Article

Sentence-CROBI: A Simple Cross-Bi-Encoder-Based Neural Network Architecture for Paraphrase Identification

Jesus-German Ortiz-Barajas 1, Gemma Bel-Enguix 2,* and Helena Gómez-Adorno 3

1 Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico
2 Instituto de Ingeniería, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico
3 Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City 04510, Mexico
* Correspondence: gbele@iingen.unam.mx

Abstract: Since the rise of Transformer networks and large language models, cross-encoders have become the dominant architecture for various Natural Language Processing tasks. When dealing with sentence pairs, they can exploit the relationships between those pairs. On the other hand, bi-encoders can obtain a vector given a single sentence and are used in tasks such as textual similarity or information retrieval due to their low computational cost; however, their performance is inferior to that of cross-encoders. In this paper, we present Sentence-CROBI, an architecture that combines cross-encoders and bi-encoders to obtain a global representation of sentence pairs. We evaluated the proposed architecture in the paraphrase identification task using the Microsoft Research Paraphrase Corpus, the Quora Question Pairs dataset, and the PAWS-Wiki dataset. Our model obtains competitive results compared with the state-of-the-art by using model ensembles and a simple model configuration. These results demonstrate that a simple architecture that combines sentence pair and single-sentence representations without using complex pre-training or fine-tuning algorithms is a viable alternative for sentence pair tasks.

Keywords: paraphrase identification; transformers; cross-encoders; bi-encoders

MSC: 68-04

1. Introduction

“Paraphrase” refers to sentences that have the same meaning as other sentences but use different words [1]. The problem of paraphrase identification is a binary classification task in which, given two texts $S_1$ and $S_2$, it must be determined whether they have the same meaning or not. Developing paraphrase identification systems is challenging because defining what constitutes a paraphrase is complex. Previous works define the paraphrase as an approximate equivalence between texts; in addition, there are different types of paraphrasing based on the level of changes that the texts could have [2]: the low paraphrase, which consists of substituting synonyms, hypernyms, hyponyms, meronyms and holonyms; and the high paraphrase, which consists of the realization of the phenomena of the low paraphrase, in addition to the morphological, lexical, semantic, syntactic and discursive phenomena. Because of the above, one option is to develop deep-learning-based approaches, which allow us to identify paraphrases of any type without extracting complex linguistic features to define text pairs.

The Transformer architecture [3] introduced a new era of Natural Language Processing (NLP) with the rise of pre-trained large language models. As a result of pre-training, they can learn universal representations of language that can be fine-tuned to specific tasks, without the need to train each model from scratch [4].

The cross-encoder model is one of the most popular approaches based on pre-trained language models. This model encodes the two texts together and applies full self-attention
to both texts at once [5]. Another pre-trained language model approach is the bi-encoder model. This approach applies self-attention separately for each text using a Siamese network and then compares them using a similarity metric [6].

Following the introduction of the BERT model [7], many approaches have emerged to increase its performance, from the modification of the pre-training stage [8] to modifications of the attention mechanisms [9], knowledge distillation [10], and other complex approaches. Our work proposes Sentence-CROBI, a simple architecture that combines the representations of cross-encoders and bi-encoders for sentence pair tasks. The results show competitive performance with state-of-the-art models when using model assembly and when using a simple approach, which offers a simple alternative for these types of tasks.

The structure of the paper is the following. In Section 2, we describe related work, where we consider previous BERT-based approaches applied to the paraphrase identification task. Section 3 describes the corpora that we used to train and evaluate the Sentence-CROBI architecture. In Section 4, we explain the proposed architecture and the experimental setup. Finally, in Sections 5 and 6, we present the results and conclusions, respectively.

2. Related Work

The Transformer network [3] is an architecture that can encode texts in parallel by using attention mechanisms instead of a sequential mechanism such as Recurrent Neural Networks. This feature enables researchers to train models with large amounts of text efficiently, marking the beginning of a new era in the artificial intelligence field where pre-trained large language models are used to solve several Natural Language Processing tasks [4].

The BERT model [7] is the most well-known language model based on the Transformer architecture using a cross-encoder approach, and it has obtained state-of-the-art results in a wide variety of tasks [11]. It consists of two versions: the base version and the large version, made up of 12 and 24 Transformer encoder blocks, respectively. The pre-training of the model consists of two tasks. The first task is the Masked Language Model, in which the [MASK] token replaces a portion of the input tokens, and the model learns to predict the actual values of those tokens. The second task is the Next Sentence Prediction, in which, from two texts $A$ and $B$, the model must identify whether $B$ is the text that comes after $A$ or not. After pre-training, the model can be fine-tuned for any NLP problem by appending an additional layer to the top of the model, using a small number of epochs and a low learning rate. After the emergence of the BERT model, the NLP community proposed different approaches to improve the performance of large language models based on the Transformer architecture using the two-stage scheme: pre-training and fine-tuning. There are four axes for these approaches.

