammar and structure

of language and develop intelligent systems that can comprehend, analyze, and extract meaning from text and speech data. As Figure 1.1, NLP has 4 essential components includes:

- A vocabulary needs to comprise words and expressions. This is called lexical analysis. It shows how to analyze, recognize, and describe word structures. It involves breaking a text down into clauses, phrases, and paragraphs.

- Syntactic Analysis: The words are commonly accepted as minor syntax units. The principles and regulations governing each specific language's sentence construction are referred to as syntax. Following syntax, the placement of words can change their meaning in the sentence. This entails studying the grammar of a sentence in order to analyze the words within it. To demonstrate how the words relate to one another, the words are turned into structure.

- Semantic Analysis: The syntactic analyzer builds to assign meanings called semantic analysis. It demonstrates the relationships between the words. Only the literal meaning of words, phrases, and sentences is the sole subject of semantics. By doing this, the dictionary definition or the true meaning is just abstracted from the context. The syntactic analyzer's assigned structures have meaning that is permanent.

- Pragmatic Analysis: The total communicative and social content and its impact on interpretation are the focus of pragmatic analysis. It entails removing or extracting the purposeful application of words in context. The first consideration in this approach is always what was stated in reinterpreting what is intended. By using a set of guidelines that define cooperative dialogues, pragmatic analysis aids users in identifying this desired outcome.

The goal of semantic analysis is to determine what a language means. However, semantics is regarded as one of the most difficult domains in NLP due to language's polysemy and ambiguity. In an effort to understand the subject of a text as well as the meaning of words, semantic tasks examine sentence structure, word interactions, and related concepts.

Recently, machine learning and deep learning development have been extremely fast, especially with NLP field. Semantic tasks are continually expanding in enormous jumps

and bounds with their capacity for computation. However, human language is incredibly complex, fluid, and inconsistent. It raises significant difficulties that have not yet been fully resolved, such as Semantic ambiguity, Synonyms, and Contextual words and phrases and homonyms.

Therefore, this study investigates to improve the semantic meaning representation of words, sentences, or documents in Natural Language Processing (NLP) tasks. This research aims to increase the performance of methods that solve tasks. Besides, this task also plays a crucial role in human-machine communication as virtual assistants or chatbots, etc., have become popular rapidly in recent times. Therefore, we expect this study to be widely applied in the research community and in practical applications to improve human life.

1.2 Research direction and our contributions

Recently, pre-trained transformer-based language models such as BERT [1], RoBERTa [2], BART [3], etc, have proven their effectiveness in a variety of semantic tasks. In previous, language models were often based on the previous context to predict the following words. Based on transformer architecture, Transformer-based language models are appealing for clinical NLP because they may be used as a shared layer for transfer learning, in which pre-training them with a large amount of text data can benefit downstream tasks where annotated training data are scarce. These tasks make the learned context information become generality. Usually, pre-trained transformer-based language models only learn focus on context information. However, context information is not enough for tasks about semantics meaning. We consider increasing other information to raise the performance of transformer-based language models. One of them is topic information. Different from context, the topic information shows the general meaning of a group of words, sentences, or documents. It has an overview of the corpus. So, the topic of information injection is the main direction of our research.

The general idea for the dissertation is that combine the representation vector of topic information and the representation vector of context information.

- **Topic information** is a subject, theme, category, or general area of interest on

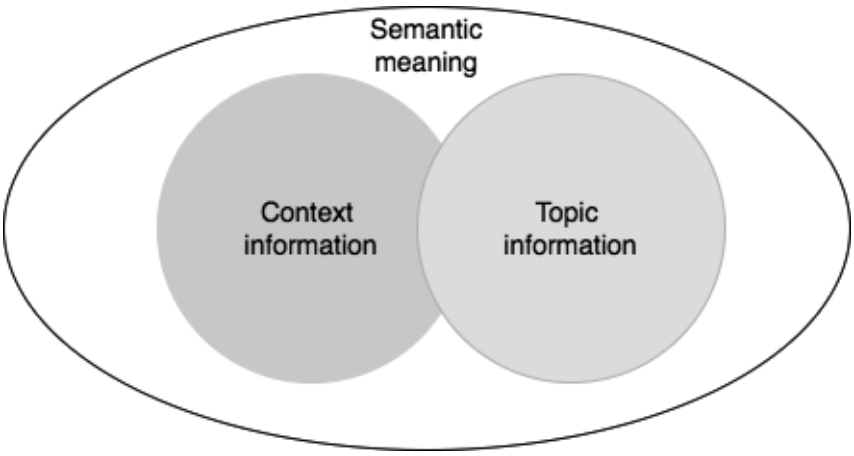


Figure 1.2: The relation of context information and topic information

which a person writes or speaks; a proposition for discussion or argument; a text or a group text. It deploys by topic modeling. Topic modeling is an unsupervised learning technique that's the ability to scan a collection of documents, identify word and phrase patterns within them, and automatically cluster word groups and related phrases that best describe a collection of documents.

Example: Give a corpus about news, a few news has keywords such as school, classroom, teacher, student,... They can have a topic about "Education". Or, a few news has keywords such as cinema, movie, TV show, actor, actress ... They can have a topic about "Entertainment".

- **Context information** is the surroundings, circumstances, environment, background, or settings that determine, specify, or clarify the meaning of an event or an entity. In NLP, context information is shown by surroundings words or surrounding sentences. Different from topic information, context is quite detailed in a small area. It is deployed by language modeling.

  Example: Give an event: "He goes to the bank". Based on surroundings words or surrounding sentences, we can determine the context of this sentence. If the next sentence is "He wants to open an account", we can understand that he will come location where is a financial establishment that invests money deposited by customers, pays it out when required, makes loans at interest, and exchanges currency. If the next sentence is "He wants to go fishing", we can understand that he will come location where is the land next to a river or lake.

We proposed divide the combination follow 2 type including: Outside interaction and Inside interaction.

- **Outside interaction**: We use the output of topic modeling and the output of the transformer-based language model to combine. The concatenation function is the foundation of the combination method. The representation vector of context concatenate with the representation vector of topic to create the new representation vector. The new vector has the dimension is that by the total of the dimension of topic vector and the dimension of context vector.

- **Inside interaction**: We use the output of topic modeling and add it to the transformer-based language model to train. The combination method is used the align function. The representation vector of topic is transferred to context space by an align function. After that, it is added to input embedding of the language model to create the new representation vector. The new vector has the dimension same as the dimension of the context vector.

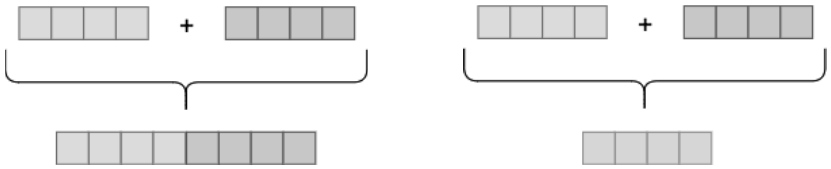


Figure 1.3: The two type of solution in our dissertation

In this research, we aim to obtain efficient knowledge injection methods for semantic tasks in NLP. Based on the results recently, our primary concern is using the Transformer architecture, pre-trained language model, and topic information from the topic model for these tasks. The effectiveness of our approaches is shown in two specific tasks, including Semantic textual similarity and Summarization. To this end, we proposed a method to improve the coherence of topic information and consider how it affects the injection of topic and context information.

- **Outside Interaction: Concatenation of topic information and context information**:In this research, we suggested a unique method for adding topic information over sub-words into Transformer-based language models to improve their

STS performance. The Sub-word Latent Topic and Sentence Transformer is the technique's name (SubTST). It takes skill to include topic- and context-based information into each word carefully. Our suggested strategy demonstrates its efficacy at the word level through our combination instead of using the topic information at the sentence level. We also give the subject model's use of the lexical unit a lot of thought. The merging of topic-based and context-based features, which received less attention in earlier techniques, has been made much easier by it. We show in the comprehensive experimental data that our suggested model significantly beats the competing approaches in both Semantic Textual Similarity and Semantic Similarity Detection tasks on most of the benchmark datasets.

- **Inside interaction: Topic based knowledge injection**: In this research, we suggest a unique approach for adding topic information into BART models to improve their ability to do abstractive summarization. The technique is known as the tBART. The BART architecture is essentially what the tBART employs. The latent subjects are acquired using sub-words in this manner as opposed to documents or words as they were in earlier research. Additionally, we use an align function to transfer the representation vector produced by the topic model into context space. A generic topic distribution adds the subject information during the encode and decode procedures. By comparing the proposed model to the benchmark datasets, XSUM and CNN/DAILY MAIL, we show that it performs noticeably better than several earlier studies.

- **Improving topic coherence and impact in the interaction**: In this study, we offer the SupLeT, a novel strategy for improving latent topics retrieved by the topic model that incorporates distance metric learning. This approach considerably enhances the topic's cohesion. Without taking into account the link between words, topic models using corpora frequently produce incoherent topics. In order to mine coherent topics, we are focused on creating a technique that considers word vectors in the exchange between words and between words and documents. As you are aware, one particular variation of the soft-clustering model is the topic model. Therefore, as support techniques, we can employ contributions about cluster validation for the topic model. We aim to bring the precise topic phrases closer while keeping

others as far away using the capability of distance metric learning (DML). Thus, the coherence of learned latent topics is shown better by the most related words of these topics. Moreover, we evaluate the effectiveness of topic coherence in topic information interaction (SubTST, tBART).

1.3 Dissertation outline

We have introduced the abstract as well as presented the research direction of our work in this Chapter. In the remainder of this dissertation, we provide the detail of the experiments and our proposed model architecture following.

- Chapter 2: We present the knowledge background for our research. This is the general premise to orientate solutions.

- Chapter 3: In this chapter, we report research on the Sub-word Latent Topic and Sentence Transformer (SubTST), an innovative and effective way of incorporating topic information with Transformer-based models. The suggested approach essentially adopts the benefits of the SBERT [4] architecture and learns latent topics at the sub-word level as opposed to the document or word level as in earlier studies.

- Chapter 4: With chapter 4, we proposed research on an innovative and effective technique for integrating topic information into the BART model. for abstractive summarization, called the tBART. The proposed model inherits the advantages of the BART, learns latent topics, and transfers the topic vector of tokens to context space by an align function.

- Chapter 5: We introduce a support method for topic model, call the SupLeT. The method use distance metric learning to refine learned latent topics.

- Chapter 6: Our dissertation comes to a close with a summary of the conclusions we reached for this dissertation as well as recommendations for future works.

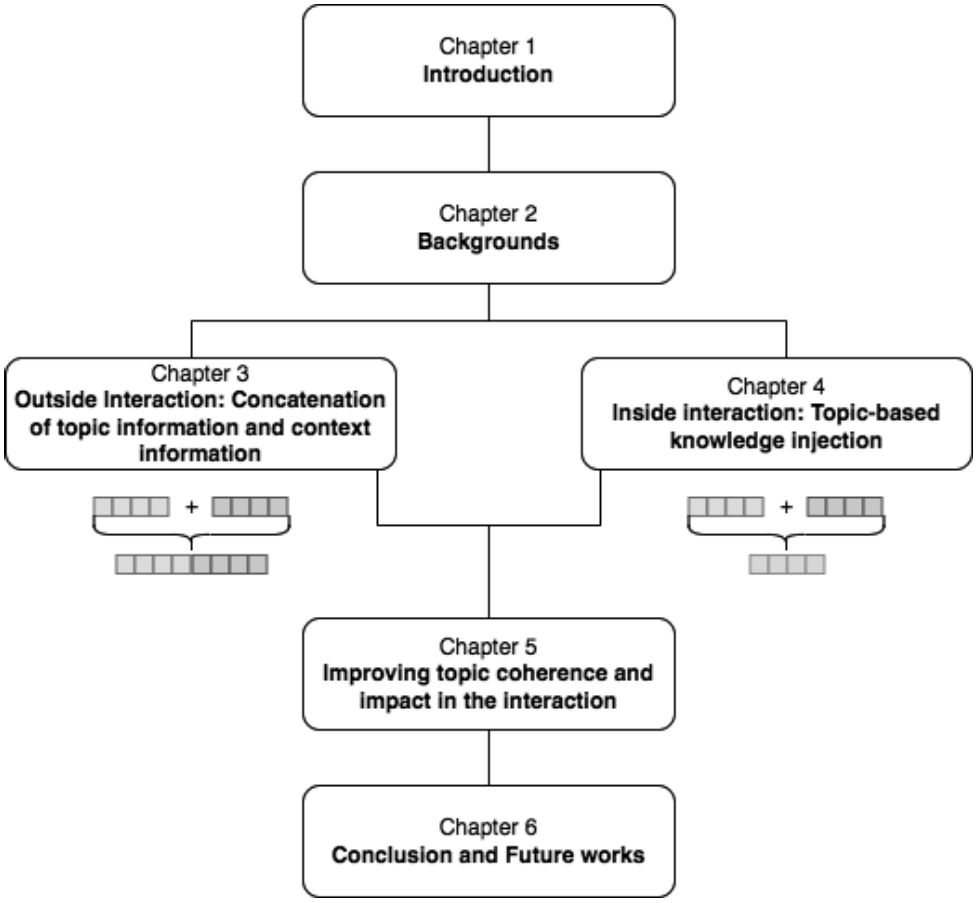


Figure 1.4: The outline of the dissertation

Chapter 2

Backgrounds

2.1 Semantics textual similarity and Summarization

In this dissertation, we focus on two tasks of semantic tasks includes: Semantic textual similarity (STS) and Summarization.

2.1.1 Semantics textual similarity

A task in the field of Natural Language Processing (NLP) called semantic similarity, also known as semantic textual similarity. This task evaluates the relationship between texts or documents using a predetermined metric. There are several applications of semantic similarity, including sentiment analysis, information retrieval, etc. We can formulate this task such as:

Inputs: Pair of text sequences $S_1 = w_1, w_2, w_3, ..., w_{N1}$ and $S_2 = w_1, w_2, w_3, ..., w_{N2}$ with $N1, N2$ is number of words in text sequence S_1, S_2

Outputs: Label (Similar/non-similar) or Score (from 1 to 5)

We can provide a classification of semantic textual similarity solutions including [5]:

- **Knowledge-based**: Semantic similarity between two phrases is calculated using knowledge-based methods. They are founded on data taken from one or more underlying knowledge sources, such as dictionaries, ontology databases, etc. The techniques display terms or concepts associated with semantic links in a systematic manner.

- **Corpus-based**: Using data gathered from sizable corpora, corpus-based algorithms quantify the semantic similarity of terms.

- **Deep learning-based**: This approach applies the developments in neural networks and deep learning to solve the problem.

- **Hybrid**: The hybrid approach exploits the advantage of methods and builds models to calculate semantic similarity.

2.1.2 Summarization

Summarization is the act of computationally condensing a set of data to produce a subset (a summary) that encapsulates the most crucial or pertinent information from the original material [6]. The produced summary should be shorter in length than the input text and include the most important information in the input text [7]. The summarization task supports the readers to get the important points of the original document without the need to read the entire document. We can formulate this task such as:

Inputs: Original document $d = w_1, w_2, w_3, ..., w_N$ with N is number of words in d

Outputs: Summary document $S_d = w_1, w_2, w_3, ..., w_{N_s}$ with N_s is number of words in summary S_d and $N_s < N$

Based on the above definition of the task, we can devise three types of solutions to summarize includes extractive, abstractive, and hybrid.

- **Extractive summarization**: With the extractive approach, the content of the summary document is extracted from the original document, but the words of summary document are not modified in any way. The content of the summary document includes keywords, keyphrases, or key sentences from the original document.

- **Abstractive summarization**: Abstractive summarization methods analyze the original content, develop a semantic meaning representation of it, and utilize this representation to provide a summary that is more akin to what a human language might comprehend.

- **Hybird**: The hybrid approach combines both the abstractive summarization and extractive summarization approaches.

2.2 Transformer and Pre-trained language models

Recently, a new powerful model, Transformer, introduced by Vaswani et al. [8] got impressive performance in machine translation tasks by using the self-attention mechanism. Similar to the previous architecture, this model also contains two components Encoder and Decoder, separately. Compared with the model Sequence-to-sequence using LSTM, this architecture is based on the attention score between pairs of words to compute the dependencies between them. Therefore, it can overcome the vanishing gradient problem with the long sentence. Besides, this architecture is proven to be effective in transferring knowledge with pre-trained language models, especially on machine reading comprehension tasks [1]. Therefore, our work focus on improving Transformer architecture with topic information for the semantic tasks.

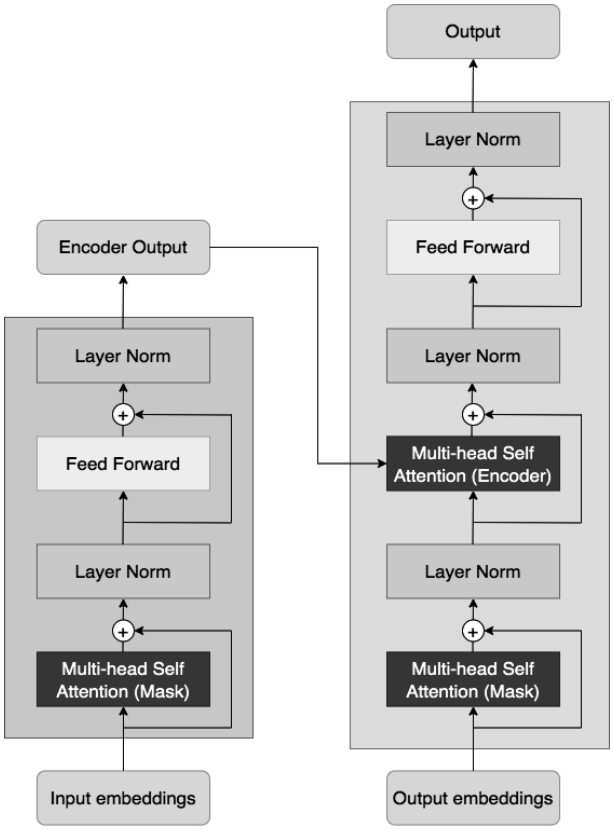


Figure 2.1: The architecture of Transformer

A variety of statistical and probabilistic techniques are used in language modeling to estimate the likelihood that a given string of words will appear in a sentence. In order to provide a foundation for their word predictions, language models examine corpora of text data. Pre-trained transformer-based language models, however, have provided the

academic and practice communities with a breakthrough for implementing automated Natural Language Processing (NLP) techniques, even with constrained time and computational resources, in recent years. These models are even more alluring because they have produced cutting-edge results in numerous NLP tasks.

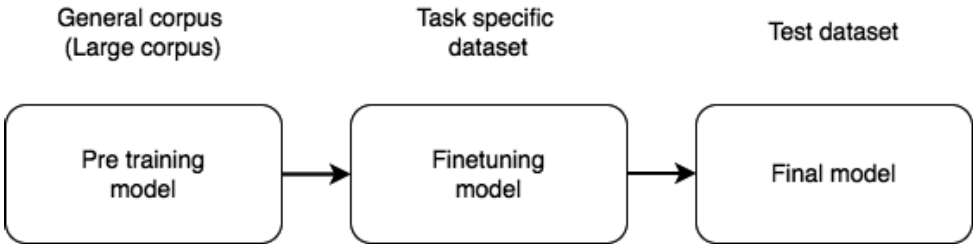


Figure 2.2: Pretrained language models progress

Pre-trained transformer-based language models are language models that have undergone extensive training with little regard for the detailed tasks they will be used. Two pre-trained tasks are often used that mask language model and next sentence prediction to pre-train. These tasks are trained on a large corpus (Wikipedia, etc). However, to use pre-trained language models, the final output layers need to be adjusted for the task; this is known as the fine-tuning stage. In this step, the pre-trained models are modified by specific tasks such as (retrieval, generation, classification, etc). And then, we have the final model to evaluate on the test dataset. This is the final step to evaluate the model after fine-tuning. The final model can use in real systems.

2.3 Topic models

A common issue in natural language processing is topic modeling. What exactly is topic modeling? Finding the abstract "themes" that appear in a group of texts is one use of a topic model. A popular text-mining technique for identifying hidden semantic patterns in a text body is topic modeling. When creating a paper, we always want to have a specific topic in mind and anticipate certain phrases that illustrate the issue to appear more or less frequently. For example, a document is in "Education" topic, the words which belong to "Education" topic such as school, classroom, student, teacher, ... can appear more often in this document. However, a document often relates to several themes in varying degrees. Thus, the topic whose proportion is more significant than other topics,

the number of words on this topic also is larger in the document. The meaning of latent topics discovered by topic modeling techniques often determine by clusters of similar words - words have relation to semantic information.

