BURT: BERT-inspired Universal Representation from Learning Meaningful Segment

Yian Li, Hai Zhao

Abstract—Although pre-trained contextualized language models such as BERT achieve significant performance on various downstream tasks, current language representation focuses on linguistic objective at a specific granularity. Thus this work introduces the universal representation learning, i.e., embeddings of different levels of linguistic unit in a uniform vector space. We present a universal representation model, BURT (BERT-inspired Universal Representation from learning meaningful segment), to encode different levels of linguistic unit into the same vector space. Specifically, we extract meaningful segments based on point-wise mutual information (PMI) to incorporate different granular objectives into the pre-training stage. We conduct experiments on datasets for English and Chinese including the GLUE and CLUE benchmarks, where our model surpasses its baselines and alternatives on a wide range of downstream tasks. We present our approach of constructing analogy datasets in terms of words, phrases and sentences and experiment with multiple representation models to examine geometric properties of the learned vector space through a task-independent evaluation. Finally, we verify the effectiveness of our method in two real-world text matching scenarios. As a result, our model significantly outperforms existing information retrieval (IR) methods and yields universal representations that can be directly applied to retrieval-based question-answering and natural language generation tasks.

Index Terms—Artificial Intelligence, Natural Language Processing, Transformer, Language Representation.

1 INTRODUCTION

Representations learned by deep neural models have attracted a lot of attention in Natural Language Processing (NLP). However, previous language representation learning methods such as Word2Vec [1], LASER [2] and USE [3] focus on either words or sentences. Later proposed pre-trained contextualized language representations like ELMo [4], GPT [5], BERT [6] and XLNet [7] may seemingly handle different sized input sentences, but all of them focus on sentence-level specific representation still for each word, leading to unsatisfactory performance in real-world situations. Although the latest BERT-wwm-ext [8], StructBERT [9] and SpanBERT [10] perform MLM on a higher linguistic level, the masked segments (whole words, trigrams, spans) either follow a pre-defined distribution or focus on a specific granularity. Besides, the random sampling strategy ignores important semantic and syntactic information of a sequence, resulting in a large number of meaningless segments.

However, universal representation among different levels of linguistic units may offer a great convenience when it is needed to handle free text in language hierarchy in a unified way. As well known that, embedding representation for a certain linguistic unit (i.e., word) enables linguistics-meaningful arithmetic calculation among different vectors, also known as word analogy. For example, vector (“King”) - vector (“Man”) + vector (“Woman”) results in vector (“Queen”). Thus universal representation may generalize such good analogy features or meaningful arithmetic operation onto free text with all language levels involved together. For example, Eat an onion : Vegetable :: Eat a pear : Fruit. In fact, manipulating embeddings in the vector space reveals syntactic and semantic relations between the original sequences and this feature is indeed useful in true applications. For example, “London is the capital of England,” can be formulated as \( v(\text{capital}) + v(\text{England}) \approx v(\text{London}) \). Then given two documents one of which contains “England” and “capital”, the other contains “London”, we consider these two documents relevant.

In this paper, we explore the regularities of representations including words, phrases and sentences in the same vector space. To this end, we introduce a universal analogy task derived from Google’s word analogy dataset. To solve such task, we present BURT, a pre-trained model that aims at learning universal representations for sequences of various lengths. Our model follows the architecture of BERT but differs from its original masking and training scheme. Specifically, we propose to efficiently extract and prune meaningful segments (n-grams) from unlabeled corpus with little human supervision, and then use them to modify the masking and training objective of BERT. The n-gram pruning algorithm is based on point-wise mutual information (PMI) and automatically captures different levels of language information, which is critical to improving the model capability of handling multiple levels of linguistic objects in a unified way, i.e., embedding sequences of different lengths.
Overall, our pre-trained models improves the performance of baselines in both English and Chinese. In English, BURT-base reaches 0.7 percent gain on average over Google BERT-base. In Chinese, BURT-wwm-ext obtains 74.5% on the WSC test set, 13.4% point absolute improvement compared with BERT-wwm-ext and exceeds the baselines by 0.2% ~ 0.6% point accuracy on five other CLUE tasks including TNEWS, IFLYTEK, CSL, ChID and CMRC 2018. Extensive experimental results on our universal analogy task demonstrate that BURT is able to map sequences of variable lengths into a shared vector space where similar sequences are close to each other. Meanwhile, addition and subtraction of embeddings reflect semantic and syntactic connections between sequences. Moreover, BURT can be easily applied to real-world applications such as Frequently Asked Questions (FAQ) and Natural Language Generation (NLG) tasks, where it encodes words, sentences and paragraphs into the same embedding space and directly retrieves sequences that are semantically similar to the given query based on cosine similarity. All of the above experimental results demonstrate that our well-trained model leads to universal representation that can adapt to various tasks and applications.

