Learning Semantic Textual Similarity via Topic-informed Discrete Latent Variables

Erxin Yu[1,2], **Lan Du**[4], **Yuan Jin**[4], **Zhepei Wei**[1,2], **Yi Chang**[1,2,3*]
[1]School of Artificial Intelligence, Jilin University
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Jilin University
[3]International Center of Future Science, Jilin University
[4]Faculty of Information Technology, Monash University, Australia
erxin.yu@outlook.com, lan.du@monash.edu,
yuan.jin@monash.edu, weizp19@mails.jlu.edu.cn, yichang@jlu.edu.cn

Abstract

Recently, discrete latent variable models have received a surge of interest in both Natural Language Processing (NLP) and Computer Vision (CV), attributed to their comparable performance to the continuous counterparts in representation learning, while being more interpretable in their predictions. In this paper, we develop a topic-informed discrete latent variable model for semantic textual similarity, which learns a shared latent space for sentence-pair representation via vector quantization. Compared with previous models limited to local semantic contexts, our model can explore richer semantic information via topic modeling. We further boost the performance of semantic similarity by injecting the quantized representation into a transformer-based language model with a well-designed semantic-driven attention mechanism. We demonstrate, through extensive experiments across various English language datasets, that our model is able to surpass several strong neural baselines in semantic textual similarity tasks.

1 Introduction

Semantic Textual Similarity (STS), which concerns the problem of measuring and scoring the relationships or relevance of pairs of text on real-valued scales, is a fundamental task in NLP. It has further driven many other important NLP tasks such as machine translation (Zou et al., 2013), text summarization (Mohamed and Oussalah, 2019), question answering (Bordes et al., 2014), and etc.

Recently, deep neural language models have achieved state-of-the-art performance for STS. The success of these models is attributed to their adoption of self-supervised learning on text representations, which overcomes the absence of large labeled STS data in many domains. The text representations learned by these models, usually consist of continuous latent variables, and can be fed pairwise into some functions (e.g. a multi-layer perceptron) to compute their semantic similarity.

Compared to the continuous ones, discrete latent variables have drawn much less attention in language modeling, despite that natural languages are discrete in nature, and there is growing evidence, from several NLP tasks (Jin et al., 2020; Bao et al., 2020), that they are equally good, if not more suitable, as the continuous counterparts. As a widely-used training technique for learning discrete latent variables, vector-quantized variational autoencoder (VQ-VAE) (Oord et al., 2017) computes the values for the variables through the nearest neighbor lookup across the quantized vectors from the shared latent embedding space. Despite its success in speech recognition and computer vision, VQ-VAE has yet to be investigated in its use and viability in general NLP tasks. In this paper, we intend to explore the use of VQ-VAE and its generated discrete latent variables for semantic textual similarity.

Two issues need to be resolved in order for the VQ-VAE to work effectively on the STS task. First, the codebook embeddings need to be carefully initialized to prevent the so-called codebook collapse, in which only a few of the embeddings are selected and learned by the model, causing a reduction in the representational capacity of the codebook. Past research has indicated that sufficiently informed and sophisticated manipulation of the embedding initialization can mitigate this problem. A second issue is that most STS scenarios are designed to teach a language model to measure the textual similarity from a local perspective (e.g. over contexts within each sentence). However, global semantics underlying a broader context has already been shown to benefit a number of NLP tasks, including language generation (Guo et al., 2020a), textual similarity calculation (Peinelt et al., 2020), keyphrase generation (Wang et al., 2019), etc. For STS, it can provide perspectives on the global correlations and dependencies among sentences, which

[*] Corresponding Author

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4937–4948
December 7-11, 2022 ©2022 Association for Computational Linguistics

can be leveraged to better distinguish the semantic nuances of sentence-level contexts.

In this paper, we propose to leverage topic modeling to tackle both issues. The topic information can provide informative guidance on the initialization and the learning of VQ-VAE's codebook. Furthermore, since the discovered topics can capture global semantics (e.g., semantics as distributions over the vocabulary shared by the whole corpus), they can be used to calibrate the bias of local semantics on the measure of textual similarity.

Our proposed topic-enhanced VQ-VAE language model features two major components to enrich the codebook representational capacity with rich contextual information. One is the topic-aware sentence encoder, which computes the sentence embedding using its (local) topic distribution and the corresponding global topic-word distributions. The other is a topic-guided VQ-VAE codebook, where we align its latent codes/embeddings with the topic embeddings jointly learned by an NTM throughout the training phase. We further incorporate the quantized sentence representations learned by our topic-enhanced VQ-VAE into transformer-based language models to guide the learning of contextual embeddings. Built on top of the multi-head attention, a semantics-driven attention mechanism is proposed to compute the attention scores based on the quantized sentence representations. The proposed attention mechanism is a flexible plug-and-play module that requires a small number of extra parameters and a minimum change to the existing multi-head attention implementation. Our contributions are summarised as:

- To the best of our knowledge, this is the first language model that explores the use of discrete latent variables learned by VQ-VAE for STS.
- We introduce a topic-enhanced VQ-VAE model that combines the merits of topic modeling and vector quantization, where the quantization is informed by the topic information jointly learned by an NTM to enhance its robustness against the collapsing problem and its capacity of capturing both the global and local semantics.
- We present a simple yet effective semantics-driven attention mechanism to inject the quantized representations into transformer models, which can serve as a plug-and-play module for almost all the existing transformer-based models.
- Comprehensive experiments on six real-world datasets for semantic textual matching demonstrate the effectiveness of our model and the significance of its major components.

