# Predicate-Argument Based Bi-Encoder for Paraphrase Identification

Anonymous ACL submission

## Abstract

Paraphrase identification involves identifying whether a pair of sentences express the same or similar meanings. While cross-encoders have achieved high performances across several benchmarks, bi-encoders such as SBERT have been widely applied to sentence pair tasks. They exhibit substantially lower computation complexity and are better suited to symmetric tasks. In this work, we adopt a bi-encoder approach to the paraphrase identification task, and investigate the impact of explicitly incorporating predicate-argument information into SBERT through weighted aggregation. Experiments on six paraphrase identification datasets demonstrate that, with a minimal increase in parameters, the proposed model is able to outperform SBERT/SRoBERTa significantly. Further, ablation studies reveal that the predicate-argument based component plays a significant role in the performance gain.

## 1 Introduction

Paraphrases are sentences that express the same or similar meanings with different wording (Bhagat and Hovy, 2013). Paraphrase pairs are either fully or largely semantically equivalent. For example:

a) *Marriage equality law passed in Rhode Island*

b) *Rhode Island becomes the 10th state to enact marriage equality*

It is generally considered to be a symmetric task where the paraphrase relation holds in both directions (Bhagat and Hovy, 2013; Yang et al., 2019).

Since word order and sentence structure are crucial in determining sentence meaning, effective paraphrase models must be structure-aware and word order sensitive. In light of this, paraphrase datasets have been created that are specifically designed to encourage models to consider structural differences (Xu et al., 2015; Zhang et al., 2019b). For example, PIT2015 (Xu et al., 2015) consists of paraphrase pairs that are lexically diverse and non-paraphrase pairs that are lexically similar but semantically dissimilar.

There are generally two pre-trained based approaches for sentence pair tasks such as paraphrase identification. The first is the cross-encoder approach, which involves concatenating the two input sentences and performing full-attention over the input. The second is the bi-encoder approach, which adopts a siamese structure and maps each sentence onto separate representations, which can then be compared using similarity measures such as cosine. Though typical cross-encoders like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) have set state-of-the-art performance on various sentence pair tasks (Zhang et al., 2021; Xia et al., 2021), they still face challenges from both extreme computational overhead for many use cases (Reimers and Gurevych, 2019; Thakur et al., 2021) and inconsistent predictions (ranging from 2.66% to 8.46% depending on specific datasets) when dealing with symmetric tasks (Chen et al., 2020).

In contrast, a bi-encoder approach such as Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) encodes sentences separately and generates high-quality embeddings for each of them. This architecture enables sentence embeddings to be pre-computed, supporting efficient indexing and comparison between different sequences. Due to the nature of bi-encoders, the symmetry property will be preserved as long as no asymmetry is introduced in subsequent layers. These properties make bi-encoders appealing for the paraphrase identification task. Accordingly, here, we focus on bi-encoders rather than cross-encoders.

One downside of SBERT is that it only adopts a very simple strategy, which is mean-pooling over all tokens, to generate sentence embeddings. As previously discussed, models should ideally be sensitive to any structural differences between two sentences. Relational Graph Convolutional Networks (RGCNs) (Schlichtkrull et al., 2018) have been used to introduce structural information (e.g. dependency/semantic parse trees) into SBERT and improvements have been reported on unsupervised similarity comparison tasks (Peng et al., 2021). One drawback of RGCNs is the size of the parameter space. For example, a single-layer RGCN can involve more than 30 million parameters. Furthermore, as we will demonstrate, the performance gain on different paraphrase identification datasets is not consistent.

An important aspect of sentence meaning concerns its predicate-argument structure. This has been utilised to generate paraphrases (Kozlowski et al., 2003) and to compare sentence meanings (Shan et al., 2009). Inspired by the Self-Explain model (Sun et al., 2020) which uses a span-based framework to generate sentence embeddings, we propose a method that effectively introduces sentence structure into SBERT via the aggregation of predicate-argument spans. This self-attention based aggregation allows us to gain benefits with minimal increased cost in terms of additional parameters. Empirical results indicate that the proposed model yields improvements on six benchmarks for paraphrase identification. Upon closer investigation, we find the predicate-argument span (PAS) component plays a crucial role in the performance gains and can be easily generalised to other models.