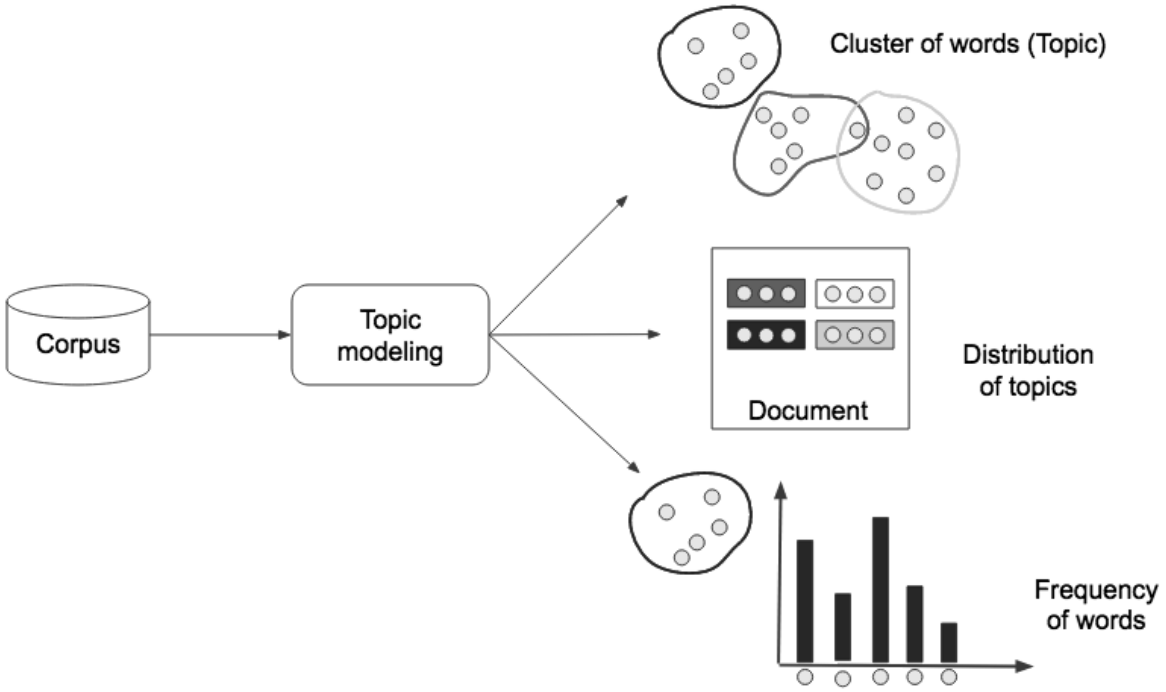


Figure 2.3: The definition of Topic Modeling

Give an example of topic modeling. We used topic modeling to identify the topics of a set of comments by recognizing patterns and recurrent terms. Let's see how the following comment about service might be grouped using an "unsupervised" learning method, such as: "My son like playing games on Roblox. It is free for regular users. However, if you play over 10 games, you need to pay the fee for members: 2 euros for 1 month. You can get a discount of 15% if you charge for one year." By recognizing phrases and idioms like free, pay, 2 euros, the fee, etc, we can cluster this comment with other comments that talk about similar things ("pricing"). In essence, topic modeling algorithms generate lists of patterns and words that they believe to be connected, leaving it up to you to decipher the significance of these relationships.

With a basic topic modeling tool, the input of model often uses Bag-of-word algorithm to represent the relationship between word and document such as word-document matrix. The representation does not depend on the order of words in the document. Or, to put it another way, the document's words are interchangeable. Moreover, there

is no relationship between the documents in a corpus; they are all independent. The fundamental presumptions of a topic model could be referred to as the exchange ability of words and documents. Latent topics on a corpus can be found using statistical methods by scrutinizing the words in the original texts. Two matrices of words and documents in latent topic space are frequently the output of topic modeling. They demonstrate the corpus's capacity for clustering. It is possible to group documents by the topic probability distribution. A topic model, however, is more than just a cluster algorithm.

2.4 Knowledge injection

Deep learning has advanced significantly in recent years, with neural network-based models obtaining the best performance in a range of applications. However, there are a number of limitations that make it challenging to apply entirely data-driven neural network models in practical settings. Among these include:

- The excessive reliance on training data.
- The lack of robustness.
- The inability to generalize.
- The challenge of providing a sufficient explanation.
- The obvious gaps in latent and common knowledge.

Due to the availability of knowledge sources that are rich and structured, the researchers is looking into Knowledge Injection to address the above issues. As a result, hybrid AI systems have been developed, which hybrid the data-driven learning of models with knowledge from outsources. Knowledge Injection systems include various neural network configurations with knowledge graphs, organized knowledge bases, and retrieval-augmented neural models, to name a few.

Other intriguing results of building AI systems with human knowledge include increased predictability and trust. In contrast to contemporary AI systems, humans frequently rely on forms of expertise that are both formally specified and implicitly understood. Therefore, we investigate methodologies and techniques that look at how knowledge injection can make up for the absence of such information in AI systems.

To gain insight about how to attain effective knowledge injection, we performed a systematic classification of existing knowledge injection based on board literature study [9]. The methods of knowledge injection can be divided following:

- **Feature fused**: An established knowledge base is used to provide features for this kind of model. By projecting into embedding with a trainable matrix and learning its meaning through a pre-training task, feature fusion often takes this into account [10, 11].

- **Embedding combined**: Embedding combined methods produces embedding from additional information that significantly improves model performance. Then, the tokens will be finetuned so that their associated embedding by attention mechanisms or other weighting procedures can be combined [12, 13].

- **Knowledge supervised**: Knowledge supervised methods choose the entities that meet a specific relation or relational triplets of knowledge graph as training data to avoid the extra training cost of the model [10]. The method of concatenating relational triplets or entities with the input sequence without sacrificing efficiency [14].

- **Retrieval based**: Instead of injecting knowledge, retrieval based methods can update perception by consulting external knowledge. They usually retrieve desired information from knowledge sources by computing the relevance between input text and knowledge [15]. They do not preserve knowledge within models.

- **Rule guided**: To consider the effective of knowledge injection in downstream tasks, rule guided methods maintain the original form of additional knowledge [16, 17]. A major advantage of such methods is that they guarantee the reliability of results using mathematical formulations and provide interpretability through an explicit reasoning process.

Chapter 3

Outside Interaction: Concatenation of topic information and context information

3.1 Introduction

In the rapid explosion of information, the importance of text understanding is more and more prominent. In most Natural Language Processing (NLP) tasks, it is essential to automatically extract and represent words, sentences, and documents in a meaningful space. With the development of many NLP techniques, there are more and more linguistic applications in practice. The most popular ones belong to search engines. In the digital world, searching action occurs every day and everywhere. A critical success factor of most information retrieval systems comes from understanding and revealing the similarity between query and samples in searching space such as online sites, documents, etc. This protocol requires a deep understanding of textual information instead of the traditional word-based overlap methods. Originating from practical and scientific value, Semantic Textual Similarity Detection has always been a fundamental task in NLP.

Semantic Textual Similarity (STS) and semantic similarity detection (SSD) are to determine the semantic correlation between a pair of text sequences. It plays an important role in Natural Language Understanding (NLU) to reveal the semantic similarity among sentences. Obviously, semantic similarity is a crucial and key component in most Informa-

tion Retrieval systems [18] as well as the other ones such as Question answering [19]. In the long-term consistent growth of Natural Language Processing, STS is also considered as the fundamental and essential task in text understanding. The typical example of the SSD task and STS task is presented in Table 3.1 by following:

- Semantic similarity detection(SSD or STS - label): In this task, each dataset provide a pair of text sequences. The goal of the task is that predict the semantic similarity between text sequence pairs in binary classification task (Similar(1)/Non-similar(0))

- Semantic textual similarity(STS or STS - score): In this task, each dataset provide a pair of text sequences. The goal of the task is that predict the semantic similarity between text sequence pairs. However, different from the SSD, the output of STS is the similarity score of each text sequence pair (range from 0 to 5).

In all samples, there is a lot of overlap in the surface of two sentences, especially the keywords. It is a barrier to the success of text understanding systems.

Table 3.1: Example of semantic similarity detection(SSD) and semantic textual similarity(STS)

Task	Sentence Pair (Input)	Output
SSD	S1: How do I get funding for my web based startup idea ? S2: How do I get seed funding pre-product ?	similar.
SSD	S1: What is ecstasy ? S2: How addictive is ecstasy ?	non - similar
STS	S1: How do I get funding for my web based startup idea ? S2: How do I get seed funding pre-product ?	4.4/5.0
STS	S1: What is ecstasy ? S2: How addictive is ecstasy ?	1.6/5.0

Traditionally, a challenge of most NLU models comes from linguistic ambiguity. It means that a word and even a sentence can be interpreted in many ways, which is based on the context and topic of the conversation. For example, the sentence “I will go to the bank” has at least two meanings. Based on the specific topic such as either economy or scenery, it is easy to interpret the correct content of the above sentence. Therefore, this typical example reveals that topic information is crucial in text understanding. Together with the surface context inside sequences, topic features are essentially useful to clarify the sentence meaning.

The strength of latent topic information as features in text understanding has been proved in many tasks such as sentiment analysis [20], classification [21], sentence generation [22] etc. Especially, in Semantic Textual Similarity, it has been pointed out in several studies [23], [24], [25], [26] and [27]. The previous works, however, utilize topic information as an external feature. It is not integrated into a language model to extract the words' meaning and relationship. On the contrary, the combination between topic and latent features is more promising and effective. By concatenating topic and sentence embedding, tBERT [28] is one of the most attractive approaches in this flow. All the above approaches are the most prominent demonstration of the power of topic information in text understanding, especially Semantic Textual Similarity.

So far, most previous studies on the STS focus on digesting the contextual meaning of text such as Sentence-BERT [4], ECNU [27] and so on. Especially, the appearance of Transformer-based Language Models is the key factor for the success in recent STS approaches. However, as we mentioned above, the integration of topic information into language models has little interest in previous works. Therefore, we propose a novel approach for enhancing the capacity of Transformer-based language models for the STS by incorporating topic information over sub-words. The method is called the Sub-word Latent Topic and Sentence Transformer (SubTST).

The following are the main contributions of our work:

- We propose a delicate integration of topic-based and context-based information into each word. Instead of utilizing the topic information at sentence-level, our proposed approach proves the effectiveness in word-level via our combination.

- We also deploy a remarkable consideration to using the lexicon unit in the topic model. It has greatly facilitated the hybrid of the topic-based features and context-based features, which has less attention in previous approaches.

- Through the detailed experimental results, we indicate that our approach prominently outperforms the competitive methods in both Semantic Textual Similarity and Semantic Similarity Detection tasks on most of the benchmark datasets.

3.2 Related Works

C

3.2.1 Sentence representation learning

Recently, the pre-trained language models and their modifications have been popularly used to represent natural language. BERT [1] is one of the most popular pre-trained language models based on the transformer network. It outperforms many traditional approaches in a lot of NLP tasks such as information retrieval, sentence classification, and so on [1]. The strength of this approach comes from the bidirectional Transformer encoder and next sentence prediction (NSP) training objective function which is useful to embed and learn semantic relation of sentence pair. Especially, BERT obtains significant results in semantic textual similarity tasks. It proves Transformer-based language models have a tremendous potential to extract and understand textual information.

Especially, Sentence-BERT (SBERT) [4] recently achieves the promising performance in the Semantic Textual Similarity. It is a variant of BERT based on encoding each sentence by siamese and triplet networks. Particularly, every sentence is separately embedded by BERT. Then, a pooling layer is added at the top of each sentence embedding to normalize the sentence representation. In this model, MEAN-strategy is used as the default setting of the pooling layer. The weights of each embedding flow are simultaneously updated by the siamese network. In the paper, the authors proposed three versions of objective function to adapt it into three kinds of tasks which include classification, regression, and triplet loss. Based on the siamese network and pooling layers, Sentence-BERT reduces quite a lot of training time than previous neural sentence embedding methods. Moreover, Sentence-BERT is popular to be considered as a fundamental language model for the Semantic Textual Similarity and Detection task.

Another problem in text understanding is the ambiguity in natural languages. Therefore, in the development of Transformer-based language models, there is much considerable effort to address the anisotropic problem. Most approaches in this branch are trying to improve the quality of textual representation by the external information. In particular, BERT-flow [29] tries to transform the sentence embedding distribution of BERT into an isotropic Gaussian distribution through normalizing flows [30] that are learned with

an unsupervised objective. Besides the flow-based approach, the whitening method in BERT-whitening [31] also achieves promising results against the previous works. Specifically, the whitening tried to transform the mean value of the sentence vectors to zero and the covariance matrix to the identity matrix. The main goal of these approaches is quite similar to our proposed architecture. Instead of adjusting the word embedding space, our model proposes a combination to expand the meaning of each word by its topic information.

In another consideration, ConSERT [32] is a self-supervised framework for sentence embedding based on contrastive learning. Without any external modification of BERT embedding, ConSERT puts its concentration on the strength of the objective function of training. Therefore, it is only useful in the specific tasks which are more suitable for contrastive learning. Despite its difference from our goal, we also consider it as a promising approach to semantic textual similarity and detection. This choice comes from the close relation between contrastive learning and semantic textual similarity.

Finally, it also exists a few works that utilize topic information for text representation. One of the most related approaches to our work is tBERT [28]. Specifically, in the tBERT model, a sentence pair is concatenated by $[SEP]$ token and processed by BERT encoder (a.k.a., cross-encoder). The representation of sentence pair is extracted by the outputs of $[CLS]$ token. Then, the document/word topic embedding vector is calculated by the average of all topic information corresponding to document/words. The final representation of tBERT is a concatenation between a sentence pair from BERT and two average topic embeddings in either document or word level. The performance of tBERT approach proves the importance of topic information in Semantic Textual Similarity and Detection task. Obviously, the interaction between topic information and textual features among surface works, however, gains less concern in this work. The topic information only contributes to the classifier layer instead of textual representation. Therefore, to address this drawback, we propose a delicate integration between topic information and textual embedding of words in the sentence. This combination provides more contexts to representation at the word-level, which is important groundwork for text understanding.

3.2.2 Topic Modeling

One of the popular methods for topic modeling is Latent Dirichlet Allocation (LDA) [33] that is based on generative probabilistic models. A fundamental premise of LDA is that a document was created by selecting a number of subjects, and then selecting a number of words for each topic. In recent years, Deep learning was used as the approach for topic model. Approaches using it in topic model had good results as neural topic models(NTM) [34]. With this model, backpropagation training is possible within the context of neural variational inference. Additionally, using a stick-breaking structure, we suggest a recurrent network that is comparable to Bayesian non-parametric topic models in that it can find a notionally unlimited number of topics. In 2019, Adversarial-neural Topic Model(ATM) [35] is the first time adversarial training for topic modeling. This model tried to capture the semantic patterns among latent topics by the generator network and discriminator network.

As an alternative, Non-negative matrix factorization(NMF) [36, 37] is an interesting approach for topic modeling. It is a method fit for short text datasets. From the perspectives of consistency across several runs and early empirical convergence, this technique offers many tangible benefits. Xiaohui et al. [38] used a factorized symmetric term correlation matrix for topic modeling. This approach aims to teach subjects by studying the words of correlation data. The method computed word correlation in texts by representing each word with its co-occurring words to derive accurate topics from term correlation data. The topic learning problem on the concept of a correlation matrix was then developed utilizing symmetric non-negative matrix factorization. However, the model is not reliable and stable. The DML-SeaNMF model [26] was proposed to discover topics from the document. The model hybrid document-word relation and word-context relation(semantics relation) as inputs of model. With the refinement of distance metric learning, the distribution of words that have the same latent topic was pulled close. So, the latent topics which were learned become more coherent and clear..

The topic model has been applied to several NLP tasks, such as: document classification [39], translation [40], summarization [41], machine translation [40], etc.

3.3 SubTST - Sub-word Latent Topics and Sentence Transformer

As we mentioned above, topical knowledge is crucial for text comprehension. However, the key question is how to combine the topic information into textual representation. To solve this problem, we propose a novel approach to delicately integrate the sub-word topic features into Transformer-based language models. The details of our architecture are shown in Figure 3.1.

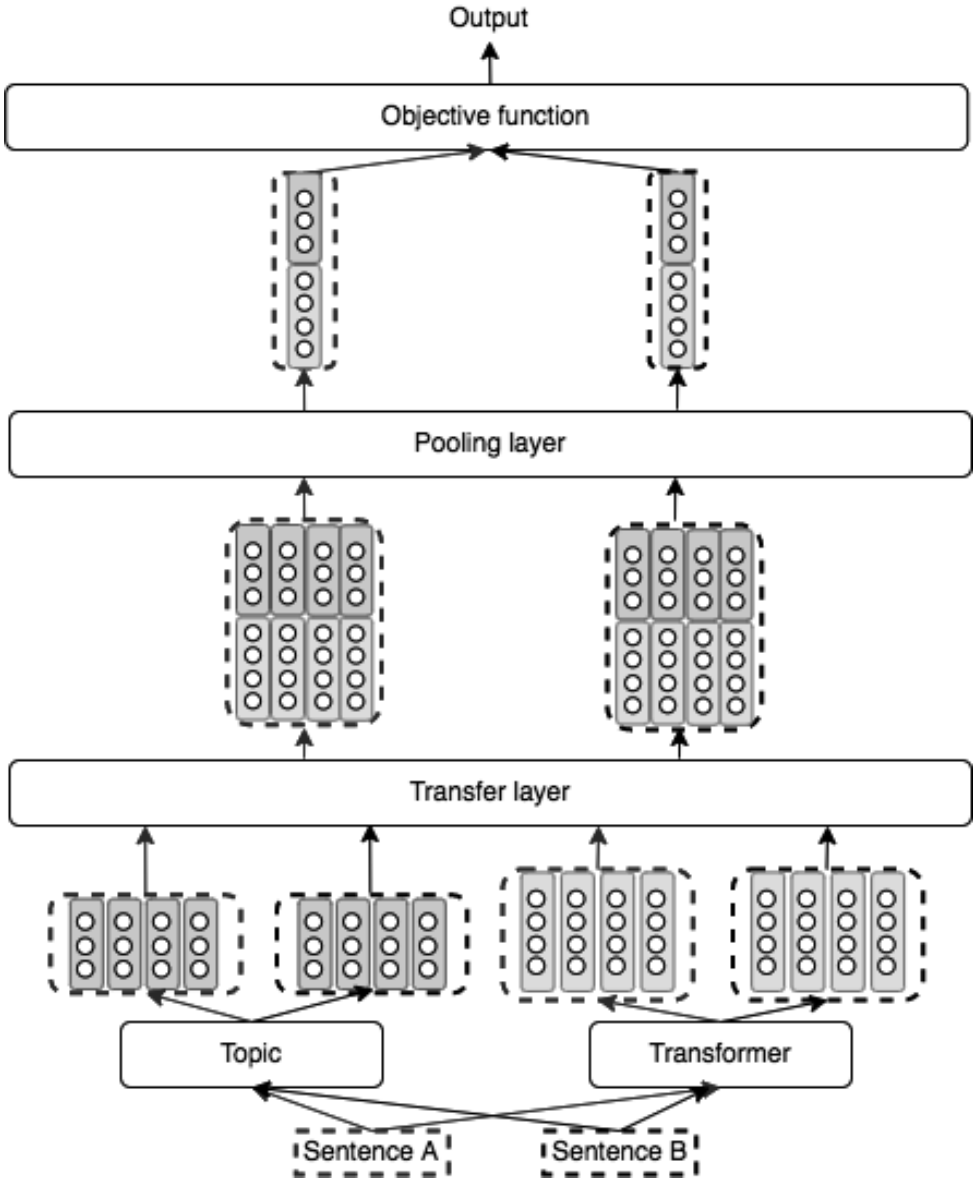


Figure 3.1: The architecture of our approach: SubTST

In particular, an input of our model is a pair of text sequences including S_A and S_B.

Each sentence N is a collection of tokens w as follows:

$$S_A = \{w_i\}_{i=1}^{N_A}$$

$$S_B = \{w_i\}_{i=1}^{N_B}$$

.

In our model, the sentence representation includes topic-based representation from topic modeling and context-based representation from language model.

