Towards Structure-aware Paraphrase Identification with Phrase Alignment Using Sentence Encoders

Qiwei Peng David Weir Julie Weeds
University of Sussex
Brighton, UK
{qiwei.peng, d.j.weir, j.e.weeds}@sussex.ac.uk

Abstract

Previous works have demonstrated the effectiveness of utilising pre-trained sentence encoders based on their sentence representations for meaning comparison tasks. Though such representations are shown to capture hidden syntax structures, the direct similarity comparison between them exhibits weak sensitivity to word order and structural differences in given sentences. A single similarity score further makes the comparison process hard to interpret. Therefore, we here propose to combine sentence encoders with an alignment component by representing each sentence as a list of predicate-argument spans (where their span representations are derived from sentence encoders), and decomposing the sentence-level meaning comparison into the alignment between their spans for paraphrase identification tasks. Empirical results show that the alignment component brings in both improved performance and interpretability for various sentence encoders. After closer investigation, the proposed approach indicates increased sensitivity to structural difference and enhanced ability to distinguish non-paraphrases with high lexical overlap.

1 Introduction

Sentence meaning comparison measures the semantic similarity of two sentences. Specifically, the task of paraphrase identification binarises the similarity as paraphrase or non-paraphrase depending on whether they express similar meanings (Bhagat and Hovy, 2013). This task benefits many natural language understanding applications, like plagiarism identification (Chitra and Rajkumar, 2016) and fact checking (Jiang et al., 2020), where it is important to detect same things said in different ways.

The difference in sentence structures is important for distinguishing their meanings. However, as shown in Table 1 and 3, many existing paraphrase identification datasets exhibit high correlation between positive pairs and the degree of their lexical overlap, such as the Microsoft Research Paraphrase Corpus (MSRP) (Dolan and Brockett, 2005). Models trained on them tend to mark sentence pairs with high word overlap as paraphrases despite clear clashes in meaning. In light of this, Zhang et al. (2019b) utilised word scrambling and back translation to create the Paraphrase Adversaries from Word Scrambling (PAWS) datasets which are mainly concerned with word order and structure by creating paraphrase and non-paraphrase pairs with high lexical overlap. As also shown in these two tables, sentence pairs in the PAWS datasets demonstrate much higher lexical overlap and lower correlation, which requires models to pay more attention to word order and sentence structure to successfully distinguish non-paraphrases from paraphrases.

Recently, various pre-trained sentence encoders have been proposed to produce high-quality sentence embeddings for downstream usages (Reimers and Gurevych, 2019; Thakur et al., 2021; Gao et al., 2021). Such embeddings are compared to derive a similarity score for different meaning comparison tasks, including paraphrase identification. Though widely used, sentence encoders still face challenges from different aspects in case of meaning comparison. Pre-trained models are observed to capture structural information to some extent (Clark et al., 2019; Hewitt and Manning, 2019; Jawahar et al., 2019). However, as we will demonstrate in this work, their direct comparison of two sentence vectors performs poorly on PAWS datasets indicating weak sensitivity to structural difference, though they achieve good performance on other general paraphrase identification datasets like MSRP. In addition, the single similarity score derived from the comparison of two vectors is difficult to interpret. This thus motivates us to find a better way of utilising sentence encoders for meaning comparison.

Elsewhere, researchers have worked on decom-

Proceedings of the 29th International Conference on Computational Linguistics, pages 4113–4123
October 12–17, 2022.

Dataset	Sentence A	Sentence B	Label
MSRP	The Toronto Stock Exchange opened on time and slightly lower.	The Toronto Stock Exchange said it will be business as usual on Friday morning.	N
	More than half of the songs were purchased as albums, Apple said.	Apple noted that half the songs were purchased as part of albums.	Y
PAWS	What factors cause a good person to become bad?	What factors cause a bad person to become good?	N
	The team also toured in Australia in 1953.	In 1953, the team also toured in Australia.	Y

Table 1: Example sentence pairs taken from both MSRP and PAWS datasets. Y stands for paraphrases while N stands for non-paraphrases.

posing sentence-level meaning comparison into comparisons at a lower level, such as word and phrase-level, which largely increased the interpretability (He and Lin, 2016; Chen et al., 2017; Zhang et al., 2019a). Alignment is the core component in these proposed systems, where sentence units at different levels are aligned through either training signals or external linguistic clues, after which a matching score is derived for sentence-level comparison. Here, we argue that, instead of comparing sentence meaning by using sentence embeddings, it would be better to combine sentence encoders with alignment components in a structure-aware way to strengthen the sensitivity to structural difference and to gain interpretability.

