A Globalization-Semantic Matching Neural Network for Paraphrase Identification

Miao Fan[1,2], Wutao Lin[1,2], Yue Feng[1,2], Mingming Sun[1,2], Ping Li[1]
1 Big Data Lab (BDL-US), Baidu Research
2 National Engineering Laboratory of Deep Learning Technology and Application, China
{fanmiao,linwutao,fengyue04,sunmingming01,liping11}@baidu.com

ABSTRACT

Paraphrase identification (PI) aims at determining whether two natural language sentences roughly have identical meaning. PI has been conventionally formalized as a binary classification task and widely used in many talks such as text summarization, plagiarism detection, etc. The emergence of deep neural networks (DNNs) renovates and dominates the learning paradigm of PI, as DNNs do not rely on lexical nor syntactic knowledge of a language, unlike traditional methods. State-of-the-art DNNs-based approaches to PI mainly adopt multi-layer convolutional neural networks (CNNs) to model paraphrastic sentences, which could discover alignments of phrases with the same length (unigram-to-unigram, bigram-to-bigram, trigram-to-trigram, etc.) at each layer. However, paraphrasing phenomena globally exist at all levels of granularity between a pair of paraphrastic sentences, i.e., *word-to-word*, *word-to-phrase*, *phrase-to-phrase*, and even *sentence-to-sentence*.

In this paper, we contribute a globalization-semantic matching neural network (**GSMNN**) paradigm which has been deployed in Baidu.com to tackle practical PI problems. Established on a weight-sharing single-layer CNN, GSMNN is composed of a multi-granular matching layer with the attention mechanism and a sentence-level matching layer. These layers are designed to capture essentially all phenomena of semantic matching. Evaluations are conducted on a public large-scale dataset for PI: Quora-QP which contains more than 400,000 paraphrasing and non-paraphrasing question pairs from Quora.com. Experimental results show that GSMNN outperforms the state-of-the-art model by a substantial margin.

KEYWORDS

Paraphrase identification; CNN; semantic matching

ACM Reference Format:
Miao Fan, Wutao Lin, Yue Feng, Mingming Sun, Ping Li. 2018. A Globalization-Semantic Matching Neural Network for Paraphrase Identification. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3269206.3272004

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3272004

1 INTRODUCTION

Paraphrase identification (PI) [18] aims at determining whether two natural language sentences (say P and Q) roughly have identical meaning. PI is conventionally formalized as a binary classification task, and is widely used in plagiarism detection [2, 25], text summarization [3, 19], evaluation of machine translation [7, 14, 31], etc. Early approaches to PI used simple surface-form word matching [20], lexical [15] features, or syntactic [27] features. The recent emergence of deep neural networks (DNNs) [4, 8, 17] renovates and dominates the learning paradigm of PI, as DNNs do not directly rely on lexical nor syntactic knowledge of a language. Moreover, DNNs have the potential to achieve outstanding performance via extracting hierarchical features automatically.

Latest research on DNNs for PI mainly adopts multi-layer convolutional neural networks (CNNs) [12] to model paraphrastic sentences, in which each layer encodes a specific *n-grams* representation. Based on the architecture, Yin and Schütze [28] and Yin et al. [29] fully explore interactions between a pair of sentences, and discover linguistic alignments of phrases with the same length (unigram-to-unigram, bigram-to-bigram, trigram-to-trigram, etc.) at each layer (see Figure 4 in Yin et al. [29]). In this paper, however, we argue that a pair of paraphrastic sentences shall not be solely identified by the alignments of phrases with the same length. The phenomena of semantic matching should globally exist at all levels of granularity between a pair of paraphrastic sentences, i.e., *word-to-word*, *word-to-phrase*, *phrase-to-phrase*, and even *sentence-to-sentence*. For example, here we have four pairs of paraphrastic sentences, **P1** & **Q1**, **P2** & **Q2**, **P3** & **Q3**, and **P4** & **Q4**:

P1: *He writes the book.*
Q1: *He pens the book.*

P2: *I can take care of myself.*
Q2: *I am perfectly capable of looking after myself, thank you!*