In the topic-based representation, each sentence is encoded into topic space. With a topic model, the final products often include: (i) the relation between tokens of the vocabulary and topics, (ii) the relation between documents of learned corpus and topics. In this work, the relation between tokens of the vocabulary and topics is used to encode input sentences. The reason for this choice comes from our consideration of enhancing the meaning of words via topic information. The topic information of each token is embedded into a vector whose dimension is the number of latent topics. Each sentence is expressed by a topic-term matrix of size $k \times N_s$, denoted by M_t where k is the number of latent topics and N_s is the number of tokens in each sentence:

$$Vt_i = TopicModel(w_i) \ \in R^k \tag{3.1}$$

$$M_t = \{Vt_i\}_{i=1}^{N_s} \ \in R^{k \times N_s} \tag{3.2}$$

In the context-based representation, the language models are provided to capture context information of tokens in the sentence. The pre-trained language models based on Transformer architecture have been widely used to learn the context information of words in many NLP tasks. We denote the output of the pre-trained Transformer-based model with a text sequence by $M_c \in R^{m \times N_s}$ where m is the internal hidden size of the transformer model and N_s is the number of tokens in an input sentence s.

$$Vc_i = Transformer(w_i) \ \in R^m \tag{3.3}$$

$$M_c = \{Vc_i\}_{i=1}^{N_s} \ \in R^{m \times N_s} \tag{3.4}$$

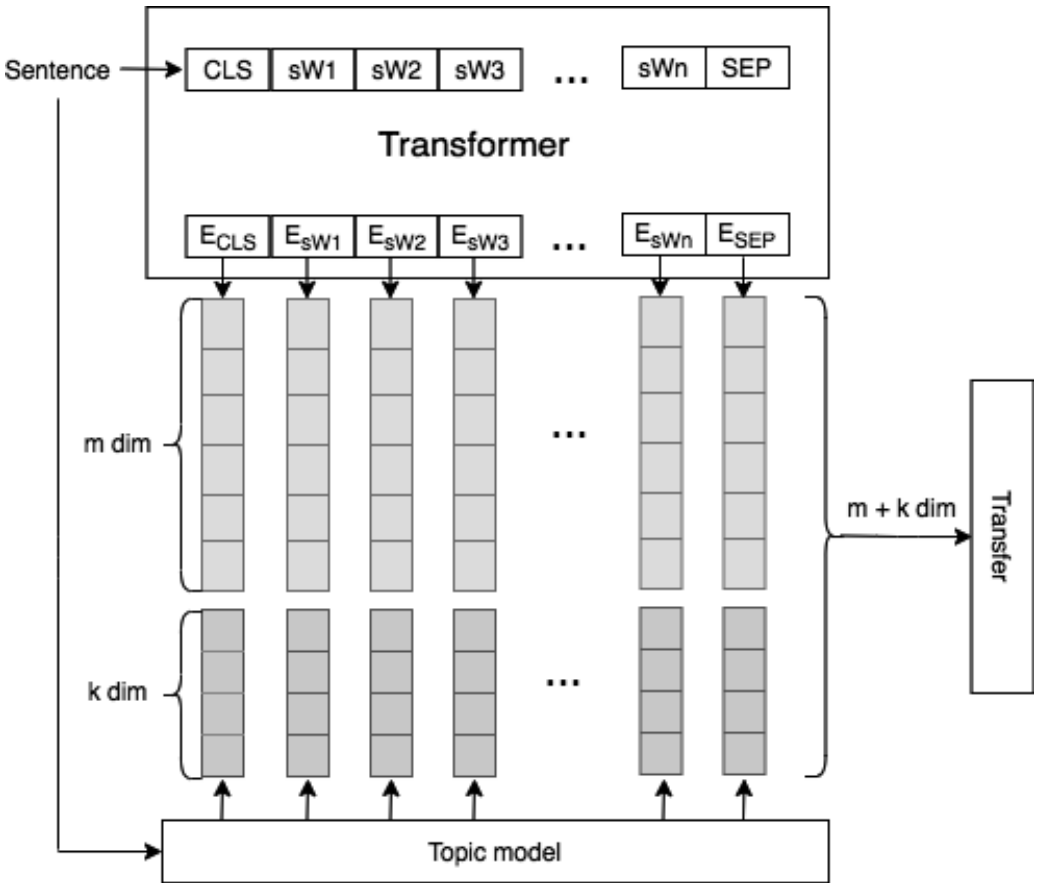


Figure 3.2: Combination of topic embedding and word embedding

After being extracted the topic-based and context-based features, these features are connected by the procedure in Figure 3.2. To aggregate the topic information with the output of Transformer-based models, we concatenate M_c and M_t into $M_{ct} \in R^{(m+k)\times N_s}$ as the following:

$$M_{ct} = \begin{pmatrix} M_c \\ M_t \end{pmatrix} = \{M_{ct_i} = CAT(Vc_i, Vt_i)\}_{i=1}^{N_s} \in R^{(m+k)\times N_s} \tag{3.5}$$

where $CAT(.)$ is the concatenation function.

So that, each token in the input sentence is encoded by a vector that has $m + k$ dimensions. This representation vector includes the context-based features on the top and topic-based features at its bottom. The number of dimensions for the vector representation depended on the hyper-parameter k - a number of pre-defined topics. We present the way to choose this parameter in the Section 3.4. After extracting the general representation, we feed M_{ct} into a transfer layer to normalize the sentence representation. It is also

illustrated in Figure 3.2. The transfer layer is constructed by the Feed-forward network with the Dropout and the Layer Normalization:

$$\begin{aligned} h\_linear_i &= W_1 M_{ct_i} + B_1 \\ h\_drop_i &= Dropout(h\_linear_i) \\ h\_norm_i &= LayerNorm(h\_drop_i) \\ h_i &= h\_norm_i \end{aligned} \tag{3.6}$$

$$H = \{h_i\}_{i=1}^{N_s} \in R^{(m+k)\times N_s} \tag{3.7}$$

Obviously, our combination of topic-based and context-based features is pretty straightforward. Therefore, our transfer layers are used to increase the interaction of topic and context to create the unification in the sentence embedding. With the linear transformer via the Feed-forward network, the signal between context and topic is blended together into the mutual representation. After applying a linear transformation to the context-topic matrix M_{ct}, we use a Dropout layer with Bernoulli distribution and the probability $p = 0.5$. Finally, we apply Layer Normalization over hidden state H to normalize the sparse distribution over mini-batch samples. After the transfer layer, the size of hidden state H keeps the same as the size of the context-topic matrix M_{ct}.

Similar to the Sentence-BERT (SBERT) [4], we add the pooling layer into the last layer of our model. It is useful to gather the information of all tokens in the sentence into one representation. Following previous works utilizing pooling layers, we also deploy two pooling strategies that are the Mean-strategy and Max-strategy. In particular, given $H \in R^{(m+k)\times N_s}$, the embedding of sentence $u \in R^{m+k}$ is estimated by:

- Mean-strategy:

$$u = MEAN(H) = \frac{\sum_{i=1}^{N_s} h_i}{N_S} \in R^{m+k} \tag{3.8}$$

- Max-strategy:

$$u = MAX(H) = max_{i=1}^{N_s} h_i \in R^{m+k} \tag{3.9}$$

Finally, our model is also flexible to adapt to many kinds of tasks. Specifically, we also provide the detailed configuration of the two most popular tasks including classification

and regression. Based on each task, the final representation and object function is defined in different ways as follows:

- **Classification**: After extracting the features of sentence in the previous parts, we combine them and their element-wise difference into the sentence pair features in Equation 3.10.

$$f_s = CAT(u_{S_A}, u_{S_B}, \mid u_{S_A} - u_{S_B} \mid) \in R^{3\times(m+k)} \tag{3.10}$$

where $CAT(.)$ is the concatenation function.

And then, we apply a linear function for f_s.

$$r = W_2 f_s + B_2 \tag{3.11}$$

where $W_2 \in R^{3\times(m+k)\times l}$ is the learnable parameter for classifier layer, $m + k$ is dimension of sentence embedding, and l is size of set labels.

Next, our model utilizes the Feed-forward layer with softmax activation to categorize a pair of sentences into its category in Equation 3.12.

$$O = Softmax(r) = \{\frac{e^{r[t]}}{\sum_{q=1}^{l} e^{r[q]}}\}_{t=1}^{l} \tag{3.12}$$

In our model, we consider Cross-Entropy as the object function in this mode.

$$loss(O) = -\sum_{j=1}^{l} log(x_j \mid x_j \in O) \tag{3.13}$$

- **Regression**: In the regression task, the similarity of sentence pairs is calculated by a cosine function following the previous works. Therefore, we also utilize the representation of sentence A and B as the input of the cosine function. We do not add any additional layer to combine sentence pair features. The detail of the similarity score is presented in Equation 3.14.

$$O = cosine\_similarity(u_{S_A}, u_{S_B}) \tag{3.14}$$

In this task, we consider Mean squared error (MSE) loss (in Equation 3.15) as objective function.

$$loss(O) = \frac{1}{T}\sum_{i=1}^{T}(O_i - output\_score_i)^2 \tag{3.15}$$

with T is batch size.

3.4 Experiments

We use our technique in a variety of tasks that are related to semantic similarity in order to assess the efficacy of the suggested approach: (i) Semantic Textual Similarity with label output/Semantic Similarity Detection (STS-label/SSD) in Section 3.4.2 and (ii) Semantic Textual Similarity with score output (STS-score) in Section 3.4.3.

In the first task, our concentration is the strength of our embedding formed by topic and context-based information together. Therefore, our main evaluation in this part surrounds the BERT-based approach and its variants against our SubSTS model. In the second one, the performance of the whole architecture is highlighted in many competitive experiments. With five benchmark datasets in Semantic Similarity Detection, our proposed models prove their strength and potential against the recent approaches.

3.4.1 Experimental Settings

Firslty, in Semantic Textual Similarity, we also employ the pre-trained BERT_{base} and BERT_{large} models[1] to encode each input sentence. However, in Semantic Similarity Detection, we also consider the BERT_{base} model. The reason of this setting comes from the consistence in evaluation as comparing with the previous works. Furthermore, we also utilize the LDA model [33] to extract the topic-based features of each token. In LDA model, the number of latent topics k plays an important role in the success of topic representation. Therefore, this hyper-parameter k was tuned for fitting with each dataset. The detail of our choice comes from the evaluation in the development set, which is presented later in each corresponding task. For optimizing function, we take advantage of Adam

[1]https://github.com/google-research/bert

optimizer [42] with a learning rate of $2e-05$. The linear learning rate warms up over 5% of training data. Dropout is 0.5 for the transfer layer. The maximum sequence length is 512, so we truncate any inputs longer than 512. These settings of the optimizer are the same in the two tasks.

Finally, we also propose some variants of our approach as follows:

- SubTST-Max, SubTST-Mean: In this mode, the topic embedding is frozen during training process. The element Max and $Mean$ point out the strategy of pooling layer of our model in Figure 3.1.

- SubTST-max-train topic, SubTST-mean-train topic: It means that the parameters of topic embedding are learnable and updated during training process by the objective loss. The meaning of max and $mean$ is similar to previous parts.

Table 3.2: The details installation of SubTST for two tasks: Semantic Textual Similarity in section and Semantic Similarity Detection

	Semantic Textual Similarity		Semantic Similarity Detection
Pre-trained model	BERT-base-cased	BERT-large-cased	BERT-base-cased
Context dims	768	1024	768
Optimizer	Adam lr: $2e-05$	Adam lr: $2e-05$	Adam lr: $2e-05$
Warm-up	5%	5%	10%
Max length	512	512	512
Number of topics (Topic dims)	100	100	90 (Quora) 80 (MSRP, CQA B) 70 (CQA A, CQA C)
Pooling layer	Mean/Max	Mean/Max	Mean/Max
Number of epochs	3	3	6
Training task	Regression	Regression	Classification

The training is conducted on an NVIDIA Quadro RTX 8000 GPUs with a batch size of 32. We run over all epochs and do not use early stops in the training process. The results of each experiment are extracted by the best performance in development set. The number of epochs is installed separately for each experimental case. The details of our experiment settings are shown in Table 3.2. However, in each task, we also present the settings which are specific to the task's characteristics.

3.4.2 Semantic Similarity Detection

Datasets and Evaluation Metrics

In the evaluation of Semantic Similarity Detection, benchmarks datasets was formatted as a classification task. All experiments in this part are done in three benchmarks datasets including Quora, MSRP, and SemEval CQA with statistics information shown in Table 3.3. The number of classes in the classification model varies depending on the properties of each dataset and is displayed as follows:

- **Quora**: The duplicated questions dataset containing over 400,000 question pairs with two labels (duplicate: 1 or non-duplicate: 0). The mission is to determine whether a pair of question is duplicated or not. Therefore, it is considered as a classification task. The train/dev/test sets are divided according to [43].

- **MSRP**: The Microsoft Research Paraphrase [44] dataset that consists of over 5000 sentence pairs collected from news. The mission of this task is to determine whether each pair of sentence is paraphrased or not. In our experiment, we also consider it as a text classification task.

- **SemEval CQA**[2] [45–47]: The SemEval CQA is the combination between three subtasks A, B, and C whose questions and answers are extracted from on Qatar Living forum.

  - SemEval Subtask A (Question-Comment Similarity): Although the original configuration of this task is on the ranking problems to calculate the relevance between questions and comments, most previous works consider it as a binary classification task. In particular, a pair of question and comment is assigned to 1 if a comment is useful to answer the question and 0 and otherwise.

  - SemEval Subtask B (Question-Question Similarity): Subtask B gives a new question and the group of the ten most related questions, rerank the related queries based on how closely they resemble the original query. Similar subtask A, a pair of question and question is assigned to 1 if a question is related to the rest question and 0 and otherwise.

[2]https://alt.qcri.org/semeval2017/task3/

– SemEval Subtask C (Question-External Comment Similarity): Subtask C is the same subtask A. With a new question, 100 comments (each related question in subtask B has 10 comments) evaluate their relevance concerning the original question.

In this experiment, we consider both our models and competitive baselines in supervised learning. It means we only use the samples on the train set for optimizing and the ones on the dev set for validating the systems. The partition of training, development, and test set is introduced in the original papers. The evaluation of SSD is based on F1-score as classification tasks.

Table 3.3: The information of benchmark datasets

Dataset	#length	#samples	#topics
Quora	13	404000	90
MSRP	22	5000	80
SemEval CQA (A)	48	26000	70
SemEval CQA (B)	52	4000	80
SemEval CQA (C)	45	47000	70

Competitive Baselines and Our method

We quantitatively evaluate the SubTST with a number of competing approaches in Semantic Similarity Detection to demonstrate the effectiveness of our suggested models. List of comparative baselines:

- Sentence-BERT (SBERT) [3] provided a method to represent as dense vectors for sentences and paragraphs based on transformer-based language models. In this experiment, the Sentence-BERT is built based on the pre-trained BERT_{base} model.
- tBERT[4] model is a topic-informed BERT-based architecture for paraphrase detection. In this experiment, we only consider tBERT formed by BERT_{base} with LDA as the topic model. All results are extracted by the original version of tBERT instead of being reproduced.
- SwissAlps [48] is a popular approach built on a siamese CNN architecture and evaluated on the SemEval CQA.

[3]https://github.com/UKPLab/sentence-transformers
[4]https://github.com/wuningxi/tBERT

- KeLP [49] - the kernel-based text sequences pair modeling has been improved, which is often considered as the competitive baselines in previous works.

- StructBERT [50] is a extend of BERT by increases language structures information (the sequential order of words and sentences) in pre-training process. We choose report the result of StructBERT$_{base}$ in experiment.

- RealFormer [51] is a modification of the Transformer-XL model that has been pre-trained to learn bidirectional contexts by maximizing expected likelihood over all possible permutations of the input sequence factorization order.

- EFL [52] is a method that convert NLP task to an entailment task and apply few-shot learning.

In our approach, each input sentence is encoded by the BERT$_{base}$, and the LDA is used for learning latent topics. The best number of latent topics for each benchmark dataset is considered based on an analysis of the tBERT's works. The detail of topics' number k is shown in Table 3.3.

With the variants of our models, we also consider four kinds of models which are similar to the ones in the STS task. It includes the frozen topic embedding (i.e SubTST-mean, SubTST-max) and learnable topic embedding (i.e SubTST-mean-train topic, SubTST-max-train). As we mentioned above, in this task, all experiments are considered in classification tasks whose objective function is Cross-Entropy. Besides, all approaches are trained over 6 epochs and are evaluated by the best performance on the development set.

Results

Table 3.4 shows a comparison between our proposed method and competitive baseline systems. Overall, our SubTST models significantly outperform the previous works in most benchmark datasets. The experimental results prominently show the effectiveness of our proposed SubTST.

Firstly, as compared with Sentence-BERT (SBERT), our proposed model with additional topic information is more effective than the vanilla version. SubTST surpasses all datasets around 2% than SBERT. It proves that the topic information is actually efficient

Table 3.4: Results of methods on datasets: MSRP, Quora, SemEval based on F1-score

	MSRP	Quora	SemEval subtask A	SemEval subtask B	SemEval subtask C
Previous researches					
SwissAlps [48]	-	-	43.3	-	-
KeLP [49]	-	-	69.87	50.64	-
tBERT [28]	88.4	90.5	76.8	52.4	27.3
StrucBERT$_{base}$ [50]	89.9	72.0	-	-	-
RealFormer [51]	90.9	88.2	-	-	-
EFL [52]	**91.0**	89.2	-	-	-
Our implementation					
Sentence-BERT - mean	80.9	89.9	76.9	47.9	32.14
Sentence-BERT - max	80.1	88.4	77.0	33.9	31.89
SubTST - mean	79.0	90.1	76.5	**61.2**	32.28
SubTST - max	80.9	89.1	77.7	44.7	32.22
SubTST - mean - train topic	**82.3**	**90.7**	**77.8**	54.2	**32.58**
SubTST - max - train topic	81.1	89.0	77.2	45.7	32.04

in SSD tasks. With the contribution of topic-based features, our proposed embeddings and architectures are more potential to enhance the ability of text representation.

Secondly, for utilizing topic information into sentence embedding, our proposed SubTST also passes tBERT in most datasets. The success of our models comes from the efficacy of using sub-word latent topics and our delicate incorporation of topic information into each word. The experimental results in Table 3.4 showed that: (i) for the SemEval subtasks A, B datasets, SubTST and all its variants obtain the significant F1 score than the tBERT; (ii) with the Quora dataset, the SubTST with frozen topic-based embedding is competitive towards the tBERT, and the performance of our model is better than tBERT in the setting of the learnable topic-based embedding.

On the MSRP dataset, our proposed approach has a little worse performance than tBERT. In fact, the reason for this weakness is from the typical characteristics of MRSP dataset. In particular, MSRP dataset comprises a limited number of samples with a large number of named entities. We use Spacy - Name entity recognition[5] to count this number. The result shows that 5000 samples have 30194 name entities (average about 6 entities/sample). This result is bigger than CQA subtask B (about 3 entities/sample). The Table 3.5 shows several error analyses when predicting by SubTST. It is difficult to address this problem by using sub-word representation. With our consideration of sub-

[5]https://spacy.io/usage/linguistic-features

word in both topic-based and context-based representation, our proposed approach is less attentive to the Named-entity problem. However, our sub-word configuration is effective to overcome the explosion in word vocabulary. Another reason is the bi-encoder in our models. In the entire self-attention mechanism, the cross-encoder is typically better than the bi-encoder [53]. Nevertheless, the cross-encoder is too low in practical application. Therefore, it exists a trade-off between performance and complexity in our approach for MSRP dataset.

Table 3.5: The analysis about Name entities of MSRP datasets

Sample	Label	Predict	#tokens of entity
Sentence A: According to the federal Centers for Disease Control and Prevention (news - web sites), there were 19 reported cases of measles in the United States in 2002. **Sentence B**: The Centers for Disease Control and Prevention said there were 19 reported cases of measles in the United States in 2002.	1	0	20
Sentence A: Americans don't cut and run, we have to see this misadventure through," she said. **Sentence B**: She also pledged to bring peace to Iraq: "Americans don't cut and run, we have to see this misadventure through.	0	0	2

Finally, we also highlight the detailed observation of all approaches in five benchmark datasets. All results are extracted by the validation process in the development set. The details of our comparison are present in Figure 3.3 over 6 epochs. We found that the SubTST achieves the peak of the F1 score after 1 or 2 epochs. In the next epochs, the change of the F1 score tends to be monotonic with a small amplitude in comparison with tBERT. Hence, it shows the stability of our SubTST approaches in the training. It is resulted from unifying the lexicon unit for topic models and Transformer-based models.