2 Related Work

2.1 Neural Language Models for STS

Neural language models have played a key role in fulfilling the STS task. Among them, the earlier CNN-based model (Hu et al., 2014; Yin et al., 2016) and the RNN-based model (Chen et al., 2018) used to achieve the state-of-the-art performance before the emergence of the transformer architecture (Vaswani et al., 2017) and the large-scale pretraining of its corresponding models. These pretrained models, with their ability to be fine-tuned on various domains and tasks, have dominated the entire NLP area, significantly improving the task performance in the specific domains. As the most prominent model, BERT (Devlin et al., 2019), along with its variants including RoBERTa (Liu et al., 2019), BERT-sim (Xia et al., 2021), ALBERT (Lan et al., 2020) and SemBERT (Zhang et al., 2020), have achieved superior results in the STS task. Despite the achieved success, current language models are mostly "black-box" models with low interpretability, which require additional explanatory components for their predictions in needed fields such as Biomedicine and Finance.

2.2 Neural Topic Models

The success of Variational AutoEncoder (VAE) (Kingma and Welling, 2013) has led to the development of a series of neural topic models (NTM) (Zhao et al., 2017, 2021b), (Refer to Zhao et al. (2021a) for a comprehensive survey on NTMs), where the posterior distribution of document topics is approximated by a neural network during the variational inference process, known as the neural variational inference (NVI) (Miao et al., 2016). The NVI also enables NTMs to be easily combined with various language models for capturing the global semantics information. These hybrid models are known as the topic-aware language models. Their combination strategies can be largely characterized by either concatenation of (document) topic distributions with the local word embeddings, or (multi-head) attention mechanisms towards the inferred topics. These models have been shown to achieve higher performance than the plain language models on various tasks, such as text summarization (Wang et al., 2020), keyphrase generation (Wang et al., 2019) and document classification (Zeng et al., 2018).

2.3 VQ-VAE

Rather than learning continuous representations, VQ-VAE (Oord et al., 2017) learns quantized representations via vector quantization. It was first applied to speech representation (Chorowski et al., 2019) and video generation (Yan et al., 2021). Recently, it has attracted lots of attention in a variety of NLP tasks. In machine translation, Kaiser et al. (2018) applied VQ-VAE to decode the target language sequence from the discrete space, making the model inference faster and the model performance only slightly compromised. In text classification, Jin et al. (2020) explored general text representations induced by VQ-VAE for the classification especially in low-resource domains. Huang and Ji (2020) designed a semi-supervised VQ-VAE for new event type induction, while Guo et al. (2020b) equipped the encode-decoder architecture with the VQ-VAE to automatically select relevant evidence. In this paper, we move one step further by guiding the vector quantization with topics learned by NTMs.

3 Modelling Framework

Given a pair of sentence $X = \{\boldsymbol{X}_i, \boldsymbol{X}_j\}$ as the input, where each sentence is represented as the bag-of-words (bow) vectors and contextualized embedding, our modelling framework, as shown in Figure 1, predicts the semantic similarity Y between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$.

3.1 Neural Topic Component

We adopt a VAE-based neural topic model (Miao et al., 2017) to learn latent topics. Different from LDA (Blei et al., 2003), NTM parameterises the latent topics $\boldsymbol{\theta}$ using a neural network conditioned on a draw from a Gaussian Softmax Construction. Here, $\boldsymbol{\theta} \in \mathbb{R}^K$ represents the topic proportions of a sentence X, where K denotes the number of topics. Let $\boldsymbol{\beta}_k \in \mathbb{R}^V$ be a distribution over a vocabulary V associated with a topic k. Following (Wang and YANG, 2020), we use word embeddings $\boldsymbol{\omega} \in \mathbb{R}^{V \times E}$, topic embeddings $\boldsymbol{\phi} \in \mathbb{R}^{K \times E}$ to compute $\boldsymbol{\beta}_k$ as $\boldsymbol{\beta}_k = Softmax(\boldsymbol{\omega} \cdot \boldsymbol{\phi}_k^T)$.

With NVI, we can use an encoder network to approximate the true posterior. Specifically, the encoder generates the variational parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ through neural networks and the latent variable $\boldsymbol{\theta}$ is sampled with the Gaussian reparameterization trick. The decoder reconstructs the BoW representation of a document by maximizing the log-likelihood of the input. The loss function (i.e., ELBO) contains a reconstruction error term and a KL divergence term as

$$L_{NTM}{=}D_{KL}[q(\boldsymbol{\theta}|\boldsymbol{x})||p(\boldsymbol{\theta}|\boldsymbol{x})]{-}E_{q(\boldsymbol{\theta}|x)}[\textstyle\sum_{m=1}^{M}\log p(w_m|\boldsymbol{\theta})],$$

where $q(\boldsymbol{\theta}|\boldsymbol{x})$ denotes the variational posterior distribution of $\boldsymbol{\theta}$ given the sentence $\boldsymbol{x} = (w_1, ..., w_m)$, approximating the true posterior $p(\boldsymbol{\theta}|x)$.

3.2 Topic-enhanced VQ-VAE

VQ-VAE (Oord et al., 2017) takes the contextualized embeddings of words within a sentence: $\boldsymbol{X}^{seq} = (\boldsymbol{x}_1^{seq}, ..., \boldsymbol{x}_m^{seq})$ into its encoder to produce the corresponding latent representations $\boldsymbol{Z}^e$. Then, the quantized representations $\boldsymbol{Z}^q$ are calculated by the nearest neighbor look-up using the predefined and shared embedding space $\boldsymbol{E}$, and further used as an input to the decoder to reconstruct the original sentence text.

However, since the original embedding space of VQ-VAE is randomly initialized, these learned embeddings can be arbitrary without clear semantic meaning. To guide the codebook learning, we incorporate the topic information into VQ-VAE by designing the following two components:

Topic Sensitive Encoder. We use a standard single-layer Transformer (Vaswani et al., 2017) as our encoder. Specifically, for a sentence consisting of a sequence of m words $\boldsymbol{X} = (w_1, ..., w_m)$, its corresponding word representations $\boldsymbol{X}^{seq}$ are generated by the transformer embedding layer, which concatenates the word embedding and position embedding. To incorporate the topic information, we leverage the sentence topic distribution $\boldsymbol{\theta}$ and the word topic weight vector $\boldsymbol{\beta}_{w_m} \in \mathbb{R}^K$ from the NTM model to compute the topical embedding of the words $\boldsymbol{X}^t = \{\boldsymbol{x}_1^t, ..., \boldsymbol{x}_m^t\} \in \mathbb{R}^{m \times K}$, where $\boldsymbol{x}_m^t \in \mathbb{R}^K = \boldsymbol{\theta} \otimes \boldsymbol{\beta}_{w_m}$. We concatenate the $\boldsymbol{X}^{seq}$ and $\boldsymbol{X}^t$ and feed them into the encoder as $\boldsymbol{Z}^e \in \mathbb{R}^{m \times E} = f_{enc}(\boldsymbol{X}^{seq} \oplus \boldsymbol{X}^t)$.