## 2 Related Work

### 2.1 Paraphrase Identification

The problem of paraphrase identification has been explored now for several decades (Mihalcea et al., 2006; Kozareva and Montoyo, 2006). Prior to the emergence of pre-trained models, bi-encoder structures were widely used. For example, Mueller and Thyagarajan (2016) applied LSTM in a siamese architecture with tied weights and used Manhattan distance to give similarity. InferSent (Conneau et al., 2017) exploited BiLSTM in a similar siamese structure with a fully-connected layer for classification over interacted sentence embeddings. Although their model was mainly proposed for transfer learning, experiments showed that it achieves good performance when directly trained on in-domain data.

Some bi-encoders do not generate single-vector sentence embeddings and allow direct comparisons between the words in the two sentences. Pang et al. (2016) proposed MatchPyramid where interaction matrix is constructed, and convolutional networks were used to extract features for final classification. PMWI (He and Lin, 2016) introduced more fine-grained comparisons between words to better dissect the meaning difference. ESIM (Chen et al., 2017) further utilised BiLSTM to bring contextualised token representations and allow rich interactions between tokens. Researchers have further improved these models by incorporating context and structure information (Liu et al., 2019a), as well as character-level information (Lan and Xu, 2018).

After the emergence of pre-trained models, cross-encoders like BERT and RoBERTa have achieved state-of-the-art performance on various sentence pair tasks including paraphrase identification. Zhang et al. (2019a) introduced pairwise word interaction mechanism into BERT. Zhang et al. (2021) improved BERT on paraphrase tasks by using CNNs to gather local information and an auxiliary task to further bring in semantic relation information. Xia et al. (2021) injected similarity matrices into BERT's attention mechanism. Though improved performance can be obtained, cross-encoders have known drawbacks. In particular, Reimers and Gurevych (2019) showed the extreme computation overhead of cross-encoders, and Chen et al. (2020) demonstrated that cross-encoders often give inconsistent predictions when reversing the input sentence order. Based on these factors, bi-encoders are often preferred for the paraphrase identification task.

### 2.2 Sentence Representation with Structures

Though pre-trained models like BERT are seen to have encoded certain structures in their contextualised representations, open questions remain about how to better utilise such information (Hewitt and Manning, 2019; Clark et al., 2019) and how useful the hidden structure is compared to externally provided sentence structures (Glavaš and Vulić, 2021; Dai et al., 2021). Recent improvements are also observed on various natural language understanding tasks by incorporating structural information into pre-trained models. SentiBERT proposed by Yin et al. (2020) incorporates constituency parse tree into BERT for sentiment analysis. Xu and Yang (2019) model each sentence as a directed dependency graph by using RGCN, and achieve improvements on pronoun resolution. Zhang et al. (2020) propose a semantics-aware BERT (SemBERT) model by further encoding semantic labels with BERT using a GRU. RGCNs have also been used by Wu et al. (2021) to introduce semantic information into RoBERTa, and achieved consistent improvements when fine-tuned on problem-specific datasets. Peng et al. (2021) propose a SBERT-RGCN model where structural information is explicitly encoded into SBERT in a similar way, achieving improvements on unsupervised sentence similarity comparison tasks. Similar efforts can be seen where researchers try to provide syntax information via self-attention mechanism (Bai et al., 2021; Li et al., 2020). Self-Explain model proposed by Sun et al. (2020) focuses on continuous text spans. It generates sentence embeddings by taking the weighted sum over all possible continuous text spans rather than individual tokens in the sentence. Though, Self-Explain achieves improvements over SentiBERT and SemBERT on sentiment analysis and language inference tasks, the continuous span strategy only captures linear structure and not differences in linguistic structure. In this paper, we draw inspiration from it, designing a similar span-based component to incorporate predicate-argument spans.

## 3 Model

Our proposed model adopts the same siamese architecture as SBERT and turns focus to the predicate-argument structure of the given sentence. As depicted in Figure 1, the model consists of different components:

**BERT:** Each sentence is first fed into the pre-trained BERT-base model to produce both a sentence representation, by applying mean-pooling over all token representations, and an original contextualised sequence-length token representation, which is used to derive predicate-argument span representations.