P3: *How much is iPhone X?*
Q3: *What is the price of iPhone X?*

P4: *Lucy gave birth to a daughter in 2008.*
Q4: *Kate is 10 years old now, and her mother is Lucy.*

The example of **P1** & **Q1** demonstrates that the word-to-word matching (e.g., *writes* ↔ *pens*) helps to identify paraphrastic sentences. Other cases of multi-granular matchings, such as word-to-phrase (e.g., *can* ↔ *am perfectly capable of*) and phrase-to-phrase (e.g., *how much* ↔ *what is the price of*) can be found in **P2** & **Q2** and **P3** & **Q3**, respectively. In addition to these multi-granular matchings, the sentence-level matching as shown by **P4** & **Q4**, is also

regarded as a type of paraphrasing phenomenon that we would hardly find explicit alignments of words or phrases, but can tell whether two sentences have the same meaning after reading them from beginning to end. To the best of our knowledge, neither the baseline nor the state-of-the-art approaches [28, 29] can capture all those semantic matchings, because the architecture of multi-layer CNN [12] they leveraged forces these CNN-based models for PI, in particular Bi-CNN-MI [28] and ABCNN [29], to align phrases with the same number of words.

In this paper, to better identify paraphrastic sentences, we contribute a globalization-semantic matching neural network (**GSMNN**) to comprehensively take those phenomena of semantic matching into full consideration. The proposed model is established on a weight-sharing single-layer CNN [13] at first. Multi-granular and sentence-level semantics/embeddings are produced by applying multiple filters with different pooling strategies to the convolutional layer. The two groups of embeddings (multi-granular embeddings and sentence-level embeddings) are then fed into a multi-granular matching layer with attention mechanism (**MGANN**) and a sentence-level matching layer (**SLMNN**), respectively. Finally, a globalization-semantic layer concatenates the two pieces of matching evidence from MGANN and SLMNN as features to train a logistic regression model for binary classification.

Evaluations are conducted on a public large-scale dataset for paraphrase identification: Quora-QP. The dataset[1] contains more than 400 thousand paraphrasing and non-paraphrasing question pairs from Quora.com, a prominent community question answering (CQA) [21–23] website. Our experimental results will show that GSMNN outperforms the state-of-the-art model ABCNN [29] in view of increasing accuracy by **4.82%** and F1-score by **4.27%** on the Quora-QP dataset. To more thoroughly explain the advantages of our approaches, we further open extensive discussions, based on quantitative and qualitative analysis, to answer several important subsequent questions: 1) why do we prefer matching semantics rather than concatenating embeddings? 2) why do we apply the attention mechanism to the multi-granular matching layer? and 3) why does GSMNN outperform ABCNN?

2 RELATED WORK

RAE [26] is a prior study which embodied the insight of analyzing information on multi-levels of granularity for paraphrase identification (PI). While RAE is a pioneer that adopted neural networks, its architecture highly depends on syntactic tree structures generated by an accurate parser which might not be available.

Hu et al. [11] exploited the convolutional neural network (CNN) [16] for PI without using extra knowledge on linguistics. The ARC-I [11] and the modern HSCNN (Hybrid Siamese CNN) [24] adopted the Siamese CNN architecture [5] to generate fix-length sentence-level embeddings as evidence, but used different loss functions, i.e., the logistic loss and a contrastive loss, respectively. However, neither ARC-I or HSCNN models the inner interactions between a pair of sentences, and the sentence-level embeddings would also make certain explicit paraphrastic information unobservable.

Later, Yin and Schütze [28] proposed Bi-CNN-MI which focuses on word-to-word and phrase-to-phrase alignments between a pair of paraphrastic sentences. To improve the performance of Bi-CNN-MI, Yin et al. [29] further enhanced Bi-CNN-MI with attention mechanism [1]. As far as we know, their attention-based model ABCNN [29] achieves state-of-the-art performance on the PI task.

Both Bi-CNN-MI and ABCNN leverage the multi-layer CNN architecture designed by Kalchbrenner et al. [12] in which each layer only encodes grams with the same length for two sentences. This structure limits the capabilities of Bi-CNN-MI and ABCNN to capture multi-granular semantic matchings, i.e., *word-to-word*, *word-to-phrase*, *phrase-to-phrase*, and even *sentence-to-sentence*, between two paraphrastic sentences. An empirical study conducted by Zhang and Wallace [30] inspired us to use another single-layer CNN architecture proposed by Kim [13], since it demonstrates that the two mainstream CNN architectures [12, 13] have comparable performance on modeling sentences. Moreover, the single-layer CNN [13] has the advantage of producing multi-granular as well as sentence-level representations by multiple filters with different pooling strategies.