Topical Latent Embedding. We get final word representations $\boldsymbol{Z}^e = \{\boldsymbol{z}_1^e, ..., \boldsymbol{z}_m^e\}$ from the topic sensitive encoder. These representations are mapped onto the nearest element of embedding $\boldsymbol{E}$. Different from previous VQ-VAE models that use a randomly initialized embedding $\boldsymbol{E}$, we leverage the topic embedding $\boldsymbol{\phi} \in \mathbb{R}^{K \times E}$ from NTM. The discretization process is defined

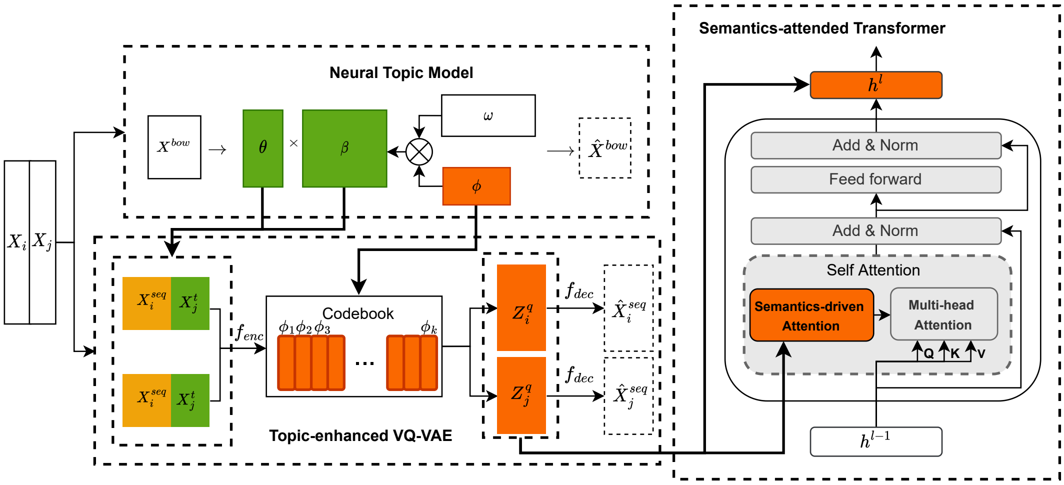


Figure 1: Our semantic textual matching framework combines the benefit of NTMs, VQ-VAE and Transformer.

as: For each $i \in \{1,\ldots,m\}$, $\boldsymbol{z}_i^q = \boldsymbol{\phi}_k \in \mathbb{R}^E$, where $k = \arg\min_{j\in\{1,\ldots,k\}} \|\boldsymbol{z}_i^e - \boldsymbol{\phi}_j\|_2$. Thus, we have quantized representations $\boldsymbol{Z}^q = \{\boldsymbol{z}_1^q, \ldots, \boldsymbol{z}_m^q\} \in \mathbb{R}^{m\times E}$ for a sentence. We then feed $\boldsymbol{Z}^q$ into the decoder which is also a single-layer transformer to reconstruct the sentence X. The overall training objective is thus defined as

$$L_{VQ-VAE} = \sum_{i=1}^{m}(-\log p(\boldsymbol{x}_i|\boldsymbol{z}_i^q) + \|sg[\boldsymbol{z}_i^e] - \boldsymbol{z}_i^q\|_2^2 + \lambda\,\|\boldsymbol{z}_i^e - sg[\boldsymbol{z}_i^q]\|_2^2). \qquad (1)$$

The first term above is the reconstruction error of the decoder given $\boldsymbol{Z}^q$. The last two terms are used to minimize the distance between the latent embedding $\boldsymbol{E}$ and the encoder output Z^e, where $\boldsymbol{\lambda}$ is a commitment loss and $sg(\cdot)$ means the stop-gradient operation. It is noteworthy that our topic-enhanced VQ-VAE model jointly trained with NTM can explicitly learn the topic assignment of each individual words, which is not feasible for NTM alone.

3.3 Semantics-driven Multi-head Attention

The multi-head attention in transformer-based models explores the relationships among tokens by calculating the token similarities, defined as

$$\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{h}^{l-1}$$
$$\text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\boldsymbol{head}_1, \cdots, \boldsymbol{head}_n)\boldsymbol{W}^O$$
$$\boldsymbol{head}_i = \text{Attention}(\mathbf{Q_i}W_i^Q, \boldsymbol{K_i}W_i^K, \boldsymbol{V_i}W_i^V) \qquad (2)$$

where $\boldsymbol{h}^{l-1} \in \mathbb{R}^{L\times(n\cdot E)}$ is the output of last layer, L is the length of sentence pair, n is the number of attention head, E is the hidden size of each attention head. $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$, $\boldsymbol{W}_i^V$ and $\boldsymbol{W}_i^O \in \mathbb{R}^{E\times E}$ are projection matrices. Scaled Dot-Product method is adopted to calculate the attention function:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \qquad (3)$$