**Predicate Argument Spans (PAS):** We use AllenNLP (Gardner et al., 2018) with its BERT-based semantic role labelling (SRL) tagger to obtain predicates and relevant arguments for all input sentences. We group the predicate and its arguments together to generate predicate-argument spans. The initial position in the sentence determines their position in the span. An example of such spans is shown below:

*He slices tomatoes in the kitchen*

From this sentence, the predicate is the verb *slices*, and the three arguments are (*he*, *tomatoes* and *in the kitchen*), involving the relations (*ARG0*, *ARG1* and *ARGM-LOC*), respectively. In this way, we form three predicate-argument spans and split them into individual words: (*He*, *slices*), (*slices*, *tomatoes*), (*slices*, *in*, *the*, *kitchen*). One sentence is likely to have multiple predicates, by adopting this strategy, we are able to obtain all potential predicate-argument spans in the given sentence. We further utilise these extracted spans to form a span-based sentence representation.

**Aggregation:** After obtaining all predicate-argument spans, we derive corresponding span representations by looking at BERT's token representations. In BERT/RoBERTa, tokenization yields sub-tokens, whereas in the created spans, we have an entire word token. To properly align them, we use the same tokenizer to break the original word into sub-tokens and represent it as a sequence of sub-tokens in the span if a sub-token exists. Given a predicate-argument span sequence $s = \{s_1, s_2, ..., s_N\}$ in the sentence, where $N$ denotes the number of spans and every span $s_i$ consists of tokens $\{x_1, ..., x_l\}$ that make up the span. For each span $s_i$, we obtain its dense vector representation $h_i$ by taking mean-pooling over all tokens in it:

$$h_i = MeanPooling(x_1, .., x_l) \tag{1}$$

Therefore, the whole representation for span sequence $s$ is represented as $h = \{h_1, h_2, ..., h_N\}$, where $h_i \in \mathbb{R}^D$.

Then, we aggregate information from all spans using a simple self-attentive mechanism. Following Sun et al. (2020), this is achieved by first assigning weights $\alpha_i$ to each span $h_i$ and combining these representations using weighted sum:

$$\begin{aligned} o_i &= W \cdot h_i + b \\ \alpha_i &= \frac{exp(o_i)}{\sum_{j=1}^{N} exp(o_j)} \end{aligned} \tag{2}$$

where $W \in \mathbb{R}^{1 \times D}$ and $b$ are learnable parameters. The span-based sentence representation $\hat{h}$ from the aggregation component is the weighted average of all predicate-argument span representations:

$$\hat{h} = \sum_{i=1}^{N} \alpha_i \cdot h_i \tag{3}$$

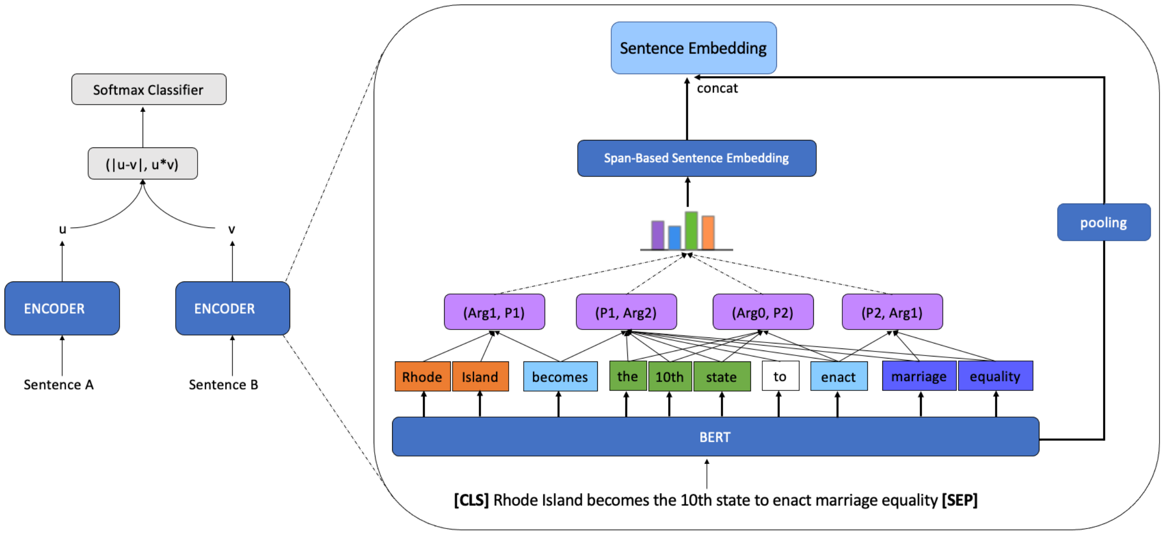


Figure 1: The proposed model in siamese structure. All parameters are shared between two encoders.