Overall, the competitive approaches involved in this paper include ARC-I [11], HSCNN [24], Bi-CNN-MI [28] and ABCNN [29].

3 MODELS

As shown by Fig. 1, the globalization-semantic matching neural network (GSMNN) that we introduce to paraphrase identification (PI), consists of two components: 1) two weight-sharing single-layer CNNs which mainly adopt the simple architecture proposed by Kim [13] to generate multi-granular and sentence-level representations for a pair of sentences, and 2) various neural layers fed by the convoluted representations for semantic matching to generate evidence for binary classification (paraphrase identification).

3.1 Sentence Representations with Shared Single-layer CNN

Suppose we have a pair of sentences (P and Q) to be identified. Let $\mathbf{p}_i \in \mathbb{R}^d$ be the d-dimensional word vector[2] corresponding to the i-th word in the sentence P, and $\mathbf{q}_j \in \mathbb{R}^d$ be the d-dimensional word vector corresponding to the j-th word in the sentence Q. Both $\mathbf{p}_i$ and $\mathbf{q}_j$ are mapped from a shared matrix $\mathbf{V} \in \mathbb{R}^{d \times l}$ which contains embeddings of the whole vocabulary of length l.

Given the fact that each sentence is a sequence of words, the sentence P which has m words is represented as

$$\mathbf{p}_{1:m} = \mathbf{p}_1 \oplus \mathbf{p}_2 \oplus \ldots \oplus \mathbf{p}_m, \tag{1}$$

where $\oplus$ is the concatenation operator, and $\mathbf{p}_{i:j}$ refers to the concatenated embeddings from the i-th word to the j-th word in the sentence P. Thus, the sentence Q of length n is represented as

$$\mathbf{q}_{1:n} = \mathbf{q}_1 \oplus \mathbf{q}_2 \oplus \ldots \oplus \mathbf{q}_n. \tag{2}$$

On the top of those sentence representations ($\mathbf{p}_{1:m}$ and $\mathbf{q}_{1:n}$), we use two weight-sharing CNNs with one layer of convolution in which multiple filters and different pooling strategies are adopted to generate various semantic features for matching the pair of sentences (P and Q).

[1] The dataset is available at http://qim.ec.quoracdn.net/quora_duplicate_questions.tsv.

[2] By default, a vector is a column vector in this article.