An important aspect of sentence meaning is its predicate-argument structure, which has been utilised in machine translation (Xiong et al., 2012) and paraphrase generation (Ganitkevitch et al., 2013; Kozlowski et al., 2003). Given the importance of detecting structural differences in paraphrase identification tasks, we propose to represent each sentence as a list of predicate-argument spans where span representations are derived from sentence encoders, and to decompose sentence-level meaning comparison into the direct comparison between their aligned predicate-argument spans by taking advantage of the Hungarian algorithm (Kuhn, 1956; Crouse, 2016). The sentence-level score is then derived by aggregation over their aligned spans. Without re-training, the proposed alignment-based sentence encoder can be used with enhanced structure-awareness and interpretability.

As pre-trained sentence encoders produce contextualised representations, two phrases of different meaning might be aligned together due to their similar syntactic structure and contexts. For example:

a) *Harris announced on twitter that he will quit.*

b) *James announced on twitter that he will quit.*

Unsurprisingly, the span *Harris announced* will be aligned to the span *James announced* with a high similarity score given that they share exactly the same context and syntactic structure. However, it might be problematic to consider this high similarity score when we calculate the overall score given clear clashes in the meaning at sentence-level. In this regard, we further explore how the contextualisation affects paraphrase identification by comparing aligned phrases based on their de-contextualised representations.

Empirical results show that the inclusion of the alignment component leads to improvements on four paraphrase identification tasks and demonstrates increased ability to detect non-paraphrases with high lexical overlap, plus an enhanced sensitivity to structural difference. Upon closer investigation, we find that applying de-contextualisation to aligned phrases could further help to recognise such non-paraphrases.

In summary, our contributions are as follows:

1) We propose an approach that combines sentence encoders with an alignment component by representing sentences as lists of predicate-argument spans and decomposing sentence-level meaning comparison into predicate-argument span comparison.
2) We provide an evaluation on four different paraphrase identification tasks, which demonstrates both the improved sensitivity to structures and the interpretability at inference time.
3) We further introduce a de-contextualisation step which can benefit tasks that aim to identify non-paraphrases of extremely high lexical overlap.

2 Related Work

2.1 Sentence Encoders

Sentence encoders have been studied extensively in years. Kiros et al. (2015) abstracted the skip-gram

model (Mikolov et al., 2013) to the sentence level and proposed Skip-Thoughts by using a sentence to predict its surrounding sentences in an unsupervised manner. InferSent (Conneau et al., 2017), on the other hand, leveraged supervised learning to train a general-purpose sentence encoder with BiLSTM by taking advantage of natural language inference (NLI) datasets. Pre-trained language models like BERT (Devlin et al., 2019) are widely used to provide a single-vector representation for the given sentence and demonstrate promising results across a variety of NLP tasks. Inspired by InferSent, Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) produces general-purpose sentence embeddings by fine-tuning BERT on NLI datasets. However, as investigated by Li et al. (2020), sentence embeddings produced by pre-trained models suffer from anisotropy, which severely limits their expressiveness. They then proposed a post-processing step to map sentence embeddings to an isotropic distribution which largely improves the situation. Similarly, Su et al. (2021) proposed a whitening operation for post-process, which aims to alleviate the anisotropy problem. Gao et al. (2021), on the other hand, proposed the SimCSE model by fine-tuning pre-trained sentence encoders with a contrastive learning objective (Chen et al., 2020) along in-batch negatives (Henderson et al., 2017; Chen et al., 2017) on NLI datasets, improving both the performance and the anisotropy problem. Though sentence encoders have achieved promising performance, the current way of utilising them for meaning comparison tasks has known drawbacks and could benefit from the fruitful developments of the alignment component, which have been widely used in modelling sentence pair relations.