The weights are learned during training. This gives the model flexibility to decide the best combination method on its own. The combination of self-attentive mechanism and predicate-argument spans allow us to construct structure-aware sentence embeddings without introducing a large number of parameters.

**Connect BERT and Aggregation:** The final sentence representation is the concatenation of both BERT mean-pooling based sentence representation and the span-based sentence representation. Sentence embeddings of the given sentence-pair are then combined using vector operations before passing to the final classifier for training as shown in Figure 1. To combine the embeddings, we use the concatenation of the element-wise multiplication $u * v$ and the absolute element-wise difference $|u - v|$. This is different to the typical aggregation strategy used with SBERT/SRoBERTa (Reimers and Gurevych, 2019) which introduces asymmetry into the task by using (u, v, |u-v|). In initial experiments, we tested the prediction consistency of SBERT on paraphrase tasks and found that, across different datasets, between 2.78% and 9.16% of test predictions change when the sentence order is reversed. Furthermore, here, we find that SBERT performs worse on paraphrase tasks with (u, v, |u-v|) compared to (|u-v|, $u * v$). Results are given in Table 5 and discussed in Section 5.1.

Finally, we note that in this siamese structure, all parameters are shared and are updated accordingly. Cross-entropy loss is used for optimisation.

## 4 Experiments

We compare our model with SBERT, SRoBERTa[1] and the SBERT-RGCN (Peng et al., 2021) which utilises RGCN to incorporate structures into SBERT with an introduction of 32 million extra parameters[2]. The original sentence-pair aggregation strategy of these models is (u, v, |u-v|). We modify this to (|u-v|, $u * v$) as discussed in Section 3, but we retain the original notation. We adopt their structures and directly fine-tune the whole model on downstream tasks from the original BERT/RoBERTa checkpoints. We considered two strategies to apply SBERT on classification inference. One involved finding the optimal similarity threshold on development set and then applying it on the test set, while the other involved directly using the trained classifier. In this paper, we adopted to the latter approach since we find it gave improved and more robust results.

### 4.1 Datasets

We evaluate our model on six binary paraphrase identification benchmarks. The statistics of these datasets are listed in Table 1. Below we give some basic descriptions:

- **Microsoft Research Paraphrase Corpus (MSRP)**: A corpus of sentence pairs obtained

[1] https://github.com/UKPLab/sentence-transformers. Due to limited computational resources, all pre-trained models are of base size.

[2] SBERT-RGCN tried both dependency and semantic parse trees. In the following experiments, we use semantic parse trees that that capture predicate-argument structures.

by clustering news articles with an SVM classifier and human annotations (Dolan and Brockett, 2005). It has 4,076 train data and 1,725 test data. In this paper, we split 10% of training data as the validation set according to GLUE (Wang et al., 2019) standardised splits.

- **TwitterURL**: To better study the realistic language usage, Lan et al. (2017) proposed the TwitterURL corpus where sentence pairs in the dataset are collected from tweets that share the same URL of news articles.

- **PIT2015**: The corpus is derived from Twitter's trending topic data, containing 18,763 sentence pairs on more than 400 distinct topics (Xu et al., 2015). Given we are dealing with binary classification, we discard questionable sentence pairs and obtain 16,510 sentence pairs in total. This dataset contains paraphrase pairs that are lexically diverse and non-paraphrase pairs that are lexically similar, but semantically dissimilar. To capture these properties, models are assumed to be structure-aware.

- **Quora Question Pairs (QQP)**: The Quora Question Pairs dataset is a collection of potential duplicate question pairs from the QA website Quora.com (Iyer et al., 2017). In this paper, we adopt the same split strategy as in Wang et al. (2017).

- **PAWS_QQP**: QQP is criticised for lacking negative examples with high lexical overlapping. Models trained on QQP tend to mark any sentence pairs with a high word overlap as paraphrases despite clear clashes in meaning. In light of these factors, Zhang et al. (2019b) proposed a new paraphrase identification dataset which has extremely high lexical overlap by applying word scrambling and back translation to sentences in QQP.

- **PAWS_Wiki**: Similar to PAWS_QQP, Zhang et al. (2019b) applied the same technique on sentences obtained from Wikipedia articles to construct sentence pairs. Both PAWS datasets aim to measure sensitivity of models on word order and sentence structure.