2 BACKGROUND

2.1 Word and Sentence Embeddings

Representing words as real-valued dense vectors is a core technique of deep learning in NLP. Word embedding models [1], [11], [12] map words into a vector space where similar words have similar latent representations. ELMo [4] attempts to learn context-dependent word representations through a two-layer bi-directional LSTM network. In recent years, more and more researchers focus on learning sentence representations. The Skip-Thought model [13] is designed to predict the surrounding sentences for an given sentence. Logeswaran and Lee (2018) [14] improve the model structure by replacing the RNN decoder with a classifier. InferSent [15] is trained on the Stanford Natural Language Inference (SNLI) dataset [16] in a supervised manner. Subramanian et al. (2018) [17] and Cer et al. (2018) [18] employ multi-task training and report considerable improvements on downstream tasks. LASER [2] is a BiLSTM encoder designed to learn multilingual sentence embeddings. Most recently, contextualized representations with a language model training objective such as OpenAI GPT [5], BERT [6], XLNet [7] are expected to capture complex features (syntax and semantics) for sequences of any length. Especially, BERT improves the pre-training and fine-tuning scenario, obtaining new state-of-the-art results on multiple sentence-level tasks. On the basis of BERT, further fine-tuning using Siamese Network on NLI data can effectively produce high quality sentence embeddings [7]. Nevertheless, most of the previous work concentrate on a specific granularity. In this work we extend the training goal to a unified level and enables the model to leverage different granular information, including, but not limited to, word, phrase or sentence.

2.2 Pre-Training Tasks

BERT is trained on a large amount of unlabeled data including two training targets: Masked Language Model (MLM) for modeling deep bidirectional representations, and Next Sentence Prediction (NSP) for understanding the relationship between two sentences. ALBERT [18] is trained with Sentence-Order Prediction (SOP) as a substitution of NSP. StructBERT [9] has a sentence structural objective that combines the random sampling strategy of NSP and continuous sampling as in SOP. However, RoBERTa [19] and SpanBERT [10] use single contiguous sequences of 512 tokens for pre-training and show that removing the NSP objective improves the performance. Besides, BERT-wwm [8], StructBERT [10], SpanBERT [9] perform MLM on higher linguistic levels, augmenting the MLM objective by masking whole words, trigrams or spans, respectively. Nevertheless, we concentrate on enhancing the masking and training procedures from a broader and more general perspective.

2.3 Analysis and Applications

Previous explorations of vector regularities mainly study word embeddings [1], [20]. After the introduction of sentence encoders and Transformer models [22], more works were done to investigate sentence-level embeddings. Usually the performance in downstream tasks is considered to be the measurement for model ability of representing sentences [3], [15], [23]. Some research proposes probing tasks to understand certain aspects of sentence embeddings [24], [25], [26]. Specifically, Rogers et al. (2020) [27] and Ma et al. (2019) [28] look into BERT embeddings and reveal its internal working mechanisms. Some work also explores the regularities in sentence embeddings [30], [31]. Nevertheless, little work analyzes words, phrases and sentences in the same vector space. In this paper, We work on embeddings for sequences of various lengths obtained by different models in a task-independent manner.

Transformer-based representation models have made great progress in measuring query-Question or query-Answer similarities. Damani et al. (2020) [35] make an analysis on Transformer models and propose a neural architecture to solve the FAQ task. Sakata et al. (2019) [36] come up with an FAQ retrieval system that combines the characteristics of BERT and rule-based methods. In this work, we also evaluate the well-trained universal representation models on FAQ task.

3 METHODOLOGY

Our BURT follows the Transformer encoder [22] architecture where the input sequence is first split into subword tokens and a contextualized representation is learned for each token. We only perform MLM training on single sequences as suggested in [10]. The basic idea is to mask some of the tokens from the input and force the model to recover them from the context. Here we propose a unified masking method and training objective considering different grained linguistic units.

Specifically, we apply an pruning mechanism to collect meaningful n-grams from the corpus and then perform n-gram masking and predicting. Our model differs from the original BERT and other BERT-like models in several ways. First, instead of the token-level MLM of BERT, we incorporate different levels of linguistic units into the training objective
3.1 N-gram Pruning

In this subsection, we introduce our approach of extracting a large number of meaningful n-grams from the monolingual corpus, which is a critical step of data processing.

First, we scan the corpus and extract all n-grams with lengths up to N using the SRILM toolkit\cite{37}. In order to filter out meaningless n-grams and prevent the vocabulary from being too large, we apply pruning by means of point-wise mutual information (PMI)\cite{38}. To be specific, mutual information \( I(x,y) \) describes the association between tokens \( x \) and \( y \) by comparing the probability of observing \( x \) and \( y \) together with the probabilities of observing \( x \) and \( y \) independently. Higher mutual information indicates stronger association between the two tokens.

\[
I(x,y) = \log \frac{P(x,y)}{P(x)P(y)} \tag{1}
\]

In practice, \( P(x) \) and \( P(y) \) denote the probabilities of \( x \) and \( y \), respectively, and \( P(x,y) \) represents the joint probability of observing \( x \) followed by \( y \). This alleviates bias towards high-frequency words and allows tokens that are rarely used individually but often appear together such as “San Francisco” to have higher scores. In our application, an n-gram denoted as \( w = (x_1, \ldots, x_{L_w}) \), where \( L_w \) is the number of tokens in \( w \), may contains more than two words. Therefore, we present an extended PMI formula displayed as below:

\[
PMI(w) = \frac{1}{L_w} \left( \log P(w) - \sum_{k=1}^{L_w} \log P(x_k) \right) \tag{2}
\]

where the probabilities are estimated by counting the number of observations of each token and n-gram in the corpus, and normalizing by the size of the corpus. \( \frac{1}{L_w} \) is an additional normalization factor which avoids extremely low scores for longer n-grams. Finally, n-grams with PMI scores below the chosen threshold are filtered out, resulting in a vocabulary of meaningful n-grams.