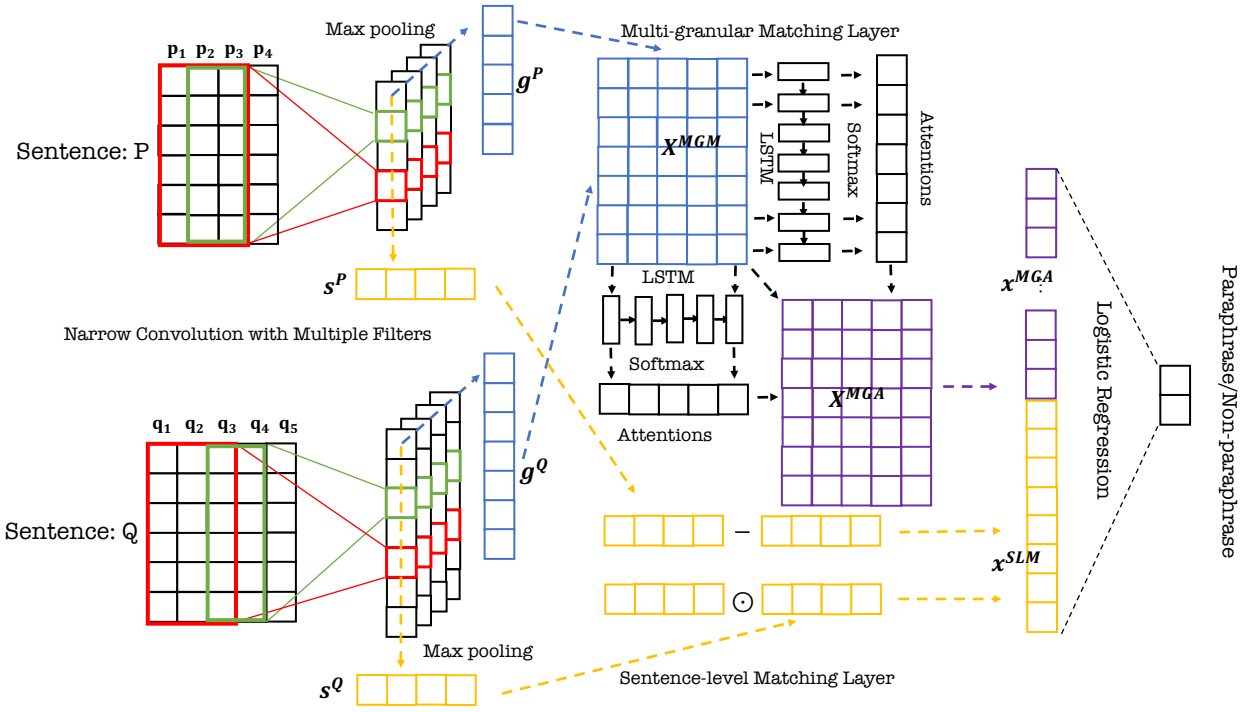


Figure 1: Architecture of the proposed globalization-semantic matching neural network (GSMNN).

3.1.1 Narrow Convolution with Multiple Filters. For this study, we prefer the narrow convolution [13] without padding, rather than the wide one-dimensional convolution used by Kalchbrenner et al. [12]. With narrow convolution, each filter can generate distributed representations of n-grams exactly with the same length. For instance, if we apply a filter $\mathbf{w}_i^h \in \mathbb{R}^{d \times h}$ which denotes the i-th filter with a window of h words, to a sequence of words $\mathbf{p}_{j:j+h-1}$ in the sentence P, a convoluted feature $c_{i,j}^{P,h}$ is produced by:

$$c_{i,j}^{P,h} = f(\mathbf{w}_i^h \cdot \mathbf{p}_{j:j+h-1} + b^h), \tag{3}$$

where $b^h \in \mathbb{R}$ is a bias term corresponding to the vector of k filters $\mathbf{w}^h = \left[\mathbf{w}_1^h, \mathbf{w}_2^h, ..., \mathbf{w}_k^h\right]^{\mathrm{T}}$, and f is a non-linear function such as *sigmoid*, *tanh* and *relu*. Here we use *tanh* as the non-linear function. The feature map $\mathbf{C}^{P,h} = \left[\mathbf{c}_1^{P,h}, \mathbf{c}_2^{P,h}, ..., \mathbf{c}_k^{P,h}\right]^{\mathrm{T}}$ encodes k kinds of h-gram embeddings of the sentence P, in which the row vector $\mathbf{c}_i^{P,h} \in \mathbb{R}^{m-h+1}$ $(i \in \{1, 2, ..., k\})$ and $\mathbf{c}_i^{P,h} = \left[c_{i,1}^{P,h}, c_{i,2}^{P,h}, ..., c_{i,m-h+1}^{P,h}\right]$. For the sentence Q of length n, the feature map is $\mathbf{C}^{Q,h} = [\mathbf{c}_1^{Q,h}, \mathbf{c}_2^{Q,h}, ..., \mathbf{c}_k^{Q,h}]^{\mathrm{T}}$, where the row vector $\mathbf{c}_j^{Q,h} \in \mathbb{R}^{n-h+1}$ $(j \in \{1, 2, ..., k\})$ and $\mathbf{c}_j^{Q,h} = \left[c_{j,1}^{P,h}, c_{j,2}^{P,h}, ..., c_{j,n-h+1}^{P,h}\right]$.

3.1.2 Pooling Strategies. Two strategies for max pooling are leveraged to generate sentence-level and multi-granular semantics on top of the two h-gram feature maps, i.e., $\mathbf{C}^{P,h}$ and $\mathbf{C}^{Q,h}$. The idea of max pooling is to keep the most significant feature: one with the highest value and to simultaneously decrease the complexity of computation.