To enable the transformer-based model to learn discrete semantics between two sentences, we design a semantics-driven attention mechanism. Specifically, given the quantized representations $\boldsymbol{Z}_i^q$, $\boldsymbol{Z}_j^q$ for sentence $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ respectively, we compute the quantized representation-based query and key matrices as follows:

$$\boldsymbol{Q}_q = \boldsymbol{Z}\boldsymbol{W}_q^Q, \quad \boldsymbol{K}_q = \boldsymbol{Z}\boldsymbol{W}_q^K, \quad \boldsymbol{Z} = \text{FFN}[\boldsymbol{Z}_i^q \oplus \boldsymbol{Z}_j^q], \qquad (4)$$

where $\boldsymbol{Z}$, $\boldsymbol{Q}_q$, $\boldsymbol{K}_q \in \mathbb{R}^{L\times E}$, $\boldsymbol{W}_q^Q$ and $\boldsymbol{W}_q^K \in \mathbb{R}^{E\times E}$, FFN means a fully connected feed-forward network. The semantics-driven attention can be defined as:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T + \boldsymbol{Q}_q\boldsymbol{K}_q^T}{\sqrt{d_k}})\boldsymbol{V}. \qquad (5)$$

It is different from the topic select-attention block introduced by (Lu et al., 2022), which is an attention block built separately on top of the multi-head attention block. In addition to the introduction of the quantized representations into the self-attention mechanism, we also leverage those representations to enhance the final output $\boldsymbol{h}^l$ of multi-layer transformer, i.e., $[\boldsymbol{h}^l; \boldsymbol{Z}_i^q \oplus \boldsymbol{Z}_j^q] \in \mathbb{R}^{L\times(n+1)E}$, which is further fed into a non-linear layer to predict label Y for semantic textual matching.

3.4 Joint Training

Our proposed framework integrates the three modules in Figure 1, i.e., the NTM, the topic-enhanced VQ-VAE, and the semantics-attended Transformer equipped with the semantics-driven attention mechanism. We first pretrain the NTM to obtain meaningful topic information used to initialize the code-

Dataset	Train/Dev/Test	Len	Label
MRPC	3,668/408/1,725	22	0/1
Quora	384,348/10,000/10,000	13	0/1
STS-B	5,749/1,500 /1,379	10	0~5
SemEval-A	20,340/3,270/2,930	41	0/1
SemEval-B	3,169/700/880	46	0/1
SemEval-C	31,690/7,000/8,800	39	0/1

Table 1: Statistics of Datasets

book of the topic-enhanced VQ-VAE. Then we joint train the NTM and topic-enhanced VQ-VAE to boost their performance. The joint loss function is defined as:

$$L = L_{VQ-VAE} + \gamma L_{NTM}, \quad (6)$$

where γ is the trade-off parameter controlling the balance between the topic model and the topic-enhanced VQ-VAE model. Finally, we incorporate the learned quantized representations from topic-enhanced VQ-VAE into the transformer block and fine-tune the transformer block on the STS task.

4 Experiment and analysis

4.1 Datasets

We conducted experiments on the following benchmark datasets used for the STS task, which include MRPC, Quora, STS-B, and SemEval CQA. The Microsoft Research Paraphrase dataset (**MRPC**) (Dolan and Brockett, 2005) contains pairs of sentences from news websites with binary labels for paraphrase detection. The **Quora** duplicate question dataset contains more than 400k question pairs with binary labels for predicting if two questions are paraphrases. We use train/dev/test set partition from (Wang et al., 2017). **STS-B** is a collection of sentence pairs extracted from news headlines and other sources. It comprises a selection of the English datasets used in the STS task which were annotated with a score from 1 to 5 denoting how similar the two sentences are. The **SemEval** community question answering has three subtasks: **(A)** Question–Comment Similarity, **(B)** Question–Question Similarity, **(C)** Question–External Comment Similarity (Nakov et al., 2015, 2016, 2017). Following their settings, we use the 2016 test set as the development set and the 2017 test set as the test set. Table 1 summarises their statistics.

4.2 Baseline Models

We implemented several strong baselines in the STS task for comparison. To evaluate the semantics-attended Transformer, we select two widely-used pretrained language models, **Bert** (Devlin et al., 2019) and **RoBERTa** (Liu et al., 2019). We used the small version to fine-tune the STS task, i.e., Bert-base and RoBERTa-base. And we modified these two models by adding the semantics-driven multi-head attention to the original multi-head attention, which is denoted as **DisBert** and **DisRoBERTa** respectively. We selected **tBERT** (Peinelt et al., 2020) as an important baseline that also makes use of topic information. Different from tBert which directly concatenates topic representation and sentence pair vector, we leverage the topic model to help VQ-VAE capture global semantics through the topic sensitive encoder and topical latent embedding. We also select the knowledge-enhanced Bert model **ERNIE** (Zhang et al., 2019) that incorporates knowledge graph into Bert and achieves improvements on language understanding tasks. To further verify the performance on the STS task, we also compared with Semantic-aware Bert (**SemBert**) (Zhang et al., 2020), which incorporates explicit contextual semantics and outperforms other Bert-based models on the STS task.

4.3 Experimental Settings

For all methods, we performed a greedy search to find their optimal hyper-parameters using the development set. For NTM, we processed the datasets with gensim tokenizer [1], and a vocabulary for each dataset was built based on its training set with stop words and words occurring less than 3 times removed. The BoW input of topic model $\boldsymbol{X}^{BoW}$ on each dataset was constructed based on the corresponding vocabulary. For the VQ-VAE model, we set the dimension of each code E to 64 and commitment loss λ = 0.0001. [2]