3.2 N-gram Masking

For a given input \( S = \{x_1, x_2, \ldots, x_L\} \), where \( L \) is the number of tokens in \( S \), special tokens \([CLS]\) and \([SEP]\) are added at the beginning and end of the sequence, respectively. Before feeding the training data into the Transformer blocks, we identify all the n-grams in the sequence using the aforementioned n-gram vocabulary. An example is shown in Figure\[2\] where there are overlap between n-grams, which indicates the multi-granular inner structure of the given sequence. In order to make better use of higher-level linguistic information, the longest n-gram is retained if multiple matches exist. Compared with other masking strategies, our method has two advantages. First, n-gram extracting and matching can be efficiently done in an unsupervised manner without introducing random noise. Second, by utilizing n-grams of different lengths, we generalize the masking and training objective of BERT to a unified level where different granular linguistic units are integrated.
3.3 Training Objective

As depicted in Figure 1, the Transformer encoder generates a fixed-length contextualized representation at each input position and the model only predicts the masked tokens. Ideally, a universal representation model is able to capture features for multiple levels of linguistic units. Therefore, we extend the MLM training objective to a more general situation, where the model is trained to predict n-grams rather than subwords.

\[
\max_{\theta} \sum_{w} \log P(w|\hat{x}; \theta) = \max_{\theta} \sum_{(i,j)} \log P(x_i, \ldots, x_j|\hat{x}; \theta) \tag{3}
\]

where \(w\) is a masked n-gram and \(\hat{x}\) is a corrupted version of the input sequence. \((i,j)\) represents the absolute start and end positions of \(w\).

4 Task Setup

To evaluate the model ability of handling different linguistic units, we apply our model on downstream tasks from GLUE and CLUE benchmark. Moreover, we construct a universal analogy task based on Google’s word analogy dataset to explore the regularity of universal representation. Finally, we present an insurance FAQ task and a retrieval-based language generation task, where the key is to embed sequences of different lengths in the same vector space and retrieve sequences with similar meaning to the given query.

4.1 General Language Understanding

Statistics of the GLUE and CLUE benchmarks are listed in Table 1. Besides the diversity of task types, we also find that different datasets concentrates on sequences of different lengths, which satisfies our need to examine the model ability of representing multiple granular linguistic units.

### Table 1

| GLUE    | Length | #Train | #Dev  | #Test | #L  |
|---------|--------|--------|-------|-------|-----|
| CoLA    | Short  | 8.5k   | 1k    | 1k    | 2   |
| SST-2   | Short  | 67k    | 872   | 1.8k  | 2   |
| MNLI    | Short-Short | 393k | 20k   | 20k   | 3   |
| QNLI    | Short-Short | 105k | 5.5k  | 5.5k  | 2   |
| RTE     | Short-Short | 2.5k | 277   | 3k    | 2   |
| MRPC    | Short-Short | 3.7k | 408   | 1.7k  | 2   |
| QQP     | Short-Short | 364k | 40k   | 391k  | 2   |
| STS-B   | Short-Short | 5.8k | 1.5k  | 1.4k  | 2   |

Following BERT, we mask 15% of all tokens in each sequence. The data processing algorithm uniformly samples one n-gram at a time until the maximum number of masking tokens is reached. 80% of the time we replace the entire n-gram with \([\text{MASK}]\) tokens. 10% of the time it is replace with random tokens and 10% of the time we keep it unchanged. The original token-level masking is retained and considered as a special case of n-gram masking where \(n = 1\). We employ dynamic masking as mentioned by Liu et al. (2019) [19], which means masking patterns for the same sequence in different epochs are probably different.

### 4.1.1 GLUE

The General Language Understanding Evaluation (GLUE) benchmark [39] is a collection of tasks that is widely used to evaluate the performance of English language models. We divide eight NLU tasks from the GLUE benchmark into three main categories.

**Sentence Pair Tasks**

- **Multi-Genre Natural Language Inference (MNLI)** [42], Stanford Question Answering Dataset (SST-2) [41], and the Chinese Winograd Schema Challenge (WSC) dataset. Samples from TNEWS and IFLYTEK are short and long sequences, respectively, and the goal is to predict the category that the given single sentence belongs to. WSC is a coreference resolution task where the model is required to decide whether two spans refer to the same entity in the original sequence.
- **Single-Sentence Classification**
  - The Corpus of Linguistic Acceptability (CoLA) [40] is to determine whether a sentence is grammatically acceptable or not. The Stanford Sentiment Treebank (SST-2) [41] is a sentiment classification task that requires the model to predict whether the sentiment of a sentence is positive or negative. In both datasets, each example is a sequence of words annotated with a label.
  - **Natural Language Inference**
    - The Multi-Genre Natural Language Inference (MNLI) [42] and Recognizing Textual Entailment (RTE) [?] are natural language inference tasks, where a pair of sentences are given and the model is trained to identify the relationship between the two sentences from entailment, contradiction, and neutral.