The strategy of *max pooling over each feature map* [6] concerns the most representative value over each kind of h-gram embeddings ($\mathbf{c}_i^{P,h}$ and $\mathbf{c}_j^{Q,h}$) for P and Q to encode the sentence-level semantics. We use $\mathbf{s}^{P,h}$ and $\mathbf{s}^{Q,h}$ to denote the sentence-level embeddings composed by h-grams, then $\mathbf{s}^{P,h} = [\max([c_{1,*}^{P,h}]), ..., \max([c_{k,*}^{P,h}])]$[3] and $\mathbf{s}^{Q,h} = [\max([c_{1,*}^{Q,h}]), ..., \max([c_{k,*}^{Q,h}])]$[4]. If we set up multiple filters $(\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^h)$ with the window size ranging from 1 to h to encode sentence-level semantics from unigrams to h-grams, the sentence-level embeddings of P and Q are represented as

$$\mathbf{s}^P = (\mathbf{s}^{P,1} \oplus \mathbf{s}^{P,2} \oplus ... \oplus \mathbf{s}^{P,h})^{\mathrm{T}}, \tag{4}$$

and

$$\mathbf{s}^Q = (\mathbf{s}^{Q,1} \oplus \mathbf{s}^{Q,2} \oplus ... \oplus \mathbf{s}^{Q,h})^{\mathrm{T}}, \tag{5}$$

where $\mathbf{s}^P, \mathbf{s}^Q \in \mathbb{R}^{kh}$ are both column vectors of length $k \times h$.[5]

The strategy of *max pooling across multiple feature maps* [9] aims at selecting the most representative value for each word or phrase across multiple feature maps. It is different from the strategy of *max pooling over each feature map* which generates static length (kh) sentence-level embeddings regardless of the lengths of P and Q, i.e., m and n. The strategy of *max pooling across multiple feature maps* can explicitly encode multi-granular information, but the length of multi-granular embeddings varies with the length of the corresponding sentence. We use $\mathbf{g}^{P,h}$ and $\mathbf{g}^{Q,h}$ to denote the multi-granular semantics composed by representations of words and

[3] $[c_{i,*}^{P,h}]$ is an equivalent expression of $\mathbf{c}_i^{P,h}$.
[4] $[c_{i,*}^{Q,h}]$ is an equivalent expression of $\mathbf{c}_i^{Q,h}$.
[5] In practice, it is possible to set different number (k) of filters for each kind of filter.

phrases, then $\mathbf{g}^{P,h} = \left[\max([c^{P,h}_{*,1}]), \max([c^{P,h}_{*,2}]), ..., \max([c^{P,h}_{*,m-h+1}])\right]$ and $\mathbf{g}^{Q,h} = \left[\max([c^{Q,h}_{*,1}]), \max([c^{Q,h}_{*,2}]), ..., \max([c^{Q,h}_{*,n-h+1}])\right]$. If we set up multiple filters $(\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^h)$ with the window size ranging from 1 to h to encode multi-granular semantics from unigrams to h-grams, the multi-granular embeddings of P and Q are, respectively, represented as

$$\mathbf{g}^P = (\mathbf{g}^{P,1} \oplus \mathbf{g}^{P,2} \oplus ... \oplus \mathbf{g}^{P,h})^{\mathrm{T}}, \tag{6}$$

and

$$\mathbf{g}^Q = (\mathbf{g}^{Q,1} \oplus \mathbf{g}^{Q,2} \oplus ... \oplus \mathbf{g}^{Q,h})^{\mathrm{T}}, \tag{7}$$

where $\mathbf{g}^P \in \mathbb{R}^{m+(m-1)+...+(m-h+1)}$ and $\mathbf{g}^Q \in \mathbb{R}^{n+(n-1)+...+(n-h+1)}$.