The first axis consists of modifying the pre-training stage. The RoBERTa model [8] was proposed as an optimized configuration of BERT. The modifications consist of performing dynamic masking of the input tokens to the model in each epoch, eliminating the auxiliary loss function for the Masked Language Model task, using longer sequences and a more extensive dataset, and training for more epochs. Similarly, the StructBERT [12] model adds two tasks to this stage to learn the structure of the language both at the word level and the sentence level. The first task consists of changing the order of the masked tokens to predict the correct word order. The second task consists of changing the order of the statements in the Next Sentence Prediction task to predict the order of the statements. The last example in this axis is the Ernie 2.0 model [13]. The authors propose a continuous multi-task learning framework to learn lexical, syntactic, and semantic information in this work. This framework allows the use of the knowledge of previous tasks for new tasks during the pre-training phase. To check the effectiveness of the proposed model, they propose a set of seven pre-training tasks divided into three sets. The first set consists of word-level tasks. The first task is knowledge masking, in which the [MASK] token replaces some named entities and phrases of the text, and the model predicts its actual value. The second task is to predict whether a word begins with a capital letter, and, finally, the last
task of this set consists of predicting whether a token appears in other document segments or not. The second set consists of structure-level tasks. These tasks are sentence reordering and sentence distance prediction. Sentence reordering consists of finding the correct order of segment permutations of the original text. Sentence distance prediction is a multi-class classification problem. The model predicts whether two text segments are adjacent in a document, whether they are in the same document but not adjacent, or whether they do not belong to the same document. The last set consists of semantic-level tasks, where the model predicts the semantic relationship of two texts and the relevance of a text in an information retrieval system.

The second axis of modifications consists of reducing the size of the models. The ALBERT model [14] uses the factorized embedding parameterization technique. This technique splits the model vocabulary into two matrices: one for the embedding layer’s vocabulary and the other for the hidden layer’s vocabulary. ALBERT also implements parameter sharing between layers to prevent the model’s growth in depth. Another proposed approach for model reduction is the BORT model [10]. It is an optimal sub-architecture of BERT obtained using a fully polynomial time approximation scheme based on three evaluation metrics: inference time, model size, and error rate. However, since the resulting model is 95% smaller than the large BERT, it is more prone to overfitting. Therefore, the authors use the Agora algorithm [15], which combines data augmentation and knowledge distillation techniques, for the fine-tuning stage.

The third axis consists of modifying the fine-tuning stage of the model to achieve better performance in the target tasks. The SMART algorithm [16] was proposed as an alternative when target task data are limited. The method uses a smoothness-inducing adversarial regularization technique to control the capacity of the model and its high complexity by adding a small perturbation to the input data. In addition, to prevent the aggressive model’s parameter update, the authors present a class of Bregman proximal point optimization techniques. These methods use a confident-region-based regularization; therefore, the model updates its parameters only based on a small neighborhood of the previous iteration. The authors apply the proposed algorithm to fine-tune the RoBERTa [8] and MT-DNN [17] models to evaluate their performance in ensemble and single model approaches.

Finally, the fourth axis consists of modifications to the Transformer architecture. In the DeBERTa model [9], the authors propose a new attention mechanism that encodes the words in two vectors: the first vector encodes the word, and the other encodes its relative location. In contrast, the vanilla Transformer architecture encodes the words by summing the content vector and the position vector. The attention mechanism calculates the attention weights in separate arrays based on both representations by separating the words into content and relative position vectors. In addition, the authors incorporate the absolute position information for the Masked Language Model task; therefore, the model takes into account the content of the word, its relative position, and its absolute position to predict the actual value of the masked token. In the same field, the Funnel-Transformer model [18] was proposed to reduce the computational cost of pre-training a language model on a vast dataset. The authors add a pooling layer after some Transformer encoder blocks to achieve this goal, reducing the hidden representations’ size by half. In the case of token-level tasks such as the Masked Language Model task during the pre-training stage, the authors add a decoder to reconstruct the final vector to the original size. In the case of sentence-level tasks, the decoder is unnecessary, and the fine-tuning process only applies to the encoder.

Additionally, there is a different axis where researchers use pre-trained large language models to obtain sentence-level representations from texts and combine them with features that do not rely on neural network models. The Lexical, Syntactic, and Sentential Encodings (LSSE) learning model [19] is a unified framework that incorporates Relational Graph Convolutional Networks (R-GCNs) to obtain different features from local contexts through word encoding, position encoding, and full dependency structures, as well as sentence-level representations obtained using the BERT model. The authors use the [CLS] token as the sentence pair representation, while the graph network learns the syntactic context by capturing the dependency structure and word order. Each context vector is compared
using a distance metric and is concatenated to the sentence pair vector to obtain the global representation.

Unlike the works described above, there is another approach based on pre-trained language models called bi-encoders. For example, in sentence pair tasks, each text is encoded by a Siamese neural network [20] separately. The Sentence-BERT model [21] uses two instances of the BERT model with shared weights, where each text is encoded independently. At the output of each BERT instance, a pooling operation is applied to the last hidden state to obtain the vectors of each text; the global representation for the sentence pair consists of some combination of the individual vectors. Although this is a more efficient approach, its performance is lower than that of cross-encoder-based approaches [5,6].

In this work, we propose Sentence-CROBI, a simple architecture that combines cross-encoder and bi-encoder approaches for sentence pair tasks.

3. Corpora

This section describes the characteristics of the corpora that we used to evaluate our architecture. We selected these datasets based on the Papers with Code (https://paperswithcode.com/ accessed on 1 February 2022) platform. It is possible to search research papers based on the task that they solve, the datasets that they use, or the proposed approach. We selected the three datasets with the highest citations for the paraphrase identification task: the Microsoft Research Paraphrase Corpus (MRPC) [22], the Quora Question Pairs (QQP) corpus, and the PAWS corpus [23].