Due to the lack of development set for PAWS_QQP, we use PAWS_Wiki's development set for early stopping since they are constructed in the same way.

| Datasets | Train | Dev | Test |
|---|---|---|---|
| MSRP | 3,668 | 408 | 1,725 |
| TwitterURL | 37,976 | 4,224 | 9,334 |
| PIT2015 | 11,530 | 4,142 | 838 |
| QQP | 384,348 | 10,000 | 10,000 |
| PAWS_QQP | 11,986 | 8,000 | 677 |
| PAWS_Wiki | 49,401 | 8,000 | 8,000 |

Table 1: Statistics of all six benchmarks used in this work.

### 4.2 Training Details

Following the SBERT training protocol, we train all models with a batch-size of 16. We tune the learning rate in the range of (2e-5, 3e-5, 5e-5) with Adam optimizer and a linear learning rate warm-up over 10% of the training data. All models are trained for four epochs and use the development set for early stopping with a patience of 5. The evaluation step depends on actual tasks but roughly we evaluate them on development set twice each epoch. The maximum sequence length is set to be 128. All experiments are conducted on NVIDIA Titan V GPUs.

### 4.3 Evaluation

The main experiment results are summarised in Table 2. We report the averaged F1 score of positive class with standard error. In the table, we see that the proposed model consistently outperforms its SBERT and SRoBERTa versions on 5 paraphrase identification tasks and show competitive, but not statistically significantly different, results on QQP. As also revealed by Zhang et al. (2019b), negative examples in QQP often have low lexical overlap, and models trained on it tend to mark any sentence pairs with high word overlap as paraphrases. We reason that the QQP task is relatively easy and does not require much structural information to achieve high scores. For tasks like PAWS_QQP and PIT2015 where structures are more important, the performance gap is more apparent. Furthermore, despite bringing in more than 30 million parameters and explicitly encoding sentence structures with a complex model, SBERT-RGCN does not significantly outperform SBERT on most of these tasks (excluding PIT2015) and underperforms our

| | QQP | TwitterURL | MSRP | PAWS_Wiki | PAWS_QQP | PIT2015 |
|---|---|---|---|---|---|---|
| SBERT | **90.78±0.09** | 70.85±0.28 | 81.67±0.46 | 81.57±0.53 | 66.01±0.45 | 52.03±1.44 |
| SBERT-RGCN | 90.41±0.09 | 70.40±0.22 | 81.70±0.17 | 81.14±0.81 | 66.22±0.75 | 59.11±0.93 |
| PAS+SBERT | 90.74±0.06 | **72.12±0.06** | **83.42±0.23** | **82.60±0.18** | **68.85±0.73** | **59.19±1.85** |
| SRoBERTa | **90.79±0.09** | 70.69±0.23 | 81.69±0.53 | 81.42±0.93 | 67.35±0.97 | 52.67±2.75 |
| PAS+SRoBERTa | 90.76±0.03 | **72.04±0.23** | **83.22±0.46** | **82.87±0.35** | **69.68±0.72** | **59.50±2.74** |

Table 2: Results on six paraphrase identification tasks, we calculate the F1 score of the positive class given most of them are imbalanced datasets. We run 5 times with random seeds and report the mean with standard error. Cells marked bold have the best performance in each column.

| | Params |
|---|---|
| SBERT-base | 109M |
| PAS only | +768 |
| PAS+SBERT | +3840 |
| SBERT-RGCN | + 32M |

Table 3: The parameter comparison between different models.

proposed model.

In summary, the proposed model shows improved performances on five out of six paraphrase tasks, demonstrating the advantages of bringing in the predicate-argument structure. Moreover, when we combine PAS with SRoBERTa, we get similar performance gains, proving the generalisation ability of our component. Similarly in Reimers and Gurevych (2019), we only observe minor differences by replacing SBERT with SRoBERTa.

The number of parameters for different approaches are shown in Table 3. We note that compared to SBERT, our proposed model introduces 3,840 additional parameters, and if we only consider the span-based component, only 768 additional parameters are introduced. In comparison, SBERT-RGCN brings in more than 32 million parameters.

## 5 Analysis

In order to better understand how the performance gain is achieved, we have carried out several experiments to investigate different aspects of the proposed model. The following experiments are conducted only with SBERT, since we would expect similar results with SRoBERTa.