2.2 Alignment in Sentence Pair Tasks

Researchers have been investigating sentence meaning comparison for years. One widely used method involves decomposing the sentence-level comparison into comparisons at a lower level. MacCartney et al. (2008) aligned phrases based on their edit distance and applied the alignment to NLI tasks by taking average of aligned scores. Shan et al. (2009) decomposed sentence-level similarity score into the direct comparison between events and content words based on WordNet (Miller, 1995). Sultan et al. (2014) proposed a complex alignment pipeline based on various linguistic features, and predicted the sentence-level semantic similarity by taking the proportion of their aligned content words. The alignment between two syntactic trees are used along with other lexical and syntactic features to determine whether two sentences are paraphrases with SVM (Liang et al., 2016).

Similar ideas are combined with neural models to construct alignments based on the attention mechanism (Bahdanau et al., 2015). They can be seen as learning soft alignments between words or phrases in two sentences. Pang et al. (2016) proposed MatchPyramid where a word-level alignment matrix was learned, and convolutional networks were used to extract features for sentence-level classification. More fine-grained comparisons between words are introduced by PMWI (He and Lin, 2016) to better dissect the meaning difference. Wang et al. (2016) put focus on both similar and dissimilar alignments by decomposing and composing lexical semantics over sentences. ESIM (Chen et al., 2017) further allowed richer interactions between tokens. These models are further improved by incorporating context and structure information (Liu et al., 2019), as well as character-level information (Lan and Xu, 2018). Recently, Pre-trained models are exploited to provide contextualised representations for the PMWI (Zhang et al., 2019a). Instead of relying on soft alignments, some other models tried to take the phrase alignment task as an auxiliary task for sentence semantic assessments (Arase and Tsujii, 2019, 2021), and to embed the Hungarian algorithm into trainable end-to-end neural networks to provide better aligned parts (Xiao, 2020). Considering pre-trained sentence encoders are often directly used to provide fixed embeddings for meaning comparison, in this work, we propose to combine them with the alignment component at inference time so that it can be used with enhanced structure-awareness without re-training.

3 Our Approach

Instead of generating a single-vector representation for meaning comparison based on sentence encoders, we propose to represent each sentence as a list of predicate-argument spans and use sentence encoders to provide its span representations. The comparison between two sentences is then based on the alignment between their predicate-argument spans. As depicted in Figure 1, the approach can be considered as a post-processing step and consists of the following main components:

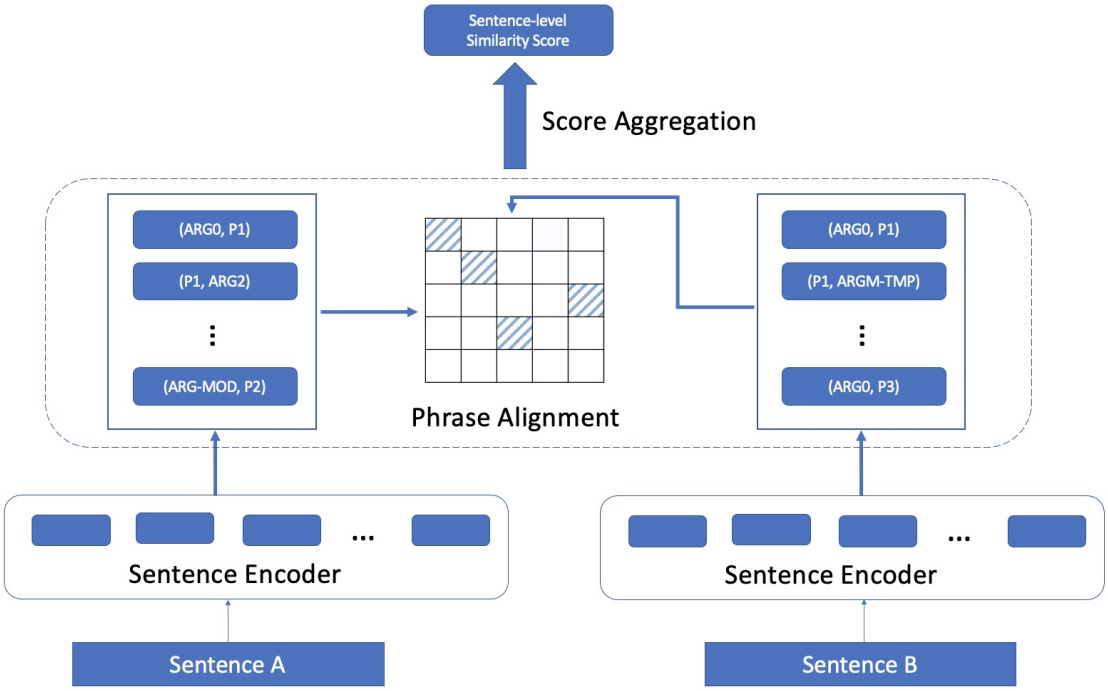

Figure 1: The proposed approach for paraphrase identification that combines sentence encoders with the phrase alignment at inference time. Predicate-argument spans are first extracted from sentences. Span representations are then derived from contextualised token representations. We perform Hungarian algorithm to align extracted phrase spans and obtain the sentence-level similarity score by aggregation over aligned spans. The alignment matrix is useful for interpretation.

Sentence Encoders: The input sentences are first fed into sentence encoders to produce contextualised token representations that will later be used to create context-aware phrase representations from the last hidden layer. The phrase representation will be the basic unit of our meaning comparison method.

Predicate Argument Spans (PAS): For each sentence, we first apply a BERT-based semantic role labelling (SRL) tagger provided by AllenNLP (Gardner et al., 2018) to obtain both predicates and relevant arguments for each sentence. To generate predicate argument spans, we group the predicate and its arguments together and order them according to their original position in the sentence. Following is an example of predicate-argument spans from a sentence:

> *James ate some cheese whilst thinking about the play.*

Two predicates, *ate* and *thinking*, are extracted by the tagger. As shown in Figure 2, a number of arguments with different relations are discovered for each predicate. We further group them into predicate-argument spans. For the given sentence, we will have three spans for the predicate *ate*: (*James*, *ate*), (*ate*, *some*, *cheese*), (*ate*, *whilst*, *thinking*, *about*, *the*, *play*) and two spans for the predicate *thinking*: (*James*, *thinking*), (*thinking*, *about*, *the*, *play*). If no predicate or associated argument is found, we take the whole sentence itself as a long span.

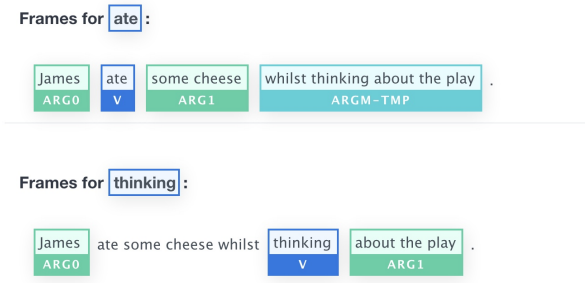

Figure 2: The extracted predicates and relevant semantic arguments for the given example sentence. Outputs are produced by the AllenNLP SRL tagger.

Phrase Alignment: After obtaining all predicate-argument spans, we derive their span representations based on the used encoder. As previous works have shown, aligning with contextual information could achieve better performance and help with disambiguation (Arase and Tsujii, 2020; Dou

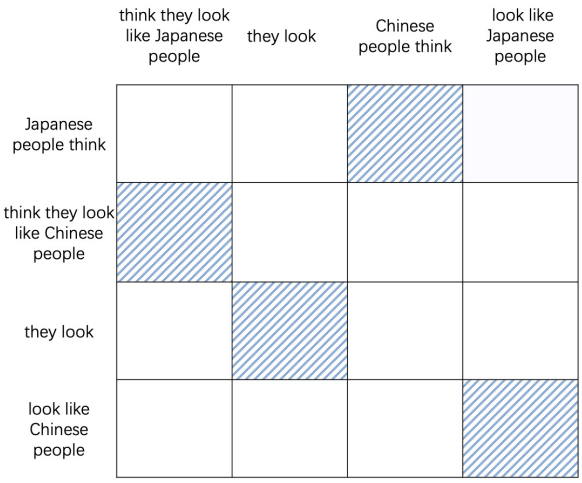

Figure 3: The predicate-argument span alignment between the example pair taken from PAWS_QQP, sentence A: *do Chinese people think they look like Japanese people?* and sentence B: *do Japanese people think they look like Chinese people?*