The Microsoft Research Paraphrase Corpus (MRPC) [22] consists of 5801 sentence pairs, collected over two years from various news websites and manually classified into two classes: Paraphrase and No Paraphrase. The corpus is partitioned into train and test subsets. The training set contains 4076 sentence pairs, where 2753 examples are paraphrases—that is, 67.5% of the pairs correspond to the Paraphrase class—and the remaining 1323 pairs of this set are non-paraphrase examples. On the other hand, the testing set consists of 1725 sentence pairs, where 66.5% are paraphrases—that is, 1147 sentence pairs. The remaining 578 pairs are non-paraphrase examples. Besides the paraphrase identification task, this corpus has been used in various tasks, such as sentence embedding computation using contrastive learning [24], zero-shot learning techniques [25], and the explainability of pre-trained language models [26].

The Quora Question Pairs (QQP) corpus consists of 795,241 question pairs labeled in a binary manner as Duplicated or Not Duplicated. It is divided into three subsets: the training set contains 363,846 question pairs, the validation set 40,430, and the testing set 390,965. The validation and training subsets have a distribution of 37% for duplicate questions and 63% for non-duplicate questions; the distribution of the test set is unknown because its labels are not publicly available. Therefore, the evaluation was performed using the GLUE Benchmark [27] server by uploading the output of our model on the test set using a specific format. To ensure the consistency of our results, we downloaded the corpus version provided by the GLUE Benchmark on their website (https://gluebenchmark.com/tasks accessed on 1 April 2022). This dataset has been used in tasks such as adversarial reprogramming [28] and model pre-training with limited resources [29].

The PAWS corpus [23]—specifically, the PAWS-Wiki subset—contains sentence pairs from Wikipedia (https://dumps.wikimedia.org accessed on 5 February 2022). It consists of 65,401 sentence pairs divided into three subsets: the training set with 49,401 instances and validation and testing sets with 8000 instances each. The distribution of the corpus includes 44% of examples labeled as Paraphrase and 56% labeled as No Paraphrase. This corpus contains examples with high lexical overlap, even for non-paraphrase sentence pairs. This characteristic makes it a challenging corpus when evaluating paraphrase detection models. Although it has been recently created, this dataset has been used in tasks such as in-context learning [30], condescending language detection [31], and intent detection [32].

Table 1 displays the statistics of the datasets described above.
Table 1. Statistics for the MRPC, QQP, and PAWS-Wiki datasets.

| Corpus            | Paraphrase Instances | Non-Paraphrase Instances | Total Instances |
|-------------------|----------------------|--------------------------|-----------------|
| MRPC (train)      | 2753                 | 1323                     | 4076            |
| MRPC (test)       | 1147                 | 578                      | 1725            |
| QQP (train)       | 134,623              | 229,223                  | 363,846         |
| QQP (val)         | 14,959               | 25,471                   | 40,430          |
| QQP (test)        | -                    | -                        | 390,965         |
| PAWS-Wiki (train) | 21,829               | 27,572                   | 49,401          |
| PAWS-Wiki (val)   | 3539                 | 4461                     | 8000            |
| PAWS-Wiki (test)  | 3536                 | 4464                     | 8000            |

Additionally, we used the Multi-Genre NLI corpus [33], which consists of labeled sentence pairs with textual entailment information in three classes: Neutral, Contradiction, and Entailment. It is composed of two subsets of training and testing. The training set contains 391,164 examples, with 130,375 examples for the Neutral class, 130,379 for the Contradiction class, and 130,411 for the Entailment class; the testing set is composed of 9714 pairs of statements, with 3094 examples of the class Neutral, 3180 for Contradiction, and 3440 labeled as Entailment. Following a two-stage fine-tuning approach, [34], we used this dataset to perform an intermediate fine-tuning stage for the proposed architecture before tuning the model in the target task.

4. Methodology

This section describes in detail the proposed architecture, the preprocessing steps that we performed to train and evaluate the model, and, finally, the experimental configuration.

4.1. Text Preprocessing

The preprocessing performed on the sentence pairs is detailed below. We converted each text to a sequence of IDs based on the BERT model [7] vocabulary. Similarly, we converted the sentence pairs to a sequence of IDs based on the ROBERTa model vocabulary [8]. Then, after encoding each text and the sentence pair, we added the classification [CLS] token and the separation [SEP] token. Following the preprocessing process, we added padding for individual texts and sentence pairs to normalize inputs to a single size. Finally, we obtained the attention mask for each text and sentence pair. This mask allows the model to distinguish between word tokens and padding tokens.

4.2. Model

In this section, we present the Sentence-CROBI architecture and its implementation. The bi-encoder component of our approach is based on the Sentence-BERT model [21]; we use a modification of the BERT model through a Siamese neural network [20] that is capable of obtaining individual vectors of fixed size from each text. We apply a pooling operation to the last hidden state of the BERT model to obtain a sentence vector for each text. We represent these sentence vectors as $u$ and $v$, respectively. We use an instance of the RoBERTa model for the cross-encoder component. This model receives the joint encoding of the sentence pair. To obtain the final representation of the sequence, we use the classification token [CLS].

After obtaining the individual representation of each text and its joint representation, we compute the Euclidean distance $D$ between the vectors $u$ and $v$. Finally, we obtain the global vector representation of the sentence pair by concatenating the classification token [CLS] from the cross-encoder representation, the vectors $u$ and $v$, and the Euclidean distance $D$. This vector is the input to a classifier composed of two fully connected networks.

We use the BERT base version composed of 12 Transformer blocks for the bi-encoder component of our architecture. Meanwhile, we use the RoBERTa large version composed of 24 Transformer blocks for the cross-encoder component.
Figure 1 shows the structure of the Sentence-CROBI architecture.