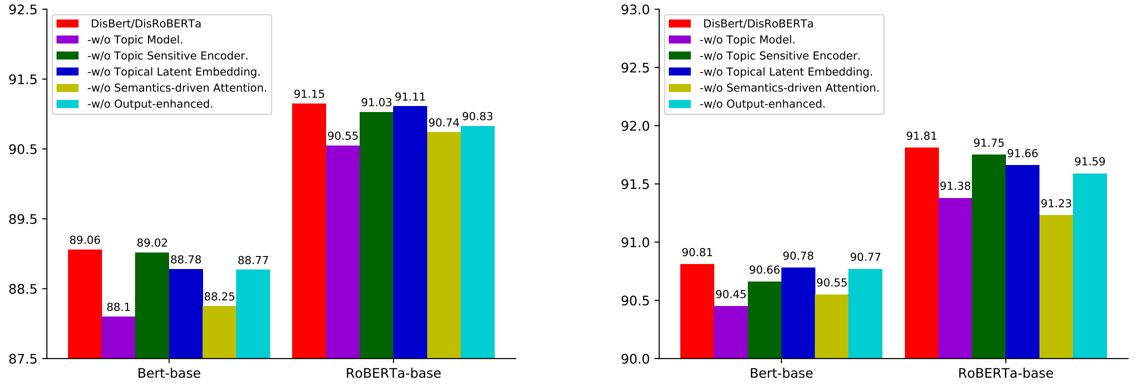
For joint training, we pretrained the NTM and selected the best model based on their reconstruction perplexity. Then we jointly trained the selected NTM and topic-enhanced VQ-VAE with γ = 1. We fine-tuned our semantics-driven language model and the vanilla one with the same parameters for a fair comparison, following their settings with the publicly available code. We use Pearson correlation coefficient (PC) and Spearman correlation coefficient (SC) for STS-B datasets. For other datasets, we report accuracy (ACC) and F1 score. More detailed parameter settings are described in the appendix.

[1] https://radimrehurek.com/gensim/ utils.html

[2] Code is available at https://github.com/ErxinYu/DisBert.

Models	MRPC		Quora		STS-B		SemEval					
							A		B		C	
Baselines	*Acc*	$F1$	*Acc*	$F1$	*PC*	*SC*	*Acc*	$F1$	*Acc*	$F1$	*Acc*	$F1$
ERNIE	83.97	88.16	90.88	90.76	85.12	83.26	77.53	75.10	68.86	51.42	94.36	27.20
tBert	84.29	88.44	90.76	90.65	85.34	83.96	78.02	76.77	71.44	52.10	94.07	27.33
SemBert	84.02	88.24	90.83	90.77	**87.13**	85.34	78.15	76.33	71.47	52.19	94.60	27.44
Bert	84.21	88.21	90.79	90.72	86.11	84.90	77.88	75.25	66.81	51.33	94.40	27.36
DisBert	**84.70**	**89.06**	**90.94**	**90.81**	86.67	**85.64**	**79.15**	**77.57**	**72.38**	**54.44**	**94.96**	**27.61**
RoBERTa	87.01	90.61	91.03	91.45	89.04	88.38	78.53	76.56	75.09	55.75	93.52	32.46
DisRoBERTa	**88.06**	**91.15**	**91.53**	**91.81**	**89.90**	**89.28**	**78.89**	**77.01**	**77.72**	**57.18**	**95.63**	**33.79**

Table 2: Comparisons of performance on six STS datasets. We report average performance for five different random seeds. The better results of different transformer-based model are highlighted in bold (according to the pairwise t-test with 95% confidence).



(a) MRPC

(b) Quora

Figure 2: Ablation study on DisBert and DisRoBERTa model. We reported F1 score for MRPC and Quora datasets.

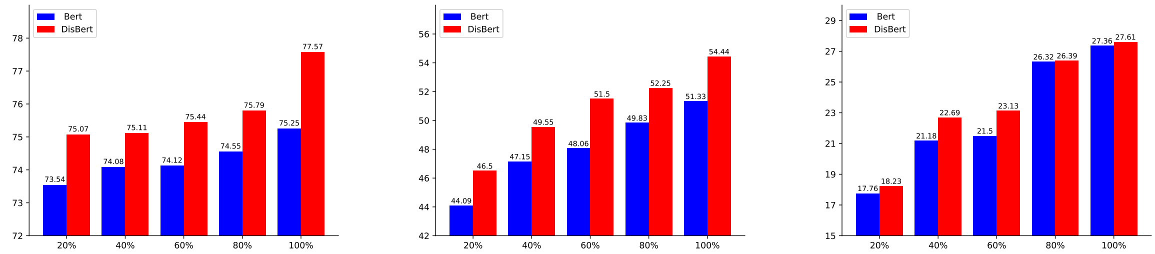
4.4 Performance on Six STS Datasets

Table 2 shows the results drove on the six datasets, for which we have the following observations.

- **Semantics-attended transformer is effective.** Our model, DisBert, which introduces the quantized representations into the transformer, significantly outperforms Bert, with the improvement ranging from 0.1% to 5.5% on the six datasets. The results demonstrate the usefulness of our semantics-driven multi-head attention mechanism. Similarly, RoBERTa equipped with vector quantized representations learned by our topic-enhanced VQ-VAE also achieves improved performance. Thus, the results on both Bert and RoBERTa verify that our semantics-driven attention is an effective plug-and-play module that can be readily applied to various transformer-based language models.
- **DisBert outperforms other bert-based models.** Our proposed methods achieve better performance than tBert does, which directly concatenates the output of the topic model and Bert. It indicates that injecting topic-enhanced VQ-VAE into Bert is a more efficient way to provide global semantics. The performance compared with ERNIE shows that our method is better at capturing sentence semantics than incorporating knowledge graphs into Bert. Compared with semBert, which is capable of explicitly absorbing contextual semantics, our model gains better performance on five out of six STS datasets. Overall, the performance gain on almost datasets demonstrates the effectiveness of our proposed framework.