3.4.3 Semantic Textual Similarity

Datasets and Evaluation Metric

Besides the above experiments, we also evaluate the performance of SubTST on the Semantic Textual Similarity (STS) task. Different from the SSD, the output of STS is the similarity score of each sentence pair. We compared our proposed approach with competitive baselines in benchmark datasets as follows:

- STS 2012-2016 tasks (STS-12, STS-13, STS-14, STS-15, STS-16) [54–58]: is a typical benchmark in Semantic Textual Similarity. It is used in the competition of

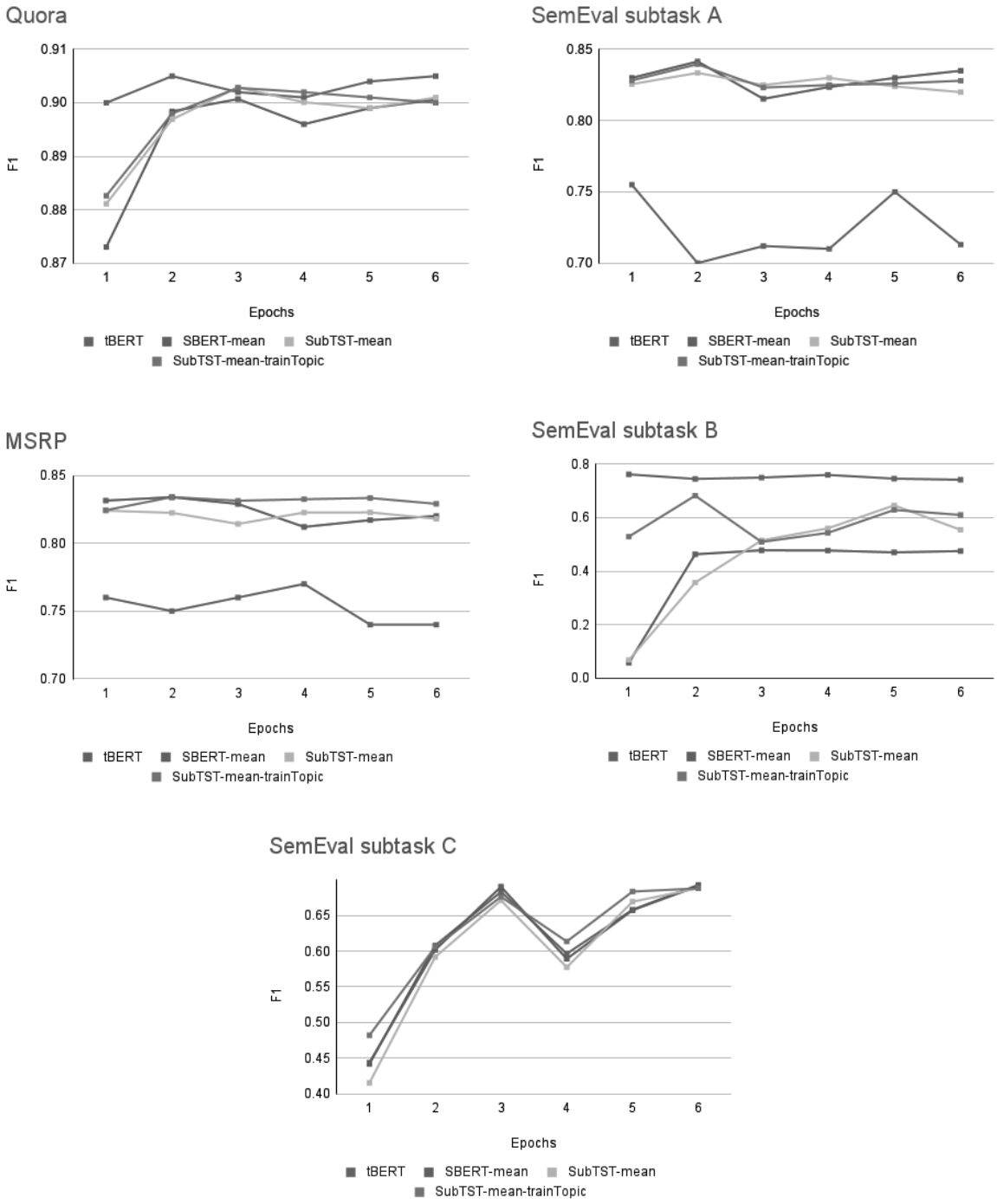


Figure 3.3: Performance of SubTST and baselines on dev set of the datasets

SemEval - International Workshop on Semantic Evaluation and updated gradually year by year.

- STS-benchmark (STS-B) [59]: STS-benchmark is the most important one where the samples are normalized and distributed carefully via the observation in previous years.

- SICK-Relatedness dataset [60]: was proposed in LREC 2014 by Marelli et al. It has several sentence pairs with a variety of lexical, syntactic, and semantic issues. The relatedness scores are numbers between 0 and 5. It was also utilized for SemEval 2014.

Overall, each sample in these datasets is a text sequences pair and a standard semantic similarity score. This score reflects the similarity between two input sentences. Each score is a decimal number from 0 to 5. The higher score it is, the more similar the sentences are. For consistent evaluation, the partition of train and test set is followed to previous works and competitions.

Following the previous works, the Spearman's rank correlation is used as the main evaluation metric. Spearman's rank correlation ρ is the most reasonable metric to evaluate methods on STS task. This measure is calculated using the cosine-similarity of the input sentence embedding and the gold similar scores, together with the strength and direction of the association. We use the version of Spearman's ρ in Spicy.

Competitive Baselines and Our method

To demonstrate the effectiveness of our suggested strategy, we contrast our model with industry benchmarks such as:

- *Avg GloVe* and *Avg BERT* are two approaches which also used word embeddings of pretrained language models to encode sentence. The sentence representation is calculated by the average vector of all words in the sentence. In our comparison, we use the results in paper of [4] instead of reproducing them.

- *BERT CLS-vector* [4] get the vector embedding of CLS-token to represent sentence embedding

- *Sentence-BERT* [4] was encode each sentence and interact weight by siamese network.

- *BERT-flow (NLI)* and *BERT-flow (target)* are two version of BERT-flow, which one using NLI (without supervision label), which one using all evaluation dataset (train + validation + test)

- *BERT-whitening (NLI)* and *BERT-whitening (target)* use BERT-whitening as base with two settings the same as BERT-flow

- *ConSERT* is a self-supervised framework for sentence embedding based on contrastive learning.

Similarity with Sentence-BERT, we pre-train SusTST on SNLI [61] and MultiNLI [62] dataset without using any STS specific training data. This process is similar to unsupervised evaluation in STS task. Therefore, we compare our methods with the approaches that are unsupervised STS.

As we mentioned above in Section3.3, the k value in Equation 3.1, number of latent topics, is an important hyper-parameter of topic model and depends on the dataset. Therefore, to choose this hyper-parameter, we do a tuning process of k in the range from $[20, 250]$ on the development set of STS-B. This is a summary of all datasets of STS and the evaluation of this dataset can apply to other STS datasets. With the result of tuning processing in Figure 3.4, we can observe that the value 100 obtains the most significant performance via the highest Spearman score. In general, we consider that the number of topics is 100 during the experiments of STS tasks. Besides, our model is trained over three epochs in the regression mode whose objective function is Mean Squared Error (MSE) in Equation 3.15.

Results

In both two configurations in BERT including BERT_{base} and BERT_{large}, our models prove their efficiency and strength against the other competitive baselines. In particular, in the benchmark dataset of SemEval competition, STS-B, our proposed approach achieves approximately 6% against Sentence-BERT (SBERT) [4]. Even that, we also outperform the competitive baselines in most datasets. This trend is similar in the SICK-R dataset.

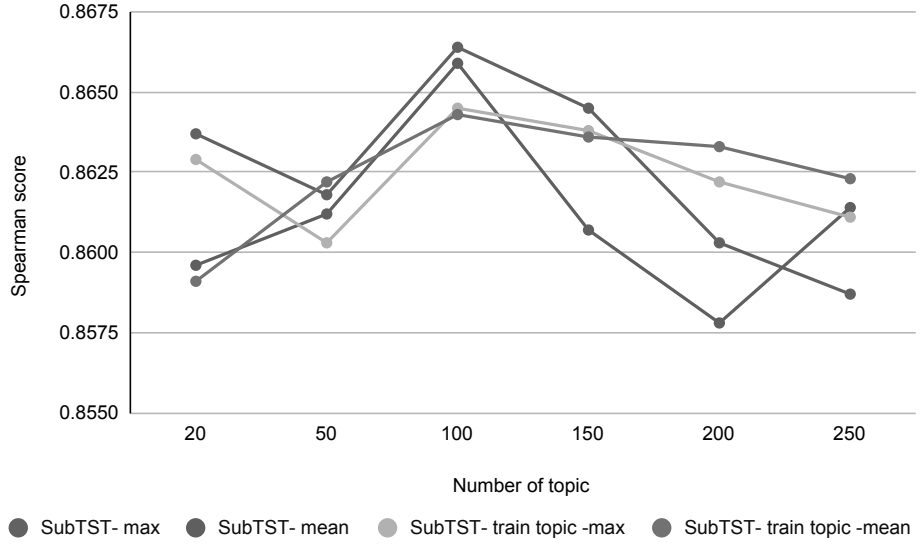


Figure 3.4: Tuning the hyper-parameter k - number of latent topics

In our comparison, we also point out the strength of our combination between topic information and context-based features from the language model. Firstly, with the same configuration, our sentence and word representation are better than ones from SBERT. As we mentioned above, SBERT whose sentence embedding only contains context-based features from BERT is quite similar to our model. The significant difference between our models and SBERT points out the contribution of our proposed integration. Obviously, with the effect of topic information, the text understanding in the sentence and word level achieves effective performance against the vanilla context-based features from SBERT.

Secondly, with the modification in the textual embedding, our model also obtains significant results against BERT-Flow and BERT-whitening. With approximately 10 percent of improvement in STS-B, our integration between topic-based and context-based is more meaningful. Even that, our method is more intelligible and natural than these previous works.

Finally, our models are also efficient against contrastive learning. Instead of utilizing contrastive learning such as ConSERT, our model still achieves significant performance with the traditional objective function. The only dataset where SubTST performs worse than ConSERT is STS-15. In STS-15, image description samples are added to the dataset. ConSERT is finetuned on various datasets including the random mix samples of seven datasets. In contrast, SubTST is only pre-trained on NLI data. SubTST, on the

Table 3.6: Results of an experiment on semantic text similarity with BERT_{base} (unsupervised; STS unlabeled texts). Numbers are showed as $\rho \times 100$.

Model	STS-B	STS-12	STS-13	STS-14	STS-15	STS-16	SICK-R
Previous research (Publicized)							
Reimers amd Gurevych (2019) [4]							
Avg. GloVe - 300 dims	58.02	55.14	70.66	59.73	68.25	63.66	53.76
Avg. BERT	46.35	38.78	57.98	57.98	63.15	61.06	58.40
BERT CLS-vector	16.50	20.16	30.01	20.09	36.88	38.03	42.63
Sentence-BERT	77.03	**70.97**	76.53	73.19	79.09	74.3	72.91
Li et al. (2020) [29]							
BERT-flow (NLI)	58.56	59.54	64.69	64.66	72.92	71.84	65.44
BERT-flow (target)	70.72	63.48	72.14	68.42	73.77	75.37	63.11
Su et al. (2021) [31]							
BERT-whitening (NLI)	67.51	61.46	66.71	66.17	74.82	72.10	64.90
BERT-whitening (target)	71.43	63.89	73.76	69.08	74.59	74.40	62.20
Yan et al. (2021) [32]							
ConSERT	73.97	64.64	78.49	69.07	**79.72**	75.95	67.31
Our implementation							
SubTST-max	83.15	65.34	78.47	**74.93**	78.97	78.87	83.22
SubTST-mean	83.12	64.52	77.98	74.79	78.14	79.58	83.07
SubTST-train topic-max	82.56	65.12	**79.49**	74.32	78.56	79.34	**83.30**
SubTST-train topic-mean	**83.16**	65.50	78.50	74.57	78.32	**79.76**	82.96

Table 3.7: Results of an experiment on semantic text similarity with BERT_{large} (unsupervised; STS unlabeled texts). Numbers are showed as $\rho \times 100$.

Model	STS-B	STS-12	STS-13	STS-14	STS-15	STS-16	SICK-R
Previous research (Publicized)							
Reimers and Gurevych (2019) [4]							
Sentence-BERT	79.23	**72.27**	78.46	74.90	80.99	76.25	73.75
Li et al. (2020) [29]							
BERT-flow (NLI)	68.09	61.72	66.05	66.34	74.87	74.47	64.62
BERT-flow (target)	72.26	65.20	73.39	69.42	74.92	77.63	62.50
Su et al. (2021) [31]							
BERT-whitening (NLI)	68.60	62.28	67.88	67.01	75.49	75.46	63.80
BERT-whitening (target)	72.48	64.34	74.60	69.64	74.68	75.90	60.80
Yan et al. (2021) [32]							
ConSERT	77.53	70.69	82.96	74.13	**82.78**	76.66	70.37
Our implementation							
SubTST-max	84.77	68.16	84.25	**78.23**	80.85	79.92	84.74
SubTST-mean	84.29	68.71	84.01	77.96	82.12	80.59	83.63
SubTST-train topic-max	**84.85**	67.58	84.57	78.16	81.77	80.65	**84.90**
SubTST-train topic-mean	84.81	68.74	**84.61**	77.98	81.62	**80.70**	84.01

other hand, outperforms ConSERT with an average relative performance improvement of around 9% among the six STS datasets.

To this end, the combination between topic information and linguistic features from language models is more informative and meaningful than the single one. It reveals that the topic information plays an important role in linguistic representation, especially via

our proposed integration.

3.4.4 Discussions

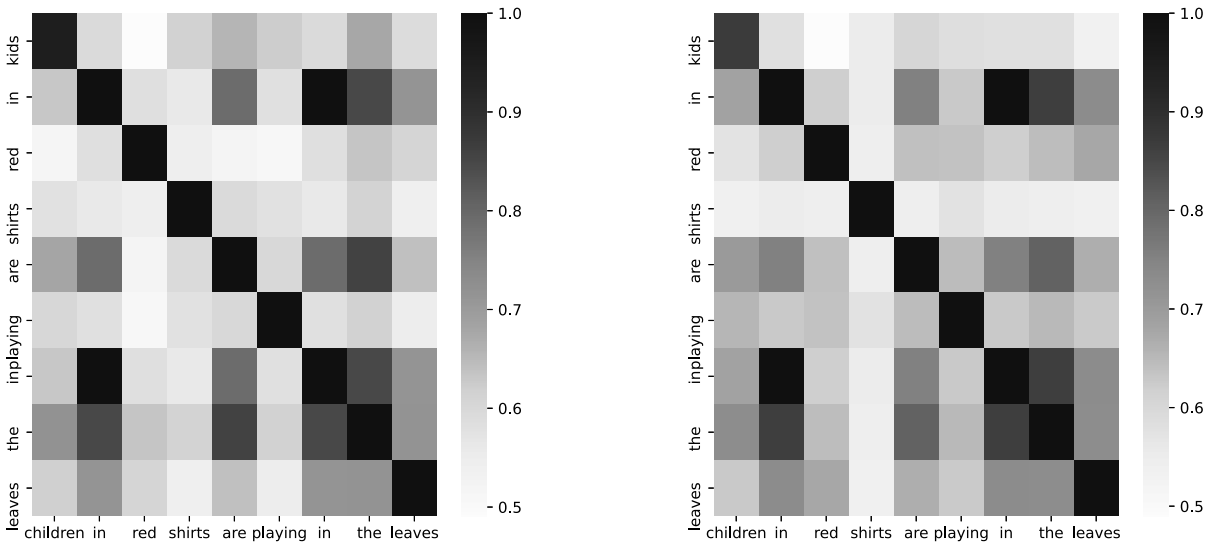
In experiments, it is easy to observe the strength of transformer models based on the bi-encoder with topic information based on sub-words. The interaction between topic information and context information brings effectiveness in semantic text understanding.

Firstly, in several previous works [29, 32], they discovered that BERT's word representation space is anisotropic. Low-frequency words are widely dispersed, while clusters of high-frequency terms surround the word's origin. High-frequency terms predominate sentence representations when token embeddings are averaged, which biases results against the genuine semantics of the sentences. Through added topic information, low-frequency words are more meaningful and are more useful to orient to the sentence meaning in the encoding process.

We assess the cosine similarity of tokens in a sentence pair to confirm the efficacy of topic information for anisotropic. Give a sentence pair in SICK data with the score is 4.7/5. Each token in a sentence is encoded by *Sentence-BERT* and *SubTST* and calculated cosine similarity with other tokens of the remaining sentence. We have two heatmaps as Figure 3.5. We can show the ability of topic information in the routing of the meaning sentence. The color of SubTST becomes stronger than SBERT. So that, we can define that the similarity becomes more clear when using topic information. The semantic relation between tokens is interpreted better in the sentence.

In some previous works such as Sentence-BERT [4], it is detailed that the complexity for finding the foremost comparable sentence match in a set of approx 11,000 text sequences is diminished from 66 hours with BERT to 5 seconds with Sentence-BERT for embedding and 0.01 seconds for computing cosine similarity. It proves that the bi-encoder models such as Sentence-BERT and our proposed SubTST models are highly ideal to put them into practice.

Normally, it is easy to understand the practical meaning of the latent topics over words/documents. However, topic modeling on sub-words can bring a lot of benefits instead of words/documents. The advantages of our consideration of sub-words are emphasized as follows:



SBERT SubTST

Figure 3.5: The representation of sentence pair base on cosine similarity. The BERT-base and MEAN-pooling are applying in this experiment.

- The model can reduce the number of unknown words (the out of vocabulary words) in the usage process. As using a topic model, the vocab of the topic model often fits with the size of the corpus. Moreover, when applying for another corpus, the number of unknown words could be very large. Using latent topics over sub-words sure significantly reduces "out of vocab".

- With transformer-based models such as BERT, sub-words are the fundamental component in processing. Besides, our consideration of sub-words in both of them is effective enough to easily integrate topic-based information into context-based features.

Certainly, based on the result of the experiments, we observe that the mean strategy is more appropriate for semantic similarity detection than the max strategy. It comes from the strength of mean pooling in sequential learning, which is also mentioned in many previous works.

3.5 Conclusion

Our proposed SubTST model is a novel approach for combining latent topic information with Transformer-based models. By integrating topic-based information into word representation, our model is effective to enhance the text representation via external features. With our delicate combination, our works propose a promising way to provide the topic information into context-based information from pre-trained language models. Besides, with the consideration of lexicon unit in both topic model and Transformer-based language models, our approach obtains the potential in both performance and complexity. In terms of semantic similarity, the experimental results reveal that our model outperforms all competitive baseline models. Furthermore, based on a bi-encoder architecture, our models offer greater benefits in practical applications with the speed-up enhancement against the cross-encoder approaches. Besides, the detailed discussion points out that our SubTST is relatively prompt to achieve peak performance and stability.

Chapter 4

Inside interaction: Topic based knowledge injection

4.1 Introductions

Automatic text summarization is the technique of efficiently extracting and compressing information from input papers while preserving their essential information. This method is crucial to the domains of several natural language processing (NLP) [63]. Currently, extractive and abstractive are two fundamental types of solutions for summarization [64]. The abstractive method creates unique words or phrases with comprehension, whereas the extractive method chooses important words, and sentences or rearranges words and sentences from the original document. The majority of methods in use today are usually designed to encrypt paragraphs and then decode them using a variety of processes. However, during the encoding and decoding procedures, there is a large amount of information loss. As a result, word embedding or contextual contents are the main focus of existing summarization research.

With the popularity of transformer-based models, the challenge of summarization is how to use the pre-trained transformer-based language model to represent and generate. It requires richer semantics information in the representation and training processes. The summary needs coherence and relatedness. And topic information injection is one of the solutions for this problem. The effectiveness by using latent topic information as features for information retrieval, recommendation system, and semantic textual similarity has

been pointed out in several studies [23], [24], [65], [26] and [27]. Topic models are more adept at picking up precise document semantics than transformers are; hence, they might be included in transformers to improve their performance even more.

In this work, we suggest a unique approach for adding topic information into BART models to improve their ability to do abstractive summarization. The method is called the tBART. The following are the main contributions of our work:

- The tBART essentially uses the BART architecture. In this method, the latent topics are learned over sub-words instead of documents/words as in previous work. In addition, we transform the representation vector generated by the topic model to convert to context space by an align function. The topic information is added in both encode and decode processes by a general topic distribution.

- We show that the suggested model significantly outperforms a number of earlier studies on the benchmark datasets: XSUM and CNN/DAILY MAIL

4.2 Related works

Recently, the encoder-decoder (or "Transformers") technique of sequence-to-sequence abstractive summarization has gained widespread recognition.

The BART model‘ [3] is a generalized pre-training model based on the Transformer model. Token masking, phrase permutation, document rotation, token deletion, and text infilling are five pre-training approaches that are introduced. Each of these methods uses a denoising autoencoder to add noise to the original text and then restore it. In BERT, tokens are randomly masked through token masking. The sentences in a document are randomly rearranged via sentence permutation. Document rotation rotates the text so that it starts with a token chosen at random from within it. Token deletion removes a token from the initial sentence at random. Text infilling puts a mask token into a randomly chosen position or replaces word sequences with a single mask token. The most accurate method is a combination of sentence permutation and text infilling. The decoder is an autoregressive model, whereas the encoder is a bidirectional model. This pre-trained BART model is tailored to a variety of tasks, including the summarizing task, for which the encoder receives a document and the decoder produces a summary of it.

A topic augmented decoder built on an RNN-based pointer-generator network was developed by See et al. [66] in 2017 and delivers a summary dependent on the input document and the latent subjects of the document. They find that latent themes reveal more general semantic information that can be used to influence the decoder's word-generation decisions.

The ability to reflect the background impact and the implicit information passed between texts is one of the main constraints of automatic summarization. As a general extractive and abstractive model for summarization, T-BERTSum [67] was proposed. To direct the acquisition of contextual information, this uses both the BERT architecture and topic data. The model demonstrates that topic embedding is combined to produce high-quality generation in a simple and efficient manner.

