Contextualized Embeddings based Convolutional Neural Networks for Duplicate Question Identification

Harsh Sakhrani[1*], **Saloni Parekh**[1*] and **Pratik Ratadiya**[2]
[1]Pune Institute of Computer Technology, Maharashtra, India
[2]vCreaTek Consulting Services Pvt. Ltd., Maharashtra, India
{harshsakhrani26, saloniparekh1609}@gmail.com, pratik.r@vcreatek.com

Abstract

Question Paraphrase Identification (QPI) is a critical task for large-scale Question-Answering forums. The purpose of QPI is to determine whether a given pair of questions are semantically identical or not. Previous approaches for this task have yielded promising results, but have often relied on complex recurrence mechanisms that are expensive and time-consuming in nature. In this paper, we propose a novel architecture combining a Bidirectional Transformer Encoder with Convolutional Neural Networks for the QPI task. We produce the predictions from the proposed architecture using two different inference setups: Siamese and Matched Aggregation. Experimental results demonstrate that our model achieves state-of-the-art performance on the Quora Question Pairs dataset. We empirically prove that the addition of convolution layers to the model architecture improves the results in both inference setups. We also investigate the impact of partial and complete fine-tuning and analyze the trade-off between computational power and accuracy in the process. Based on the obtained results, we conclude that the Matched-Aggregation setup consistently outperforms the Siamese setup. Our work provides insights into what architecture combinations and setups are likely to produce better results for the QPI task.

1 Introduction

Paraphrase Identification is a task that aims to recognize whether two input sequences are semantically equivalent or not, and is a binary classification problem. Question Paraphrase Identification is a variation of Paraphrase Identification where the two sequences are interrogative in nature. It is a critical task for Question-Answering forums like Quora [1], which can benefit from merging questions with similar meanings and organizing them more efficiently. Moreover, it can also help in the retrieval of questions that are semantically equivalent to a question asked by the user.

Deep Learning-based mechanisms for Paraphrase Identification can be classified into two major frameworks. The first framework is the Siamese Architecture that has a common encoder that is applied separately to both the input sentences to generate sentence encodings in the same embedding space [Yu *et al.*, 2014; Bowman *et al.*, 2016]. The framework has the advantage of being lightweight and simple to train because of shared network parameters. However, the disadvantage is that there is no explicit interaction between the two sentences in the encoding process. Some methods suggest the use of an attention mechanism to improve this interaction [Tan *et al.*, 2016]. Such a method uses one sequence's representation to attend to another but fails to capture the word-level interactions.

To overcome this disadvantage, a second framework called Matched Aggregation has been proposed [Wang and Jiang, 2016; Wang *et al.*, 2017]. In this framework, the attention mechanism is utilized at the word level to get the matching words between the two sentences. The matching information is then aggregated into a vector to make the final classification decision. This framework allows for word-level interaction, ensuring significantly improved results. However, most of the previous Matched Aggregation approaches relied on a combination of recurrence and complex attention ensembles to match the two sentences at the word level [Lan and Xu, 2018].

To this end, we propose a contextual embedding-based convolution architecture for question paraphrase or duplicate question identification. We propose the use of the Bidirectional Transformer Encoder, based on [Devlin *et al.*, 2019], that is capable of generating contextual word embeddings utilizing the self-attention mechanism. In addition to this, we also make use of Convolutional Neural Networks [Kim, 2014] to extract the semantic level features from the trained contextual embeddings. We perform extensive experiments utilizing the proposed architecture in both setups- Siamese and Matched Aggregation. The system is evaluated on the Quora Question Pairs Dataset [2] and produces state-of-the-art results.

Specifically, we make the following contributions through this paper:

- We present a new approach for question paraphrase

[*]Equal Contributors
[1]https://www.quora.com/
[2]https://www.kaggle.com/c/quora-question-pairs

identification in two different inference setups - Siamese and Matched Aggregation, leveraging the Bidirectional Transformer Encoder's pretraining on the Next Sentence Prediction task.

- We achieve state-of-the-art results on the Quora Question pairs dataset on which we prove that the addition of convolutional layers helps to improve the performance as against using a vanilla transformer architecture.
- Furthermore, we also experiment with partial and complete fine-tuning and empirically show that one doesn't need to increase the trainable parameters beyond a certain extent to achieve improved results, thereby also assessing the trade-off between computational power and accuracy.

