Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration

Shufan Wang and **Laure Thompson** and **Mohit Iyyer**
University of Massachusetts, Amherst
{shufanwang, laurejt, miyyer}@cs.umass.edu

Abstract

Phrase representations derived from BERT often do not exhibit complex phrasal compositionality, as the model relies instead on lexical similarity to determine semantic relatedness. In this paper, we propose a contrastive fine-tuning objective that enables BERT to produce more powerful phrase embeddings. Our approach (Phrase-BERT) relies on a dataset of diverse phrasal paraphrases, which is automatically generated using a paraphrase generation model, as well as a large-scale dataset of phrases in context mined from the Books3 corpus. Phrase-BERT outperforms baselines across a variety of phrase-level similarity tasks, while also demonstrating increased lexical diversity between nearest neighbors in the vector space. Finally, as a case study, we show that Phrase-BERT embeddings can be easily integrated with a simple autoencoder to build a phrase-based neural topic model that interprets topics as mixtures of words and phrases by performing a nearest neighbor search in the embedding space. Crowdsourced evaluations demonstrate that this phrase-based topic model produces more coherent and meaningful topics than baseline word and phrase-level topic models, further validating the utility of Phrase-BERT.

Model	Nearest neighbors of “pulls the trigger"
GloVe	his trigger, the trigger, a trigger
BERT	pulled the trigger, squeezed the trigger, scoots closer
Span-BERT	pulled the trigger, pulling the trigger, seize the day
Sent-BERT	pulling the trigger, pulled the trigger, the trigger
Phrase-BERT	**picks up his gun, squeezes off a quick burst of shots, takes aim**

Table 1: Nearest neighbors of the phrase “pulls the trigger”. While baselines rely heavily on lexical overlap, Phrase-BERT’s neighbors are both semantically similar and lexically diverse.

1 Introduction

Learning representations of phrases is important for many tasks, such as semantic parsing (Socher et al., 2011), machine translation (Ramisch et al., 2013), and question answering (Seo et al., 2018). While pretrained language models such as BERT (Devlin et al., 2018) have significantly pushed forward the state of the art on a variety of NLP tasks, they still struggle to produce semantically meaningful embeddings for shorter linguistic units such as sentences and phrases. An out-of-the-box BERT sentence embedding model often underperforms simple baselines such as averaging GloVe vectors in semantic textual similarity tasks (Reimers and Gurevych, 2019). Furthermore, Yu and Ettinger (2020) have shown that phrasal representations derived from BERT do not exhibit complex phrasal compositionality.

In this paper, we develop Phrase-BERT, which fine-tunes BERT using contrastive learning to induce more powerful phrase embeddings. Our approach directly targets two major weaknesses of out-of-the-box BERT phrase embeddings: (1) BERT never sees short texts (e.g., phrases) during pretraining, as its inputs are chunks of 512 tokens; and (2) BERT relies heavily on lexical similarity (word overlap) between input texts to determine semantic relatedness (Li et al., 2020; Yu and Ettinger, 2020; Zhang et al., 2019). To combat these issues, we automatically generate a dataset of lexically-diverse phrasal paraphrase pairs, and we additionally extract a large-scale dataset of 300 million phrases in context from the Books3 dataset from the Pile (Gao et al., 2020). We then use this paraphrase data and contextual information to fine-tune BERT with an objective that intuitively places phrase embeddings close to both their paraphrases and the contexts in which they appear (Figure 1).

Phrase-BERT outperforms strong baselines such as SpanBERT (Joshi et al., 2019) and Sentence-BERT (Reimers and Gurevych, 2019) across a suite of phrase-level semantic relatedness tasks. Additionally, we show that its nearest neighbor space exhibits increased lexical diversity, which signals that compositionality plays a larger role in its vector space (Table 1). Such phrasal diversity is an important component of models built for corpus exploration such as phrase-based topic modeling (Wang et al., 2007; Griffiths et al., 2007). To investigate Phrase-BERT's potential role in such applications, we integrate it into a neural topic model that represents topics as mixtures of words, phrases, and even sentences. A series of human evaluations reveals that our phrase-level topic model produces more meaningful and coherent topics compared to baseline models such as LDA (Blei et al., 2003) and its phrase-augmented variants. We have publicly released code and pretrained models for Phrase-BERT to spur future research on phrase-based NLP tasks.[1]

2 Related work

Our work relates to a long history of learning dense phrase representations, and in particular to approaches that leverage large-scale pretrained language models. Like most prior approaches, Phrase-BERT learns a *composition* function that combines component word embeddings together into a single phrase embedding. This function has previously been implemented with rule-based composition over word vectors (Yu and Dredze, 2015) and recurrent models (Zhou et al., 2017) that use a pair-wise GRU model using datasets such as PPDB (Pavlick et al., 2015). Other work learns *task-specific* phrase embeddings such as those for semantic parsing (Socher et al., 2011), machine translation (Bing et al., 2015) and question answering (Lee et al., 2021); in contrast, Phrase-BERT produces general-purpose embeddings useful for any task.

The advent of huge-scale pretrained language models such as BERT (Devlin et al., 2018) has opened a new direction of phrase representation learning. Yu and Ettinger (2020) highlight BERT's struggles to meaningfully represent short linguistic units (words, phrases). Several papers hypothesize that this is because BERT is trained on longer texts (512 tokens) and with a pairwise text objective that may be irrelevant for shorter texts (Reimers and Gurevych, 2019; Liu et al., 2019; Toshniwal et al., 2020). Without task-specific fine-tuning, the performance of BERT on phrases and sentences is often worse than simple baselines such as mean-pooling over GloVe vectors (Reimers and Gurevych, 2019; Li et al., 2020). Furthermore, Li et al. (2020) draw theoretical connections between BERT's pretraining objective and its non-smooth anisotropic semantic embedding space, which make it more reliant on lexical overlap to determine phrase and sentence similarity. Previously proposed methods to address these issues include predicting spans during pretraining instead of words (Joshi et al., 2019), fine-tuning BERT on shorter texts (Reimers and Gurevych, 2019), and adding an explicit postprocessing step to induce a continuous and isotropic semantic space (Li et al., 2020). As we show in the rest of this paper, Phrase-BERT produces more semantically meaningful phrase representations than these competing approaches while also promoting a lexically diverse vector space.