### 4.1.2 CLUE

The Chinese General Language Understanding Evaluation (ChineseGLUE or CLUE) benchmark [46] is a Chinese version of the GLUE benchmark for language understanding. We also find nine tasks from the CLUE benchmark can be classified into three groups.

**Single Sentence Tasks** We utilize three single-sentence classification tasks including YouTiao Text Classification for News Titles (TNEWS), IFLYTEK [47] and the Chinese Winograd Schema Challenge (WSC) dataset. Examples from TNEWS and IFLYTEK are short and long sequences, respectively, and the goal is to predict the category that the given single sentence belongs to. WSC is a coreference resolution task where the model is required to decide whether two spans refer to the same entity in the original sequence.

**Sentence Pair Tasks** The Ant Financial Question Matching Corpus (AFQMC), Chinese Scientific Literature (CSL) dataset and Original Chinese Natural Language Inference (OCNLI) [48] are three pairwise textual classification tasks. AFQMC
contains sentence pairs and binary labels, and the model is asked to examine whether two sentences are semantically similar. Each example in CSL involves a text and several keywords. The model needs to determine whether these keywords are true labels of the text. OCNLI is a natural language inference task following the same collection procedures of MNLI.

**Machine Reading Comprehension Tasks CMRC 2018** [49], ChID [50], and C³ [51] are span-extraction based, cloze style and free-form multiple-choice machine reading comprehension datasets, respectively. Answers to the questions in CMRC 2018 are spans extracted from the given passages. ChID is a collection of passages with blanks and corresponding candidates for the model to decide the most suitable option. C³ is similar to RACE and DREAM, where the model has to choose the correct answer from several candidate options based on a text and a question.

### 4.2 Universal Analogy

As a new task, universal representation has to be evaluated in a multiple-granular analogy dataset. The purpose of proposing a task-independent dataset is to avoid determining the quality of the learned vectors and interpret the model based on a specific problem or situation. Since embeddings are essentially dense vectors, it is natural to apply mathematical operations on them. In this subsection, we introduce the procedure of constructing different levels of analogy datasets based on Google’s word analogy dataset.

#### 4.2.1 Word-level analogy

Recall that in a word analogy task [1], two pairs of words that share the same type of relationship, denoted as $A : B :: C : D$, are involved. The goal is to solve questions like “$A$ is to $B$ as $C$ is to $?$”, which is to retrieve the last word from the vocabulary given the first three words. The objective can be formulated as maximizing the cosine similarity between the target word embedding and the linear combination of the given vectors:

$$d^* = \arg \max_{d^*} \cosine(c + b - a, d)$$

$$\cosine(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

where $a, b, c, d$ represent embeddings of the corresponding words and are all normalized to unit lengths.

To facilitate comparison between models with different vocabularies, we construct a closed-vocabulary analogy task based on Google’s word analogy dataset through negative sampling. Concretely, for each question, we use GloVe to rank every word in the vocabulary and the top 5 results are considered to be candidate words. If GloVe fails to retrieve the correct answer, we manually add it to make sure it is included in the candidates. During evaluation, the model is expected to select the correct answer from 5 candidate words. Examples are listed in Table 2.

#### 4.2.2 Phrase/Sentence-level analogy

To investigate the arithmetic properties of vectors for higher levels of linguistic units, we present phrase and sentence analogy tasks based on the proposed word analogy dataset. We only consider a subset of the original analogy task because we find that for some categories, such as “Australia” : “Australian”, the same template phrase/sentence cannot be applied on both words. Statistics are shown in Table 3.

**Semantic** Semantic analogies can be divided into four subsets: “capital-common”, “capital-world”, “city-state” and “male-female”. The first two sets can be merged into a larger dataset: “capital-country”, which contains pairs of countries and their capital cities; the third involves states and their cities; the last one contains pairs with gender relations. Considering GloVe’s poor performance on word-level “country-currency” questions (<32%), we discard this subset in phrase and sentence-level analogies. Then we put words into contexts so that the resulting phrases and sentences also have linear relationships. For example, based on relationship Athens : Greece :: Baghdad : Iraq, we select phrases and sentences that contain the word “Athens” from the English Wikipedia Corpus.