The rest of the paper is structured as follows: Section 2 talks about the previous approaches which have been used to tackle the problem, Section 3 explains the proposed methodology in detail, Section 4 talks about the description of the dataset, Section 5 presents the results and analysis of the proposed system against previous approaches while Section 6 concludes our paper.

2 Background Work

The paraphrase identification task, a well-studied Natural Language Processing (NLP) problem, uses Natural Language Sentence Matching (NLSM) to decide whether two sentences are paraphrased or not. Identifying duplicate questions can be thought of as a special use case of this task.

Over time, researchers have provided multiple approaches to solve this problem by trying to break down the complex internal structures of sentences and the interactions between them. The two of the most used approaches are mentioned in this section. The first method employs a "Siamese" setup, which is made up of two sub-networks with common parameters. Each sub-network responds to a single sentence and the outputs corresponding to them are mathematically operated on to obtain the final result.

[Chen *et al.*, 2018] used the Manhattan LSTM model and merged the outputs of the two subnetworks for the final classification decision. [Brahma, 2018] augmented the embeddings obtained from a Siamese architecture with two features, based on exact and paraphrase match between the words in two sentences. [Godbole *et al.*, 2018] combined the features from Gated Recurrent Unit sub-networks with additional handcrafted feature representations and used machine learning algorithms like Random Forest Classifier and Support Vector Machines (SVM) to obtain the results. [Chopra *et al.*, 2020] used a combination of similarity scores obtained from a Siamese network over LSTM and three other scores (two variants of cosine similarity and customized fuzzy match) and developed a meta classifier using SVMs.

The second setup, Matched Aggregation understands the relationship between the two sentences at the word level. Usually, the solutions in this space resort to attention to aggregate the matching information. [Wang *et al.*, 2017] proposed the Bilateral Multi-Perspective Matching (BiMPM) model that matches two sentences by a "bilateral matching with attention" mechanism in multiple perspectives. They proposed four types of representations instead of the attention-weighted representation to improve the results. [Tomar *et al.*, 2017] used character n-grams instead of word embeddings to improve the performance of BiMPM and thus matched the sentences from multiple perspectives. [Gong *et al.*, 2018] introduced the Interactive Inference Network (IIN), which can hierarchically extract semantic features from the interaction space. [Tan *et al.*, 2018] applied 4 different attention mechanisms on the obtained contextual embeddings and aggregated the matching information by further making use of bidirectional RNNs. [Choi *et al.*, 2019] constructed multi-layer LSTMs where memory cell states from the previous layer are used to control the vertical information flow thus filtering the information from lower layers and reflecting it through a soft gating mechanism. [Xu *et al.*, 2020] used an enhanced attention mechanism that helped strengthen the interaction between sentences via adding alignment context into local context in the convolution operation and combining multi-grained similarity features in different filter sizes. [Zhang *et al.*, 2020] used BERT to obtain the dynamic pre-trained word embeddings. Then inter and intra-asking emphasis is obtained by summing inter-attention and self-attention, respectively. The idea is that, the more a word interacts with others, the more important the word is. Finally, an eight-way combination is used to generate multi-fusion asking emphasis and multi-fusion word representation.

These approaches, however, have some drawbacks of their own. In the Siamese setting, there is no interaction between the two sentences in the encoding process. On the other hand, the Matched Aggregation approach makes use of recurrence in the encoding process and complex attention mechanisms to improve the word-level interaction between the two sentences. Some methods which use this approach perform matching only in a single direction, neglecting the information in the sentence pairs. We aim to tackle these drawbacks by employing a combination of transformer architecture and CNNs that provides for bidirectionality and context between the two phrases while relying just on attention and convolutional operations. Transfer learning has been on the rise in NLP [Malte and Ratadiya, 2019], and we intend to leverage the same for our benefit.

3 Proposed Methodology

In this section, we explain the proposed model architecture. First, we give a brief description of the Question Paraphrase Identification Task, followed by definitions of the building blocks of the proposed architecture. This is followed by the two inference setups and the hyperparameter setup for our experiment.

3.1 Task Description

For the question paraphrase identification task, formally, each question pair can be represented as a triplet (A,B,y), where A = (a_1 ,..., a_i ,..., a_N) is a question sequence of length N, B = (b_1 ,...,b_j ,..., b_M) is a question sequence of length M, and y $\in \{0, 1\}$ is the label that represents the relationship between A and B. Here, y = 1 denotes that A and B are identical, while y = 0 denotes that they are semantically dissimilar.