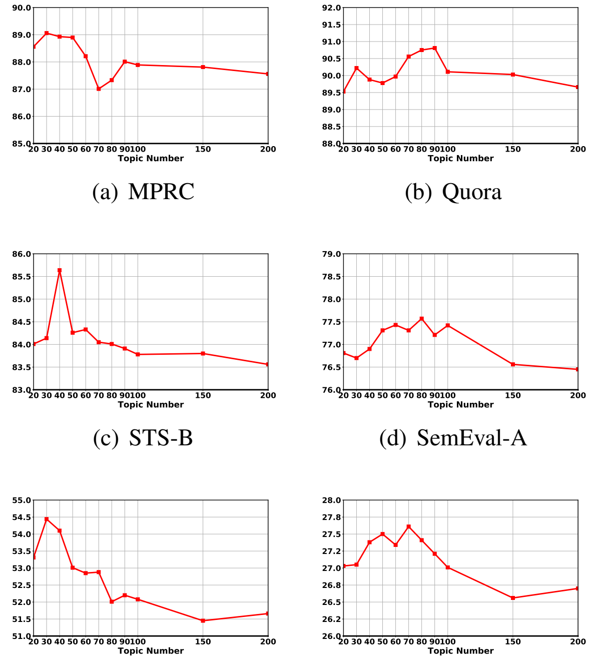
4.5 Ablation Study

We compared different variants of DisBert and DisRoBERTa in order to study how topic-informed discrete latent variables affect performance. **w/o Topic Model** means using the vanilla VQ-VAE model to get discrete latent variables without topic information. **w/o Topic Sensitive Encoder** refers to removing topical embedding $\boldsymbol{X}^t$ in the encoding phase and **w/o Topical Latent Embedding** means randomly initializing embedding instead of the topic embedding $\boldsymbol{\beta}$. **w/o Semantics-driven Attention** stands for the variant excludes the attention scores calculated by discrete variables (i.e.,



(a) SemEval-A (b) SemEval-B (c) SemEval-C

Figure 3: Performance of Bert and DisBert with different amounts of training data of SemEval datasets.



(a) MPRC (b) Quora

(c) STS-B (d) SemEval-A

(e) SemEval-B (f) SemEval-C

Figure 4: The impact of topic number K on DisBert, where the horizontal axis shows the number of topics and the vertical axis shows the performance on STS datasets. Spearman correlation coefficient is reported in STS-B datasets and F1 scores are reported for other datasets.

$Q_q K_q^t$) in Eq (5). **w/o Output-enhanced** is the variant does not concatenate quantized variable Z^q with h^l in the output layer.

As shown in Figure 2 we have the following observation: (1) Topic information is informative for the STS task for both DisBert and DisRoBERTa. The performance of DisBert has dropped 0.96, 0.04, and 0.28 on MRPC datasets without the topic model, topic sensitive encoder, and topical latent embedding respectively. (2) The well-designed semantics-driven multi-head attention benefits the STS task. Without such attention, performance dropped for both DisBert and DisRoBERTa on both datasets. Meanwhile, when the output is equipped with quantized representations, performance is significantly improved, which further verifies that the learned topic-informed discrete latent variable can capture semantics and improve the STS task.

	DisBert		DisRoBERTa	
	Multistage	Joint	Multistage	Joint
MRPC	88.78	**89.06**	91.04	**91.15**
Quora	90.45	**90.81**	90.67	**91.81**
STS-B	85.32	**85.64**	88.89	**89.28**
SemEval-A	77.49	**77.57**	76.89	**77.01**
SemEval-B	54.21	**54.44**	56.95	**57.18**
SemEval-C	27.50	**27.61**	33.57	**33.79**

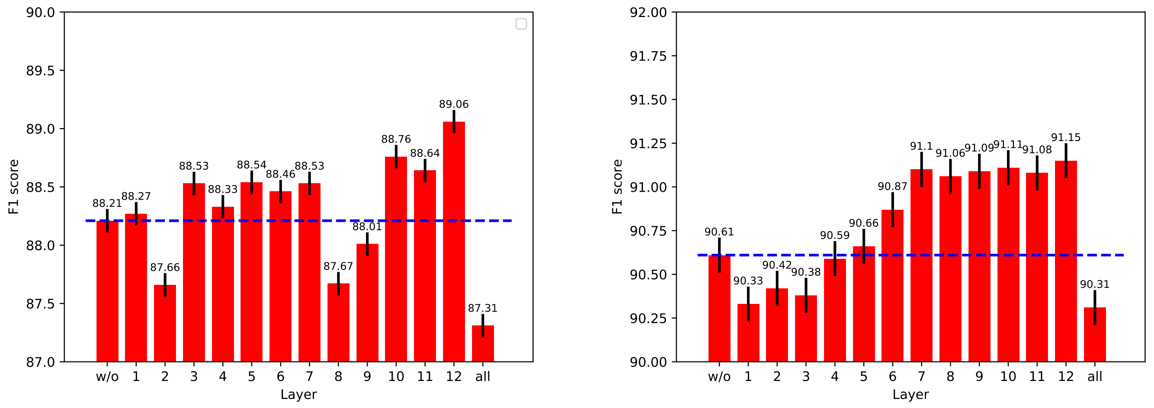
Table 3: Joint training evaluation on six datasets. Spearman correlation coefficient is reported in STS-B datasets and F1 scores are reported for other datasets.

4.6 Joint Training vs Multistage Training

After we pretrained the NTM, there are two ways of training our topic-enhanced VQ-VAE. One is to utilize the topic embeddings from the pretrained NTM as the codebook initialization in VQ-VAE and train the latter while holding the former fixed (denoted by **Multistage** in Table 3), the other is to joint train the pretrained NTM and VQ-VAE (denoted by **Joint**). The results of these two training methods derived from the six STS datasets are shown in Table 3. We can see that compared with multistage training, joint training yields better results on all datasets for both DisBert and DisRoBERTa. The comparison shows that jointly training the topic embedding matrix ϕ benefits the STS task.

4.7 Results with Varying Hyperparameters.

Impact of Training Data Amounts. In figure 3, we compared the results of Bert and DisBert with different training data. We randomly selected 20% to 100% data from the training set as training data. We can observe that DisBert outperforms Bert consistently across all training data sizes, which



(a) Bert-based

(b) RoBERTa-based

Figure 5: Performance of DisBert and DisRoBERTa with semantics-driven multi-head attention on different layers. **w/o** means the origin Bert/RoBERTa without aforesaid attention and **all** means such attention is applied to all layers.