3 Phrase Embeddings from BERT

We design two separate fine-tuning tasks on top of BERT to improve its ability to produce meaningful phrase embeddings. Since the pretrained BERT model relies heavily on lexical overlap to determine phrase similarity, our first fine-tuning objective relies on an automatically generated dataset of **lexically diverse phrasal paraphrases** to encourage the model to move beyond string matching. The second objective encourages the model to encode contextual information into phrase embeddings by relying on a phrase-in-context dataset we extract of **phrases in context** from the huge-scale Books3 corpus (Gao et al., 2020). In both cases, we rely on contrastive objectives similar to Sentence-BERT (Reimers and Gurevych, 2019) for fine-tuning (Fig. 1).

Using BERT to embed phrases: Given an input phrase X of length N tokens, we compute a representation $\boldsymbol{x}$ by averaging the final-layer token-level vectors yielded by BERT (Devlin et al., 2018)[2] after passing the tokens of X to the model as input: $\boldsymbol{x} = \sum_{i=1}^{N} \text{BERT}(X_i)/N$. As all of BERT's pretraining examples are 512 tokens long, the above method is reliable for short documents, but it struggles to model the semantics of words and phrases,

[1] https://github.com/sf-wa-326/phrase-bert-topic-model

[2] We use the 12-layer BERT-base-uncased for all experiments.

as shown by Yu and Ettinger (2020) and also by our evaluations in Section 4.1.

Creating lexically diverse phrase-level paraphrases: Our first fine-tuning objective encourages BERT to capture semantic relatedness between phrases without overly relying on lexical similarity between those phrases. To accomplish this, we create a dataset by extracting 100K phrases from WikiText-103 (Merity et al., 2017) using the shift-reduce parser from CoreNLP (Manning et al., 2014).[3] Then, given a phrase p "complete control" from the sentence "The local authorities have complete control over the allocation of building materials", we create a positive example p^+ by passing p through the GPT2-based *diverse* paraphrasing model released by Krishna et al. (2020). This model was trained by fine-tuning GPT2-large (Radford et al., 2019) on a filtered version of the PARANMT-50M sentence-level paraphrase dataset (Wieting and Gimpel, 2017), using an encoder-free seq2seq modeling approach as proposed by Wolf et al. (2019).

We decode from this model using nucleus sampling with the nucleus probability mass of 0.8 (Holtzman et al., 2019), applying lexical constraints to avoid producing any non-stopword tokens that also occur in p. This yields phrases such as "full power of the system", which are quasi-paraphrases of p with no lexical overlap. We create a negative example p^- by first randomly sampling a non-stopword from p and then replacing it with a random token from the vocabulary. In the case of "complete control", we might sample "complete" and replace it with a randomly selected token "fluid". Then, we feed the corrupted phrase into the paraphraser and decode just as we did to produce the positive example, which removes lexical overlap but preserves the distorted meaning. This produces phrases like "no change to the water level", which has no semantic relation to p.

Collecting phrases in context: The above dataset focuses exclusively on phrases *out of context*. In other words, a model trained to distinguish negative phrases from positive phrases does not observe any surrounding context in which these phrases are used. As these contexts also provide useful information about the meaning and usage of phrases, we create a second dataset to inject contextual information into BERT's phrase embeddings. Concretely, we extract phrases along with their surrounding context from the Books3 Corpus (Gao et al., 2020), a large-scale 100GB collection of books spanning a variety of topics and genres. As before, we extract phrases by running constituency parsing on a random subset of the dataset; we remove all phrases that are more than ten tokens long and then select the top 100K most frequent phrases. We also store a single positive context c^+ of length 120 tokens in which p occurs, replacing the occurrence of p within c^+ with a MASK token.

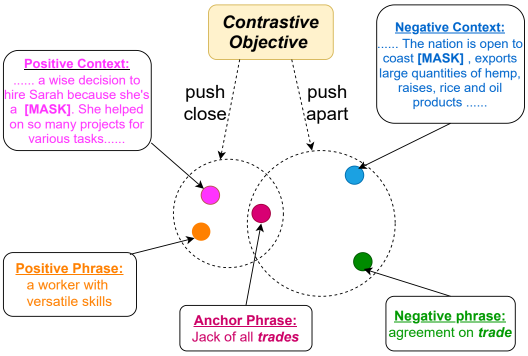
Contrastive Objective
push close
push apart
Positive Context:
...... a wise decision to hire Sarah because she's a [MASK]. She helped on so many projects for various tasks......
Negative Context:
...... The nation is open to coast [MASK] , exports large quantities of hemp, raises, rice and oil products ......
Positive Phrase:
a worker with versatile skills
Anchor Phrase:
Jack of all trades
Negative phrase:
agreement on trade

Figure 1: We propose Phrase-BERT to correct BERT's embedding space for phrases by placing semantically similar phrases and contexts closer together (e.g., the anchor phrase "Jack of all trade" and the positive paraphrase "a worker with versatile skills") and by removing lexical cues with a contrastive learning objective.

3.1 Fine-tuning BERT with a contrastive objective using the constructed datasets

We fine-tune BERT on both datasets with the same contrastive objective, following similar procedures as Sentence-BERT (Reimers and Gurevych, 2019). For the first dataset, we encourage the model to produce similar embeddings for p and p^+ while pushing the embeddings for p and p^- far apart. We embed each phrase in the triplet (p, p^+, p^-) by mean-pooling BERT's token-level representations as described previously, which gives us three vectors $(\boldsymbol{p}, \boldsymbol{p}^+, \boldsymbol{p}^-)$ Then, we compute the following triplet loss:

$$J = \max(0, \epsilon - \|\boldsymbol{p} - \boldsymbol{p}^-\| + \|\boldsymbol{p} - \boldsymbol{p}^+\|) \quad (1)$$

where $\|\cdot\|$ denotes the L2 norm and ϵ is a margin (set to 1 in our experiments).