![Figure 1](image)

Figure 1. Diagram of the Sentence-CROBI model. CLS corresponds to the classification token of the cross-encoder component. $U$ and $V$ correspond to the individual vector representations of each text, denoted by Sentence 1 and Sentence 2, respectively. $D$ is the Euclidean distance between vectors $U$ and $V$.

The Siamese component of the Sentence-CROBI architecture produces contextual word vectors. We obtain sentence vectors by applying a mean pooling operation to the contextual word embedding matrix, where each row represents a word in the input text. The proposed architecture takes the last hidden state of BERT as contextual word embeddings.

The final component of our proposed model is the classifier. It is a fully connected network with two layers. First, it receives the global sentence pair representation as input, and a dropout layer is applied with a probability of 0.1. Dropout is a regularization technique to avoid overfitting of the network; it consists of randomly setting some values of its input to zero. Then, it passes through a fully connected layer of 1793 units with a hyperbolic tangent as the activation function. Finally, the output layer consists of 2 neurons with a linear function as an activation function.

We use the cross-entropy as a loss function during the training of the Sentence-CROBI architecture. The function’s objective is to compare the probability of the predicted class to that of the actual class of the training instance. The model’s prediction is then penalized based on the distance from the actual value. Equation (1) defines the cross-entropy function, where

- $y^i$ is the actual label;
- $\hat{y}^i$ denotes the probability predicted by the model;
- $N$ is the size of the test set.

$$CE = \sum_{i=1}^{N} y^i \log (\hat{y}^i) + (1 - y^i) \log (1 - \hat{y}^i)$$  

(1)

4.3. Fine-Tuning

To fine-tune the model, we use two approaches. The first is the original approach proposed for the BERT model: it consists of initializing the model’s parameters based on
the pre-training stage and training the model for a few epochs on the target task using a small learning rate. However, one of the issues with this approach is that when the target task dataset is small, the model is prone to overfitting [35]. Because the Microsoft Research Paraphrase Corpus has only 4076 training examples, we apply a second approach by using an intermediate-related target task to fine-tune the model. The intermediate target task has more labeled data [34] and allows the model to increase its robustness and effectiveness. In this work, we use the Multi-Genre NLI described in Section 3 for intermediate training of the Sentence-CROBI architecture before fine-tuning on the Microsoft Paraphrase corpus.

4.4. Ensemble Learning

To improve the classifier’s performance in the paraphrase identification task, we use the Bagging technique [36], which reduces the generalization error by combining several models. This technique consists of training different models separately and combining each output set to vote on test data and obtain the final prediction.

In the case of neural networks, differences in random initialization or in batch generation cause independent errors in each member of the ensemble; therefore, the ensemble will perform significantly better than its members [37].

In this work, we perform the ensemble learning technique by fine-tuning several instances of the Sentence-CROBI architecture using different random seeds to initialize each model. After the fine-tuning stage, we compute the output probabilities of each test example for each independent instance of the Sentence-CROBI model. We obtain $k$ output matrices, where $k$ is the number of independent instances of the model, and the dimension of each matrix is $N \times 2$, where $N$ is the number of examples on the test set, and 2 corresponds to the number of classes. We compute the probability average of the $k$ predictions, and the classification is based on the class with the highest probability.

4.5. Training Details

Following the fine-tuning procedure in the ROBERTa model [8], we train our models with a batch size in the range of {16,32}. We use a learning rate in the range of {$1 \times 10^{-5}$, $2 \times 10^{-5}$, $3 \times 10^{-5}$} with the Adam optimizer, with a warm-up ratio of 0.06 and a linear decay to zero. We train all models for a maximum of 10 epochs and perform pseudo early stopping to use the model with the best performance on the validation data. The maximum length is 35 for individual texts and 128 for text pairs. We use HuggingFace’s Transformers library to implement the Sentence-CROBI model [38]. Our code implementation is publicly available on Github (https://github.com/jgermanob/Sentence-CROBI created on 14 September 2022)

5. Results

We present the Sentence-CROBI model’s results for the corpora described in Section 3 and their comparison with the state-of-the-art models described in Section 2. The evaluation metrics used are Accuracy and F1-score in the Paraphrase class.

Tables 2 and 3 report the results obtained from each paper for the BORT, StructBERT, Funnel-Transformer, ALBERT, and Ernie 2.0 models. In the case of the SMART algorithm, we use the results reported by the authors when fine-tuning the RoBERTa and MT-DNN models using their approach.

Tables 4 and 5 report the results that we obtained using the public implementation for each model in the state-of-the-art. We report the average of five runs using different random seeds.

Table 2 shows the state-of-the-art results obtained from the GLUE Benchmark leaderboard on the Microsoft Research Paraphrase Corpus and the results with the Sentence-CROBI architecture. We order the different approaches based on the F1-score metric in descending order. The state-of-the-art results correspond to some ensemble learning approaches; nevertheless, the authors do not provide details on their ensemble learning process.

For the case of the Sentence-CROBI architecture, we use 15 different models for the Bagging algorithm as an ensemble learning technique. All models correspond to an
independent run using different random seeds. Five models correspond to fine-tuning the model on the MRPC corpus after performing intermediate fine-tuning of the model on the MNLI corpus—that is, we initialize the model’s weights based on the pre-training stage, fine-tune the model on the intermediate task, and finally fine-tune the model on the target task. Another five models are analogous but use the PAWS-Wiki dataset as the intermediate task. The remaining five models correspond to the MRPC corpus’s fine-tuning without any intermediate fine-tuning. After completing all runs, we average the output probabilities to obtain the final prediction.