3.2 Architecture Details

We propose a combination of the Bidirectional Transformer Encoder and Convolutional Neural Networks as the main feature extraction block. Contextual embeddings are derived from the transformer encoder which are then passed as an input to the convolutional network.

Bidirectional Transformer Encoder (BTE)

The transformer encoder block is inspired from the BERT model architecture [Devlin *et al.*, 2019]. The internal architecture is significantly dominated by the Transformer encoder [Vaswani *et al.*, 2017]. The model was pre-trained on large text corpora and can generate word-level contextualized representations for a given sequence. As shown in Figure 1, the encoder block comprises a Multi-Head Self Attention Layer and a position-wise fully connected feed-forward layer. A residual connection is employed between each of these two layers, followed by layer normalization.

The encoder employs the "Scaled Dot Product Attention" mechanism to encode each token based on all the other relevant tokens in the sequence. The formula for calculating attention focused weights is shown in the following equation:

$$Z = softmax(\frac{Q \times K^T}{\sqrt{(d_k)}})V \quad (1)$$

where, $Q = (W_q E)$ is the Query vector, $K = (W_k E)$ is the Key vector, $V = (W_v E)$ is the Value vector. Here E is the embedding vector and W_q, W_k and W_v are trainable Weight matrices.

The attention mechanism is further refined by employing "Multi Headed" attention. The idea is to leverage the attention mechanism multiple times by calculating multiple Z matrices for different sets of Key, Query, and Value vectors. All the output Z matrices are then concatenated and further multiplied by an additional weight matrix to get the corresponding Multi Headed Attention output which is then fed to the feed-forward layer.

We use the publicly available weights of the Bidirectional Transformer Encoder[3], which has **12** Transformer Encoders, **12** self-attention heads and an embedding size of **768**.

Convolutional Neural Networks (CNN)

The CNN architecture we adopt is shown in Figure 2. Let $x_i \in R^{768}$ be the token embedding of the i^{th} token in the sequence. The resulting output representations from the encoder (of length n) can be interpreted as:

$$x_{1:n} = x_1 \oplus x_2 \oplus ... \oplus x_n \quad (2)$$

where $\oplus$ is the concatenation operator. The convolution operation applies a filter $w \in R^{g,768}$ to a g-word window to produce a new feature. A feature e_i is generated from a window of words $x_{i:i+g-1}$ by:

$$e_i = f(w \cdot x_{i:i+g-1} + b) \quad (3)$$

where b is the bias term and f is the ReLU activation function. This filter is then applied to each possible window of words in the sequence $\{x_{1:g}, x_{2:g+1}, ... , x_{n-g+1:n}\}$ to produce a feature map:

$$e = [e_1, e_2, ..., e_{n-g+1}] \quad (4)$$

The feature map is then subjected to a max-over-time pooling operation, with the maximum value serving as the feature corresponding to a particular filter.

This process is for extracting a single feature using a single filter. To obtain multiple features, the model employs multiple filters (of varying window sizes). The CNN in our case is made up of 400 parallel convolutional filters of four distinct sizes $(768 \times 2, 768 \times 3, 768 \times 4, 768 \times 5)$, each with 100 filters. These features are then concatenated and passed through a fully connected softmax layer to obtain the corresponding label. We also adopt a dropout layer for regularization purposes.