Moreover, Wang and his colleagues proposed the topic assistant model (TA) [68] for the transformer-based models. They used a topic model to learn latent semantics. The latent semantics are applied as an assistant model in the training process through three modules including Semantic-informed attention (SIA), Topic embedding with masked attention (TEMA), and Document-related modulation (DRM). Since TA is a plug-and-play model that does not alter the original Transformer network's structure, it is user-friendly and compatible with a variety of Transformer-based models. Transformer+TA can be readily fine-tuned by users using a pre-trained model; TA merely adds a few extra parameters.

Although these models have shown the benefits of merging topic models and S2S learning, incorporating topic data into Transformer-based summarization algorithms is still a relatively unexplored field of research.

4.3 Our approach

Topic knowledge is crucial for understanding texts, as we discussed previously. However, the essential question is how to incorporate the topical information into textual representation. We provide a novel method to gently include the sub-word subject information into Transformer-based language models in order to solve this issue. The figure 4.1 presents our architecture's specifics.

The tBART have three main components include:

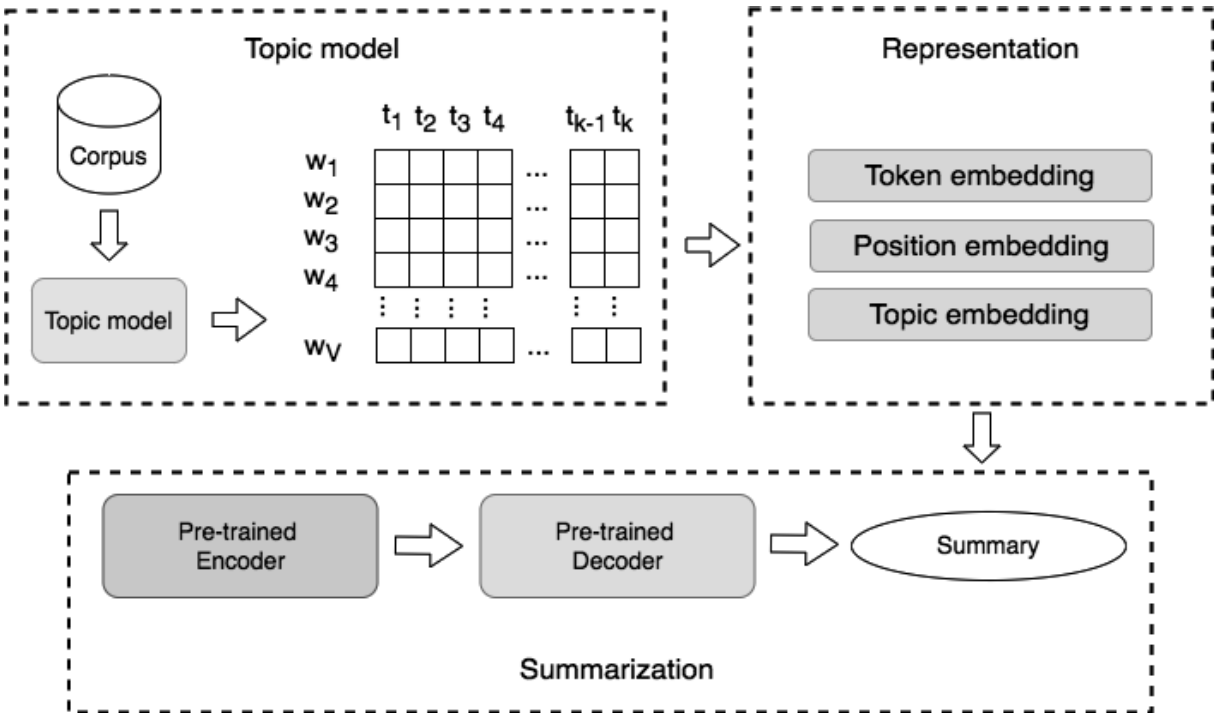


Figure 4.1: The architecture of tBART

- Topic model: It has the goal that learning the latent topics.
- Representation: It consists the embedding of context, position and topic.
- Summarization: It has an abstractive summarization based on the above two components.

4.3.1 Topic model component

The "Topic model" components is the pre-processing for tBART. To learn latent topics, we use a topic model as the core of component. A word-document matrix or a bag-of-words is frequently used as the input to a basic topic modeling method like LDA [33] or NMF [69] to express the relationship between words and documents. The representation is independent of the document's word order. Or, to put it another way, the document's words are interchangeable. Moreover, there is no relationship between the documents in a corpus; they are all independent. Latent topics on a corpus can be found based on statistical methods by looking at the words used in the original texts. Words and documents are represented by topic modeling's outputs in their own latent topic spaces. In this component, the output is the relation of word and latent topics. The representation

of output is the matrix $W \in R^{V \times k}$ with V - the size of vocabulary; k - the number of latent topics.

4.3.2 Representation component

To increase topic information, the modification of input embedding is necessary. In BART model, the input embedding include token embedding and position embedding. However, we specially add topic embedding in the representation of the input text. The "Representation" component is represent an input text $S = \{w_i\}_{i=1}^{n}$ with m - the internal hidden size of the transformer model n - the length of input text S. We have the token embedding and position embedding with input text S:

$$Token\ embedding = \{Ew_i\}_{i=1}^{n} \in R^{m \times n} \tag{4.1}$$

$$Position\ embedding = \{Ei\}_{i=1}^{n} \in R^{1 \times n} \tag{4.2}$$

Each sequence in the topic-based format is encoded into a topic space. The outcomes of the topic model frequently include (i) the relationship between vocabulary tokens and subjects and (ii) the relationship between learned corpus articles and topics. The relationship between vocabulary tokens and themes - W - is exploited in this study to encode input text. This decision was made after taking into account how topic information could improve the meaning of tokens. The topic information of each token is embedded into a vector whose dimension is the number of latent topics - k. Each input text is distinguished by a topic-term matrix of size $k \times n$, denoted by topic embedding where k is the number of latent topics and n is the number of tokens in each input text:

$$Et_i = W(w_i) \in R^{k} \tag{4.3}$$

$$Topic\ weight = \{Et_i\}_{i=1}^{n} \in R^{k \times n} \tag{4.4}$$

To have topic embedding, we apply an align function between token embedding and topic weight. With the align function, the interaction between topic and context becomes stronger. The token embedding is multiplied with topic weight to create the *Align weight* by Equation 4.5. The matrix shows the effect of the topic and context. It is the alignment

from context space to topic space and vice versa. After that, we convert *Topic weight* to context space base on the align matrix by Equation 4.6.

$$Align\ weight = TRANS(Token\ embedding) \times Topic\ weight \tag{4.5}$$

$$Topic\ embedding = Topic\ weight \times TRANS(Align\ weight) \tag{4.6}$$

where $TRANS$ is the transpose function.

So that, the representation of an input is show such as:

$$Input\ embed = \ Token\ embedding + Position\ embedding + Topic\ embedding \tag{4.7}$$

4.3.3 Summarization component

In this component, the BART is apply as the core of the component. In BART, the encoder is Bidirectional Encoder of BERT model‘ [1] and the decoder is Autoregressive Decoder of GPT model [70].

The BERT encoder outputs a vector comprising sentence-level information in addition to an embedding vector for each token in each text sequence in its input. By learning for both token- and sentence-level tasks in this way, the decoder becomes a solid starting point for any upcoming fine-tuning tasks. The previously stated and illustrated masked sequences are used for the pre-training. BART empowers the BERT encoder by using more difficult types of masking mechanisms in its pre-training while BERT was taught using a straightforward token masking technique. Each encoder layer has mask multi-head self attention layer and feed forward layer. After each step, layer norm was applied to normalize.

The GPT model's decoder utilized an architecture resembling that of the original Transformers' decoder section. GPT sequentially stacks 12 of these decoders such that changing the current token computation only affects prior tokens. Above is a picture of

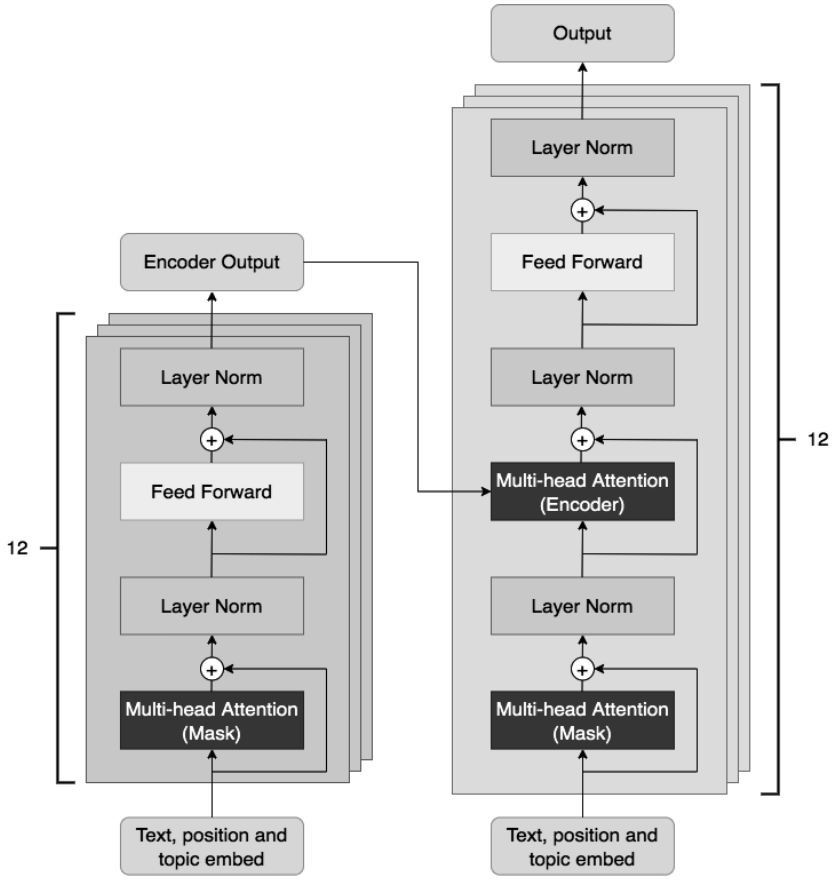


Figure 4.2: Encoder - Decoder architecture

the architecture. The GPT decoder also employs the masked multi-headed self-attention block and a feed-forward layer, as seen in the original Transformer decoder. The multi-head attention of the transformers is chosen to help the decoder learn the soft alignment between the summary document and the original document in order to successfully decode the sequence and more accurately capture the encoded information.

In some related research as Topic Assistant [68], the supporting topic information is learned from the original document in the encoder. After that, this information is presented by vector embedding and added to the decoder. However, we used a general topic space to represent topic information in our approach. Moreover, the topic information is added to both encoder and decoder. So that, the topic information of encoder and decoder are uniformity when the output of encoder was used in decoder. We discovered that the model can benefit from the knowledge communicated between these two jobs without significantly altering its architecture to give a more comprehensive sequence.

4.4 Experimental

4.4.1 Experimental setup

We evaluate the performance of tBART on two datasets XSUM ans CNN/Daily Mail with statistics information shown in Table 4.1.

- XSUM: It is a dataset for evaluating abstractive single-document summarizing methods. 226,711 news articles and a one-sentence summary make up the dataset. The articles span a wide range of topics and were compiled from BBC pieces published between 2010 and 2017.

- CNN/Daily Mail: The English-language CNN/DailyMail Dataset is made up of just over 300,000 unique news stories that were authored by reporters for CNN and the Daily Mail. Although the initial version was developed for automated reading, comprehension, and abstractive question answering, the current version supports both extractive and abstractive summarization.

Table 4.1: The information of benchmark datasets

Dataset	train/dev/test	#avg length of doc	#avg length of summary
XSUM	204045 11332 11334	431.07	23.26
CNN/Daily Mail	287113 13368 11490	781.6	55.6

We quantitatively compare the tBART with several previous methods based on the ROUGE score(ROUGE-1, ROUGE-2, ROUGE-L). The baselines include: Transformer‘ [71]; BART [3]; BERTSum [71]; PTGEN and PTGEN+Cov [66]; T-BERTSum [67]; BERTSum+TA and BART+TA [68].

We chose the pre-trained of BART includes (i) *facebook/bart-large-cnn* ;(ii) *facebook/bart-large-xsum* to apply for “Summarization” component. The LDA is used for learning latent topics, which is better than other topic models such as GSDMM [72] as mentioned in the study on tBERT [28], SubTST [65]. We set $k = 1$ for the number of latent topics. The greatest summary generation suggestion is when k=1. When K is greater than 1, the

model will become erratic; we have seen that word's capacity to express many themes is insufficient. Overall, though, many themes won't veer off-topic much, which is more like the summary document than the outcome of setting k to 0. Each word's probability distribution across topics was determined by us, and any new document can deduce its topic distribution.

4.4.2 Experimental results

We make a comparison between the proposed method and baseline systems that is shown the results in Table 4.2 and 4.3.

Table 4.2: Our approach - tBART on XSUM dataset with ROUGE score results

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	29.41	9.77	23.01
BART	45.14	22.27	37.25
BERTSum	38.81	16.50	31.27
PTGEN	29.70	9.21	23.24
PTGEN + Cov	28.10	8.02	21.72
T-BERTSum	39.90	17.48	32.18
BertSum + TA	39.77	17.39	32.39
BART + TA	45.76	22.68	38.03
tBART	**45.84**	**22.73**	**38.90**

Table 4.3: Our approach - tBART on CNN/Daily mail dataset with ROUGE score results

Methods	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	40.21	17.76	37.09
BART	44.16	21.28	40.90
BERTSum	42.13	19.60	39.18
PTGEN	36.44	15.66	33.42
PTGEN + Cov	39.53	17.28	36.38
T-BERTSum	42.12	20.45	39.74
BertSum + TA	43.06	20.58	39.67
BART + TA	44.47	21.39	41.32
tBART	**44.55**	**21.40**	**41.61**

As shown in Table 3.7 and 3.3, the first part of table is the baslines without topic information support. The second part is the baselines with topic information support.

The last part is our approach - tBART. Overall, the tBART significantly outperforms baseline models in XSUM and CNN/Daily Mail benchmark datasets. The experimental results prominently show the effectiveness of tBART.

The tBART model outperforms conventional transformer-based models in a variety of evaluation criteria, showing that the topic may effectively collect more important details and summarize reliable material without resorting to conventional methods. No matter how our model is compared to the baselines, the score demonstrates its superiority, which suggests the need for the theme to be introduced to direct the generation.

When compared with other models with topic information support, tBART also outperforms them. The additional topic information in representation was directed to semantics in sentences. The topic is raised for all encode and decode processes. It achieves much more efficiency than just using for decoder such as Topic assistant (+ TA). With T-BERTSum, our model is higher than 3 - 6 points on the ROUGE-1 score.

Table 4.4: Comparison of original document, gold summary and generated summaries of baselines and our approach

Gold summary	BART	tBART
youtube user serpentor filmed his feline friend in action footage shows the tabby producing bizarre noises as she petted the video has been seen many times.	a user filmed his feline friend in action footage shows the tabby pet producing a range of gurgling noises the show has been seen for more than 1600 times	the **youtube** user **serpentor** filmed his feline friend in action footage shows the tabby pet producing a range of gurgling noise the show has been seen for many times
A shot was reportedly fired at a car outside a primary school in Liverpool as parents were taking their children inside, police have said.	A man has been arrested on suspicion of attempted murder after a shot was fired at a car at a primary school in Liverpool.	A man has been arrested on suspicion of attempted murder after a shot was fired at a car **outside** a primary school in Liverpool

The Table 4.4 provides a few generated summaries by BART and tBART. As can be seen, topic information is used to generate some commonly overlooked words, such as “outside” and “youtube”. It demonstrated the value of subject knowledge during the generation process.

4.5 Conclusion

This chapter presents a new method for incorporating latent topic information with BART model, called the tBART. This method aims to add information and guide semantic meaning in the generation process. The experimental results show that our model outperforms all baseline models in summarization. Hence, this indicates the effectiveness of our proposed method. In addition, the tBART is built based on BART architecture, so it has more advantages in practical applications. Our work also reveals the effectiveness of latent topics in semantic tasks. In the future, we towards develop the latent topic online in the learning process and increase the quality of topic information for knowledge injection.

Chapter 5

Improving topic coherence and impact in the interaction

5.1 Introduction

To have a better way to manage and use large digital documents, it needs techniques to automatically discover, search, index the collections. Using probabilistic models and modern machine learning and statistics techniques, researchers developed methods for identifying word trends in document collections. These are referred to as "theme models." The collecting and analysis of news flow, recommend systems, and the identification of related stack overflow inquiries are all examples of applications for topic modeling. All of these focus on revealing the latent thematic structure in the text because it is thought that whatever text we produce, whether it be a tweet, post, or research paper, is composed of topics like sports, physics, aerospace, etc.

Topic modeling is a common issue in natural language processing. It converts a document into a list of general topics that appear in several documents. A popular text-mining technique for identifying hidden semantic patterns in text content is topic modeling. If a document contains a particular set of words, readers expect a corresponding topic and vice versa. With a basic topic modeling method as LDA or NMF, the input of the model often uses Bag-of-word to represent the relationship between word and document such as a word-document matrix. The representation does not depend on the order of words in the document. Or, to put it another way, the document's words are interchangeable.

Moreover, there is no relationship between the documents in a corpus; they are all independent. Based on statistical algorithms, latent topics on a corpus can be discovered by analyzing the words of the original texts. The outputs of topic modeling represent words and documents in their own latent topic space.

With the expansion of social networks, we can extract latent information by applying various text mining techniques from the vast number of instructive posts, comments, and questions. In fact, this data source consists of a large number of short texts that raises a big challenge for mining. In short texts, each sentence just has few words that make existing statistical machine learning methods for natural language processing (NLP) become ineffective because of the ambiguity induced by the less information of word co-occurrence.

After years, there are several methods for extracting latent topics from the text such as Latent Dirichlet Allocation (LDA) [73], non-negative matrix factorization (NMF) [69], Pseudo-document-based Topic Model (PTM) [74], and GPUDMM [75]. In topic modeling, the number of words in a document strongly affects the performance of the models. The above methods can perform well on long texts, however, the performance significantly decreases on the short text. Thus, we can find an supporting method to improve the topic models.

In this study, we offer the SupLeT, a novel strategy for improving latent topics retrieved by the topic model that combines distance metric learning (DML) [76, 77]. This method helps to significantly improve the topic coherence and document classification accuracy.

The following are the main contributions of our work in this chapter:

- Without considering the link between words, topic models using corpora frequently produce incoherent topics. We are interested in coming up with a topic model technique that could mine coherent themes by taking word vectors into account in the relationships between words and between words and documents. As you know, a specific type of soft-clustering model is known as a topic model [78]. So, as support approaches, we can use contributions pertaining to cluster validation for topic models. We aim to keep certain words on the same topic as close as possible while trying to keep others as far away as possible by using the ability of distance

metric learning (DML). The most similar words to the acquired latent subjects so better illustrate their coherence.

- Our experimental results, which we present in this chapter, demonstrate that the suggested approach significantly outperforms baselines in terms of (i) topic coherence and (ii) the usefulness of topic-based representation for document classification and semantic similarity detection on short text datasets. Moreover, we evaluate the effectiveness of topic coherence in topic information interaction.

The chapter is organized as follows. The related researches about topic model and distance metric learning are shown in Section 5.2. Section 5.3 briefly introduces SeaNMF, NMF, and Distance metric learning. Section 5.4 presents our approach named SupLeT. We demonstrate our experiments in Section 5.5. Finally, we conclude our work in Section 5.6.

5.2 Related Works

5.2.1 Topic modeling

One of the popular methods for topic modeling is Latent Dirichlet Allocation (LDA) [73] that is based on generative probabilistic models. A fundamental premise of LDA is that a document was created by selecting a number of subjects, and then selecting a number of words for each topic.

Another method of LDA is LeadLDA [79] - a topic modeling method for microblog posts. LeadLDA converts microblog posts to as conversation tree to increase context information and reduce the sparse data problem. In specific, the model extracts main messages - which start a topic(i.e, key aspects of previously focused topics, new topic) in conversation, called lead message and follower messages - which do not give the new topic, only echo topic(i.e, reply post, repost post). The model has difficulty with the ambiguity caused by the little information from the appearance.

In recent years, Deep learning was used as the approach for topic model. Approaches using it in topic model had good results as neural topic models(NTM) [34]. With this model, backpropagation training is possible within the context of neural variational infer-

ence. Additionally, using a stick-breaking structure, we suggest a recurrent network that is comparable to Bayesian non-parametric topic models in that it can find a notionally unlimited number of topics. In 2019, Adversarial-neural Topic Model(ATM) [35] is the first time adversarial training for topic modeling. This model tried to capture the semantic patterns among latent topics by the generator network and discriminator network.