### Table 2

Examples from our word analogy dataset. The correct answers are in bold.

| A : B :: C | Candidates |
|------------|------------|
| boy : girl :: brother | daughter, sister, wife, father, son |
| big : worse :: big | bigger, larger, smaller, biggest, better |
| Beijing : China :: Paris | France, Europe, Germany, Belgium, London |
| Chile : Chilean :: China | Japanese, Chinese, Russian, Korean, Ukrainian |

### Table 3

Statistics of our analogy datasets. #p and #q are the number of pairs and questions for each category. #c is the number of candidates for each dataset. #l (p/s) is the average sequence length in phrase/sentence-level analogy datasets.

| Dataset          | #p | #q | #c | #l (p/s) |
|------------------|----|----|----|----------|
| capital-common   | 23 | 506| 5  | 6.0/12.0 |
| capital-world    | 116| 4524|5 | 6.0/12.0 |
| city-state       | 67 | 2467| 5 | 6.0/12.0 |
| male-female      | 23 | 506 | 5 | 4.1/10.1 |
| present-participle| 33 | 1056| 2 | 4.8/8.8 |
| positive-comparative| 37 | 1322| 2 | 3.4/6.1 |
| positive-negative| 29 | 812 | 2 | 4.4/9.2 |
| All              | 328| 11193| - | 5.4/10.7 |

2. https://dumps.wikimedia.org/enwiki/latest
TABLE 4
Details of the templates.

| Category       | Topics                                                                 |
|----------------|------------------------------------------------------------------------|
| Daily Scenarios| Traveling, Recipe, Skin care, Beauty makeup, Pets                      |
| Sport & Health | Outdoor sports, Athletics, Weight loss, Medical treatment             |
| Reviews        | Movies, Music, Poetry, Books                                           |
| Persons        | Entrepreneurs, Historical/Public figures, Writers, Directors, Actors   |
| General        | Festivals, Hot topics, TV shows                                       |
| Specialized    | Management, Marketing, Education, Workplace skills                     |
| Others         | Relationships, Technology, Literature                                  |

| All            | 07                                      |

**Syntactic** We consider three typical syntactic analogies: Tense, Comparative and Negation, corresponding to three subsets: “present-participle”, “positive-comparative”, “positive-negative”, where the model needs to distinguish the correct answer from “past tense”, “superlative” and “positive”, respectively. For example, given phrases “Pigs are bright” : “Pigs are brighter than goats” : “The train is slow”, the model need to give higher similarity score to the sentence that contains “slower” than the one that contains “slowest”. Similarly, we add synonyms and synonymous phrases for each question to evaluate the model ability of learning context-aware embeddings rather than interpreting each word in the question independently. For instance, “pleasant” ≈ “not unpleasant” and “unpleasant” ≈ “not pleasant”.

**4.3 Retrieval-based FAQ**

The sentence-level analogy discovers relationships between sentences by directly manipulating sentence vectors. Especially, we observe that sentences with similar meanings are close to each other in the vector space, which we find consistent with the target of information retrieval task such as Frequently Asked Question (FAQ). Such task is to retrieve relevant documents (FAQs) given a user query, which can be accurately done by only manipulating vectors representing the sentences, such as calculating and ranking vector distance in terms of cosine similarity. Thus, we present an insurance FAQ task in this subsection to explore the effectiveness of BURT in real-world retrieval applications.

An FAQ task involves a collection of Question-Answer (QA) pairs denoted as \{(Q_1, A_1), (Q_2, A_2), ..., (Q_N, A_N)\}

where \(N\) is the number of QA pairs. The goal is to retrieve the most relevant QA pairs for a given query. We collect frequently asked questions and answers between users and customer service from our partners in a Chinese online financial education institution. It contains over 4 types of insurance questions, e.g., concept explanation (“what”), insurance consultation (“why”, “how”), judgement (“whether”) and recommendation. An example is shown in Figure 3. Our dataset is composed of 300 QA pairs that are carefully selected to avoid similar questions so that each query has only one exact match. Because queries are mainly paraphrases of the standard questions, we use query-Question similarity as the ranking score. The test set consists of 875 queries and the average lengths of questions and queries are 14 and 16, respectively. The evaluation metric is Top-1 Accuracy (Acc.) and Mean Reciprocal Rank (MRR) because there is only one correct answer for each query.

**4.4 Natural Language Generation**

Moving from word and sentence vectors towards representation for sequences of any lengths, a universal language model may have the ability of capturing semantics of free text and facilitating various applications that are highly dependent on the quality of language representation. In this subsection, we introduce a retrieval-based Natural Language Generation (NLG) task. The task is to generate articles based on manually created templates. Concretely, the goal is to retrieve one paragraph at a time from the corpus which best describes the standard questions, e.g., concept explanation (“what”), insurance consultation (“why”, “how”), judgement (“whether”) and recommendation. An example is shown in Figure 3. We collect frequently asked questions and answers between users and customer service from our partners in a Chinese online financial education institution. It contains over 4 types of insurance questions, e.g., concept explanation (“what”), insurance consultation (“why”, “how”), judgement (“whether”) and recommendation. An example is shown in Figure 3. Our dataset is composed of 300 QA pairs that are carefully selected to avoid similar questions so that each query has only one exact match. Because queries are mainly paraphrases of the standard questions, we use query-Question similarity as the ranking score. The test set consists of 875 queries and the average lengths of questions and queries are 14 and 16, respectively. The evaluation metric is Top-1 Accuracy (Acc.) and Mean Reciprocal Rank (MRR) because there is only one correct answer for each query.