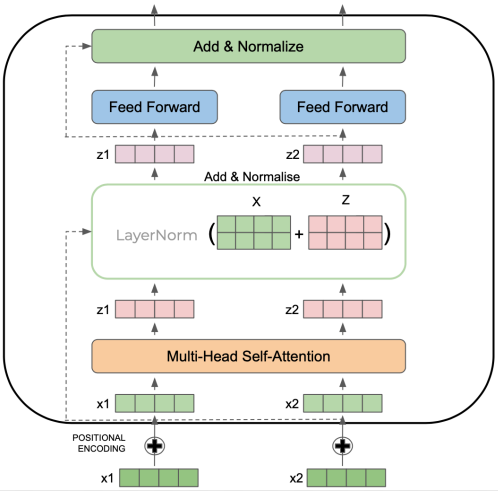

Figure 1: Bidirectional Transformer Encoder

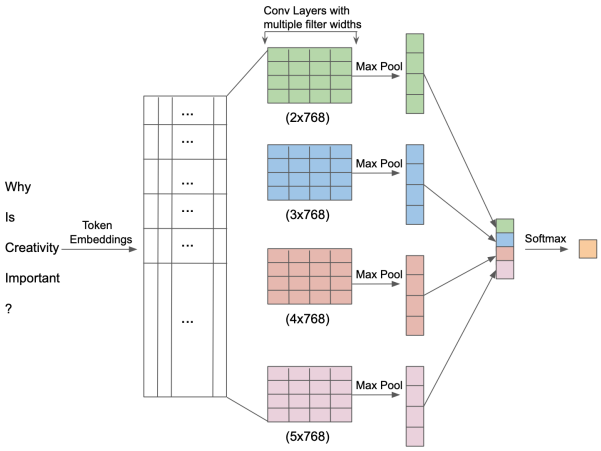

Figure 2: CNN Model Architecture

3.3 Inference Methodology

As mentioned earlier, we experiment with two inference setups for the proposed architecture:

[3]https://huggingface.co/bert-base-uncased

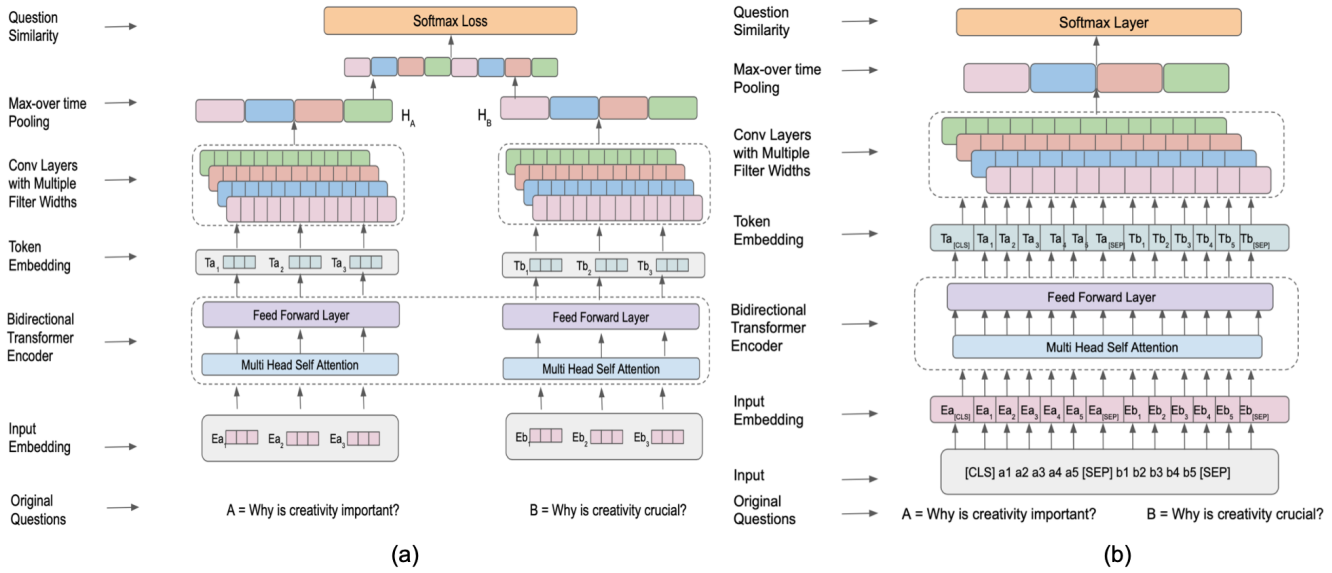

Figure 3: Our Proposed Methodology - a combination of Bidirectional Transformer Encoder and CNN: (a) Siamese Framework. (b) Matched Aggregation Framework.

1. Siamese Network
2. Matched Aggregation Framework

In the case of Siamese Architecture, as shown in Figure 3(a), our proposed approach works in the following steps: The input question is first tokenized, and an input embedding is created for each token $a_i \in A$ and $b_i \in B$. These embeddings are then fed to the Encoder along with their corresponding attention masks. The attention mask is a boolean vector that represents the tokens that should or should not be attended to by the model. The resulting output representations from the encoder are then passed through multiple parallel convolutional filters and max-pooling layers to obtain multiple output features. Concatenating these multiple output features yields condensed vectors H_A and H_B, which represent questions A and B, respectively. The condensed vectors are then concatenated and passed through a softmax layer to get the corresponding label. Our proposed methodology for the Siamese Network is explained in Algorithm 1.