Cluster	Words
7	dead protesters attack crash weapons kill killed
16	bus day transport by car bike has what
20	government vote year elections obama winner
26	in for you on next out it up before us with

Sentence Pair	Label
S1: Five killed$_{[7]}$ in$_{[26]}$ Belgian coach$_{[16]}$ crash$_{[7]}$. S2: Teenagers among$_{[26]}$ 5 dead$_{[7]}$ in$_{[26]}$ Belgian bus$_{[16]}$ crash$_{[7]}$.	3.8
S1: Mugabe declared$_{[20]}$ winner$_{[20]}$ of disputed elections$_{[20]}$ S2: Zimbabwe Mugabe declared$_{[20]}$ winner$_{[20]}$ in disputed vote$_{[20]}$	4.0

Table 4: Case Study on STS-B datasets.

suggests that we can leverage the topic-enhanced discrete variable in all data sizes, even when the training data is scarce.

Impact of Topic Numbers. Figure 4 shows the performance of DisBert given varying topic numbers. As we can see, the curves on all datasets are not monotonic and the best accuracy is achieved with different numbers of topics, e.g., $k = 30$ on the MRPC dataset. We also found that on larger training datasets DisBert requires larger topic numbers and vice versa, e.g., $k = 90$ for Quora and $k = 30$ for SemEval-B. It is not unexpected, as larger datasets can cover more topics. However, how to let the data choose the right number of topics in NTM is beyond the scope of this paper.

Impact of Bert Layers. Figure 5 shows the effect of semantics-driven multi-head attention on different layers. We found that the aforesaid attention is not always helpful for all layers and best performances are achieved when applying such attention to the last layer of DisBert/DisRoBERTa on MRPC datasets. Meanwhile, the performance drops when such attention is applied to all layers.

4.8 Case Studies

In Table 4, we conduct a case study on STS-B datasets to show how the model works with topical information. The left part of the table shows four word clusters generated by Topic-enhanced VQ-VAE. As described in Section 3.2, the topic-enhanced VQ-VAE model clusters words in the discrete space given by the codebook. It can be seen that words in the same cluster form a meaningful topic: Cluster 7 is about protest, Cluster 6 is about transportation, Cluster 20 is about government election, and Cluster 26 is one of the clusters capturing all the functional words. The right part of the table shows two pairs of sentences used in the STS tasks with the assignments of the four topics to their words. We can observe that sentences, which are predicted to be similar by our model, tend to have more words assigned to similar topics, even if those words are morphologically different. For instance, “Killed” and “dead” are both from Cluster 7, “coach” and “bus” are from Cluster 16, and “elections” and “vote” are from Cluster 20. The heat map calculated by semantics-driven multi-head attention is shown on Appendix A.3 for more discussion.

5 Conclusion

In this paper, we developed a topic-enhanced VQ-VAE model to effectively train discrete latent variables by informing the vector quantization with semantics (i.e., latent topics) learned from a broader context by a neural topic model. Then we further designed a semantics-driven multi-head attention mechanism for enriching the contextual embeddings learned by Transformer with topical information, which is calculated based on the quantized

Dataset	Topic Model	VQ-VAE	Semantics-driven Bert	Total	Bert
MRPC	247	218	220	685	200
Quora	3,512	5,123	18,232	26,927	17,805
STS-B	197	186	239	622	214

Table 5: Limitation—Training time (in seconds) of different models in three datasets. We trained the topic model for 500 epochs, VQ-VAE for 20 epochs, semantics-driven Bert, and Bert for 10 epochs. All models are trained in the same environment with a single GeForce RTX 3090 card.

representations from topic-enhanced VQ-VAE. Its plug-and-play characteristic allows it to be readily incorporated into various transformed-based language models. Through experiments on different scale datasets for the STS task, we proved that building the semantic information learned by NTM via vector quantization into the multi-head attention block can improve the STS performance of transformer-based language models.

6 Limitations

The limitations of this work, to the best of our knowledge, can be summarized into two aspects:

(1) Compared to the origin Bert/RoBERTa, our model DisBert/DisRoBERTa needs a longer time to train since there are three components in our framework, i.e., topic model, VQ-VAE model, and the transformer-based model. From table 5, we can observe that in a small dataset like MRPC and STS-B, the training time is almost three times longer. While in the large datasets Quora, our model took an extra 2.5 hours. Therefore in future work, we need to improve the efficiency of our model.

(2) The improvement of our method is limited when sentences are long enough or training datasets are large, e.g., Quora and SemEvalC shown in table 2. Our model is capable of complete global semantics, hence it works better for small datasets and short sentences which contain limited semantics.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61976102 and No.U19A2065) and the Science and Technology Development Program of Jilin Province (No.20210508060RQ), and the Fundamental Research Funds for the Central Universities, JLU.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *ACL*, pages 85–96.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *EMNLP*, pages 615–620.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *ACL*, pages 2406–2417.

Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. 2019. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *IWP*.

Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. 2020a. Recurrent hierarchical topic-guided rnn for language generation. In *ICML*, pages 3810–3821.

Daya Guo, Duyu Tang, Nan Duan, Jian Yin, Daxin Jiang, and Ming Zhou. 2020b. Evidence-aware inferential text generation with vector quantised variational autoencoder. In *ACL*, pages 6118–6129.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NeurIPS*, pages 2042–2050.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *EMNLP*, pages 718–724.

Shuning Jin, Sam Wiseman, Karl Stratos, and Karen Livescu. 2020. Discrete latent variable representations for low-resource text classification. In *ACL*, pages 4831–4842.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *ICML*, pages 2390–2399.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *ICLR*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *ICLR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arxiv*.

Ruiying Lu, Bo Chen, Dan dan Guo, Dongsheng Wang, and Mingyuan Zhou. 2022. Topic-aware contextualized transformers. In *ECIR*.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. 2017. Discovering discrete latent topics with neural variational inference. In *ICML*.

Yishu Miao, Lei Yu, and Phil Blunsom. 2016. Neural variational inference for text processing. In *ICML*.

Muhidin Mohamed and Mourad Oussalah. 2019. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. Semeval-2017 task 3: Community question answering. *SemEval-2017 workshop*.

Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, James Glass, and Bilal Randeree. 2015. Semeval-2015 task 3: Answer selection in community question answering. *SemEval-2015 workshop*.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 task 3: Community question answering. In *SemEval-2016 workshop*.

A Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *NeurIPS*, pages 6309–6318.

Nicole Peinelt, Dong Nguyen, and Maria Liakata. 2020. tBERT: Topic models and BERT joining forces for semantic similarity detection. In *ACL*, pages 7047–7055.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Xinyi Wang and YI YANG. 2020. Neural topic model with attention for supervised learning. In *AISTATS*, pages 1147–1156.

Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R Lyu, and Shuming Shi. 2019. Topic-aware neural keyphrase generation for social media language. In *ACL*, pages 2516–2526.

Zhengjue Wang, Zhibin Duan, Hao Zhang, Chaojie Wang, Long Tian, Bo Chen, and Mingyuan Zhou. 2020. Friendly topic assistant for transformer based abstractive summarization. In *EMNLP*, pages 485–497.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*, pages 4144–4150.

Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2021. Using prior knowledge to guide bert's attention in semantic textual matching tasks. *Proceedings of the Web Conference 2021*.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *TACL*.

Jichuan Zeng, Jing Li, Yan Song, Cuiyun Gao, Michael R Lyu, and Irwin King. 2018. Topic memory networks for short text classification. In *EMNLP*, pages 3120–3131.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *ACL*, pages 1441–1451.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *AAAI*, pages 9628–9635.

He Zhao, Lan Du, and Wray L. Buntine. 2017. A word embeddings informed focused topic model. In *ACML*.

He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. 2021a. Topic modelling meets deep neural networks: A survey. In *IJCAI Survey Track*.

He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray Buntine. 2021b. Neural topic model via optimal transport.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *EMNLP*, pages 1393–1398.

A Example Appendix

A.1 Dataset Examples

Table 6 shows examples from different datasets. Labels indicate if the second sentence is a paraphrase (for paraphrasing tasks) or relevant (for QA tasks). In STS-B datasets, the label stands for sentence pair similarity score, and the higher score, the more similar the two sentences are.

A.2 Hyper-Parameter settings

Table 7 shows the hyperparameter we chose in our model. All hyper-parameters were chosen through greedy search based on development set performance.

For topic model, we choose number of topics in (20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200), batch size in (128, 256, 512), and learning rate in (1e-3, 2e-3, 3e-3).

For the VQ-VAE model, we choose the batch size in (32, 64, 128), and the learning rate in (2e-5, 3e-5).

For the Bert model, We choose the batch size in (32, 64, 128), and the learning rate in (2e-5, 3e-5).

A.3 Heat Map

Figure 6 shows the heat map of sentence pair 1 in the case study, which is calculated by our semantics-driven multi-head attention. We can observe that words from the same cluster usually pay high attention to each other because they correspond to the same hidden vector in the codebook. Such attention could help Bert model focus on the words which may have similar semantics.

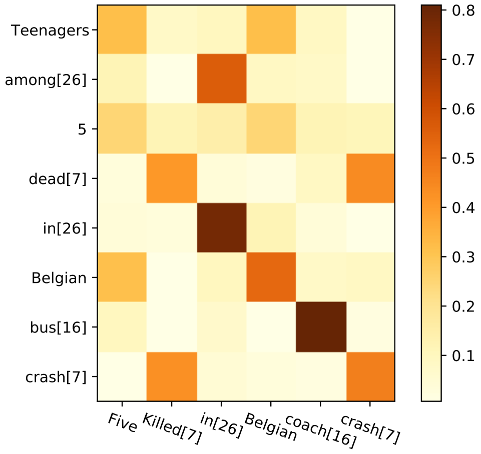


Figure 6: Heat Map calculated by semantics-driven multi-head attention.

Dataset	Sentence Pair	Label
MRPC	S1:The world's two largest automakers said their U.S. sales declined more than predicted last month as late summer sales frenzy caused more of an industry backlash than expected. S2: Domestic sales at both GM and No. 2 Ford Motor Co. declined more than predicted as a late summer sales frenzy prompted a larger-than-expected industry backlash.	1
Quora	S1: Currently , all Supreme Court Justices come from very elite law schools, is it similar for the best lawyers in private practice . S2: What 's your type of jungle -LRB- concrete or nature -RRB- and why ?	1
STS-B	S1: A man is spreading shreded cheese on a pizza. S2: A man is spreading shredded cheese on an uncooked pizza.	3.8
SemEval-A	S1: Massage oil is there any place i can find scented massage oils in qatar? S2: Whats the name of the shop?	0
SemEval-B	S1: Music tastes so; what kind of music do you like? S2: After school program and summer camp Can anyone please tell me if there is any after school program and summer camp in Doha? – for a grade two girl (seven years old).	0
SemEval-C	S1: Best sunglass store in Qatar Can somebody suggest the best store where i can get the best sunglasses for ladies. Where can i find a store with the best variety? S2: Fashion wear - Primark Lingerie - Agent Provocateur Lingerie - Intimo Lingerie - Peach John	1

Table 6: Dataset Examples.

Models	MRPC	Quora	STS-B	SemEval		
				A	B	C
Topic Model						
Topic number K	30	90	40	80	30	70
Vocabulary size V	5,179	37,013	4,369	12,102	3,763	13,272
Topic Embedding hidden size E	64	64	64	64	64	64
Batch size	256	512	256	256	256	256
Epochs	500	150	500	200	500	200
Learning rate	1e-3	2e-3	1e-3	1e-3	1e-3	1e-3
VQ-VAE Model						
Max sequence length	128	128	128	256	256	256
Vocabulary size	10,734	61,040	9513	31,060	8,259	31,060
Batch size	32	128	32	64	32	64
Epochs	10	10	10	10	10	10
Learning rate	2e-5	3e-5	2e-5	2e-5	2e-5	2e-5
DisBert/DisRoBERTa Model						
Max sequence length	128	128	128	256	256	256
Batch size	32	128	32	32	32	64
Epochs	10	5	10	10	10	10
Learning rate	2e-5	3e-5	2e-5	3e-5	2e-5	3e-5

Table 7: Parameter settings.