---

**Fig. 3.** Examples of Question-Answer pairs from our insurance FAQ dataset. The correct match to the query is highlighted.

**Query:** 80-year-old people get accident insurance?

| Q: 受益人能否修改? | A: 可以修改受益人。 |
|-------------------|-------------------|
| Can I change the beneficiary? | Yes, the beneficiary can be modified. |
| Q: 高龄老人能否购买意外险? | A: 你可以选择老年人的意外产品。 |
| Can seniors buy accident insurance? | You can choose accident insurance for the elderly. |
| Q: 如何进行线上理赔? | A: 首先给保险公司打电话报案申请理赔。第二… |
| How to make an online claim? | First, call the insurance company to report the case and apply for a claim. Second, ... |

3. https://www.cluebenchmarks.com/rc.html
4. https://gluebenchmark.com
The problem is solved in two steps. First, an index for all the documents is built using BM25. For each query, it will return a set of candidate documents that are related to the topic. Second, we use representation models to re-rank all the documents is built using BM25. For each query, it was assessed by two native Chinese speakers, who were asked to examine whether the retrieved paragraphs were “relevant” to the topic and “conveyed the meaning” of the given sentence.

## 5 Implementation

### 5.1 Data Processing

We download the English and Chinese Wikipedia Corpus and pre-process with `process_wiki.py` which extracts text from xml files. Then for the Chinese corpus, we convert the data into simplified characters using OpenCC. In order to extract high-quality n-grams, we remove punctuation marks and characters in other languages based on regular expressions, and finally get an English corpus of 2,266M words and a Chinese corpus of 380M characters. We calculate PMI scores of all n-grams with a maximum length of N = 10 for each document instead of the entire corpus considering that different documents usually describe different topics. We manually evaluate the extracted n-grams and find nearly 50% of the top 2000 n-grams contain 3 ~ 4 words (characters for Chinese), and only less than 0.5% n-grams are longer than 7. Although a larger n-gram vocabulary can cover longer n-grams, it will cause too many meaningless n-grams. Therefore, for both English and Chinese corpus, we empirically retain the top 3000 n-grams for each document, resulting in vocabularies of n-grams with average lengths of 4.6 and 4.5, respectively. Finally, for English, we randomly sample 10M sentences rather than use the entire corpus to reduce training time.

### 5.2 Pre-training

As in BERT, sentence pairs are packed into a single sequence and the special [CLS] token is used for sentence-level predicting. While in accordance with Joshi et al. (2020) [10], we find that single sentence training is better than the original sentence pair scenario. Thus in our experiments, the input is a continuous sequence with a maximum length of 512.

Instead of training from scratch, we initialize both English and Chinese models with the officially released checkpoints (bert-base-uncased, bert-large-uncased, bert-base-chinese) and BERT-wwm-ext, which is trained from the Chinese BERT using whole word masking on extended data [6]. Base models are comprised of 12 Transformer layers, 12 heads, 768 dimensional hidden states and 110M parameters in total. The English BERT-large has 24 100M parameters in total. The English BERT-large has 24 128,560 parameters in total. The English BERT-large has 24
Transformer layers, 16 heads, 1024 dimensional hidden states and 340M parameters in total. We use Adam optimizer with initial learning rate of 5e-5 and linear warmup over the first 10% of the training steps. Batch size is set to 16 and dropout rate is 0.1. Each model is trained for one epoch.

### 5.3 Fine-tuning

Following BERT, in the fine-tuning procedure, pairs of sentences are concatenated into a single sequence with a special token [SEP] in between. For both single sentence and sentence pair tasks, the hidden state of the first token [CLS] is used for softmax classification. We use the same sets of hyperparameters for all the evaluated models. All experiments on the GLUE benchmark are ran with a total train batch sizes between 8 and 64 and learning rates of 3e-5 for 3 epochs. For tasks from the CLUE benchmark, we set batch sizes to 8 and 16, learning rates between 1e-5 and 3e-5, and train 5 epochs on WSC and 2~5 epochs on the rest tasks.

### 5.4 Downstream-task Models

On GLUE and CLUE, we compare our model with three variants: pre-trained models (Chinese BERT/BERT-wwm-ext, English BERT-base/BERT-large), models trained with the same number of additional steps as our model (MLM), and models trained using random span masking with the same number of additional steps as our model (Span). For the Span model, we simply replace our n-gram module with the masking strategy as proposed by [10], where the probability of span length is based on a geometric distribution \( l \sim Geo(p) \). We follow the parameter setting that \( p = 0.2 \) and maximum span length \( l_{\text{max}} = 10 \).