Our model obtains competitive results in comparison to the state-of-the-art. There is only a difference of 1.23 in Accuracy and 0.75 in F1-score with the BORT model [10].

Table 2. Results on the Microsoft Research Paraphrase Corpus obtained from the GLUE Benchmark leaderboard.

| Model                   | Accuracy | F1-Score | Difference Compared with Sentence-CROBI (Accuracy/F1-Score) |
|-------------------------|----------|----------|-------------------------------------------------------------|
| BORT [10]               | 92.30    | 94.10    | 1.23/0.75                                                  |
| MT-DNN SMART [16]       | 91.60    | 93.70    | 0.53/0.35                                                  |
| RoBERTa SMART [16]      | 91.60    | 93.70    | 0.53/0.35                                                  |
| StructBERTRoBERTa [12]  | 91.50    | 93.60    | 0.43/0.25                                                  |
| Funnel-Transformer [18] | 91.20    | 93.40    | 0.13/0.05                                                  |
| ALBERT [14]             | 91.20    | 93.40    | 0.13/0.05                                                  |
| Sentence-CROBI          | 91.07    | 93.35    | -                                                           |
| Ernie 2.0 [13]          | 87.40    | 90.20    | −3.67/−3.15                                                |

Table 3 shows the state-of-the-art and the Sentence-CROBI results in the Quora Question Pairs dataset. Our proposed model obtains competitive results. However, there is a more significant gap compared to the best approach, with a difference of 0.6 in Accuracy and 1.6 in F1-score. The main difference with this corpus is the evaluation process, because all the state-of-the-art approaches follow a single-task fine-tuning approach. We use the Bagging algorithm as an ensemble learning technique and five runs with different random seeds to obtain the final prediction. In addition, the dataset is challenging because of the difference between the distributions in the subsets.

Table 3. Results on the Quora Question Pairs dataset obtained from the GLUE Benchmark leaderboard.

| Model                   | Accuracy | F1-Score | Difference Compared with Sentence-CROBI (Accuracy/F1-Score) |
|-------------------------|----------|----------|-------------------------------------------------------------|
| Funnel-Transformer [18] | 90.70    | 75.40    | 0.6/1.6                                                     |
| StructBERTRoBERTa [12]  | 90.70    | 74.40    | 0.6/0.6                                                     |
| ALBERT [14]             | 90.50    | 74.20    | 0.4/0.4                                                     |
| RoBERTa SMART [16]      | 90.01    | 74.00    | −0.09/0.2                                                   |
| MT-DNN SMART [16]       | 90.20    | 73.90    | 0.1/0.1                                                     |
| Ernie 2.0 [13]          | 90.10    | 73.80    | 0.0/0.0                                                     |
| Sentence-CROBI          | 90.10    | 73.80    | -                                                           |
| BORT [10]               | 85.90    | 66.00    | −4.2/−7.8                                                   |

Table 4 shows the results for the PAWS-Wiki corpus. The authors do not originally use this corpus in their work; for this reason, we use the public implementation of each of the state-of-the-art models. In this configuration, we do not use any intermediate fine-tuning task, and we report the mean over five runs with different random seeds. Our proposed model obtains the second-best performance using this dataset, with a small difference of 0.13 in both Accuracy and F1-score.
Figure 2 displays a bar chart showing each model’s best performance on the Microsoft Research Paraphrase Corpus and Quora Question Pair dataset. We obtain these performance metrics from the GLUE Benchmark leaderboard for the state-of-the-art models. The Sentence-CROBI model corresponds to the ensemble learning technique described above in the case of MRPC and a single fine-tuning approach for the QQP dataset. All models achieve a higher F1-score than 90 in MRPC. However, in the QQP dataset, only the BORT model obtained an F1-score lower than 70. The difference in the BORT model’s performance on both datasets suggests instability in its fine-tuning algorithm because of the model’s size.

Table 4. Results on the PAWS-Wiki dataset.

| Model                             | Accuracy | F1-Score | Difference Compared with Sentence-CROBI (Accuracy/F1-Score) |
|-----------------------------------|----------|----------|------------------------------------------------------------|
| RoBERTa SMART [16]               | 94.93    | 94.34    | 0.13/0.13                                                  |
| Sentence-CROBI                    | 94.80    | 94.21    |                                                            |
| DeBERTa [9]                       | 94.69    | 94.12    | −0.11/−0.09                                               |
| ALBERT [14]                       | 94.70    | 94.08    | −0.1/−0.13                                               |
| MT-DNN SMART [16]                 | 94.16    | 93.52    | −0.64/−0.69                                               |
| Ernie 2.0 [13]                    | 93.86    | 93.18    | −0.94/−1.03                                               |
| StructBERT [12]                  | 93.13    | 92.41    | −1.67/−1.8                                                |

Finally, Table 5 shows the results obtained in the Microsoft Paraphrase corpus with a simple model configuration—that is, without intermediate fine-tuning tasks or ensemble learning strategies. We report the mean over five runs with different random seeds. Under these conditions, the Sentence-CROBI architecture obtains the third-best performance compared to state-of-the-art models. The difference regarding the best performance, obtained by the DeBERTa model [9], is 0.21 in Accuracy and 0.08 in F1-score.
Table 5. Results on the Microsoft Research Paraphrase Corpus following a single-model approach.