Algorithm 1 Siamese Network

Input: T_n tuples, each of form (S_a, S_b, $label$);
Output: Trained model

for $tuple \in T_n$ **do**
  $h_a \leftarrow BTE + CNN(tuple[0])$
  $h_b \leftarrow BTE + CNN(tuple[1])$
  $label \leftarrow tuple[2]$
  $input \leftarrow h_a \oplus h_b$
  $output \leftarrow$ softmax($input$)
  backpropogate(CROSS_ENTROPY_LOSS($output$,$label$))
end for

In the Matched Aggregation approach as shown in Figure 3(b), the two questions are "packed together" into a single sequence. The questions are differentiated by the presence of a special token ($[SEP]$) between the two questions. Furthermore, a binary mask is also associated with every token indicating whether it belongs to question A or question B. Since the two questions are packed into a single sequence, one single condensed vector is obtained (corresponding to the sequence) which is then passed through a softmax layer to get the corresponding label. Our proposed methodology for the Matched Aggregation Framework is explained in Algorithm 2. The difference in the encoding process is the significant distinguishing factor between the two proposed approaches.

Algorithm 2 Matched Aggregation Framework

Input: T_n tuples of form (S_a, S_b, $label$);
Output: Trained model

for $tuple \in T_n$ **do**
  $inp \leftarrow$ [CLS] + $tuple$[0] + [SEP] + $tuple$[1] + [SEP]
  $label \leftarrow tuple[2]$
  $h_{int} \leftarrow BTE + CNN(inp)$
  $output \leftarrow$ softmax(h_{int})
  backpropogate(CROSS_ENTROPY_LOSS($output$,$label$))
end for

3.4 Hyperparameter Setup

The hyperparameter setup is identical for both the Siamese Network and the Matched Aggregation Framework except for the *max_len* argument. The *max_len* of the input sequence was chosen to be **64** for the Matched Aggregation Framework and **32** for the Siamese Framework. To counter the slight imbalance in the dataset, we adopt the *Weighted Random Sampler* from PyTorch. We use the Cross-Entropy Loss function to calculate the loss. Adam was used as an optimizer, the

Identification of Duplicate Questions
S_1: How can I be a good geologist?
S_2: What should I do to be a great geologist?
Label: 1 (Duplicate Questions)
S_1: What are some good rap songs to dance to?
S_2: What are some of the best rap songs?
Label: 0 (Distinct Questions)

Table 1: Sample dataset inputs

batch size was chosen to be **8**, and the learning rate was set to 10^{-5}. The models were trained for **12** epochs on Nvidia GeForce RTX 2080 Ti GPU.

4 Dataset Description

The Quora Question Pairs dataset comprises of more than 400,000 question pairs along with their indicative paraphrase label. There is no official train/validation/test split. However, a standard split used by [Wang *et al.*, 2017] has since been used by other studies as well [Tomar *et al.*, 2017; Gong *et al.*, 2018; Tan *et al.*, 2018]. We use the same split for fair result comparison. Both the validation and test set comprise of 10000 samples each, 5000 paraphrase pairs, and 5000 non-paraphrase pairs. Table 1 shows some sample inputs from the dataset.

5 Results

For the comparison against previous techniques, we report the Classification accuracy in Table 2 for duplicate question identification on the Quora Question Pairs Test Set. Due to the fair balance of samples between the classes, most papers utilize test accuracy as the only evaluation metric. It can be seen that our BTE+CNN approach in a matched aggregation framework has produced the best results.

We also conduct experiments to determine how the number of tunable encoders impacts the overall performance of the model. Experimental results suggest that there is an increase in the performance with the number of tunable encoders, however, it is not substantial enough to compensate for the massive increase in the training cost and time. We also carry out experiments to analyze the influence of CNN on the model architecture and the overall performance. When compared to Mean Pooling, the incorporation of CNN in the design results in considerably superior performance. We also report the F1 scores for the experiments we conducted to provide a more comprehensive understanding of our methodology. All these results are tabulated in Table 3.