We also evaluate the aforementioned models on our universal analogy task. Baseline models include Bag-of-words (BoW) model from pre-trained word embeddings: GloVe, sentence embedding models: InferSent, GenSen, USE and LASER, pre-trained contextualized language models: BERT, ALBERT, RoBERTa and XLNet. To derive semantically meaningful embeddings, we fine-tune BERT and our model on the Stanford Natural Language Inference (SNLI) [16] and the Multi-Genre NLI Corpus [7] using a Siamese structure following Reimers and Gurevych (2019) [7].
| Method      | Acc. | MRR |
|-------------|------|-----|
| TF-IDF      | 73.7 | 0.813 |
| BM25        | 72.1 | 0.802 |
| LASER       | 79.9 | 0.856 |
| BERT        | 76.8 | 0.831 |
| MLM         | 78.3 | 0.843 |
| Span        | 78.6 | 0.846 |
| BURT        | 82.2 | 0.872 |
| BERT-wwm-ext| 76.7 | 0.834 |
| MLM         | 76.7 | 0.834 |
| Span        | 79.3 | 0.856 |
| BURT-wwm-ext| 80.7 | 0.863 |

**TABLE 10**
Comparison of models performance on the FAQ dataset.

**TABLE 11**
Results on NLG according to human judgment. “R” and “CM” represent the percentage of paragraphs that are “relevant” and “convey the meaning”, respectively.

| Method      | R  | CM  |
|-------------|----|-----|
| Judge1      | 60.3 | 63.9 |
| Judge2      | 61.8 | 61.6 |
| Avg.        | 61.1 | 62.8 |

For FAQ and NLG, we compare our models with statistical methods such as TF-IDF and BM25, a sentence representation model LASER [2], the pre-trained BERT/BERT-wwm-ext and models trained with additional steps (MLM, Span). We observe that further training on Chinese SNLI and MNL datasets underperforms BERT on the FAQ dataset. Therefore, we only consider pre-trained models for these two tasks.

### 6 Experiments

#### 6.1 General Language Understanding

Table [5] and show the results on the GLUE and CLUE benchmarks, where we find that training BERT with additional MLM steps can hardly bring any improvement except for the WSC task. In Chinese, the Span model is effective on WSC but is comparable to BERT on other tasks.

BERT-wwm-ext is better than our model on classification tasks involving pairs of short sentences such as AFQMC and OCNLI, which may be due to its relative powerful capability of modeling short sequences. Overall, both BURT and BURT-wwm-ext outperform the baseline models on 4 out of 6 tasks with considerable improvement, which sheds light on their effectiveness of modeling sequences of different lengths. The most significant improvement is observed on WSC (3.5% over the updated BERT-wwm-ext and 0.7% over the Span model), where the model is trained to determine whether the given two spans refer to the same entity in the text. We conjecture that the model benefits from learning to predict meaningful spans in the pre-training stage, so it is better at capturing the meanings of spans in the text. In English, our approach also improves the performance of BERT on various tasks from the GLUE benchmark, indicating that our proposed PMI-based masking method is general and
independent with language settings.

6.2 Universal Analogy

Results on analogy tasks are reported in Table 8. Generally, semantic analogies are more challenging than the syntactic ones and higher-level relationships between sequences are more difficult to capture, which is observed in almost all the evaluated models. On word analogy tasks, all well-pre-trained language models like BERT, ALBERT, RoBERTa and XLNet hardly exhibit arithmetic characteristics and increasing the model size usually leads to a decrease in accuracy. However, our method of pre-training using n-grams extracted by the PMI algorithm significantly improves the performance on word analogies compared with BERT, obtaining 72.8% (BURT-base) and 79.4% (BURT-large) accuracy, respectively. Further training BURT-large on SNLI and MNLI results in the highest accuracy (80.5%).

Despite the leading performance on word-level analogy datasets of GloVe, InferSent and USE, they do not generalize well on higher level analogy tasks. We conjecture their poor performance is caused by synonyms and paraphrases in sentences which lead the model to produce lower similarity scores to the correct answers. In contrast, Transformer-based models are more advantageous in representing higher-level sequences and are good at identifying paraphrases and capturing relationships between sentences even if they have less lexical overlap. Moreover, fine-tuning pre-trained models achieves considerable improvements on high-level semantic analogies. Overall, SBURT-base achieves the highest average accuracy (60.7%).

Examples from the Negation subset are shown in Table 7. Notice that the word ”not” does not explicitly appear in the correct answers. Instead, ”inefficient” and ”unaware” are indicators of negation. As expected, BOW will give a higher similarity score for the sentence that contain both ”not” and ”inefficient” because the word-level information is simply added and subtracted despite the context. By contrast, contextualized models like BURT capture the meanings and relationships of words within the sequence in a comprehensive way, indicating that it has indeed learned universal representations across different linguistic units.

6.3 Retrieval-based FAQ

Results are reported in Table 10. Although nearly 62% of the paragraphs retrieved by BM25 are relevant to the topic, only two-thirds of them actually convey the original meaning of the template. Despite LASER’s comparable performance to BURT on FAQ, it is less effective when different granular linguistic units are involved at the same time. Re-ranking using BURT substantially improves the quality of the generated paragraphs. We show examples retrieved by BM25, LASER, the Span model and BURT in Table 9 denoted by B, L, S and U, respectively. BM25 tends to favor paragraphs that contain the keywords even though the paragraph conveys a different meaning, while BURT selects accurate answers according to semantic meanings of queries and documents.
We show examples in Figure 6 where BURT successfully retrieves the correct answer while TF-IDF and BM25 fail. Both sentences “Can 80-year-old people get accident insurance?” and “Can life insurance last until the age of 80?” contain the word “80”, which is a possible reason why TF-IDF tends to believe they highly match with each other, ignoring that the two sentences are actually describing two different issues. In contrast, using vector-based representations, BURT considers “seniors” as a paraphrase of “80-year-old people”. As depicted in Figure 6, queries are close to the correct responses and away from other sentences.