Similarly, for the second dataset, we compute the triplet loss, $\max(0, \epsilon - \|\boldsymbol{p} - \boldsymbol{c}^-\| + \|\boldsymbol{p} - \boldsymbol{c}^+\|)$,

[3] We extract all NP, VP, ADJP, and ADVP phrases and then filter to select the most frequent 100K. More details on this process can be found in the Appendix §A.1.

for each data instance $(\boldsymbol{p}, \boldsymbol{c}^+, \boldsymbol{c}^-)$, or embedding vectors encoded by Phrase-BERT from the phrase-context triple (p, c^+, c^-), where c^- is a randomly sampled context.

Implementation details: We fine-tune Phrase-BERT on an NVIDIA RTX 2080ti GPU for 1 epoch. We use a batch size of 16 and optimize using Adam (Kingma and Ba, 2014) with a learning rate of $2e-5$. The initial 10% of training steps are used as warm-up steps, following the linear warm-up schedule used by Reimers and Gurevych (2019).

4 Experimental setup

We evaluate our phrase embeddings on a diverse collection of phrase-level semantic relatedness tasks following previous works on evaluating phrase embeddings (Turney, 2012; Yu and Dredze, 2015; Asaadi et al., 2019; Yu and Ettinger, 2020). Due to a lack of benchmarks like SentEval (Conneau et al., 2018) at the phrase level, we create filtered versions of some datasets by removing lexical overlap cues.

4.1 Datasets

We compare the performance of Phrase-BERT against baselines on a variety of phrases tasks involving phrases of different length and types.

Turney: The dataset of Turney (2012) contains 2,180 examples that test bigram compositionality by asking models to select which of five *unigrams* is most similar in meaning to a given bigram.

BiRD: The bigram-relatedness judgment dataset (Asaadi et al., 2019) is a correlation task that consists of 3,455 pairs of bigram phrases, each of which has a corresponding human rating of similarity between 0 and 1.

PPDB: We create a paraphrase classification dataset from PPDB 2.0 (Pavlick et al., 2015) that contains 23,364 phrase pairs[4] by sampling examples from PPDB-small, the highest-quality subset of PPDB 2.0 according to human raters. Given a pair, we apply the paraphrase classification model described later in this section to input phrase embeddings to determine if the inputs are paraphrases. Negative examples are created by randomly sampling phrases from the dataset. The average phrase length in this dataset is 2.5 tokens.

Dataset	Example	Label
Turney	(learned person, pundit)	“match”
BiRD	(business development, economic growth)	0.586
PPDB-filtered	(global affairs, world affairs)	“positive”
	(world affairs, domestic affairs)	‘negative”
PPDB	(actively participate, play an activate role)	“positive"
PAWS-short	(a variable version of the basic lyrics, a basic version of the variable lyrics)	“negative"

Table 2: Datasets and examples used in our phrase embedding evaluation.

PPDB-filtered: Since the above PPDB dataset contains a large amount of lexical overlap between paraphrase pairs, it can be solved with superficial heuristics. We follow Yu and Ettinger (2020) by creating a more challenging version by filtering out lexical overlap cues. Specifically, we control the amount of word overlap in each positive and negative pair to be exactly the same. We also ensure that each overlapping token in a pair occurs in both positive and negative pairs so that the model cannot rely on cues from word identity. This dataset has 19,416 phrase pairs.

PAWS-short: The previous datasets test include mainly bigrams and short phrases, motivating us to evaluate our models on a dataset with longer text. PAWS is a challenging dataset for paraphrase classification on text pairs where even negatives have high lexical overlap (Zhang et al., 2019). However, it contains sentences and short paragraphs in addition to phrases. We filter PAWS to only include examples shorter than 10 tokens in length while ensuring class balance between paraphrase and non-paraphrase pairs. We follow the split released by the authors and extract 1,300 total examples, with an average length of 9.4 tokens.

4.2 Baselines

We compare Phrase-BERT against phrase embeddings derived from baselines that include averaged GloVe vectors[5] as well as the base versions of BERT (Devlin et al., 2018), SpanBERT (Joshi et al., 2019), and Sentence-Bert (Reimers and Gurevych, 2019). Except for GloVe and Span-BERT, We ob-

[4] We use a 70/15/15 train/dev/test split.

[5] https://nlp.stanford.edu/projects/glove/

Model	Turney	BiRD	PPDB-filtered	PPDB	PAWS-short
GloVe	37.8	0.560	44.2	47.2	50.0
BERT	42.6	0.444	60.1	86.2	50.0
SpanBERT	38.7	0.258	57.3	95.1	50.1
Sentence-BERT	51.8	0.687	64.2	95.8	50.0
Phrase-BERT-phrase	55.4	0.682	68.0	96.9	50.0
Phrase-BERT-context	55.0	0.672	65.0	95.7	49.2
Phrase-BERT	**57.2**	**0.688**	**68.0**	**97.6**	**58.9**

Table 3: Phrase-BERT outperforms other baselines on all phrase-level semantic relatedness datasets. The improvements on PPDB-filtered and PAWS-short show that Phrase-BERT captures phrase semantics without over-reliance on lexical overlap.

tain phrase embeddings from GloVe by averaging pretrained token embeddings; for Span-BERT, we use the concatenation of the phrase boundary representations following Joshi et al. (2019). we use the mean-pooled representation over the final-layer outputs from these models as phrase representations, following the observation by Reimers and Gurevych (2019) and Yu and Ettinger (2020) that this method outperforms other possibilities (e.g., using the [CLS] representation). We also compare the full Phrase-BERT model with two ablated versions: Phrase-BERT-phrase (removing the phrase-context fine-tuning) and Phrase-BERT-context (removing the phrase-level paraphrase fine-tuning).

Paraphrase classification model: The paraphrase classification datasets (PPDB-filtered, PPDB, and PAWS-short) require task-specific fine-tuning. We use the same setup as Adi et al. (2016) and Yu and Ettinger (2020). In short, we add a simple classifier on top of the concatenated embedding of a phrase pair, implemented using an multilayer perceptron with a hidden layer of size 256 and an ReLu activation before the classification layer.