### 5.1 Ablation Study

Our proposed model is made of different components and so it is important to dissect the impact of each component so as to explain the improved performance. Given that the final sentence representation is the concatenation of both mean-pooling based BERT representation and the weighted sum of span representations, we first assess their performances individually on six datasets. Furthermore, it is necessary to assess the impact of adopting the weighted sum strategy when we derive span-based sentence representations. We experimented with simple averaging over all spans and compared it with the weighted sum where the model learns to combine different spans.

The ablation experiment results are shown in Table 4. The SBERT-only component appears to perform the poorest, and the complete model achieves the highest performance on five out of six tasks. By only using the span-based sentence representation, we are able to achieve significant improvements over SBERT on most of these tasks. The improvements are more substantial when concatenating with SBERT sentence representations. We observe considerable performance decreases on most tasks when switching from weighted sum to simple averaging, which further verifies the benefits of adopting learnable weights.

The original asymmetric sentence aggregation strategy (u, v, |u-v|) of SBERT assumes an ordering of the sentences by concatenating two individual sentence embeddings. $u * v$ has been widely used elsewhere (Conneau et al., 2017; Cer et al., 2018) and we found that concatenating this with |u-v| gave the best performance on all tasks. The results are summarised in Table 5. Therefore, we use (|u-v|, $u * v$) as our concatenation method for all of our other experiments.

### 5.2 Span Strategy Analysis

The impact of incorporating predicate-argument spans into SBERT in terms of the performance on

|  | QQP | TwitterURL | MSRP | PAWS_Wiki | PAWS_QQP | PIT2015 |
|---|---|---|---|---|---|---|
| PAS+SBERT | 90.74±0.06 | 72.12±0.06 | 83.42±0.23 | 82.60±0.18 | 68.85±0.73 | 59.19±1.85 |
| - SBERT-only | **90.78±0.09** | 70.85±0.28 | 81.67±0.46 | 81.57±0.53 | 66.01±0.45 | 52.03±1.44 |
| - PAS only | 90.70±0.08 | **71.64±0.14** | **82.91±0.12** | **82.26±0.34** | **67.38±0.22** | **54.95±1.45** |
| - PAS only<br>(simple average) | 90.11±0.13 | 71.09±0.30 | 82.13±0.14 | 81.85±0.26 | 66.55±0.41 | 51.82±1.31 |

Table 4: Experimental results for ablation study. The second row gives the result for the complete model and following rows for different components. We calculate F1 score of the positive class and report the mean with standard error across 5 runs with random seeds. Cells marked bold perform the best among different components.

|  | QQP | TwitterURL | MSRP | PAWS_Wiki | PAWS_QQP | PIT2015 |
|---|---|---|---|---|---|---|
| (u, v, \|u-v\|) | 90.52±0.08 | 70.83±0.27 | 80.68±0.36 | 80.90±0.78 | 65.91±0.47 | 45.71±1.25 |
| (\|u-v\|) | 65.46±1.80 | 58.17±2.36 | 80.48±0.21 | 61.92±0.97 | 64.91±4.39 | 34.25±0.54 |
| (\|u-v\|, u*v) | **90.78±0.09** | **70.85±0.28** | **81.67±0.46** | **81.57±0.53** | **66.01±0.45** | **52.03±1.44** |

Table 5: Results on SBERT with different concatenation strategies. F1 score of the positive class with standard error across 5 random runs is reported. Cells marked bold give the best performance.

various paraphrase identification tasks has been investigated in the above experiments. We now address the question of whether it is the use of specifically predicate-argument based spans that is critical, or whether this is simply a result of the fact that we are benefiting from the use of representations based on spans rather than all tokens. To verify this, we further conduct experiments with different span strategies. We pick three paraphrase identification datasets for this purpose (MSRP, PAWS_QQP and PIT2015) since performance gaps between PAS+SBERT and SBERT are more apparent in previous experiments.