and Neubig, 2021). We take the mean-pooling over all tokens in the span to produce a contextualised span representation for later alignment. The tokenization strategy in BERT generates sub-tokens, whereas in the produced spans, we have word tokens. To align them properly, we use the same tokenizer to break the original word into sub-tokens and represent it as a list of sub-tokens in the span if a sub-token exists. Given two collections of predicate-argument span representations, $p = \{p_1, p_2, ..., p_M\}$ and $q = \{q_1, q_2, ..., q_N\}$, we are trying to find the best alignments between them. This can be viewed as a standard assignment problem that has been extensively handled by Hungarian algorithm (Kuhn, 1956). A similarity matrix, C, is constructed for each pair of sentences where the row has one collection of spans and the column has another. The value for each entry cell, C_{mn}, is the cosine similarity score between the two span, p_m and q_n. The task of finding the best alignment is to find alignments among two collections that give the maximum score:

$$max \sum_{m} \sum_{n} C_{mn} X_{mn} \quad (1)$$

X is a boolean matrix where $X[m, n] = 1$ if span m is assigned to span n. We apply the modified Jonker-Volgenant algorithm[1] (Crouse, 2016) to find the best alignments that maximise the overall score. After discovering the optimal X, we obtain a collection of aligned span pairs associated with their cosine similarity scores, $A = \{A_1, A_2, ..., A_l\}$. One alignment example taken from PAWS is given in Figure 3.

Score Aggregation: To produce a sentence-level similarity score for the given pair, we simply take the mean-pooling over scores of all aligned parts:

$$Score_{ij} = MeanPooling(A_1, .., A_l) \quad (2)$$

The similarity score between sentence i and sentence j is the average score of their aligned spans, and will be used for determining whether the sentence pair is paraphrase or non-paraphrase. The alignment matrix, as shown in Figure 3, is useful to explain how the overall score is derived and why.

4 Experiments

We follow the same two-step procedure in previous work for evaluation (Li et al., 2020; Thakur et al., 2021). For vanilla sentence encoders, we first generate fixed sentence embeddings, and then derive sentence-level similarity scores by calculating the cosine similarity between two embeddings. For sentence encoders combined with the alignment component, we derive sentence-level similarity scores by aggregation over cosine scores of all aligned spans where span representations are derived from sentence encoders. Otherwise specifically stated, the alignment is performed between predicate-argument spans (PAS). Their performances under these two scenarios are evaluated and compared. We here experiment with three widely used sentence encoders, BERT-base (Devlin et al., 2019), Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) and SimCSE (Gao et al., 2021).

Datasets	Train	Dev	Test
PAWS_QQP	11,986	-	677
PAWS_Wiki	49,401	8,000	8,000
MSRP	3,668	408	1,725
TwitterURL	37,976	4,224	9,334

Table 2: Statistics of all four datasets used in this work.

4.1 Datasets

In this work, we evaluate the proposed approach on four different paraphrase identification tasks.

[1]We use its Scipy implementation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linear_sum_assignment.html

The statistics of these datasets are listed in Table 2. Below we give some basic descriptions:

- **PAWS_QQP**: In order to assess the sensitivity to word order and syntactic structure, Zhang et al. (2019b) proposed a paraphrase identification dataset which has extremely high lexical overlap by applying back translation and word scrambling to sentences taken from the Quora Question Pairs (Wang et al., 2017).
- **PAWS_Wiki**: Similar to PAWS_QQP, the same technique is applied to sentences obtained from Wikipedia articles to construct sentence pairs (Zhang et al., 2019b). Both PAWS datasets aim to measure sensitivity of models on word order and sentence structure.
- **Microsoft Research Paraphrase Corpus (MSRP)**: This corpus constructs sentence pairs 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 adopt the same split strategy as stated in GLUE (Wang et al., 2019).
- **TwitterURL**: Lan et al. (2017) proposed the TwitterURL corpus where sentence pairs in the dataset are collected by linking tweets that share the same URL of news articles. The corpus contains multiple references of both formal well-edited and informal user-generated texts.