Our main observations after performing this study have been:

1. The model can recognize stronger relationships between the two sentences due to the Bidirectional Transformer Encoder's pre-training on the Next Sentence Prediction task and its capacity to pack two sentences together. This differs from the Siamese network approach in which the two sentences do not interact explicitly during the encoding process.
2. CNNs seem to be more effective than mean pooling because their design enhances the quality of feature representations by introducing more semantic relationships. In the instance of the Siamese setup, the combination of the Bidirectional Transformer Encoder and CNNs with just four trainable encoders outperforms a fully fine-tuned Bidirectional Transformer Encoder alone. This demonstrates how CNN aids the Siamese framework in capturing the information.
3. Experimenting with a different number of trainable parameters suggests that the higher the number of trainable parameters, the sharper the performance would be. For instance, in the case of Matched Aggregation Framework, the Bidirectional Transformer Encoder-CNN hybrid that outperforms previous state-of-the-art baselines has only four trainable encoders. Even though slightly better results were yielded by training Bidirectional Transformer Encoder in its entirety (twelve trainable encoders), it comes at the cost of having $\sim$4x more trainable parameters, introducing a huge computational overload.
4. We consider the BTE-CNN hybrid with 12 encoders

Model	Accuracy (%)
Siamese-LSTM [Wang *et al.*, 2017]	82.58
MP-LSTM [Wang *et al.*, 2017]	83.21
L.D.C. [Wang *et al.*, 2016]	85.55
BiMPM [Wang *et al.*, 2017]	88.17
PWIM [He and Lin, 2016]	83.40
ESIM [Chen *et al.*, 2017]	85.00
ESIM+Syn T.LSTM [Chen *et al.*, 2017]	85.40
InferSent [Conneau *et al.*, 2017]	86.60
SSE [Nie and Bansal, 2017]	87.80
GenSen [Subramanian *et al.*, 2018]	87.01
LSTM+ElBiS [Choi *et al.*, 2018]	87.30
CDTFME-Aver [Xie *et al.*, 2019]	88.00
pt-DecAttword [Tomar *et al.*, 2017]	87.54
pt-DecAttchar [Tomar *et al.*, 2017]	88.40
CAS-LSTM [Choi *et al.*, 2019]	88.40
Bi-CAS-LSTM [Choi *et al.*, 2019]	88.60
REGMAPR [Brahma, 2018]	88.64
DIIN [Gong *et al.*, 2018]	89.06
MwAN[Tan *et al.*, 2018]	89.12
DIIN (Ensemble)[Gong *et al.*, 2018]	89.84
MFAE(ELMo) [Zhang *et al.*, 2020]	89.61
MFAE(BERT) [Zhang *et al.*, 2020]	89.79
MFAE(BERT Ens) [Zhang *et al.*, 2020]	90.54
BTE + CNN (4 trainable encoders)[4]	**90.58**
BTE + Mean Pooling (12 trainable encoders)[4]	**90.78**
BTE + CNN (12 trainable encoders)[4]	**90.80**

Table 2: Classification Accuracy for Duplicate Question Identification on Quora Question Pairs

[4]This model belongs to the Matched Aggregation Framework

Setup	Model	No. of Tr. Encoders	Accuracy (%)	F1-Score	No. of Tr. Params
Siamese	BTE+Mean Pooling	4	81.11	0.8008	29M
	BTE+Mean Pooling	12	82.10	0.8085	109M
	BTE+CNN	2	86.68	0.8606	16M
	BTE+CNN	4	**86.88**	**0.8613**	30M
	BTE+CNN	12	86.56	0.8562	110M
Matched Aggregation	BTE+Mean Pooling	4	90.41	0.8981	29M
	BTE+Mean Pooling	12	90.78	0.9027	109M
	BTE+CNN	2	90.10	0.8950	16M
	BTE+CNN	4	90.58	0.9002	30M
	BTE+CNN	12	**90.80**	**0.9022**	110M

Table 3: Performance Comparison in the two setups - Siamese and Matched Aggregation. Tr stands for Trainable.