5 Results & Discussion

In this section, we highlight takeaways from our results on the phrase-level semantic relatedness benchmarks as well as measurements of lexical diversity. We also provide an ablation study that confirms the benefits of both fine-tuning objectives.

5.1 Phrase-BERT effectively captures phrase semantics

From Table 3, we observe that Phrase-BERT consistently outperforms BERT and other baseline models across all five evaluation datasets. Among the baselines, Sentence-BERT also yields notable improvements over BERT, demonstrating the relationship between phrase and sentence-level semantics. However, Phrase-BERT still outperforms Sentence-BERT, especially in tasks where the input is very short, such as the phrase-unigram tasks from Turney. Moreover, despite the masked span prediction objective of SpanBERT, which intuitively may increase its ability to represent phrases, the model consistently underperforms on all tasks.

5.2 Phrase-BERT does not rely solely on lexical information to understand phrases.

Several previous works report that pretrained transformer-based representations overly use lexical overlap to determine semantic relatedness (Yu and Ettinger, 2020; Li et al., 2020; Reimers and Gurevych, 2019). Our experiments quantitatively show that for both short and long phrases, BERT and other baselines heavily rely on lexical overlap and not compositionality to encode phrase relatedness. Despite high accuracies on the full PPDB dataset (where examples with lexical overlap are not filtered out), baselines significantly underperform Phrase-BERT on the two datasets in which lexical overlap cues are removed for paraphrase classification (PPDB-filtered, PAWS-short). On the other hand, Phrase-BERT's strong across-the-board performance demonstrates that it is able to go beyond string matching. Additionally, both of Phrase-BERT's objectives are complementary: Phrase-BERT-phrase (trained with paraphrase data only) and Phrase-BERT-context (trained with context data) are both consistently worse than Phrase-BERT.

5.3 Evaluating lexical diversity in the phrase embedding space

For many practical use cases of phrase embeddings (e.g, corpus exploration, or tracking how phrasal semantics change over time), it is useful to visualize the nearest neighbors of particular phrases (Mikolov et al., 2013; Dieng et al., 2019; Bommasani et al., 2020). However, if these nearest neighbors contain heavy lexical overlap, they may not add any new information and may miss important meaning from phrases. For example, the phrase “natural language processing” has no lexical overlap with “computational linguistics”, but both should be nearest neighbors. To measure this, we obtain the top-10 nearest neighbors for a query phrase in the embedding space and measure the lexical diversity within this set.[6] We report three different metrics: (1) the **percentage of unique unigrams** in each of the phrase’s nearest neighbors normalized by the phrase’s length, which is inspired by sentence-level translation diversity metrics (Vijayakumar et al., 2018); (2) **LCS-precision**, which measures the longest common substring between the source phrase and the top-k nearest neighbors (lower = more diverse); and (3) the average **Levenshstein edit distance** (Levenshtein, 1966) between a phrase and each of its top-k nearest neighbors. Table 4 shows that Phrase-BERT exhibits slightly higher lexical diversity than Sentence-BERT, which is the most competitive model on semantic relatedness tasks after Phrase-BERT.

5.4 Ablating the two objectives

As shown in Table 3, Phrase-BERT-phrase also performs reasonably well in many phrase semantics tasks, especially the PPDB paraphrase classification tasks. However, without training on the context data, which is much longer (128 tokens), it underperforms on the PAWS-short dataset, which consists of longer inputs. Phrase-BERT-phrase is also worse at inducing a lexically diverse embedding space, as indicated by the high LCS-precision. Meanwhile, fine-tuning using only the context objective (Phrase-BERT-context) yields the highest lexical diversity (Table 4) at the cost of a worse semantic space, which is perhaps because of the diverse content in the extracted contexts.

6 Using Phrase-BERT for topic modeling

We have shown that Phrase-BERT produces meaningful embeddings of variable-length phrases and a lexically diverse nearest neighbor space. In this section, we demonstrate Phrase-BERT’s utility in the downstream application of phrase-based corpus exploration. Capturing both phrase semantics and phrasal diversity is an important aspect for topic models that incorporate phrases in topic descriptions (Wang et al., 2007; Griffiths et al., 2007; El-Kishky et al., 2014). We show that Phrase-BERT can be integrated with an autoencoder model to build a phrase-based neural topic model (PNTM). Despite its simple architecture, PNTM outperforms other topic model baselines in our human evaluation studies in terms of topic coherence and topic-to-document relatedness (Table 5).

Model	**% new tokens**	**LCS-precision**	**Levenshtein-Distance**
Sentence-BERT	5.0	51.1	8.5
Phrase-BERT	5.3	47.6	8.7
Phrase-BERT-phrase	4.8	52.3	8.2
Phrase-BERT-context	5.4	44.8	9.0

Table 4: Lexical diversity among the top-10 nearest neighbors in the phrase embedding space.

the high seas fleet, wartime, kamikaze, the imperial japanese navy, the outbreak of world war ii, guadalcanal
an award, critically acclaimed, woman of the year, best actor, awards and nominations, the winner, best actress
hindi, the indian ocean, subcontinent, the central bay of bengal, bengali, india ’s, bihar
rhythmic, monosyllable, beats, the song ’s composition, drumbeat, rhythmically, the song ’s lyrics
stalking, the mystery, paranormal, linked to the paranormal, fox mulder david duchovny, cases linked to the paranormal, the conspiracy, mulder and scully
a separate species, phylogenetic, taxonomic, clade, a genus, taxonomical, phylogenetically

Table 5: A sample of six topics induced by PNTM with $K = 1000$ on Wikipedia. Topics are interpreted as mixtures of words and phrases, which enables fine-grained exploration of document collections.

6.1 Building a topic model with Phrase-BERT

We integrate Phrase-BERT into previous unigram-based neural topic models (Iyyer et al., 2016; Akoury et al., 2020) that try to reconstruct a document representation through an interpretable bottleneck layer. Unlike prior implementations, computing text representations using Phrase-BERT allows us to produce high quality topic descriptions (with a mixture of words and phrases) using a simple nearest neighbor search in the embedding space [7].