Datasets	Lexical Overlap		
	Positive	Negative	Overall
PAWS_QQP	95.24%	96.79%	96.35%
PAWS_Wiki	84.50%	84.99%	84.77%
MSRP	55.95%	42.60%	51.48%
TwitterURL	29.28%	7.73%	11.94%

Table 3: The lexical overlap between sentence pairs across different datasets. We report both the overall figure and the figures for each class. We calculate the lexical overlap in terms of Jaccard Similarity with ngram=1.

The percentage of lexical overlap between sentence pairs in terms of their labels are summarised in Table 3. It shows that sentence pairs taken from the PAWS datasets generally have higher lexical overlap. Compared to datasets like MSRP and TwitterURL, where negative examples have lower lexical overlap than positive examples, the two PAWS datasets exhibit similar degrees of lexical overlap regardless of their labels. In light of this, we expect that models that are sensitive to word order and sentence structure would demonstrate greater improvements on the PAWS datasets in comparison to models without such sensitivity. Specifically, we put our focus on the PAWS datasets and explore whether different models capture structural differences.

4.2 Implementation Details

For sentence encoders used in this work, we generate sentence embeddings according to their default strategies. For BERT-base[2] and SBERT[3], we use the mean-pooling over the last hidden layer as its sentence representation, and for SimCSE[4], we use the CLS token after the trained MLP layer. For all experiments in this work, no training process is involved. In order to calculate evaluation metrics like accuracy and F1 score, we find optimal thresholds for different metrics on the development set, and apply them on test sets to binary the cosine similarity as paraphrase or non-paraphrase. Given PAWS_QQP does not have development set, we randomly sample 20% of its training data as the development set following the same class distribution. All experiments are conducted on RTX 3090 GPUs.

4.3 Evaluation

The main results are summarised in Table 4, and we report the F1 score of the positive class as well as the overall accuracy. It shows that, with our proposed approach, the performance of different sentence encoders can generally be improved. In addition, significant improvements are observed on PAWS datasets after we introduce the alignment component. This demonstrates the effectiveness of our proposed alignment-based sentence encoder and validates the improved sensitivity to word order and sentence structure. Furthermore, we find that the performance of different models, regardless of combining with the alignment component, is similar or competitive on MSRP and TwitterURL datasets. It suggests that both datasets are inadequate when used to detect the model's structure-

[2]We use its huggingface implementation: https://huggingface.co/bert-base-uncased

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

[4]https://github.com/princeton-nlp/SimCSE

	PAWS_QQP (F1/ACC)	PAWS_Wiki (F1/ACC)	TwitterURL (F1/ACC)	MSRP (F1/ACC)	AVG (F1/ACC)
BERT	37.13/72.97	61.28/56.75	63.24/85.65	80.50/70.38	60.54/71.44
+ Alignment	**47.46/75.18**	**63.08/62.58**	**65.26/86.52**	**80.96/70.61**	**64.19/73.72**
SBERT	33.95/74.74	61.83/60.63	65.61/87.04	81.68/73.39	60.61/73.95
+ Alignment	**52.75/77.70**	**62.52/64.51**	**66.60/87.33**	**82.10/73.80**	**65.99/75.84**
SimCSE	36.16/75.48	61.32/62.58	**69.20/87.74**	**82.80/74.61**	62.37/75.10
+ Alignment	**57.49/79.17**	**65.00/65.99**	67.83/87.27	81.70/73.68	**68.01/76.53**

Table 4: Results on four paraphrase identification tasks, we report both the F1 score of the positive class and the overall accuracy. Cells marked bold have the best performance in each column.