Predictions by the MA and Siamese Frameworks
S_1: How do I improve improvisation skills on drums? S_2: What should I do to improve my drumming skills? **MA**: 0 (Distinct Questions) **Siam**: 1 (Duplicate Questions)

Table 4: Predictions. MA: Matched Aggregation Framework and Siam: Siamese Framework

in both the frameworks and discover that the Matched Aggregation framework was able to correctly recognize 62% of the samples that were incorrectly classified by the Siamese framework on the test set. Table 4 shows a sample that was wrongly classified by the Siamese Framework as a duplicate pair, but correctly classified by the Matched Aggregation Framework as a distinct pair.

6 Conclusion

In this paper, we introduced a novel contextual embedding-based CNN architecture for duplicate question pairs identification on the Quora Question Pairs dataset. We combined together Bidirectional Transformer Encoder and Convolutional Neural Networks and leveraged its representation extraction ability in two setups - Siamese and Matched Aggregation. More importantly, we discover that our approach of encoding the two sentences together proves to be effective in identifying the duplication. Results show that the proposed approach produces the best results yet, despite being a non-ensemble approach with significantly less training cost. Another point to note is that, while training the Encoder in its entirety ensures even better results, the current trained setup almost matches the same results while having $\sim$4x lesser trainable parameters. The future direction of this work could include investigating the performance of the model on multilingual and domain-specific data.

References

[Bowman *et al.*, 2016] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1466–1477, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Brahma, 2018] Siddhartha Brahma. Regmapr - text matching made easy. *ArXiv*, abs/1808.04343, 2018.

[Chen *et al.*, 2017] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 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, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[Chen *et al.*, 2018] Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. Quora question pairs. *University of Waterloo*, 2018.

[Choi *et al.*, 2018] Jihun Choi, Taeuk Kim, and Sang-goo Lee. Element-wise bilinear interaction for sentence matching. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 107–112, 2018.

[Choi *et al.*, 2019] Jihun Choi, Taeuk Kim, and Sang-goo Lee. Cell-aware stacked lstms for modeling sentences. In *Asian Conference on Machine Learning*, pages 1172–1187. PMLR, 2019.

[Chopra *et al.*, 2020] Ankush Chopra, Shruti Agrawal, and Sohom Ghosh. Applying transfer learning for improving domain-specific search experience using query to question similarity. In *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 1–8, 2020.

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

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics.

[Godbole *et al.*, 2018] Ameya Godbole, Aman Dalmia, and Sunil Sahu. Siamese neural networks with random forest for detecting duplicate question pairs. 01 2018.

[Gong *et al.*, 2018] Yichen Gong, Heng Luo, and Jian Zhang. Natural language inference over interaction space. In *International Conference on Learning Representations*, 2018.

[He and Lin, 2016] Hua He and Jimmy Lin. 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, San Diego, California, June 2016. Association for Computational Linguistics.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[Lan and Xu, 2018] Wuwei Lan and Wei Xu. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[Malte and Ratadiya, 2019] Aditya Malte and Pratik Ratadiya. Evolution of transfer learning in natural language processing. *arXiv preprint arXiv:1910.07370*, 2019.

[Nie and Bansal, 2017] Yixin Nie and Mohit Bansal. Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[Subramanian *et al.*, 2018] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*, 2018.

[Tan *et al.*, 2016] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Improved representation learning for question answer matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 464–473, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Tan *et al.*, 2018] Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. Multiway attention networks for modeling sentence pairs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4411–4417. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[Tomar *et al.*, 2017] Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. Neural paraphrase identification of questions with noisy pretraining. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 142–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Wang and Jiang, 2016] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. 11 2016.

[Wang *et al.*, 2016] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 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, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[Wang *et al.*, 2017] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150, 2017.

[Xie *et al.*, 2019] Yuqiang Xie, Yue Hu, Luxi Xing, and Xiangpeng Wei. Dynamic task-specific factors for meta-embedding. In *International Conference on Knowledge Science, Engineering and Management*, pages 63–74. Springer, 2019.

[Xu *et al.*, 2020] Shiyao Xu, Shijia E, and Yang Xiang. Enhanced attentive convolutional neural networks for sentence pair modeling. *Expert Systems with Applications*, 151:113384, 2020.

[Yu *et al.*, 2014] L. Yu, K. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. *ArXiv*, abs/1412.1632, 2014.

[Zhang *et al.*, 2020] Rong Zhang, Qifei Zhou, Bo Wu, Weiping Li, and Tong Mo. *What Do Questions Exactly Ask? MFAE: Duplicate Question Identification with Multi-Fusion Asking Emphasis*, pages 226–234. 01 2020.