[6] We randomly choose source phrases from the Storium dataset (Akoury et al., 2020), which contains a diverse set of stories that does not appear in either the pretraining or fine-tuning data of Phrase-BERT, and use a vocabulary of 125K most frequent words and phrases from this dataset to compute the nearest neighbors.

[7] We also conduct experiments of training other versions PNTM, replacing the embedding function Phrase-BERT by

6.1.1 Model description

The bottleneck layer in our PNTM is implemented through a linear combination of rows in a $K \times d$ dimensional topic embedding matrix $\mathbf{R}$, where K denotes the number of topics, each row of $\mathbf{R}$ corresponds to a different topic's embedding and $d = 768$. Concretely, assume we have an input document X with tokens $X_1, X_2, \dots, X_n$. We encode the document by passing its tokens through Phrase-BERT to obtain a single vector representation $\boldsymbol{x}$. We then score $\boldsymbol{x}$ against K different learned topic embeddings,[8] which produces a distribution t over topics: $t = \text{softmax}(\mathbf{R}\boldsymbol{x})$. Given the distribution t, we then compute a reconstructed vector $\tilde{\boldsymbol{x}}$ as a linear combination of the rows in $\mathbf{R}$: $\tilde{\boldsymbol{x}} = \mathbf{R}^{\mathsf{T}}t$.

Intuitively, we want the model to push $\tilde{\boldsymbol{x}}$ as close to the input $\boldsymbol{x}$ as possible as this forces salient information to be encoded into the rows of $\mathbf{R}$. We accomplish this through optimizing a contrastive loss function, which minimizes the inner product between $\tilde{\boldsymbol{x}}$ and $\boldsymbol{x}$ while maximizing the inner product between $\tilde{\boldsymbol{x}}$ and the representation $\boldsymbol{z}$ of some randomly sampled document Z:

$$L(\theta) = \sum_{i}^{N} \max(0, 1 - \tilde{\boldsymbol{x}} \cdot \boldsymbol{x} + \boldsymbol{x} \cdot \boldsymbol{z}_i), \qquad (2)$$

where the summation is over N negative documents (N=5 in our experiments), and θ denotes the model parameters. Finally, to discourage duplicate topics, we follow Iyyer et al. (2016) by adding an orthogonality penalty term $H(\theta) = \|\mathbf{R}\mathbf{R}^T - \mathbf{I}\|$, where $\mathbf{I}$ is the identity matrix.

6.1.2 Interpreting learned topic embeddings

After training PNTM over all of the documents in a target collection, we obtain topics (lists of words and phrases that are most closely associated with a particular topic embedding) by performing a nearest neighbor computation with items in the vocabulary. Assume we have a vocabulary V of words and phrases derived from the target collection (the phrase extraction process is detailed in appendix A.2). We pass each item in our vocabulary through Phrase-BERT which is trained to place words and phrases of variable length in the same semantically-meaningful vector space. The resulting vectors then form an embedding matrix $\mathbf{L}$ of size $|V| \times d$ whose rows contain the corresponding output of the Phrase-BERT function. We can efficiently vectorize the topic interpretation by computing $\mathbf{R}\mathbf{L}^{\mathsf{T}}$, which results in a $K \times |V|$ matrix where each row corresponds to the inner products between a topic embedding and the vocabulary representations.

6.2 Human evaluations on learned topics

We compare Phrase-BERT against a slate of both neural and non-neural topic model baselines, including prior phrase-based topic models, on three datasets from different domains, using various topic sizes. Overall, we identify three key takeaways from our experiments: (1) Phrase-BERT produces more coherent topics than other phrase-based topic models and is competitive with word-level topic models; (2) Phrase-BERT's topics remain coherent even with large numbers of topics (e.g., 500-1000), unlike word-level models; and (3) despite the increase in vocabulary, Phrase-BERT's assignment of topics to documents is not impaired.

6.2.1 Experiment Setup

We experiment with three datasets (denoted as **Wiki**, **Story**, and **Reviews**) across three different domains (Wikipedia (Merity et al., 2017), fictional stories (Akoury et al., 2020), and online user product reviews (He and McAuley, 2016)). The datasets differ considerably in terms of document length and vocabulary. [9]

We compare PNTM against four strong topic modeling baselines, two of which also incorporate phrases into topic interpretation (all neural models were trained on a single Nvidia RTX 2080Ti GPU in less than 3 hours):

LDA: LDA is a popular probabilistic generative model that learns document-topic and topic-word distributions (Blei et al., 2003). We use Mallet (McCallum, 2002) with a parameter update every 20 intervals for 1000 total passes. We empirically observe that these hyperparameters produce the highest quality topics.

pLDA: Mimno (2015) incorporate phrases into LDA by simply converting them into unique

other composing functions such as BERT and SpanBERT. This leads to issues such as incoherence and the lack of lexical diversity in topic descriptions, further highlighting the strength of Phrase-BERT in capturing phrase semantics. Examples of topics obtained with these models are provided in Appendix §A.5

[8] This computation is identical to a dot product attention mechanism (Bahdanau et al., 2014).

[9] Details of the datasets can be found in the Appendix §A.2.

word types (e.g., "critically acclaimed" to "critically_acclaimed") and then run LDA as usual. We denote this method as *phrase-LDA* (pLDA). [10]

TNG: We also compare our approach to the Mallet implementation of the topical n-gram model (TNG) of Wang et al. (2007), which learns to associate documents with topics while inducing a combined unigram and phrase vocabulary.

UNTM: Finally, we implement a unigram version of PNTM, setting the word embedding function to simply average pretrained GloVe vectors. This model was also used by Akoury et al. (2020) and is based on the dictionary learning autoencoder originally proposed by Iyyer et al. (2016).