Models	PAWS_QQP (F1/ACC)	PAWS_Wiki (F1/ACC)
BERT-TokenLevel	40.13/73.41	62.33/62.36
BERT-RandomSpan	19.91/73.71	61.30/57.19
BERT-ContinuousRandom	39.86/74.74	61.25/57.66
BERT-PAS	**47.46/75.18**	**63.08/62.58**
SBERT-TokenLevel	47.51/75.04	61.65/64.15
SBERT-RandomSpan	38.89/73.12	61.07/59.08
SBERT-ContinuousRandom	46.56/74.74	61.28/58.49
SBERT-PAS	**52.75/77.70**	**62.52/64.51**
SimCSE-TokenLevel	50.74/74.00	62.03/63.28
SimCSE-RandomSpan	34.31/73.56	61.24/57.70
SimCSE-ContinuousRandom	40.74/77.25	61.30/57.24
SimCSE-PAS	**57.49/79.17**	**65.00/65.99**

Table 5: Evaluation using different span types for alignment. We report the F1 score of the positive class and the overall accuracy.

awareness for the structural information is not required to achieve high scores on them. Accordingly, compared to its alignment version, the lack of sensitivity to structural differences translates the slightly better performance on TwitterURL and MSRP obtained by SimCSE into much worse performance on the PAWS datasets. This further supports our previous arguments and demonstrates the advantages of introducing the alignment component to enhance structure-awareness.

5 Analysis

To better understand the improvements, we have conducted several experiments to investigate different aspects of the proposed approach. Given we are mostly interested in the performance on the two PAWS datasets, we only experiment and report the results on the PAWS_QQP and PAWS_Wiki in the following experiments.

5.1 Comparison to Other Span Strategies

In this experiment, we consider three more scenarios with different span types, and investigate the impact of the predicate-argument span. The alignment between different tokens are widely used in previous works, so here, instead of aligning predicate-argument spans, we directly conduct alignment at token-level. Two further strategies are explored regarding phrase-level alignment. Firstly, we randomly sample words from the sentence to make a span, where the words in each span might not necessarily be sequential. In the RandomSpan strategy, no linguistically-meaningful structures are preserved. Secondly, we randomly sample continuous word sequences to build a span, where the span must contain sequential texts. In this ContinuousRandom strategy, only sequential relations are preserved. 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. As demonstrated in Table 5, the alignment between predicate-argument spans outperforms all the others. In other words, the model's sensitivity to word order and structural differences can be greatly improved by comparing two sentences' predicate-argument structures.

5.2 Large Improvements in Recall

We have observed significant improvements on PAWS datasets by introducing the alignment between predicate-argument spans in previous experiments. It is crucially important to understand how the improvement is obtained. In this experiment, we look into the recall of positive and negative pairs. In PAWS_Wiki, we find that almost all sentence pairs are classified as positive by vanilla models given the near-zero recall for the negative class as shown in Table 6. Despite utterly incorrect pre-

	PAWS_QQP (recall of +)	PAWS_Wiki (recall of -)
BERT	32.46	0.09
+ Alignment	**36.65**	**25.96**
SBERT	24.08	9.14
+ Alignment	**47.64**	**29.53**
SimCSE	25.65	0.27
+ Alignment	**50.26**	**52.28**

Table 6: Results on PAWS_QQP and PAWS_Wiki. For PAWS_QQP, we report the recall of positive class and for PAWS_Wiki, we report the recall of negative class.

Models	PAWS_QQP (F1/recall of +)	PAWS_Wiki (F1/recall of -)
BERT-Alignment	47.46/36.65	63.08/25.96
+ decontext	**52.50/43.98**	**63.39/45.09**
SBERT-Alignment	52.75/47.64	62.52/29.53
+ decontext	**65.43/64.40**	**66.63/64.38**
SimCSE-Alignment	57.49/50.26	65.00/52.28
+ decontext	**65.16/68.06**	**67.32/54.14**

Table 7: The results on PAWS datasets after applying de-contextualisation. We report the F1 score of the positive class on both datasets, the recall of positive class on PAWS_QQP, and the recall of negative class on PAWS_Wiki.

dictions, it spuriously lowers the performance gap (on PAWS_Wiki) in terms of the F1 score of the positive class as shown in Table 4. After applying the alignment process to sentence encoders, we notice significant improvements in the recall of negative class. About 70% of sentence pairs in the PAWS_QQP have negative labels, which makes vanilla models difficult to distinguish paraphrases from non-paraphrases and mark most of sentence pairs as negative, as evidenced by the low recall for positives in the table. Similarly, we observe significant improvements in recall after introducing the alignment component. The large improvements in recall demonstrate the enhanced ability to distinguish non-paraphrases from paraphrases. Moreover, as shown in Table 4, the improvements in recall are not at the expense of their general performance, since we are still improving on F1 scores and the overall accuracy.