| Model               | Accuracy | F1-Score | Difference Compared with Sentence-CROBI (Accuracy/F1-Score) |
|---------------------|----------|----------|------------------------------------------------------------|
| DeBERTa [9]         | 89.30    | 91.96    | 0.21/0.08                                                  |
| Ernie 2.0 [13]      | 89.11    | 91.89    | 0.02/0.01                                                  |
| **Sentence-CROBI**  | **89.09**| **91.88**| -                                                          |
| RoBERTa SMART [16]  | 88.83    | 91.75    | −0.26/−0.13                                                |
| MT-DNN SMART [16]   | 87.71    | 90.84    | −1.38/−1.04                                                |
| ALBERT [14]         | 87.58    | 90.83    | −1.51/−1.05                                                |
| StructBERT [12]     | 86.56    | 90.06    | −2.53/−1.82                                                |

Figure 3 displays a bar chart showing each model’s average performance over five runs using different random seeds on the Microsoft Research Paraphrase Corpus and PAWS-Wiki corpus. The configuration for all the models is a single fine-tuning approach, without any intermediate task or ensemble learning technique. BORT and Funnel-Transformer do not appear in this chart because there is no public implementation. The Sentence-CROBI architecture is 0.56 above the average in F1-score for MRPC, which is 91.31. In the same corpus, 4 of 7 models, our approach included, have performance higher than 91. Meanwhile, the average F1-score for the PAWS-Wiki dataset is 93.69, and our proposed model achieves a value 0.51 above this. Similar to MRPC, our model is one of the four models with an F1-score higher than 94.

Figure 3. Average performance metrics over five runs with different random seeds of the proposed architecture and the state-of-the-art on the Microsoft Research Paraphrase Corpus and the PAWS-Wiki corpus, using a single-model configuration without intermediate fine-tuning and ensemble learning techniques.

5.1. Statistical Significance Tests

We perform a statistical significance test to compare the performance of the Sentence-CROBI architecture with the state-of-the-art. We select the non-parametric Wilcoxon signed test [39] because the distribution of our data is unknown [40]. To compute the significance tests, we use the Python library SciPy [41]. The null hypothesis is that the differences follow a symmetric distribution around zero. First, the absolute values of the differences are ranked. Then, each rank is given a sign according to the sign of the difference. The
threshold that we use to accept or reject the null hypothesis is \( \alpha = 0.05 \). We use the MRPC and PAWS-Wiki corpora to perform this test, without intermediate fine-tuning or ensemble learning. Table 6 shows the results of the Wilcoxon signed test between the proposed architecture and the state-of-the-art methods. It is possible to observe that none of the comparisons is statistically significant, since the \( p \)-values of all the comparisons are not less than the threshold \( \alpha \).

### Table 6. Significance tests using the Wilcoxon signed test between the proposed architecture and the state-of-the-art models. We compare the \( p \)-values with a threshold \( \alpha = 0.05 \) to accept or reject the null hypothesis.

| Model 1                          | Model 2                          | MRPC \( p \)-Value | PAWS-Wiki \( p \)-Value |
|----------------------------------|----------------------------------|---------------------|-------------------------|
| Sentence-CROBI                   | ALBERT [14]                      | 0.0625              | 0.3125                  |
| ALBERT [14]                      | Ernie 2.0 [13]                   | 0.8125              | 0.0625                  |
| Ernie 2.0 [13]                   | StructBERT [12]                  | 0.0625              | 0.0625                  |
| StructBERT [12]                  | RoBERTa SMART [16]               | 0.3125              | 0.3125                  |
| RoBERTa SMART [16]               | MT-DNN SMART [16]                | 0.0625              | 0.0625                  |

Additionally, we performed a statistical significance test using the Wilcoxon signed test with the methods described in the state-of-the-art. As in the tests with the Sentence-CROBI architecture, we used a threshold of \( \alpha = 0.05 \) to accept or reject the null hypothesis. The datasets used are MRPC and PAWS-Wiki. Following the same approach as the significance tests with our model, we do not perform any intermediate fine-tuning stage or ensemble learning strategy. Table 7 shows the results of the tests. In the same way, it is possible to observe that, for the two datasets used, there is no significant difference between the results.

### Table 7. Significance tests using the Wilcoxon signed test between the state-of-the-art models. We compare the \( p \)-values with a threshold \( \alpha = 0.05 \) to accept or reject the null hypothesis.

| Model 1                          | Model 2                          | MRPC \( p \)-Value | PAWS-Wiki \( p \)-Value |
|----------------------------------|----------------------------------|---------------------|-------------------------|
| ALBERT [14]                      | DeBERTa [9]                      | 0.0625              | 1.0                     |
| DeBERTa [9]                      | Ernie 2.0 [13]                   | 0.0625              | 0.0625                  |
| Ernie 2.0 [13]                   | StructBERT [12]                  | 0.1875              | 0.0625                  |
| StructBERT [12]                  | RoBERTa SMART [16]               | 0.0625              | 0.0625                  |
| RoBERTa SMART [16]               | MT-DNN SMART [16]                | 1.0                 | 0.0625                  |
| DeBERTa [9]                      | Ernie 2.0 [13]                   | 0.8125              | 0.0625                  |
| Ernie 2.0 [13]                   | StructBERT [12]                  | 0.0625              | 0.0625                  |
| StructBERT [12]                  | RoBERTa SMART [16]               | 0.4375              | 0.125                   |
| RoBERTa SMART [16]               | MT-DNN SMART [16]                | 0.0625              | 0.0625                  |
| MT-DNN SMART [16]                |                                  |                     |                         |