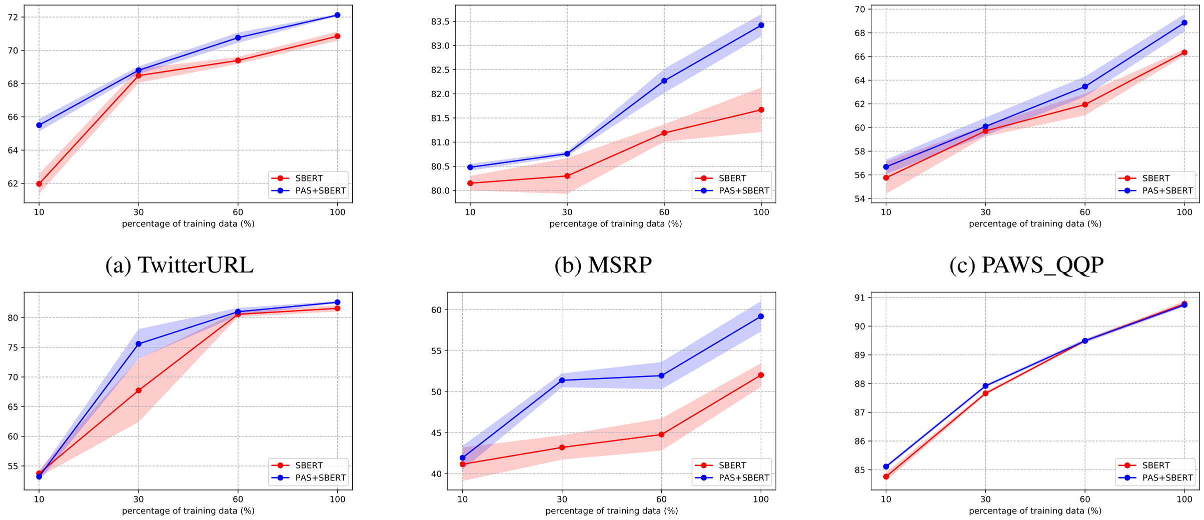
| Task | Span Type | Span only | Self-Explain* | SBERT |
|---|---|---|---|---|
| MSRP | PAS | **82.91±0.12** | 81.23±0.27 | 81.67±0.46 |
|  | Continuous Random Span | 81.40±0.43 |  |  |
|  | Random Span | 81.86±0.47 |  |  |
| PAWS_QQP | PAS | **67.38±0.22** | 66.88±0.46 | 66.01±0.45 |
|  | Continuous Random Span | 65.45±0.44 |  |  |
|  | Random Span | 65.75±0.74 |  |  |
| PIT2015 | PAS | **54.95±1.45** | 47.60±1.01 | 52.03±1.44 |
|  | Continuous Random Span | 51.62±1.92 |  |  |
|  | Random Span | 50.85±2.11 |  |  |

Table 6: Evaluation for different span strategies using our span-only component on three datasets. We calculate the F1 score of the positive class and report the mean with standard error across 5 runs with random seeds. Cells marked bold have the best performance in the row. * denotes the Self-Explain based bi-encoder.

Here we experiment with two other span strategies. The first, inspired by the Self-Explain model (Sun et al., 2020), is the continuous random span, where instead of following the predicate-argument structure, we randomly sample continuous word sequences from the sentence to build a span. The length of the sampled spans is arbitrary. To make a fair comparison, the number of sampled spans is the same as that of the predicate-argument spans in the sentence. The other one is random span, where we do not necessarily sample continuous words, but allow word leaps from one to another. In this strategy, we have the opportunity to get both continuous and discontinuous word sequences to form spans, which better matches the scenario of PAS. The only difference between these two strategies and PAS is the words in the span.

We also experiment with a bi-encoder approach more directly based on the Self-Explain cross-encoder model (Sun et al., 2020). This model extracts all possible continuous text spans and obtains span representations by taking the first and last token in the span, passing them through a complex mapping function. Unlike our PAS model, this model brings in 2.36 million more parameters compared to SBERT.

Table 6 shows the results. In order to focus on the impact of different span strategies, we only use the PAS component and do not concatenate it with SBERT sentence representations in this experiment. As shown in the table, the PAS-based model outperforms the Self-Explain inspired bi-encoder model and achieves the best performance among all other span-based models. The continuous random span and the random span model have



(a) TwitterURL

(b) MSRP

(c) PAWS_QQP

(d) PAWS_Wiki

(e) PIT2015

(f) QQP

Figure 2: Performance of SBERT and our proposed model on six benchmarks with different training data size. X-axis: Percent of supervised training data. Y-axis: F1 score of the positive class. The coloured bands indicate the standard error across 5 random runs.

comparable performances with SBERT. This is expected because they do not introduce linguistically-meaningful structures and the impact of contextualisation makes them similar to SBERT despite the absence of some tokens. Despite introducing 2.36 million more parameters, the Self-Explain inspired bi-encoder model does not show consistent improvements over SBERT on these datasets, which further suggests the importance of the predicate-argument structure in this paraphrase identification task.