5.3 De-contextualisation

As pre-trained sentence encoders produce contextualised representations, two phrases of different meaning might be aligned for similar syntactic structure and contexts with a high similarity score. In the example shown in Section 1, *Harris announced* will be aligned with *James announced* with a high similarity score given their identical syntactic structure and contexts. However, does such high similarity score make sense when it comes to the task of paraphrase identification? Comparing the meaning of two phrases in the context of their use often helps disambiguate. In this case, the highly similar context instead downplays the difference, while it is the minor difference that changes the whole sentence meaning. This problem is exacerbated in the PAWS datasets given that both PAWS_QQP and PAWS_Wiki have extremely high lexical overlap, with 96.35% and 84.77% respectively, as shown in Table 3. Such high lexical overlap indicates a similar context, and thus a high similarity score between aligned phrases. In this experiment, we align phrases based on their contextualised representations as before but de-contextualise them by sending these phrases, without context, through sentence encoders to produce context-agnostic representations. A similarity score at sentence-level is then derived from cosine similarities between context-agnostic representations.

We show the results in Table 7. It clearly shows that, in spite of losing contextual information, the model with de-contextualisation process appears to improve the performance significantly. Additionally, it suggests that contextualisation might be harmful in situations where we focus on small differences that might change the meaning of the whole.

6 Conclusion

In this work, we propose an approach that combines sentence encoders with an alignment component by representing sentences as lists of predicate-argument spans and decomposing sentence-level meaning comparison into predicate-argument span comparison. Experiments with three widely used sentence encoders show that such method leads to improvements on various paraphrase identification tasks and increases the sensitivity to word order and structural differences between two sentences. The alignment matrix can further be utilised for interpretation. We then demonstrate that applying de-contextualisation to aligned phrases could help to recognise non-paraphrases of extremely high lexical overlap. Our future work includes exploring other alignment algorithms and more application scenarios for alignment-based sentence encoders.

Acknowledgement

We thank all anonymous reviewers for their insightful comments. We would like to further thank Bowen Wang and Wing Yan Li for helpful discussions and proofreading.

References

Yuki Arase and Jun'ichi Tsujii. 2019. Transfer fine-tuning: A BERT case study. 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 5393–5404, Hong Kong, China. Association for Computational Linguistics.

Yuki Arase and Junichi Tsujii. 2021. Transfer fine-tuning of bert with phrasal paraphrases. *Computer Speech Language*, 66:101164.

Yuki Arase and Jun'ichi Tsujii. 2020. Compositional phrase alignment and beyond. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1611–1623.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

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.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

A Chitra and Anupriya Rajkumar. 2016. Plagiarism detection using machine learning-based paraphrase recognizer. *Journal of Intelligent Systems*, 25(3):351–359.

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.

David F. Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew E 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.

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.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *arXiv preprint arXiv:1705.00652*.

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.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Shan Jiang, Simon Baumgartner, Abe Ittycheriah, and Cong Yu. 2020. *Factoring Fact-Checks: Structured Information Extraction from Fact-Checking Articles*, page 1592–1603. Association for Computing Machinery, New York, NY, USA.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 3294–3302.

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.

Harold W Kuhn. 1956. Variants of the hungarian method for assignment problems. *Naval research logistics quarterly*, 3(4):253–258.

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.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Chen Liang, Praveen K Paritosh, Vinodh Rajendran, and Kenneth D Forbus. 2016. Learning paraphrase identification with structural alignment. In *IJCAI*, pages 2859–2865.

Linqing Liu, Wei Yang, Jinfeng Rao, Raphael Tang, and Jimmy Lin. 2019. 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.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 802–811, Honolulu, Hawaii. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

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.

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.

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.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Dls @ cu: Sentence similarity from word alignment. In *SemEval@ COLING*, pages 241–246.

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, Online. Association for Computational Linguistics.

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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349.

Han Xiao. 2020. Hungarian layer: A novel interpretable neural layer for paraphrase identification. *Neural Networks*, 131:172–184.

Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911.

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.