Since there is no statistical significance between our proposed approach and the state-of-the-art models, the Sentence-CROBI architecture has an advantage due to two factors. The first one is its implementation facility that relies only on using two pre-trained models, one with a cross-encoder approach and the other with a bi-encoder approach, and combining both representations to obtain a global vector; there are no modifications to the pre-trained models’ architecture or during the pre-training stage. The second one is the fine-tuning procedure: our model takes the most straightforward scheme, with only a few epochs and a low learning rate, to adjust the model to the target task, using a standard loss function as the cross-entropy for classification tasks.
5.2. Error Analysis

We perform a quantitative error analysis of our architecture’s performance on the Microsoft Research Paraphrase Corpus, which we report in Table 2; in this setting, we perform ensemble learning by using the Bagging technique and 15 instances of our model with different random seeds. Five correspond to an intermediate fine-tuning stage using the MNLI corpus; five correspond to an intermediate fine-tuning stage using the PAWS-Wiki corpus, and the remaining instances correspond to fine-tuning the model on MRPC without using intermediate tasks. Figure 4 shows the confusion matrix obtained by our model using the configuration described above. The Sentence-CROBI model correctly predicts 1081 of 1147 paraphrase instances, corresponding to 94.24% of the examples of this class. On the other hand, it correctly predicts 490 of 578 non-paraphrase samples, corresponding to 84.77% instances of this class.

![Sentence-CROBI's confusion matrix on the Microsoft Research Paraphrase Corpus using ensemble learning.](image)

We also perform a qualitative error analysis based on the first five false positive and false negative examples predicted by the Sentence-CROBI model.

Table 8 shows the false positive examples. In general, it is possible to notice that all examples share the subject. For instance, in the first pair is “Ballmer”. In the second pair, the first sentence refers to a female subject, while the second refers to a person who plays a schoolgirl character, and both subjects go to see a specialist because they are sick. The difference between the sentences in the third to fifth pairs is the specificity in describing the performed actions, but the subjects are the same.

Table 8. False positive examples predicted by the Sentence-CROBI model. False positives correspond to non-paraphrase instances classified by the model as paraphrases.

| Text 1 | Text 2 |
|--------|--------|
| Ballmer has been vocal in the past warning that Linux is a threat to Microsoft. “She first went to a specialist for initial tests last Monday, feeling tired and unwell”. Garner said the self-proclaimed mayor of Baghdad, Mohammed Mohsen al-Zubaidi, was released after two days in coalition custody”. “It appears from our initial report that this was a textbook landing considering the circumstances”, “Burke said”. Powell recently changed the story, telling officers that Hoffa’s body was buried at his former home, where the search was conducted Wednesday”. | “In the memo, Ballmer reiterated the open-source threat to Microsoft”. “The star, who plays schoolgirl Nina Tucker in Neighbours, went to a specialist on 30 June feeling tired and unwell”. Garner said self-proclaimed Baghdad mayor Mohammed Mohsen Zubaidi was released 48 h after his detention in late April. “Said Mr. Burke: It was a textbook landing considering the circumstances”. Powell changed the story earlier this year, telling officers that Hoffa’s body was buried at his former home, where the aboveground pool now sits”. |
Table 9 shows the false negative examples predicted by our model. Our approach struggles with sentences with a high word overlapping rate between them. For instance, in the first pair, the first sentence talks about the possibility of a man becoming sick, while the second talks about the fact that there is only a sick man. The third pair is different because of the number of bodies that they refer to. Finally, in the fourth and fifth examples, the model cannot identify correctly that the subjects are different.

| Text 1                                                                 | Text 2                                                                 |
|-----------------------------------------------------------------------|------------------------------------------------------------------------|
| “A Washington County man may have the county’s first human case of West Nile virus, the health department said Friday”. | The county’s first and only human case of West Nile this year was confirmed by health officials on 8 September. |
| “Snow’s remark “has a psychological impact”, said Hans Redeker, head of foreign-exchange strategy at BNP Paribas”. | “Snow’s remark on the dollar’s effects on exports “has a psychological impact”, said Hans Redeker, head of foreign-exchange strategy at BNP Paribas”. |
| “Another body was pulled from the water on Thursday and two seen floating down the river could not be retrieved due to the strong currents, local reporters said”. | “Two more bodies were seen floating down the river on Thursday, but could not be retrieved due to the strong currents, local reporters said”. |
| “Amgen shares gained 93 cents, or 1.45 percent, to $65.05 in afternoon trading on Nasdaq”. | Shares of Allergan were up 14 cents at $78.40 in late trading on the New York Stock Exchange. |
| “In his speech, Cheney praised Barbour’s accomplishments as chairman of the Republican National Committee”. | Cheney returned Barbour’s favorable introduction by touting Barbour’s work as chair of the Republican National Committee. |

6. Conclusions

We present the Sentence-CROBI model, a simple language-model-based architecture that combines cross-encoders and bi-encoders to compute a vector representation in sentence pair tasks. Our model works by combining the output representations of cross-encoders and bi-encoders. Therefore, it does not rely on complex architecture modifications, adding more tasks to the pre-training stage, reducing the model’s size, or modifying the fine-tuning algorithm.