### 5.3 Training Size Analysis

In order to examine the stability of our model and the impact of the predicate-argument structure when different sizes of training data are available, we conduct experiments with different training data scales. We randomly sample from 10% to 100% data (10%, 30%, 60%, 100%) from the training set as training data. We show the results in Figure 2. In spite of limited increased parameters, the proposed model appears to yield consistent improvements across different training scales. We also note that, whilst our proposed model performs comparably to SBERT on QQP when trained with the complete data-set, we can see that when only a small proportion of training data (e.g. 10%, 30%) is available, our model demonstrates improvements over SBERT. Thus the introduction of predicate-argument structures may be more beneficial with limited annotated training data.

## 6 Conclusion

In this work, we propose a method which effectively introduces sentence structure to a sentence embedding via the aggregation of predicate-argument spans (PAS). Experiments with SBERT and SRoBERTa show that such method brings improvements on six paraphrase identification tasks. Compared to models based on RGCNs, our method obtains more consistent benefits with minimal increased cost in terms of numbers of parameters. Upon closer investigation, we show that the PAS component and its learnable weights play a substantial impact in the performance gain. This PAS component, as demonstrated with SRoBERTa, can be easily extended to other models that require the generation of sentence embeddings. Our future work will include enhancing the structural difference between sentences by taking use of the argument tag information.

## References

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Hannah Chen, Yangfeng Ji, and David K Evans. 2020. Pointwise paraphrase appraisal is potentially problematic. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 150–155.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1816–1829.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, pages 937–948.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. 2017. First quora dataset release: Question pairs. *data. quora. com*.

Zornitsa Kozareva and Andrés Montoyo. 2006. Paraphrase identification on the basis of supervised machine learning techniques. In *International conference on natural language processing (in Finland)*, pages 524–533. Springer.

Raymond Kozlowski, Kathleen F McCoy, and K Vijay-Shanker. 2003. Generation of single-sentence paraphrases from predicate/argument structure using lexico-grammatical resources. In *Proceedings of the second international workshop on Paraphrasing*, pages 1–8.

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234.

Wuwei Lan and Wei Xu. 2018. Character-based neural networks for sentence pair modeling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 157–163.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2020. Improving bert with syntax-aware local attention. *arXiv preprint arXiv:2012.15150*.

Linqing Liu, Wei Yang, Jinfeng Rao, Raphael Tang, and Jimmy Lin. 2019a. Incorporating contextual and syntactic structures improves semantic similarity modeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1204–1209.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Aaai*, volume 6, pages 775–780.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Qiwei Peng, David Weir, and Julie Weeds. 2021. Structure-aware sentence encoder in bert-based Siamese network. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 57–63, Online. Association for Computational Linguistics.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Jian-fang Shan, Zong-tian Liu, and Wen Zhou. 2009. Sentence similarity measure based on events and content words. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 7, pages 623–627. IEEE.

Zijun Sun, Chun Fan, Qinghong Han, Xiaofei Sun, Yuxian Meng, Fei Wu, and Jiwei Li. 2020. Self-explaining structures improve nlp models. *arXiv preprint arXiv:2012.01786*.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Zhaofeng Wu, Hao Peng, and Noah A Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242.

Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2021. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *Proceedings of the Web Conference 2021*, pages 2466–2475.

Wei Xu, Chris Callison-Burch, and William B Dolan. 2015. Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 1–11.

Yinchuan Xu and Junlin Yang. 2019. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 96–101.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709.

Da Yin, Tao Meng, and Kai-Wei Chang. 2020. Sentibert: A transferable transformer-based architecture for compositional sentiment semantics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3695–3706.

Kun Zhang, Le Wu, Guangyi Lv, Meng Wang, Enhong Chen, and Shulan Ruan. 2021. Making the relation matters: Relation of relation learning network for sentence semantic matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14411–14419.

Yinan Zhang, Raphael Tang, and Jimmy Lin. 2019a. Explicit pairwise word interaction modeling improves pretrained transformers for english semantic similarity tasks. *arXiv preprint arXiv:1911.02847*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019b. Paws: Paraphrase adversaries from word scrambling. 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 1298–1308.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware bert for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.