Following Chang et al. (2009), we perform two different sets of human evaluation experiments on Amazon Mechanical Turk: (1) **word intrusion**, which measures topic coherence, and (2) **topic-to-document relatedness**, which evaluates whether a topic assigned to a document is actually relevant. We implement the **word intrusion** task by giving crowd workers a list of six words or phrases and asking them to choose which one does not belong with the rest. The intruder is a highly-probable word or phrase sampled from a *different* topic. We evaluate topic coherence through the *model precision* metric (Chang et al., 2009), which is simply the fraction of judgments for which the crowd worker correctly chose the intruder, averaged over all of the topics in the model. Similarly, for the **topic-to-document relatedness** task, we present crowdworkers with a passage from Wikipedia and two topics from the model, one of which is the most probable topic assigned by the model. Then, we ask them to choose which topic best matches the passage and report the fraction of workers agreeing with the model.[11]

6.2.2 Topic modeling results

PNTM produces more coherent topics than other phrase-based topic models. Table 6 contains the results of the word intrusion task run over all three datasets. In these experiments, we set the number of topics K to 50 for all models and use the same vocabulary for pLDA and PNTM (as TNG induces phrases, we cannot control for its vocabulary). Compared to the other phrase-based models (TNG and pLDA), PNTM achieves substantially higher model precision. Notably, both neural topic models achieve higher model precision than LDA-based counterparts, and UNTM yields slightly more coherent topics than PNTM.[12]

Model	Wiki	Story	Reviews	Average
PNTM	**83.3**	**76.7**	**77.3**	**79.1**
pLDA	55.3	48.7	50.7	51.6
TNG	37.3	58.7	60.0	52.1
UNTM	76.9	70.0	62.0	69.6
LDA	48.7	48.0	52.7	49.8

Table 6: PNTM achieves higher model precision on word intrusion than other phrase-based topic models (top), and its topics are of similar quality to word-level neural topic models (UNTM).

PNTM maintains high topic coherence when trained with more topics. One conceivable advantage of PNTM is that incorporating phrases into its vocabulary allows it to model topics at a finer granularity than unigram models. To test this hypothesis, we compare PNTM to its unigram-level neural counterpart UNTM across different values of K ranging from 50 to 1000. Figure 2 plots the model precision derived from word intrusion experiments from both systems on Wikipedia, and Table 5 includes six topics sampled from the $K = 1000$ model. While model precision is similar between the two models when $K = 50$, PNTM produces higher quality topics as K increases. This increase in topic quality in PNTM signals that incorporating phrases into topic descriptions enable more topics to capture coherent and meaningful information.

PNTM is competitive with existing topic models on topic-to-document relatedness. Table 8 shows that all five models achieve similar results with TNG performing slightly worse than the rest. A worker accuracy of close to 90% signals that PNTM topics are assigned to relevant documents. Overall, PNTM is the only phrase-based model to achieve high scores in both word intrusion and topic-to-document relatedness tasks, showing that

[10] We use the same phrase vocabulary used for our PNTM model (Table 9) and the same hyperparameters as LDA.

[11] For all crowd experiments, we obtain three judgments per example, and we only allow qualified workers from English-speaking countries to participate. We restrict our tasks to workers with at least 97% HIT approval rate and more than 1,000 HITs approved on Mechanical Turk. Workers are paid $0.07 per judgment, which we estimate is a roughly $10-12 hourly wage.

[12] Two or more workers agree on the same choice 90.5% of the time, indicating high degree of agreement; more details are in Appendix Table 10

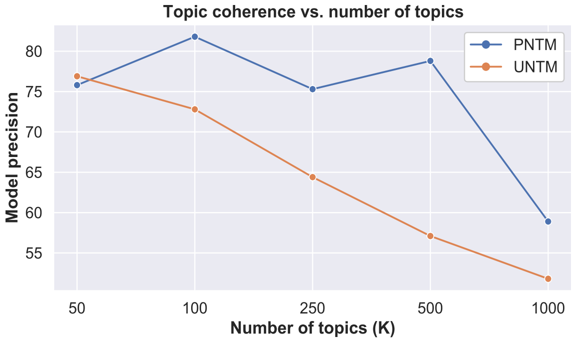

Figure 2: Word intrusion experiments show that PNTM achieves higher model precision (y-axis) than UNTM when the number of topics (K) is increased.

it learns higher quality topics without sacrificing relevance. The Krippendorf's α in Table 8 is a reliability statistic that measures inter-annotator agreement.

PNTM exhibits topic correspondence between related datasets. We observe that the topics induced by PNTM on different but related datasets have correspondences when trained using the same random seed. More concretely, each topic in PNTM is associated with an index (denoting the corresponding row of the $\mathbf{R}$ matrix), and we observe correspondences between topics with the same index trained on different datasets. Table 7 contains examples of this phenomenon; for instance, a topic on nightlife from a model trained on the "fantasy classic" Storium genre contains *bar, drinking, the tavern*, while the topic with the same index from a model trained on "occult pulp horror" stories contains *nightclub, clubbing, partygoer*. This ability provides practitioners with potentially new ways of exploring and comparing different collections of text, and it is not something easily implemented within LDA-based models. We theorize that such correspondences are possible because the learned topic embeddings do not move far away from their random initializations, which could be an effect of the orthogonality regularization.[13]

PNTM also exhibits other useful properties such as the ability to interpret topics with phrases of various length (including even sentences). Qualitative examples of these phenomena are provided in Appendix §A.4

Genre	Corresponding topics
Fantasy Classic	curious, scanning, surveying, his vision, being watched
Space Adventure	sensor, inspect, monitoring, check it out, spectrographic
Fantasy Classic	bar, drinking, tavern, bartender, the tavern
Pulp Horror	nightclub, clubbing, partygoer, nightlife, his drink

Table 7: Examples of topic correspondences between models trained on different genres of stories. Matching topics have the same index in the $\mathbf{R}$ matrix.

Model	Accuracy	Krippendorf's α
PNTM	89.3	0.7084
pLDA	89.3	0.6259
TNG	78.7	0.4876
UNTM	80.7	0.7981
LDA	90.0	0.7084

Table 8: Results of our topic-to-document relatedness experiments. PNTM achieves competitive worker accuracy to all other models, which indicates that its topic distribution assignments are not harmed by the inclusion of phrases to its vocabulary.