Our proposed architecture achieved competitive results with the state-of-the-art models in all the evaluated datasets. The most significant difference is when we evaluate the Quora Question Pairs dataset. The Funnel-Transformer model outperforms our model by 1.6 regarding the F1-score. On the other hand, the least significant difference is concerning the PAWS-Wiki dataset, where the RoBERTa model fine-tuned using the SMART algorithm outperformed our model by 0.13 in terms of the F1-score.

The proposed model performs best when no intermediate fine-tuning tasks or ensemble learning techniques are used. These results suggest that combining cross-encoders and bi-encoders could improve the model’s performance in sentence pair tasks without any auxiliary technique. Moreover, there is no statistical significance between our proposed approach and the state-of-the-art models. These results represent our model’s advantage, because its success does not rely on adding more pre-training tasks, modifying the Transformer architecture, or creating new fine-tuning algorithms. In the same way, it is easy to implement using existing tools, and it is possible to adapt the model to different tasks with minor changes. The changes only consist of replacing the combination strategy of the cross-encoder and bi-encoder representations, the last layer on the model, and the loss function. This configuration follows the current paradigm of the Natural Language Processing field, where pre-trained models are adapted to a wide variety of tasks without designing each model from scratch.

This paper is the first approach that combines bi-encoder and cross-encoder representations for sentence pair tasks. Therefore, future work includes exploring different combinations of these two models and measuring their impact on the current state-of-the-art datasets and new scenarios.
Author Contributions: Conceptualization, J.-G.O.-B.; methodology, J.-G.O.-B., G.B.-E. and H.G.-A.; software, J.-G.O.-B.; validation, G.B.-E. and H.G.-A.; formal analysis, J.-G.O.-B.; investigation, J.-G.O.-B.; resources, J.-G.O.-B.; data curation, J.-G.O.-B., G.B.-E., and H.G.-A.; writing—original draft preparation, J.-G.O.-B.; writing—review and editing, G.B.-E. and H.G.-A.; visualization, J.-G.O.-B.; supervision, G.B.-E. and H.G.-A.; project administration, G.B.-E. and H.G.-A.; funding acquisition, G.B.-E. and H.G.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by PAPIIT projects TA400121 and TA101722, CONACYT CB A1-S-27780, and CONACYT PNPC scholarship with No. CVU 1086461.

Data Availability Statement: Publicly available datasets were used in this study: The Microsoft Research Paraphrase Corpus (https://www.microsoft.com/en-us/download/details.aspx?id=52398 accessed on 1 March 2022), the Quora Question Pairs corpus (https://gluebenchmark.com/tasks accessed on 1 March 2022), the PAWS-wiki corpus (https://github.com/google-research-datasets/paws accessed on 1 March 2022), and the Multi-Genre NLI corpus (https://cims.nyu.edu/~sbowman/multinli/ accessed on 1 March 2022).

Acknowledgments: The authors thank CONACYT for the computing resources provided through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercomputo del INAOE.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Bhagat, R.; Hovy, E. What is a Paraphrase? *Comput. Linguist.* 2013, 39, 463–472.
2. Montoya, M.M.; da Cunha, I.; López-Escobedo, F. Un corpus de paráfrasis en español: Metodología, elaboración y análisis. *Rev. Lingüística Teor. Appl.* 2016, 54, 85–112.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. *Adv. Neural Inf. Process. Syst.* 2017, 30, 6000–6010.
4. Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; Huang, X. Pre-trained Models for Natural Language Processing: A Survey. *Sci. China Technol. Sci.* 2020, 63, 1872–1897.
5. Humeau, S.; Shuster, K.; Lachaux, M.A.; Weston, J. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. *arXiv* 2019, arXiv:1905.01969.
6. Peng, Q.; Weir, D.; Weeds, J.; Chai, Y. Predicate-Argument Based Bi-Encoder for Paraphrase Identification. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Dublin, Ireland, 2022; pp. 5579–5589. https://doi.org/10.18653/v1/2022.acl-long.362.
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.
8. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* 2019, arXiv:1907.11692.
9. He, P.; Liu, X.; Gao, J.; Chen, W. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv* 2020, arXiv:2006.03654.
10. de Wynter, A.; Perry, D.J. Optimal Subarchitecture Extraction For BERT. *arXiv* 2020, arXiv:2010.10499.
11. Rogers, A.; Kovalova, O.; Rumshisky, A. A Primer in BERTology: What We Know about How BERT Works. *Trans. Assoc. Comput. Linguist.* 2020, 8, 842–866. https://doi.org/10.1162/tacl_a_00349.
12. Wang, W.; Bi, B.; Yan, M.; Wu, C.; Xia, J.; Bao, Z.; Peng, L.; Si, L. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. *arXiv* 2020, arXiv:1908.04577.
13. Sun, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. Ernie 2.0: A Continual Pre-training Framework for Language Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8968–8975.
14. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* 2020, arXiv:1909.11942.
15. de Wynter, A. An Algorithm for Learning Smaller Representations of Models with Scarc Data. *arXiv* 2020, arXiv:2010.07990.
16. Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Zhao, T. SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2177–2190. https://doi.org/10.18653/v1/2020.acl-main.197.
37. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 15 June 2022).

38. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

39. Wilcoxon, F. Individual comparisons of grouped data by ranking methods. J. Econ. Entomol. 1946, 39, 269–270.

40. Dror, R.; Baumer, G.; Shlomov, S.; Reichart, R. The Hitchhiker’s Guide to Testing Statistical Significance in Natural Language Processing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 1383–1392. https://doi.org/10.18653/v1/P18-1128.

41. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat. Methods 2020, 17, 261–272. https://doi.org/10.1038/s41592-019-0686-2.