7 Visualization

7.1 Single Pattern

Mikolov et al. (2013) [20] use PCA to project word embeddings into a two-dimensional space to visualize a single pattern captured by the Word2Vec model, while in this work we consider embeddings for different granular linguistic units. All pairs in Figure 4 belong to the “male-female” category and subtracting the two vectors results in roughly the same direction.

7.2 Clustering

Given that embeddings of sequences with the same kind of relationship will exhibit the same pattern in the vector space, we obtain the difference between pairs of embeddings for words, phrases and sentences from different categories and visualize them by t-SNE. Figure 5 shows that by subtracting two vectors, pairs that belong to the same category automatically fall into the same cluster. Only the pairs from “capital-country” and “city-state” cannot be totally distinguished, which is reasonable because they all describe the relationship between geographical entities.

7.3 FAQ

We show examples in Figure 6 where BURT successfully retrieves the correct answer while TF-IDF and BM25 fail. Both sentences “Can 80-year-old people get accident insurance?” and “Can life insurance last until the age of 80?” contain the word “80”, which is a possible reason why TF-IDF tends to believe they highly match with each other, ignoring that the two sentences are actually describing two different issues. In contrast, using vector-based representations, BURT considers “seniors” as a paraphrase of “80-year-old people”. As depicted in Figure 6, queries are close to the correct responses and away from other sentences.

8 Conclusion

This paper formally introduces the task of universal representation learning and then presents a pre-trained language model for such a purpose to map different granular linguistic units into the same vector space where similar sequences have similar representations and enable unified vector operations among different language hierarchies.

In detail, we focus on the less concentrated language representation, seeking to learn a uniform vector form across different linguistic unit hierarchies. Far apart from learning either word only or sentence only representation, our method extends BERT’s masking and training objective to a more general level, which leverages information from sequences of different lengths in a comprehensive way and effectively learns a universal representation from words, phrases and sentences.

Overall, our proposed BURT outperforms its baselines on a wide range of downstream tasks with regard to sequences of different lengths in both English and Chinese languages. We especially provide an universal analogy task, an insurance FAQ dataset and an NLG dataset for extensive evaluation, where our well-trained universal representation model holds the promise for demonstrating accurate vector arithmetic with regard to words, phrases and sentences and in real-world retrieval applications.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013.
[2] M. Artetxe and H. Schwenk, “Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond,” Trans. Assoc. Comput. Linguistics, vol. 7, pp. 599–610, 2019. [Online]. Available: https://transacl.org/ojs/index.php/tacl/article/view/1742
[3] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Lmitiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, “Universal sentence encoder for English,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 169–174.
[4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in Proceedings of NAACL-HLT, 2018, pp. 2227–2237.
[5] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper.pdf, 2018.
[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.
[7] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xinet: Generalized autoregressive pretraining for language understanding,” arXiv preprint arXiv:1906.08237, 2019.
[8] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu, “Pre-training with whole word masking for chinese bert,” 2019.
[9] W. Wang, B. Bi, M. Yan, C. Wu, J. Xia, Z. Bao, L. Peng, and L. Si, “Strucbert: Incorporating language structures into pre-training for deep language understanding,” in International Conference on Learning Representations, 2020.
[10] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, “Spanbert: Improving pre-training by representing and predicting spans,” Transactions of the Association for Computational Linguistics, vol. 8, pp. 64–77, 2020.
[11] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543.
[12] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” 2016.
Yian Li received her Bachelor’s degree in Electronic and Information Engineering from Huazhong University of Science and Technology in 2019. She is working towards the M.S. degree in Department of Computer Science, Shanghai Jiao Tong University. Her research interests focuses on natural language processing, especially in language representation and pre-training language model.
Hai Zhao received the BEng degree in sensor and instrument engineering, and the MPhil degree in control theory and engineering from Yan-shan University in 1999 and 2000, respectively, and the PhD degree in computer science from Shanghai Jiao Tong University, China in 2005. He is currently a full professor at department of computer science and engineering, Shanghai Jiao Tong University after he joined the university in 2009. He was a research fellow at the City University of Hong Kong from 2006 to 2009, a visiting scholar in Microsoft Research Asia in 2011, a visiting expert in NICT, Japan in 2012. He is an ACM professional member, and served as area co-chair in ACL 2017 on Tagging, Chunking, Syntax and Parsing, (senior) area chairs in ACL 2018, 2019 on Phonology, Morphology and Word Segmentation. His research interests include natural language processing and related machine learning, data mining and artificial intelligence.