Some techniques used topic model variations for short text data to mitigate the consequences of topic modeling's sparsity problems. Biterm topic model (BTM) [80] is a method that learn topic based on a model about the generation of term co-occurrence patterns in the corpus. In order to address the issue of sparse word co-occurrence patterns at the document level, the advanced BTM employs an explicit model of word co-occurrence and aggregated patterns in the entire corpus. Next then, Variational Auto-Encoder Topic Model (VAETM for short) [81] is propose as solution for short text. The model used large-scale information to combine the word embedding and entity embedding as input of model. This combination tried reduce the lack of word co-occurrence patterns when apply transitional methods for short text.

In another way, Non-negative matrix factorization(NMF) [36,37] is an interesting solution for topic modeling. It is a method fit for short text datasets. From the perspectives of consistency across several runs and early empirical convergence, this technique has many practical benefits. Xiaohui et al. [38] used a factorized symmetric term correlation matrix for topic modeling. This approach aims to teach subjects by studying the concept of correlation data. The method computed term correlation in short texts by representing each term with its co-occurring terms in order to derive accurate topics from term correlation data. The topic learning problem on the concept of a correlation matrix was then developed utilizing symmetric non-negative matrix factorization. However, the model is not reliable and stable. The SeaNMF model [69] was proposed to learn topics from the short text in WWW 2018. The model combines document-word relation and word-context relation(semantics relation) as inputs. This relation build by the skip-gram view on the corpus. The model is solved using a block coordinate descent algorithm. It is effective to solve topic model problems for sparse data as short text.

5.2.2 Distance metric learning

Over last years, distance metric learning robustness effect for many pattern recognition problems. The idea of distance metric learning is to use the distance between samples to improve the performance of learning methods. One of its applications is to improve nearest neighbors classifiers (k-nearest neighbors classification). The most popular algorithms are LMNN and NCA. In 2005, Neighborhood Component Analysis(NCA) [82] was introduced by Jacob Goldberger. The k-Nearest Neighbor classification algorithm uses the Mahalanobis distance measure, which can be learned using the NCA technique. In order to reduce the leave-one-out error anticipated by the nearest neighbor classification, it attempts to learn a linear transformation. Our classification model, in contrast to existing approaches, is non-parametric and makes no assumptions on the structure of the class distributions or the borders between them.

Another distance metric learning approach with the explicit goal of making closest neighbors classifiers more accurate is called Large margin nearest neighbor(LMNN) [76, 77]. This method's goal is to optimize number of nearest neighbors has the same class and try to keep samples from different class as far away as possible by a large margin. The learning process uses 2 penalties(Pull and Push) for samples in the local neighborhood.

5.3 Background

5.3.1 Non-negative Matrix Factorization - NMF

The process of non-negative matrix factorization involves splitting the original matrix into two smaller matrices, with the advantage that none of the three matrices include any negative elements. It is helpful while evaluating high-dimensional data objects. The NMF model in topic models is on par with the LDA model in terms of generative probabilistic modeling. With a group of documents that has N documents and the number of terms in vocabulary is M, we will have a term-document matrix A. The column of A showed a bag of terms and a document on vocabulary. Using NMF for this matrix A was built for two output matrices W, H. Matrix A. approximates the product of these two matrices. Objective function of NMF can formula such as:

$$O_{NMF} = min_{W,H \geq 0} ||A - WH^T||_F^2 \tag{5.1}$$

More detail, matrix A has size M *words* $\times$ N *documents*. We have two matrices after factoring with the K subject. The word distribution in the topic is represented by the matrix W. Each column represents the presence of a vocabulary topic. The size of W is M *words* $\times$ K *topic*. The subject distribution in documents is displayed in the matrix H. Each row represents the document's latent topic space. Size of H is N *documents* $\times$ K *topic*.

5.3.2 Semantics-assisted NMF - SeaNMF

SeaNMF [69] is a non-negative matrix factorization-based model for extracting ideas from short texts. In order to use the information in its learning process, SeaNMF increased the semantic information. Pointwise mutual information (PMI) [83] is explained by the representation of semantic information. The term-document matrix A and the semantic correlation matrix S were used as the model's inputs by SeaNMF. The link between keywords and their contexts is displayed in the matrix S. (word-word relation). The SeaNMF model has factorized to produce three output matrices including: W, W_c, and H, given input matrices and the K number of topics. The word distribution in the topic is represented by the matrix W. Each column represents the presence of a vocabulary topic. The distribution subject in documents is displayed in the matrix H. Each row represents the document's latent topic space. There is a new output matrix in SeaNMF called W_c. The word in semantics context is represented by the matrix W_c.

5.3.3 Distance metric learning

Over the last years, distance metric learning affected many pattern recognition problems. The idea of distance metric learning is to use the distance between samples to improve the performance of learning methods. One of its applications is to improve nearest neighbors classifiers (k-nearest neighbors classification) [84]. The most popular algorithms are LMNN and NCA.

Large margin nearest neighbor - LMNN

One of the most popular Mahalanobis distance learning techniques [85, 86] is the large margin nearest neighbor, or LMNN [77]. The approach was made to function with nearest neighbor classifiers. The effectiveness of the nearest neighbor classifier may be enhanced. The foundation of LMNN is the notion that samples' labels will be more trusted if their closest neighbors share those labels.

Give a set of samples: $X = \{x_1, x_2, x_3, ..., x_n\}$ and their labels: $Y = \{y_1, y_2, y_3, ..., y_n\}$. Consider three samples x_i, x_j, x_k: x_j is target neighbor of x_i, x_k is impostor.

$S = \{(x_i, x_j) : y_i = y_j;\ x_j\ is\ neighbor\ of\ x_i\}$

$R = \{(x_i, x_k) : y_k \neq y_i;\ x_k\ is\ neighbor\ of\ x_i\}$

The distance between each sample in dataset X is used to generate a perimeter after the target neighbor has been determined. In this perimeter, there was no sample difference label, therefore LMNN attempted to learn a distance. As a result, a margin is created using the perimeter's radius. Any sample from another class that crosses this line is referred to as a *impostor*. Now, LMNN moves the target neighbor closer while attempting to keep imposters at a minimum distance.

Two penalties are used by LMNN during the learning process. The first one penalizes distant target neighbors (ε_{pull}) and the second one penalizes nearby impostors(ε_{push}).

Combining the two penalties mentioned above with the parameter t, which manages the "pull/push" trade-off, yields the LMNN objective function:

$$O_{LMNN} = \min \{(1-t)\varepsilon_{pull} + t\varepsilon_{push}\} \quad t \in [0, 1] \tag{5.2}$$

Neighborhood Components Analysis - NCA

In 2005, Neighborhood Component Analysis(NCA) [82] was introduced by Goldberger. With the intention of minimizing the leave-one-out error anticipated by the nearest neighbor classification, NCA seeks to learn a linear transformation. Our classification model, in contrast to existing approaches, is non-parametric and makes no assumptions on the structure of the class distributions or the borders between them.

They use the decomposition $M = L^T L$ and define the probability p_{ij} that x_i is the

neighbor of x_j by calculating the softmax likelihood of the Mahalanobis distance:

$$p_{ij} = \frac{exp(-||Lx_i - Lx_j||_2^2)}{\sum_{l \neq i} exp(-||Lx_i - Lx_j||_2^2)} \quad p_{ii} = 0 \tag{5.3}$$

The stochastic nearest neighbors rule's likelihood that x_i would be correctly classified is then:

$$p_i = \sum_{j:j \neq i, y_j = y_i} p_{ij} \tag{5.4}$$

Finding the matrix L that optimizes the total likelihood of being properly classified is the goal of optimization.

$$L = argmax \sum_i p_i \tag{5.5}$$

5.4 Support learning for topic modeling

In this section, we put up a fresh idea for raising the standard of the lesson learnt. The strategy is built on distance metric learning's capacity to support the model.

5.4.1 The general idea

As mentioned in the "Background" section, topic models typically employ an unsupervised approach to effectively learn latent topics. To improve quality, we want to continuously improve the latent topics they have learned. Our method was built base on the idea of topic models which use the factorization method. Thus, in this research, we focus to develop 2 methods: non-negative matrix factorization(NMF) and semantics-assisted non-negative matrix factorization(SeaNMF).

Note that, NMF uses the term-document matrix A as the input of the model. Additionally, SeaNMF uses two matrices as its input: the semantic matrix S and the word-document matrix A. Bag-of-words was employed by word-document matrix A to depict the relationship between word and document. By calculating PMI, a measure of association, the semantic matrix S was created. Two matrices are initialized using a corpus as their foundation. However, an assumption is given: inputs matrices are not the most optimal when learning latent topics. So, we need to define a transformation f which will create appropriate inputs for topic modeling.

$$f(X) \rightarrow X' \tag{5.6}$$

To create new input matrices that have high quality, the method needs a benchmark for the learning process. Normally, the learning methods use object function (unsupervised learning) or label (supervised learning) to do it. With the SeaNMF model, the model used a new objective function which is built by the objective function of the NMF model. Most methods for topic modeling are unsupervised learning methods. Some cases learned topic based on label class of document classification problem. From this, we proposed to choose an object to be the label to learn. The difference from the label of the normal topic model, latent topics are used as goal labels in our method. The correct learned topics support fixing the wrong learned topics.

With the above ideas, we propose a method that incorporates the topic model with distance metric learning (DML) for topics refinement. The purpose of this approach is that: (i) for each word, assign the most likely topic determined by the topic model, which is referred to as a soft label for such a word; (ii) with soft labels and DML, learns a transformation f to update input matrices of topic modeling method. As a result, the new input carried over the substance of the latent topic from the previous phase after being refined. Next section, we will discuss this in more detail.

5.4.2 The proposed method

Overview of proposed method

The proposed method is presented in Figure 5.1 and Figure 5.2. Our model consists of three main components: Topic learning, Pooling, and Distance metric learning. The first is "Topic learning". Its function is to learn latent topics from corpus by a topic model (eg. NMF model and SeaNMF model - which were introduced in section 5.3). Based on the results of "Topic learning", the "Pooling" continue do its mission. In this component, we extract temporary labels for refinement. The method that learns latent topics often is the unsupervised learning method. It does not have the labels in the training data. As a result, labels are needed to be the basis for the next component. Finally,the "Distance metric learning" component is to base on learned latent topics to optimize input matrices

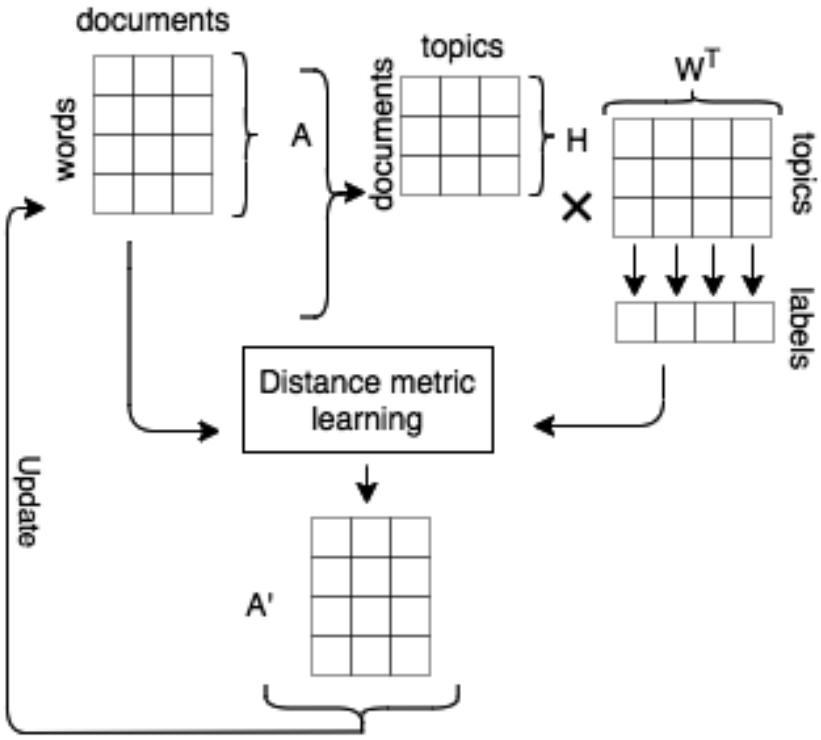


Figure 5.1: Support learning for Topic Model with the standard NMF model

in refinement process. After learning by three components, SupLeT finished after T time steps.

Topic learning component

The "Topic learning" component is to apply a topic modeling for a corpus to learn latent topics. With the NMF model, the input is a term-documents matrix A. As the above introduction, the matrix A represents the relation of word and document by bag-of-words algorithm. Another example is the SeaNMF model. With a corpus, the term-document matrix A and semantic matrix S were built as the input of the SeaNMF model. To improve the NMF model, the semantic information is added in the SeaNMF model by the matrix S. The skip-gram model on the corpus is used to learn the semantic relationships between words and their situations. It was shown that the skip-gram is useful for identifying word semantic links and fitting factorization techniques. Matrix S is the result of this component. In this component, topic modeling receives input matrices to generate the output. The relationships word-topic and document-topic were represented by the output. We can analyze an example of the SeaNMF model. With two input matrices A and S,

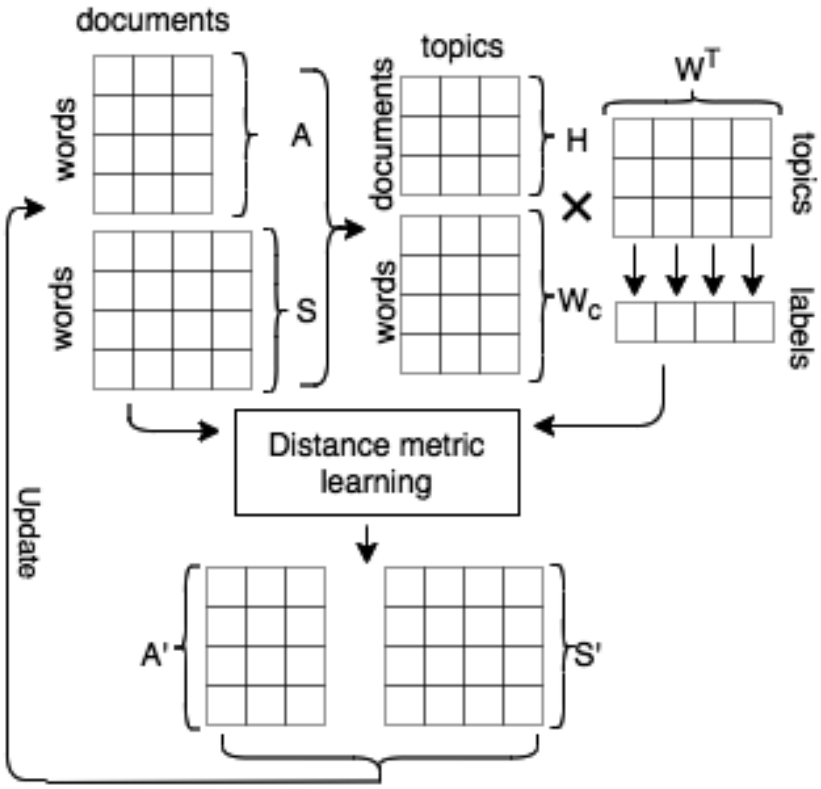


Figure 5.2: Support learning for Topic Model with the standard SeaNMF model

SeaNMF model divided into three lower-rank matrices - W (word-topic), W_c (context-topic) and H (document-topic). When starting, matrices W, W_c, and H used initialize by random non-negative real numbers. And they are updated in each iteration. The number of iteration T' set up by the user. The final state of lower-rank matrices determined when the objective function converges or the number of iteration get over T'.

Pooling component

In supervised topic modeling approaches, goal labels of documents in document classification problem often choose to use in the learning process. The labels can be considered "topic" at a high-level because the number of classes in document classification is not so much. In each this class, there is a lot of sub-class (class in low-level). The sub-class is hidden under high-level topics and quite difficult to find them. In our research, we use sub-class as the label in the learning process. In other words, the latent topic learned by topic modeling is chosen to become a label in the SupLeT model.

As we see, the relationship between words, documents, and latent topics is a result of

topic modeling. Thus, one of the output matrices of the SeaNMF model is the source to extract labels. We used a term-topic matrix W as an input of the "Pooling" component. In this component, the goal is to extract labels for the next component. The idea of this component is that used learned topics to improve the topics in the refinement process. Therefore, we use the fittest latent topic for each word as the label and called with the name "Soft label".

Let W be a representation matrix of the relationship between word and topic. Each row of matrix W represents the probability of a word with K topic in latent topic space. So, a word was represented by a vector K-dimension. $i-th$ elements in vector show the connection of this word with $i-th$ topic. If $i-th$ element is higher than $j-th$ element, the word represents for $i-th$ topic better than $j-th$ topic. Based on this character, a soft label can be defined as follow:

$$\begin{aligned} &Label\ of\ W(row,:) = Topic\ k \\ &\qquad if\ value\ W(row,k)\ is\ maximize\ in\ W(row,:) \\ &\qquad 0 \leq k < K \end{aligned}$$

For example: given a matrix term-topic W with number latent topic is 3. So, the label set contains three elements as follow: 0,1 and 2 fit with numerical order column in matrix W. Each row in W matrix is a word in the vocabulary and we can extract the word's label as follow:

$$W = \begin{bmatrix} 0.024 & \mathbf{0.0265} & 0.0153 \\ \mathbf{0.209} & 0.0214 & 0.0245 \\ 0.019 & 0.0223 & \mathbf{0.25} \\ ... & & \\ 0.00226 & \mathbf{0.224} & 0.0256 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0 \\ 2 \\ ... \\ 1 \end{bmatrix}$$

However, those labels are not permanent. They were created by the topic model in the "Topic learning" component. And, they change after each time running this model. In the SupLeT model, a time step finishes when all components ("Topic learning", "Pooling" and "Distance metric learning") complete running time. Soft labels only exist in a time step and continuity changes in the next time step.

Distance metric learning component

After running the "Pooling" component, every word in the vocabulary has an associated soft label. With these soft labels, the "Distance metric learning" component can action. The goal of this component is to transform two matrices A and S with the SeaNMF model and matrix A with the NMF model for fitting into topic modeling. We continue with an example about the SeaNMF model.

As introduced in section 5.3, distance metric learning is an approach to develop learning methods based on learning distance. The methods following this approach are very diverse and abundant such as Neighborhood Component Analysis - NCA and Large margin nearest neighbor - LMNN. Each method fit to increase the quality of the type of learning method. For example, NCA and LMNN often use for clustering problems. In some surveys, document clustering and topic modeling are two closely related tasks. The approach of topic modeling also is quite similar to soft clustering. Thus, we choose a learning distance method as the Large margin nearest neighbor(LMNN). LMNN is an approach driven by the nearest neighbor to improve the performance of clustering and classification [87]. So, LMNN can support the topic modeling method to increase quality.

In "Large margin nearest neighbor" section, LMNN was introduced that it uses input include: dataset $X = \{x_1, x_2, x_3, ..., x_n\}$ and their labels: $Y = \{y_1, y_2, y_3, ..., y_n\}$. When we use our methodology, dataset X is the corpus's vocabulary, and each word is a sample. Each sample in the collection does, however, have two representations, which correspond to two matrices A and S:

- With term-document matrix A: It defined as a matrix that show the relationship of word and document. A sample(a word) is represented by a vector N-dimensions. Note that, N is the number of documents in the corpus.

- With semantic correlations matrix S: It defined as a matrix that show the relationship of word and their contexts. A sample(a word) is represented by a vector M-dimensions. Note that, M is the size of the vocabulary.