7 Conclusion

We propose Phrase-BERT, which induces powerful phrase embeddings by fine-tuning BERT with two contrastive objectives on datasets of lexically diverse phrase-level paraphrases and phrases-in-context. Phrase-BERT consistently outperforms strong baseline models on a suite of phrasal semantic relatedness tasks, even when lexical overlap cues are removed. These results suggest that Phrase-BERT looks beyond simple lexical overlap to capture complex phrase semantics. Finally, we integrate Phrase-BERT into a neural topic model to enable phrase-based topic interpretation, and show that the resulting topics are more coherent and meaningful than competing methods.

Ethical considerations

For all datasets and experiments, we use publicly available datasets from sources such as Wikipedia, Storium stories, and Amazon public reviews. We respect the privacy of all data contributors. In all crowdsourced evaluations, we strive to pay Mechanical Turkers with competitive payments.

We modify BERT embeddings in this project. Pretrained language models such as BERT are known to produce embeddings that raise ethical

[13] The average L2 distance between the learned topic vectors and their random initializations is 2.72, while the average L2 distance amongst the learned topic vectors themselves is 3.65.

concerns such as gender (Gala et al., 2020) and racial biases (Merullo et al., 2019; Bommasani et al., 2020), and can also output other offensive text content. Practitioners may consider employing a post-processing step to filter out potentially offensive content before releasing the final output.

Acknowledgments

We are grateful to Kalpesh Krishna, Xiang Lorraine Li, Varun Manjunatha, Simeng Sun, Zhichao Yang, and to the UMass NLP group for many helpful discussions. We would also like to thank He He for recommending the use of PAWS as evaluation data, Tu Vu for the valuable suggestions on paper writing, and the anonymous reviewers for their insightful feedback. SW and MI were supported by award IIS-1955567 from the National Science Foundation (NSF).

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks.

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *emnlp*.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. 2019. Big BiRD: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 505–516, Minneapolis, Minnesota. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1587–1597, Beijing, China. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. 2009. Reading tea leaves: How humans interpret topic models.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loic Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adji B Dieng, Francisco J R Ruiz, and David M Blei. 2019. Topic modeling in embedding spaces. *arXiv preprint arXiv:1907.04907*.

Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R. Voss, and Jiawei Han. 2014. Scalable topical phrase mining from text corpora. *Proc. VLDB Endow.*, 8(3):305–316.

Dhruvil Gala, Mohammad Omar Khursheed, Hannah Lerner, Brendan O'Connor, and Mohit Iyyer. 2020. Analyzing gender bias within narrative tropes. In *Workshop on NLP and CSS at EMNLP*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. 2007. Topics in semantic representation. *Psychological review*, 114(2):211.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *North American Association for Computational Linguistics*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Empirical Methods in Natural Language Processing*.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. In *Association for Computational Linguistics (ACL)*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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. *CoRR*, abs/1907.11692.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. *http://mallet. cs. umass. edu*.

Stephen Merity, Caiming Xiong, James Bradbury, and R. Socher. 2017. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843.

Jack Merullo, Luke Yeh, Abram Handler, Alvin Grissom II, Brendan O'Connor, and Mohit Iyyer. 2019. Investigating sports commentator bias within a large corpus of american football broadcasts. In *Empirical Methods in Natural Language Processing*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

David Mimno. 2015. Using phrases in mallet topic models.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Carlos Ramisch, Aline Villavicencio, and Valia Kordoni. 2013. Introduction to the special issue on multiword expressions: From theory to practice and use. *ACM Trans. Speech Lang. Process.*, 10(2).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Minjoon Seo, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2018. Phrase-indexed question answering: A new challenge for scalable document comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 559–564, Brussels, Belgium. Association for Computational Linguistics.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. ICML'11, page 129–136, Madison, WI, USA. Omnipress.

Shubham Toshniwal, Haoyue Shi, Bowen Shi, Lingyu Gao, Karen Livescu, and Kevin Gimpel. 2020. A cross-task analysis of text span representations. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 166–176, Online. Association for Computational Linguistics.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *J. Artif. Int. Res.*, 44(1):533–585.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Seventh IEEE international conference on data mining (ICDM 2007)*, pages 697–702. IEEE.

John Wieting and Kevin Gimpel. 2017. Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *arXiv preprint arXiv:1711.05732*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers.

Mo Yu and Mark Dredze. 2015. Learning composition models for phrase embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

Zhihao Zhou, Lifu Huang, and Heng Ji. 2017. Learning phrase embeddings from paraphrases with GRUs. In *Proceedings of the First Workshop on Curation and Applications of Parallel and Comparable Corpora*, pages 16–23, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Dataset	# Docs	# Words	# Phrases	Tok/doc
Wikipedia	304K	47.2K	75K	396
Storium	419K	44.0K	75K	190
Reviews	10K	32.4K	75K	101

Table 9: Corpus statistics for our three datasets, including the number of unique word and phrase types in our precomputed vocabulary. Note that we cap the number of unique phrases to the 75K most frequent.

A Appendix

A.1 Source phrase extraction from Books3 Corpus

The Books3 Corpus (Gao et al., 2020) is a huge-scale collection of books from a variety of genres. We mine the Books3 to extract phrases by selecting constituency chunks. Particularly, we use the fast Stanford shift-reduce parser[14] from Manning et al. (2014), collecting all verb, noun, adjective, and adverb phrases from the data and keep the top 100K phrases with the highest frequency. We do not keep prepositional phrases as we find high overlap between prepositional phrases with noun phrases empirically.

A.2 Phrase vocabulary extraction from dataset

Given the Wikipedia corpus (Merity et al., 2017), we first include all word types detected by spaCy's English tokenizer (Honnibal et al., 2020) that occur more than five times in the corpus. We augment this vocabulary with phrases by extracting constituent chunks from the output of a constituency parser. We use the the same shift-reduce parser as in appendix Section A.1. Specifically, we extract all verb, noun, adjective, and adverb phrases from the data, and add the most frequent 75K phrases into our **L** matrix for topic interpretation as in Section 6.1.2. We omit prepositional phrases as they overlapped significantly with noun phrases. We perform the same vocabulary creation steps for the other two datastest (**Story** and **Reviews**) to extract all datasets (Table 9)

A.3 Agreement among crowdsourced workers

Evaluations from crowdsourced human evaluations show high inter-annotator agreement, indicated by close to 90% of 2 or more workers agreeing on the same choice.