And their set labels Y is soft label obtain soon the result of the "Pooling" component. With dataset X and their labels Y, the k-nearest neighbor algorithm was applied. At the same time, the object *impostor* and $targetneighbor$ also determine. In this case, the

target neighbor is the documents that have the same latent topic and cluster. And, an impostor is the documents which have the difference latent topic and in a cluster. A document can either an be an impostor in a cluster or a target neighbor in other clusters. When we had full things needed, the learning process of "Distance metric learning" can start. Given a pair of words (w_i, w_j), representation vector of d_i is $\vec{w_i}$, w_j is $\vec{w_j}$. The distance between word w_i and word w_j is calculated based on the Mahalanobis distance as shown in (7):

$$d_M(w_i, w_j) = \sqrt{(\vec{w_i} - \vec{w_j})^T Q(\vec{w_i} - \vec{w_j})} \tag{5.7}$$

The value of $\vec{w}$ is extracted from term-document matrix A or semantic matrix S. The penalizes (ε_{pull}) and (ε_{push}) for impostor and target neighbor are calculated such as equation (8) and equation (9).

$$\varepsilon_{pull} = \sum_{x_i, x_j \in S} d_M^2(x_i, x_j) \tag{5.8}$$

$$\varepsilon_{push} = \sum_{x_i, x_j, x_k \in R} [1 + d_M^2(x_i, x_j) - d_M^2(x_i, x_k)]_+ \tag{5.9}$$

After that, matrix Q is determined by the objective function of LMNN:

$$\min \{(1-t)\varepsilon_{pull} + t\varepsilon_{push}\} \quad t \in [0, 1] \tag{5.10}$$

After finding matrix Q, transformation matrix L which use to transform object is determine based on equation $Q = L^T L$. We used two times LMNN to find two transformation matrices for A and S. The updated of matrix A - A' and matrix S - S' transformed into metric space by:

$$A' = AL_1^T \tag{5.11}$$

$$S' = SL_2^T \tag{5.12}$$

In the next time step, the new matrix A' and S' will be used as input of the SeaNMF model. And the condition about non-negative also checks with A' and S'. With size of A' is M *words* $\times$ N *documents*, S' is M *words* $\times$ M *words*:

$$A'(i,j) = \begin{cases} 0 \Leftrightarrow A'(i,j) < 0 \\ A'(i,j) \Leftrightarrow A'(i,j) \geq 0 \end{cases} \quad 0 \leq i < M; 0 \leq j < N \tag{5.13}$$

$$S'(i,j) = \begin{cases} 0 \Leftrightarrow S'(i,j) < 0 \\ \\ S'(i,j) \Leftrightarrow S'(i,j) \geq 0 \end{cases} \qquad 0 \leq i,j < M \tag{5.14}$$

A time step in a loop comes to an end here. The procedure employs time steps of T.

However, the best state may be not the last time step. With SupLeT, we set a condition to show the best state of model: use measure evaluate of topic model *Topic coherence*. Topic coherence [88] is a popular measurement used to evaluate topic models. In the "Experiments" section, we will introduce it in more detail. When SupLeT runs, each time steps created output matrices W, W_c and H. Based on matrix W and co-occurrence matrix, topic coherence was calculated by the average of PMI of latent topics after running SeaNMF. And it determined the best state in a loop: the time step which has the maximum topic coherence score is the best state. Matrices output of this time step will be saved to use for other tasks. We used both of two ways (T time steps and the best state) to refinement topic modeling in 5.5 section.

5.5 Experiments

5.5.1 Topic coherence

Datasets

All experiments were conducted with the benchmark datasets as reference [69, 89–91]. The datasets we used include:

- **TagNews**[1]: It is a component of the TagMyNews dataset. It is news that has been culled from popular newspaper websites' RSS feeds. Sport, Business, Entertainment, US, World, Health, and Science & Technology are the categories.

- **Question 2002**[2]: The learning question classification studies used this dataset by Xin Li, Dan Roth [90].

- **StackOverflow**[3]: The dataset used by Jiaming Xu et al. [89] in VSM-NLP workshop NAACL 2015. It is questioned in StackOverflow from July 31st to August

[1]https://github.com/isthegeek/News-Classification
[2]https://cogcomp.org/Data/QA/QC/
[3]https://github.com/jacoxu/StackOverflow?

14th, 2012.

- **Yahoo**: This dataset is a part of Yahoo dataset for research - Yahoo! Answers Manner Questions, version 2.0[4]. The questions and answers posted to Yahoo! Answers are all accessible to any web user who wants to peruse or download them. A portion of the Yahoo! Answers corpus makes up the data that we have gathered. In our dataset, we divide the question and answer into two corpora: Yahoo questions and Yahoo answers. Each pair question-answer has subject from ten different categories.

- **Yelp review**[5]: It is a subset of the reviews on Yelp. It was initially created for the Yelp Dataset Challenge, which gives students the opportunity to explore and share their findings after conducting research or analysis on Yelp's data. The label of each sample is 1,2,3,4 and 5 for sentiment analysis.

- **MSRP**[6]: The Microsoft Research dataset for sentences with two labels each contains sentences from news stories in pairs.

- **Quora**[7]: The dataset for Quora question pairs consists of question pairs with two labels. The objective is to determine what pair.

Evaluation metrics

To assess the effectiveness of our model, we used experiments to evaluate the following points: (i) the coherence of the topics; (ii) the efficiency of topic-based representation for categorizing documents and identifying semantic similarities.

In the first experiment, a measure of topic coherence used to represent the coherence of the topic in terms of their interpretability. A common approach of evaluation for topic models is topic coherence [88]. This measurement is word-based.The overall concept is to show the top-ranked subject words to human annotators who label topics with coherence scores after using the input from automatic coherence calculation methods. Topic coherence score calculated by the average/median of the pairwise word-similarity

[4]https://webscope.sandbox.yahoo.com/catalog.php?datatype=1
[5]https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset
[6]https://www.microsoft.com/en-us/download/details.aspx?id=52398
[7]https://www.kaggle.com/c/quora-question-pairs

scores of the top-ranked words (typically 5 or 10) in the topic. Normally, pointwise mutual information (PMI) often used to calculate topic coherence.

We consider an equation to calculate topic coherence. With a topic, k, the equation used to calculate topic coherence of topic k is :

$$TC_k = \frac{2}{n(n-1)} \sum_{1 \leq i \leq j \leq n} log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{5.15}$$

where n is top-n words in topic k. $p(w_i, w_j)$ is the probability of word w_i, w_j co-occurring. $p(w_i)$ and $p(w_j)$ is marginal probability of w_i, w_j. The methods used the average PMI on all topics to evaluate.

$$Topic\ Coherence = \frac{\sum_{k=1}^{K} TC_k}{K} \tag{5.16}$$

Coherent topics, or subjects with high topic coherence scores, are produced by a good model. T.Shi [69] compares SeaNMF with other cutting-edge models using these metrics.

Besides, document classification and semantic similarity detection can be used to evaluate the topic model. The goal is to consider the effect of the learned topic on the performance of these tasks. With document classification, we only use the latent topic as a feature of the classification model. The difference from the first experiment, all elements of the dataset used, include content and label (this label is not the soft label in our method). To assess the categorization system's quality, fivefold cross-validation was performed. A dataset has a 4:1 random split between training and test data. To classify document in this experiment, the classify model used 1 layer Fully Connected. The quality is measured by 3 measures: Precision, Recall, and F1-score.

With semantic similarity detection, we applied the tBERT [91] model to compare semantic similarity between a pair of text. This model combined the topic model with BERT to improve performance. The result is well than the standard BERT. Based on this model, LDA is changed by NMF, SeaNMF, and SupLeT to appraise the quality of the improving topic model on benchmark datasets MSRP and Quora. The performance is measured by F1-score.

Results

With two methods for distance metric learning, we compared them on three datasets: Agnews, StackOverflow, and Tagnews to choose the best method for SupLeT. The experiment used the Topic coherence to evaluate. We analyze these methods based on average score and maximum score on three times loop. The result of this experiment is shown in Table 5.1. On all three datasets, LMNN is outstanding to NCA with the average score and maximum score. In our experiments, LMNN is the chosen method used in the support learning process.

Table 5.1: Topic coherence of SupLeT based on SeaNMF model with 2 type of distance metric learning: NCA and LMNN

		Agnews	Stack Overflow	Tagnews
NCA	Max	3.708	1.889	3.297
	Avg	3.331	1.623	3.256
LMNN	Max	4.768	3.133	3.464
	Avg	4.361	2.652	3.413

In the next experiment, we continuously use topic coherence as a measure to evaluate. To prove the effectiveness of distance metric learning in the support process, we applied both Non-negative metric factorization (NMF) and Semantics-assisted NMF (SeaNMF). And they were compared with a popular method on topic modeling - Latent Dirichlet allocation(LDA).

- **Latent Dirichlet allocation (LDA)**: an illustrious baseline method in topic modeling. In this paper, we use the implementation[8] of LDA on scikit-learn.
- **SupLeT - NMF**: this is a version of support learning for topic modeling with NMF. In this version, LMNN was used for the support process with a transformer matrix (L).
- **SupLeT - SeaNMF**: this is a version of support learning for topic modeling with SeaNMF. In this version, LMNN was used for the support process with two transformer matrices (L_1, L_2).

[8]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

The number of topics is 100 in this experiment. The number of top-keywords is 10. We analyze the result of the experiment with the parameters in Table 5.2.

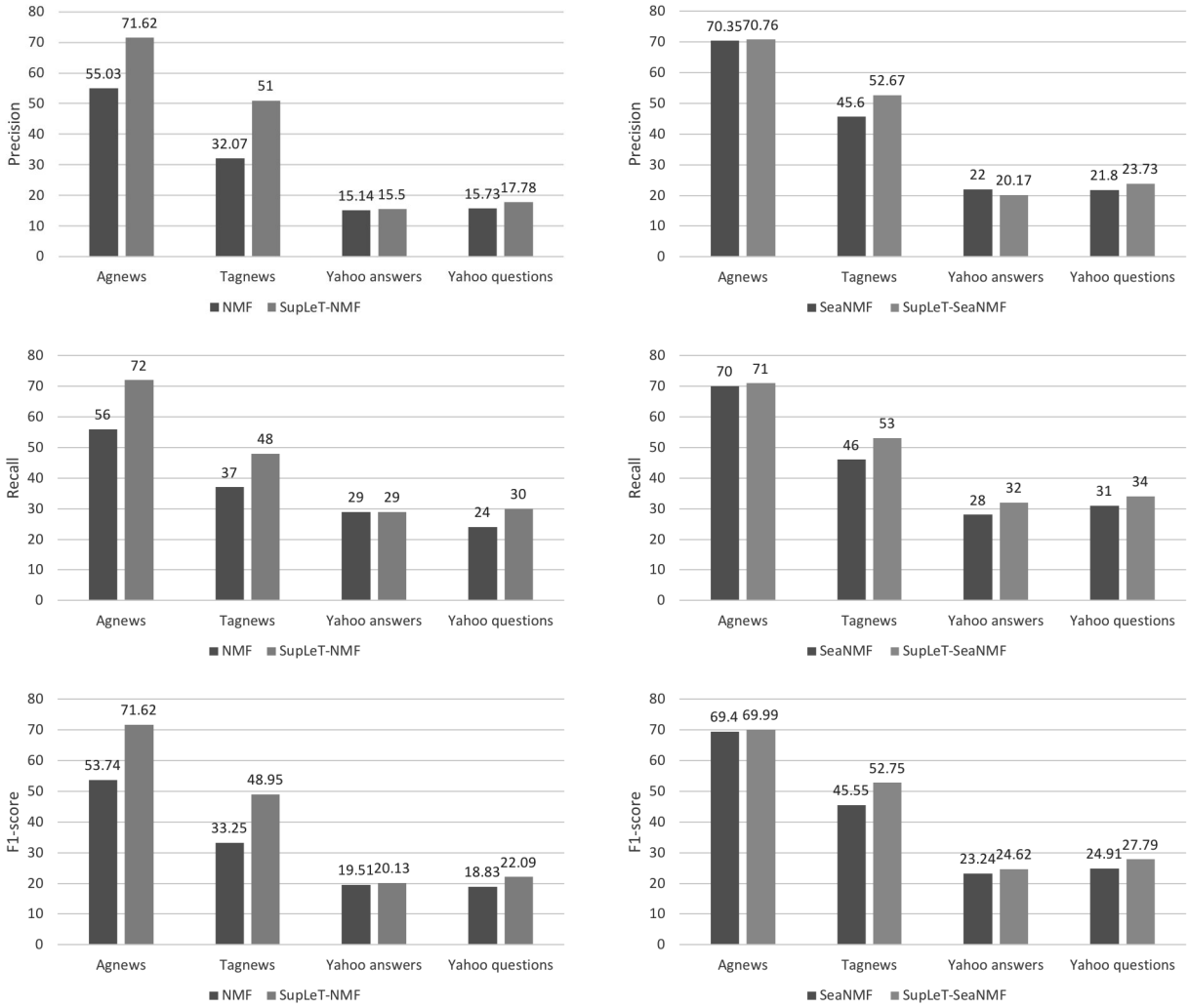
Table 5.2: Topic coherence result on datasets

	Agnews	Stack Overflow	Tagnews	Yahoo answers	Yahoo questions	Yelp review
LDA	1.187	0.675	2.023	0.807	0.904	0.129
NMF	1.709	1.000	2.484	1.386	1.853	0.485
SeaNMF	3.814	1.838	3.287	4.843	4.379	4.872
SupLeT - NMF	3.471	1.940	1.907	3.284	3.491	2.363
SupLeT - SeaNMF	**4.768**	**3.133**	**3.464**	**6.771**	**4.415**	**4.886**

From Table 5.2, we notice that our approach outperforms the baseline model LDA. Using distance metric learning to support the topic model showed significant improvement when compared with baseline. With the standard NMF and SeaNMF, SupLeT - NMF, and SupLeT - SeaNMF displayed the advantage. All 6 datasets which we were used in this experiment are short text datasets. They are the best suitable for SeaNMF, which works well on short context data. However, our approach can improve the performance of SeaNMF and NMF by about 1-2 units on topic coherence score. It implies distance metric learning can learn more coherent latent topics.

In addition to the topic coherence, document classification used to compare the methods in our experiment. We can see the result in Figure 5.3. All four datasets: Agnews, Tagnews, Yahoo answers, and Yahoo questions presented the effectiveness of our approach. With traditional methods as LDA, the approach has a significant improvement on three evaluation scores: Precision, Recall, and F1-score. The SupLeT-NMF and SupLeT-SeaNMF perform better than their standard model. It showed that distance metric learning adjusted effectiveness input matrix for the topic model can more easy to learn. With some time to learn, support learning methods condensed important information about the topic and added it to the input matrix. The result on Yahoo answers and Yahoo questions is not outstanding because the context of them is very short. All the standard methods and the update methods also difficult to catch.

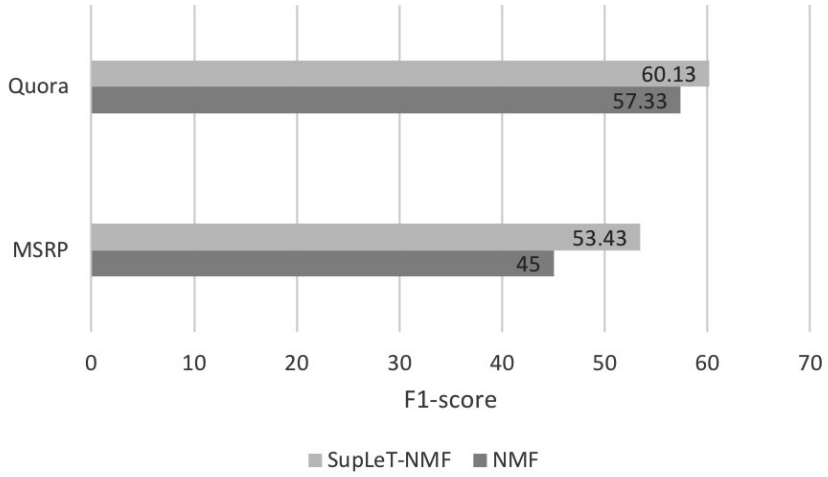
Semantics similarity detection is another task which we used to analyze the effectiveness of support learning for the topic model. The result of the experiment is showed in Figure 5.4. The SupLeT proved that the latent topic information increased in refinement learning. With two datasets MSRP and Quora, SupLeT outperforms the baseline models.



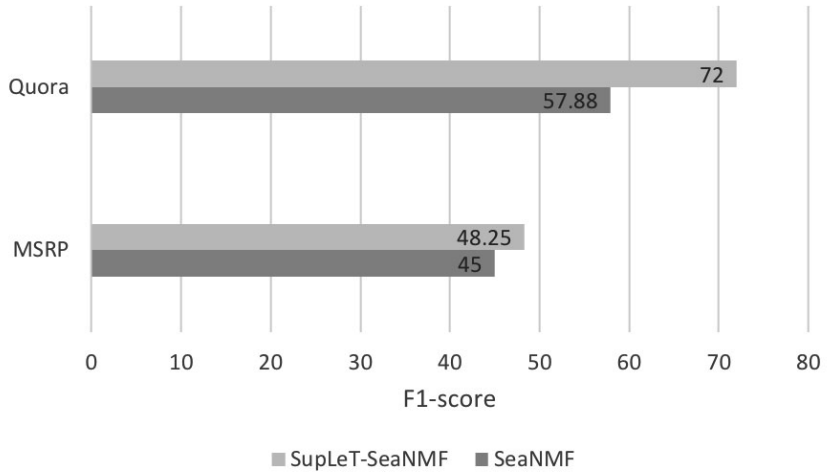
SupLeT-NMF and NMF

SupLeT-SeaNMF and SeaNMF

Figure 5.3: Document classification result on datasets



SupLeT-NMF and NMF



SupLeT-SeaNMF and SeaNMF

Figure 5.4: Model performance on MSRP and Quora dataset

The average length on a sample of MSRP is 37.29 tokens and the average length on a sample of Quora is 24.76 tokens. With the ability of the standard models (SeaNMF for short text), SupLeT-SeaNMF can work well on Quora and SupLeT-NMF fit with the MSRP dataset.

We identify related topics from SupLeT-SeaNMF and SeaNMF based on the top-10 keywords after learning latent topics on TagNews and StackOverflow datasets. Table 5.3 displays the top ten keywords in the retrieved list for the chosen topics. As we can see, "Sport" and "Japan news" are two of TagNews's latent topic. The topics chosen from StackOverflow concern "Visual Studio".

In the top-10 keywords, we can see a number of related words that represent topics

of SupLeT-SeaNMF are more than in SeaNMF. In "Sport" category, we consider 2 to groups: SupLeT-SeaNMF 6 and SeaNMF 34. "6" and "34" are numeral order topic in latent topic groups. As we can see, SupLeT-SeaNMF 6 has 8/10 keywords which have close relation with "Sport" such as basketball, soccer, league. Only two words "global" and "uconn" can be difficult to see the relation with the category. However, SeaNMF 34 only has 6/10 keywords. It proved that SupLeT-SeaNMF completes the goal of the model. Latent topics were refined and had clear representation.

Table 5.3: Top 10 keywords of several discovered latent topics by SupLeT-SeaNMF and SeaNMF

Dataset	Category	Index of topic	Top 10 key words	
TagNews	Sport	SupLeT-SeaNMF 6	league	semi-finals
			basketball	soccer
			play	uconn
			global	winning
			champions	fans
		SeaNMF 34	keeps	nbc
			winning	basketball
			semi-finals	drought
			champions	play-off
			roundup	share
	Japan	SupLeT-SeaNMF 42	japan	shut
			nuclear	rescue
			trust	reactors
			crisis	radioactivity
			government	quake
		SeaNMF 10	japan	deals
			street	stocks
			wall	dow
			nuclear	rescue
			worries	quake
StackOverflow	Visual Studio	SupLeT-SeaNMF 2	Visual	screen
			Studio	IFEnd
			Window	Refactoring
			FreezingTFS	Structured
			Might	IntelliSense
		SeaNMF 2	Visual	projects
			Studio	Can
			project	Keyboard
			Code	build
			Using	Setup

5.5.2 Impact of topic coherence in interaction

To evaluate the impact of the imporve topic cohenrence in interaction, we try to change the topic model method of SubTST and tBART by SupLeT-SeaNMF in Chapter 3 and Chapter 4. With SubTST, we choose the setting includes :(i) train topic:the parameters of topic embedding are learnable and updated during training process by the objective loss, (ii) mean: Mean point out the strategy of pooling layer of SubTST. The results are showed in Table 5.4,5.5 .

Table 5.4: Experimental results on semantic textual similarity with BERT_{base} and two option of topic models (unsupervised; STS unlabeled texts).

Model	STS-B	STS-12	STS-13	STS-14	STS-15	STS-16	SICK-R
Topic model: LDA							
SubTST	83.16	65.50	78.50	74.57	78.32	79.76	82.96
Topic model: SupLeT - SeaNMF							
SubTST	83.20	65.58	78.61	74.73	78.56	79.80	83.25

Table 5.5: Results of methods on CNN/Daily Mail datasets based on ROUGE score

	XSUM			CNN/Daily Mail		
Methods	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Topic model: LDA						
tBART	45.84	22.73	38.90	44.55	21.40	41.61
Topic model: SupLeT - SeaNMF						
tBART	45.90	22.76	38.93	44.60	21.41	41.65

As shown in Table 5.4 and 5.5, the effectiveness of the improve topic coherence are showed. Overall, the topic information interaction with SupLeT is higher than using the LDA model. The enhancement of topic coherence has clarified the semantics of words in the topic space. The increased coherence makes it easier to represent topic information. The vector representation of topic information is also more accurate. It is necessary for topic interaction. In the interaction, the quality of each element needs to be seriously considered. So that, if can improve the element's performance, the interaction also has a certain development about performance.

About SubTST, STS benchmarks datasets's structure is the pair of sentences. The length of each sentence is quite short. This is the available for SupLeT-SeaNMF, because SeaNMF is a topic model for short text. So, SubTST with SupLeT-SeaNMF has a big

variability. Oppositely, the benchmark dataset of tBART is long documents (the average length is 431 for XSum, and 781 for CNN/Daily Mail). The support of SuperT-SeaNMF becomes weak than when used for SubTST.