Model	Wiki	Story	Reviews
PNTM	92.0	90.0	96.0
pLDA	96.0	94.0	82.0
TNG	92.0	94.0	96.0
UNTM	86.0	96.0	90.0
LDA	88.0	88.0	90.0

Table 10: Annotator agreement statistics (percentage of questions where two or more workers agree on the same answer) of our word intrusion experiments across datasets and models.

Interpreting with words / phrases
missourian, american history, county route, alabama, confederate, a state highway
Interpreting with sentences
1. At its 1864 convention , the Republican Party selected Johnson , a War Democrat from the Southern state of Tennessee , as his running mate .
2. Burnett also raised a Confederate regiment at Hopkinsville , Kentucky , and briefly served in the Confederate States Army .
3. Parker was nominated for Missouri 's 7th congressional district on September 13 , 1870 , backed by the Radical faction of the Republican party .

Table 11: Sentence-level interpretation makes it clear that this topic is about Civil War-era American history, while word and phrase interpretation offers a more high-level view.

A.4 Qualitative Evaluation on PNTM: Interpreting topics with sentences

Another capability that sets PNTM apart from existing models is *sentence-level* topic interpretation, which offers an even more fine-grained understanding of learned topics. This functionality has potential to help with automatic topic labeling, which traditionally has been a manual process because the most probable words in a topic are not necessarily the most descriptive words of a particular high-level theme. Since the underlying BERT model of PNTM's embedding function is fine-tuned on both sentence and phrase-level data, its representations are semantically meaningful across multiple scales of text. We also do not have to retrain the model to interpret topics with sentences; rather, we just have to encode the training sentences (or potentially sentences from an external corpus) with our embedding model (PNTM) and then add them to the vocabulary (i.e., as additional rows in the **L** matrix).

Table 11 contains one such example, which is a

[14] https://nlp.stanford.edu/software/srparser.html

topic from a PNTM model trained with $K = 50$ on Wikipedia. When interpreted with just words and phrases, the topic looks like it focuses on Southern and Midwestern U.S. states and their history. However, when interpreting the same topic with sentences from the training set, we observe that the most probable sentences for this topic all reference the Civil War / Reconstruction era of U.S. history. These kinds of observations might influence not only a practitioner's labeling of a particular topic, but also how they use the topic model itself.

A.5 Topics from PNTM with different embedding functions

We present topic samples from three versions of PNTM, using BERT, SpanBERT, and Phrase-BERT respectively as the embedding function. Other than the embedding function used, the three topic models have the same architecture and are trained with the same hyperparameters. The training dataset is **Wiki** (the same dataset in Section 6.2.1), with the number of training epochs = 300 and the number of topics = 50.

Qualitatively, we observe that the Phrase-BERT-based topic model produces the highest quality topics that are both lexically diverse and also coherent, in Table 14. The topics from the BERT-based topic model (Table 12) have lower quality as the over-reliance on word content overlap makes some topics less informative (e.g., “his album", “the album", “an album" ...). However, the topic descriptions are largely interpretable as the words and phrases used are still semantically coherently. The SpanBERT-based topic model, on the other hand, produces even lower quality topics as the topic descriptions are incoherent in many cases, as shown in Table 13.

winning, semifinalist, finisher, a race, raceme, race, the race, the race 's, side rowing competition, formula one
bullfighter, bullfighting, showman, wwe 's, wwe smackdown, wwe day, wrestle, wrestler, the wwe championship, wrestling
the gatehouse, the plant, the farm, the estate, the building, the fort, the castle, landscaping, was built, the monument
the beatles, his album, the album 's, discography, the album, an album, beatles, this album, the beatles ', their album
tropical cyclones, a tropical depression, developed into a tropical depression, a tropical storm warning, a tropical cyclone, tropical storm arlene, tropical storm status, tropical storm, the tropical storm, a tropical storm
a mother, her parents, his parents, her father, her father 's, her mother 's, his mother, her mother, his mother 's, her parents '

Table 12: A sample of six topics induced by PNTM with BERT as the embedding model

two episodes, expressible, side rowing race, tourmaline, followed throughout the united kingdom, the island 's, drive, sidecar, flywheel, cockpit
lieutenant colonel, new mexico, midshipman, postmenopausal, generalship, generalissimo, ambassadorship, valedictorian, the wwe championship, the spanish 2̆013 american war
ellipse, opulent, meetinghouse, embark, rapidity, swiftly, the 13th century, institution, rapidly, gradual
songbook, the novel, a novel, this book, her book, storybook, fiction book, novelette, novelization, novelisation
major intersections, rainstorm, thunderstorm, torrential, a storm, the race 's, cloudy, high winds, major hurricanes, windstorm
satanic, luciferin, lynchpin, judas, blackmailer, kidnapping, satanist, bosch, vaulting, afire

Table 13: A sample of six topics induced by PNTM with SpanBERT as the embedding model

the semifinal, olympic, marathon, raceme, bicyclist, semifinalist, side rowing race, racer, place finish, side rowing competition
powerful, wrestler, the forces, most powerful, demonic, the organization, a force, power, an organization, dark forces
newly built, terrace, atrium, architecture, foyer, the building, the city centre, facade, architecturally
musician, his music, musical, concerto, chorale, live performances, a concert, accompaniment, pianistic, antiphonal
tropical depression, a tropical cyclone, a category 2 hurricane, a tropical storm, a category 1 hurricane, tropical storm status, tropical storm, a tropical disturbance, developed into a tropical depression, a tropical depression
a police officer, criminology, criminalisation, criminal cases, illegality, law enforcement, criminalization, felony, criminality, misdemeanor

Table 14: A sample of six topics induced by PNTM with Phrase-BERT as the embedding model