5.6 Conclusion

A technique to hone latent subjects is presented in this chapter. Our strategy suggests combining topic modeling and distance learning (NMF and SeaNMF). The learning distance process uses the large margin nearest neighbor (LMNN). Latent themes are used as labels by LMNN, and the word "sample" is used. The topic model's input matrices are updated by this learning process, which also produces a transformation matrix. On datasets, we contrasted SupLeT with conventional approaches. Experimental findings demonstrated that our model outperforms the standard models. In future works, SupLeT needs to change the calculation method of the "Distance metric learning" component. Currently, the time for learning of "Distance metric learning" component in SupLeT is not small. We should work to speed up processing and enhance the model's functionality in order to use it in a real-time system. At the same time, we try to develop the model to apply for other tasks such as aspect mining [92] and sentiment analysis [93].

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this thesis, we study the task of Semantic Parsing in NLP, which plays a key role in building human language interfaces, or human-machine communication. The main findings and our contributions are discussed and discuss and summarized as follows:

- Topic information is an important role in Natural language understanding. Differently from context, the topic brings meaning to covering general documents. It provided a general view for a group of words, and texts to clearly understand the meaning.

- Based on the analysis, topic information great support for transformer-based language models about semantics. Language models learn context information. However, context is local information, topic is global information. The combination of topic and context give a full view of the semantics of the text.

- With outside interaction, this method gives a combination based on concatenating two representation vectors. The topic information is added to the output of the transformer-based language model to represent input text. The fine-tuning process is based on Siamese Networks. The outside interaction keeps the original transformer-based language model. In this method, the topic vector only affects the surface of the language model by the special structure.

- With inside interaction, this method gives a combination based on transferring the

representation vectors and adding in the same space. The representation vector of the topic is transferred to context space by an align function. After that, it is injected into the input representation of the transformer-based model.

- The quality of the topic affects the performance of the interaction. Based on SupLeT, we have the high-quality topics. The high-quality topic improves both SubTST(outside interaction) and tBART(inside interaction).

6.2 Future work

Based on the current results, there are some potential directions that can be further studied in the future work:

- In this dissertation, topic information is embedded with subwords-unit to represent the sentence vector. However, words, phrases, or all sentences have meaningful than subwords. Furthermore, these representations can be injected into the self-attention mechanism of each input sentence. Improving topic representation with the hierarchical level can more support other semantic tasks.

- The two direct tasks of this research are semantic textual similarity and summarization. However, topic information interaction also supports other semantic tasks such as generation, question answering, or information retrieval. The application of the advance of the research can have effects on the tasks.

- Based on the analysis of SubTST and tBART, the topic model is built as a pre-train model. If the context learning process and topic learning process can take at the same time by the neural topic model technical, the training time can reduce. This is one of our future works.

Publications and Awards

Journals

- Thanh, Nguyen Ha, Phuong Minh Nguyen, Thi-Hai-Yen Vuong, Minh Q. Bui, Minh-Chau Nguyen, Binh Dang, Vu D. Tran, Le-Minh Nguyen and Ken Satoh. "Transformer-Based Approaches for Legal Text Processing." The Review of Socionetwork Strategies (2022): 1-21.. Transformer-based approaches for legal text processing. *The Review of Socionetwork Strategies*, 16(1):135–155, Apr 2022. ISSN 1867-3236. doi: 10.1007/s12626-022-00102-2. URL *https://doi.org/10.1007/ s12626-022-00102-2.*.

- Binh Dang, Tung Le, and Le-Minh Nguyen. SubTST: A Consolidation of Subword Latent Topics and Sentence Transformer in Semantic Representation, *Applied Intelligence*,2022, *https://doi.org/10.1007/s10489-022-04184-x*.

Conference papers

- Thanh, Nguyen Ha, Binh Dang and Le-Minh Nguyen. Deep Learning Approach for Vietnamese Consonant Misspell Correction.*Proceeding of the 16th International Conference of the Pacific Association for Computational Linguistics*, PACLING (2019). doi: 10.1007/978-981-15-6168-9_40. URL *https://doi.org/10.1007/978-981-15-6168-9_40*.

- Dang,Binh Tran, Nguyen Ha Thanh and Le-Minh Nguyen. Latent Topic Refinement based on Distance Metric Learning and Semantic-assisted Non-negative Matrix Factorization. *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, PACLIC (2020). URL *https://aclanthology.org/2020.paclic-1.8/*.

- Binh Dang, Chau Nguyen and Le-Minh Nguyen. An approach for Personalized Legal Information Retrieval System, *Proceedings of the 15th International Workshop on Juris-informatics*, JURISIN (2021).

- Thanh, Nguyen Ha, Minh Q. Bui, Chau Nguyen, Tung Le, Phuong Minh Nguyen, Binh Dang, Vuong Thi Hai Yen, Teeradaj Racharak, Nguyen Le Minh, Duc-Vu Tran, Phan Viet Anh, Nguyen Truong Son, Huy-Tien Nguyen, Bhumindr Butr-indr, Peerapon Vateekul and Prachya Boonkwan. A summary of the alqac 2021 competition. *Proceeding of the 13th International Conference on Knowledge and Systems Engineering (KSE)*, KSE (2021). doi: 10.1109/KSE53942.2021.9648724. URL *https://doi.org/10.1109/KSE53942.2021.9648724*.

- Nguyen Ha Thanh, Dang Tran Binh, Bui Minh Quan, Nguyen Le Minh. Evaluate and Visualize Legal Embeddings for Explanation Purpose. *Proceeding of the 13th International Conference on Knowledge and Systems Engineering*, KSE (2021). doi: 10.1109/KSE53942.2021.9648655. URL *https://doi.org/10.1109/KSE53942.2021.9648655*.

- Bui, Minh Q., Vu D. Tran, Nguyen Ha Thanh, Binh Dang and Le-Minh Nguyen. How Curriculum Learning Performs on AMR Parsing. *Proceeding of the 13th International Conference on Knowledge and Systems Engineering*, KSE (2021). doi: 10.1109/KSE53942.2021.9648646 . URL *https://doi.org/10.1109/KSE53942.2021.9648646*.

- Dang,Binh Tran, Dang, Thai Tran and Le-Minh Nguyen. SubTST: A Combination of Sub-word Latent Topics and Sentence Transformer for Semantic Similarity Detection. *Proceedings of the 14th International Conference on Agents and Artificial Intelligence*, ICAART (2022). doi: 10.5220/0010775100003116, URL *https://doi.org/10.5220/0010775100003116* .

- Binh Dang and Le-Minh Nguyen. tBART: Abstractive summarization based on the joining of Topic model and BART. *Proceeding of the 14th International Conference on Knowledge and Systems Engineering*, KSE (2022).

Awards

- Ranked second place among all Task 3 (Legal Information Retrieval) competitors of legal competition COLIEE in two years 2019 and 2020.

- Organizing committee of the legal Workshop of KSE 2021: Automated Legal Question Answering Competition (ALQAC 2021).

- Runner-Up Student Paper Award in KSE 2021 with paper "Evaluate and Visualize Legal Embeddings for Explanation Purpose"

Bibliography

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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: Association for Computational Linguistics, 2019, pp. 4171–4186.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Association for Computational Linguistics, 2020, pp. 7871–7880.

[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP 2019)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3982–3992.

[5] D. Chandrasekaran and V. Mago, "Evolution of semantic similarity - A survey," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 41:1–41:37, 2021.

[6] D. R. Radev, E. H. Hovy, and K. R. McKeown, "Introduction to the special issue on summarization," *Comput. Linguistics*, vol. 28, no. 4, pp. 399–408, 2002.

[7] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, 2017.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[9] J. Yang, G. Xiao, Y. Shen, W. Jiang, X. Hu, Y. Zhang, and J. Peng, "A survey of knowledge enhanced pre-trained models," *CoRR*, vol. abs/2110.00269, 2021.

[10] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, "Sentilare: Sentiment-aware language representation learning with linguistic knowledge," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 2020, pp. 6975–6988.

[11] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A continual pre-training framework for language understanding," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 8968–8975.

[12] M. E. Peters, M. Neumann, R. L. L. IV, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," 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 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 43–54.

[13] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: enhanced language representation with informative entities," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-*

August 2, 2019, Volume 1: Long Papers, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 1441–1451.

[14] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "KEPLER: A unified model for knowledge embedding and pre-trained language representation," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 176–194, 2021.

[15] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[16] B. Gangopadhyay, S. Hazra, and P. Dasgupta, "Semi-lexical languages: a formal basis for using domain knowledge to resolve ambiguities in deep-learning based computer vision," *Pattern Recognit. Lett.*, vol. 152, pp. 143–149, 2021.

[17] S. Amizadeh, H. Palangi, A. Polozov, Y. Huang, and K. Koishida, "Neuro-symbolic visual reasoning: Disentangling "visual" from "reasoning"," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 279–290.

[18] H. Liu, Y. Feng, M. Zhou, and B. Qiang, "Semantic ranking structure preserving for cross-modal retrieval," *Appl. Intell.*, vol. 51, no. 3, pp. 1802–1812, 2021.

[19] K. O'Shea, K. A. Crockett, Z. Bandar, and J. O'Shea, "Erratum to: An approach to conversational agent design using semantic sentence similarity," *Appl. Intell.*, vol. 40, no. 1, p. 199, 2014.

[20] A. Amara, M. A. H. Taieb, and M. B. Aouicha, "Multilingual topic modeling for tracking COVID-19 trends based on facebook data analysis," *Appl. Intell.*, vol. 51, no. 5, pp. 3052–3073, 2021.

[21] X. Du, R. Zhu, F. Zhao, F. Zhao, P. Han, and Z. Zhu, "A deceptive detection model based on topic, sentiment, and sentence structure information," *Appl. Intell.*, vol. 50, no. 11, pp. 3868–3881, 2020.

[22] C. Gao and J. Ren, "A topic-driven language model for learning to generate diverse sentences," *Neurocomputing*, vol. 333, pp. 374–380, 2019.

[23] Z. Qin, M. Thint, and Z. Huang, "Ranking answers by hierarchical topic models," in *Next-Generation Applied Intelligence, 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2009*, ser. Lecture Notes in Computer Science, B. Chien, T. Hong, S. Chen, and M. Ali, Eds., vol. 5579. Tainan, Taiwan: Springer, 2009, pp. 103–112.

[24] M. Ovsjanikov and Y. Chen, "Topic modeling for personalized recommendation of volatile items," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010*, ser. Lecture Notes in Computer Science, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., vol. 6322. Barcelona, Spain: Springer, 2010, pp. 483–498.

[25] Q. H. Tran, V. D. Tran, T. T. Vu, M. L. Nguyen, and S. B. Pham, "JAIST: Combining multiple features for answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, 2015, pp. 215–219.

[26] T.-B. Dang, H.-T. Nguyen, and L.-M. Nguyen, "Latent topic refinement based on distance metric learning and semantics-assisted non-negative matrix factorization," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*. Hanoi, Vietnam: Association for Computational Linguistics, Oct. 2020, pp. 70–75.

[27] G. Wu, Y. Sheng, M. Lan, and Y. Wu, "ECNU at SemEval-2017 task 3: Using traditional and deep learning methods to address community question answering task," in *Proceedings of the 11th International Workshop on Semantic Evaluation*

(SemEval-2017). Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 365–369.

[28] N. Peinelt, D. Nguyen, and M. Liakata, "tBERT: Topic models and BERT joining forces for semantic similarity detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 7047–7055.

[29] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9119–9130.

[30] L. Dinh, D. Krueger, and Y. Bengio, "NICE: non-linear independent components estimation," in *Processdings of the 3rd International Conference on Learning Representations, (ICLR 2015)*, 2015.

[31] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," 2021.

[32] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "ConSERT: A contrastive framework for self-supervised sentence representation transfer," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Aug. 2021, pp. 5065–5075.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[34] Y. Miao, E. Grefenstette, and P. Blunsom, "Discovering discrete latent topics with neural variational inference," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, pp. 2410–2419.

[35] R. Wang, D. Zhou, and Y. He, "Atm: Adversarial-neural topic model," *Information Processing & Management*, vol. 56, p. 102098, 2019.

[36] J. Choo, C. Lee, C. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *IEEE transactions on visualization and computer graphics*, vol. 19, pp. 1992–2001, 2013.

[37] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Weakly supervised nonnegative matrix factorization for user-driven clustering," *Data Min. Knowl. Discov.*, vol. 29, pp. 1598–1621, 2015.

[38] X. Yan, J. Guo, S. Liu, X. Cheng, and Y. Wang, "Learning topics in short texts by non-negative matrix factorization on term correlation matrix," in *Proceedings of the 13th SIAM International Conference on Data Mining*, 2013, pp. 749–757.

[39] Z. Wang, C. Wang, H. Zhang, Z. Duan, M. Zhou, and B. Chen, "Learning dynamic hierarchical topic graph with graph convolutional network for document classification," in *The 23rd International Conference on Artificial Intelligence and Statistics, (AISTATS 2020)*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. Online: PMLR, 2020, pp. 3959–3969.

[40] J. Zhang, L. Li, A. Way, and Q. Liu, "Topic-informed neural machine translation," in *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers . (COLING 2016)*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 1807–1817.

[41] X. Fu, J. Wang, J. Zhang, J. Wei, and Z. Yang, "Document summarization with vhtm: Variational hierarchical topic-aware mechanism," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 7740–7747, 2020.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Processdings of the 3rd International Conference on Learning Representations, (ICLR 2015)*, Y. Bengio and Y. LeCun, Eds., 2015.

[43] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4144–4150.

[44] W. B. Dolan and C. Brockett, "Automatically constructing a corpus of sentential paraphrases," in *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[45] P. Nakov, L. Màrquez, W. Magdy, A. Moschitti, J. Glass, and B. Randeree, "SemEval-2015 task 3: Answer selection in community question answering," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado, USA: Association for Computational Linguistics, 2015, pp. 269–281.

[46] P. Nakov, L. Màrquez, A. Moschitti, W. Magdy, H. Mubarak, A. A. Freihat, J. Glass, and B. Randeree, "SemEval-2016 task 3: Community question answering," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, CA, USA: Association for Computational Linguistics, 2016, pp. 525–545.

[47] P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, and K. Verspoor, "SemEval-2017 task 3: Community question answering," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 27–48.

[48] J. M. Deriu and M. Cieliebak, "SwissAlps at SemEval-2017 task 3: Attention-based convolutional neural network for community question answering," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 334–338.

[49] S. Filice, G. Da San Martino, and A. Moschitti, "KeLP at SemEval-2017 task 3: Learning pairwise patterns in community question answering," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 326–333.

[50] W. Wang, B. Bi, M. Yan, C. Wu, J. Xia, Z. Bao, L. Peng, and L. Si, "Structbert: Incorporating language structures into pre-training for deep language understanding," in *8th International Conference on Learning Representations, ICLR 2020*. Addis Ababa, Ethiopia: OpenReview.net, 2020.

[51] R. He, A. Ravula, B. Kanagal, and J. Ainslie, "Realformer: Transformer likes residual attention," in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, ser. Findings of ACL, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., vol. ACL/IJCNLP 2021. Online: Association for Computational Linguistics, 2021, pp. 929–943.

[52] S. Wang, H. Fang, M. Khabsa, H. Mao, and H. Ma, "Entailment as few-shot learner," *CoRR*, vol. abs/2104.14690, 2021.

[53] S. Humeau, K. Shuster, M.-A. Lachaux, and J. Weston, "Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring," in *International Conference on Learning Representations*, 2020.

[54] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "SemEval-2012 task 6: A pilot on semantic textual similarity," in *Proceedings of 6th International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 2012, pp. 385–393.

[55] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "*SEM 2013 shared task: Semantic textual similarity," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 32–43.

[56] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2014 task 10: Multilingual semantic textual similarity," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, Aug. 2014, pp. 81–91.

[57] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, G. Rigau, L. Uria, and J. Wiebe, "SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability," in *Proceedings of the 9th International Workshop on Semantic Eval-*

uation (SemEval 2015). Denver, Colorado: Association for Computational Linguistics, Jun. 2015, pp. 252–263.

[58] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe, "SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 497–511.

[59] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14.

[60] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A SICK cure for the evaluation of compositional distributional semantic models," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014, pp. 216–223.

[61] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 632–642.

[62] A. Williams, N. Nangia, and S. R. Bowman, "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, (NAACL-HLT 2018)*, M. A. Walker, H. Ji, and A. Stent, Eds. New Orleans, Louisiana, USA: Association for Computational Linguistics, 2018, pp. 1112–1122.

[63] M. F. Mridha, A. A. Lima, K. Nur, S. C. Das, M. Hasan, and M. M. Kabir, "A survey of automatic text summarization: Progress, process and challenges," *IEEE Access*, vol. 9, pp. 156 043–156 070, 2021.

[64] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 9815–9822.

[65] B. Dang, T. Dang, and L. Nguyen, "Subtst: A combination of sub-word latent topics and sentence transformer for semantic similarity detection," in *Proceedings of the 14th International Conference on Agents and Artificial Intelligence, ICAART 2022, Volume 3, Online Streaming, February 3-5, 2022*. SCITEPRESS, 2022, pp. 91–97.

[66] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1073–1083.

[67] T. Ma, Q. Pan, H. Rong, Y. Qian, Y. Tian, and N. Al-Nabhan, "T-bertsum: Topic-aware text summarization based on BERT," *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 3, pp. 879–890, 2022.

[68] Z. Wang, Z. Duan, H. Zhang, C. Wang, L. Tian, B. Chen, and M. Zhou, "Friendly topic assistant for transformer based abstractive summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 485–497.

[69] T. Shi, K. Kang, J. Choo, and C. K. Reddy, "Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 1105–1114.

[70] R. Alec, N. Karthik, and S. Tim, "Improving language understanding by generative pre-training." 2018.

[71] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3730–3740.

[72] J. Yin and J. Wang, "A dirichlet multinomial mixture model-based approach for short text clustering," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 233–242.

[73] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[74] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, "Topic modeling of short texts: A pseudo-document view," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2105–2114.

[75] C. Li, H. Wang, Z. Zhang, A. Sun, and Z. Ma, "Topic modeling for short texts with auxiliary word embeddings," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016, pp. 165–174.

[76] K. Q. Weinberger, B. John, and K. S. Lawrence, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems 18*, 2006, pp. 1473–1480.

[77] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, 2009.

[78] E. H. Ramirez, R. Brena, D. Magatti, and F. Stella, "Probabilistic metrics for soft-clustering and topic model validation," in *2010 IEEE/WIC/ACM International Con-*

ference on Web Intelligence and Intelligent Agent Technology, vol. 1, 2010, pp. 406–412.

[79] J. Li, M. Liao, W. Gao, Y. He, and K.-F. Wong, "Topic extraction from microblog posts using conversation structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2114–2123.

[80] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22Nd International Conference on World Wide Web*, 2013, pp. 1445–1456.

[81] X. Zhao, D. Wang, Z. Zhao, W. Liu, C. Lu, and F. Zhuang, "A neural topic model with word vectors and entity vectors for short texts," *Information Processing & Management*, vol. 58, p. 102455, 2021.

[82] J. Goldberger, S. Roweis, G. Hinton, and S. Ruslan, "Neighbourhood components analysis," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, 2004, pp. 513–520.

[83] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2014, pp. 2177–2185.

[84] J. L. Suárez, S. García, and F. Herrera, "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges," *Neurocomputing*, vol. 425, pp. 300–322, 2021.

[85] B. Nguyen and B. De Baets, "An approach to supervised distance metric learning based on difference of convex functions programming," *Pattern Recognition*, vol. 81, pp. 562–574, 2018.

[86] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, pp. 3600–3612, 2008.

[87] Q. Hu, P. Zhu, Y. Yang, and D. Yu, "Large-margin nearest neighbor classifiers via sample weight learning," *Neurocomputing*, vol. 74, pp. 656–660, 2011.

[88] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, "Automatic evaluation of topic coherence," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 100–108.

[89] J. Xu, P. Wang, G. Tian, B. Xu, J. Zhao, F. Wang, and H. Hao, "Short text clustering via convolutional neural networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015, pp. 62–69.

[90] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics*, vol. 1, 2002, pp. 1–7.

[91] N. Peinelt, D. Nguyen, and M. Liakata, "tBERT: Topic models and BERT joining forces for semantic similarity detection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7047–7055.

[92] J. He, L. Li, Y. Wang, and X. Wu, "Targeted aspects oriented topic modeling for short texts," *Applied Intelligence*, vol. 50, no. 8, pp. 2384–2399, Aug. 2020.

[93] F. Huang, C. Yuan, Y. Bi, J. Lu, L. Lu, and X. Wang, "Multi-granular document-level sentiment topic analysis for online reviews," *Applied Intelligence*, Oct 2021.