3.2 Semantic Matching Layers for PI

After the weight-sharing single-layer CNN produces multi-granular ($\mathbf{g}^P$ and $\mathbf{g}^Q$) and sentence-level ($\mathbf{s}^P$ and $\mathbf{s}^Q$) representations of sentences P and Q, a key component of our approach is to discover aligned semantics between a pair of sentences for paraphrase identification. Given that paraphrase identification is conventionally regarded as a binary classification task, the features of semantic matching are fed into a discriminative model such as the logistic regression, so that they can be directly valued. Suppose that $\mathcal{Y} = \{0, 1\}$ is the set of labels, and $y \in \mathcal{Y}$ denotes the random variable of class. We use $\mathcal{X}$ to represent the set of observed data in which $\mathbf{x} \in \mathcal{X}$ is a feature vector of a pair of sentences. The probabilistic definition of the discriminative model with parameters θ for binary classification is

$$\hat{y} = \arg\max Pr(y|\mathbf{x}; \theta), \tag{8}$$

in which $\hat{y}$ is the result of paraphrase identification given the column vector $\mathbf{x}$ of a pair of sentences.

3.2.1 Multi-granular Matching with Attention. Concatenation is an intuitive option to build the feature vector $\mathbf{x}$ in Equation 8. We name it *multi-granular concatenation neural network* abbreviated as MGCNN. In MGCNN, the feature vector $\mathbf{x}^{MGC}$ concatenates the pair of multi-granular embeddings $\mathbf{g}^P$ and $\mathbf{g}^Q$:

$$(\mathbf{x}^{MGC})^{\mathrm{T}} = (\mathbf{g}^P)^{\mathrm{T}} \oplus (\mathbf{g}^Q)^{\mathrm{T}}. \tag{9}$$

An alternative approach which may perform better than MGCNN is to model interactions between each value in $\mathbf{g}^P$ and $\mathbf{g}^Q$. We use $\mathbf{X}^{MGM}$ to record the semantic matching between $\mathbf{g}^P$ and $\mathbf{g}^Q$. It is defined as

$$\mathbf{X}^{MGM} = \mathbf{g}^P(\mathbf{g}^Q)^{\mathrm{T}}, \tag{10}$$

in which $\mathbf{X}^{MGM} \in \mathbb{R}^{[m+(m-1)+...+(m-h+1)]\times[n+(n-1)+...+(n-h+1)]}$. Each entry $x^{MGM}_{i,j} \in \mathbf{X}^{MGM}$ indicates the degree of semantic matching between the i-th word/phrase in $\mathbf{g}^P$ and the j-th word/phrase in $\mathbf{g}^Q$. If we map the 2D matrix $\mathbf{X}^{MGM}$ into a 1D feature vector $\mathbf{x}^{MGM} = \left[x^{MGM}_{1,1}, x^{MGM}_{1,2}, ..., x^{MGM}_{2,1}, x^{MGM}_{2,2}, ...\right]^{\mathrm{T}}$ and directly acquire the weights θ in Eq. (8), the model is called MGMNN (*multi-granular matching neural network*) for paraphrase identification.

Given the facts that not all entries in $\mathbf{X}^{MGM}$ could help identify paraphrastic sentences and $x^{MGM}_{i,j} \in \mathbf{X}^{MGM}$ are contextual related to each other, we equip MGMNN with the attention mechanism [1] to make the new model concentrate on fewer but more discriminating features for PI. The enhanced model is called MGANN (*multi-granular attention neural network*) which adopts two LSTMs [10] to re-value each row vector $\left[x^{MGM}_{i,*}\right]$ and column vector $\left[x^{MGM}_{*,j}\right]^{\mathrm{T}}$ in $\mathbf{X}^{MGM}$ with regard to contextual representations in rows and columns, i.e.,

$$(\mathbf{e}^{ro}_i, \mathbf{o}^{ro}_i) = \mathrm{LSTM}(\left[x^{MGM}_{i,*}\right]^{\mathrm{T}}, \mathbf{e}^{ro}_{i-1}, \mathbf{o}^{ro}_{i-1}), \tag{11}$$

and

$$(\mathbf{e}^{co}_j, \mathbf{o}^{co}_j) = \mathrm{LSTM}(\left[x^{MGM}_{*,j}\right]^{\mathrm{T}}, \mathbf{e}^{co}_{j-1}, \mathbf{o}^{co}_{j-1}). \tag{12}$$

$\mathbf{e}^{ro}_i$ and $\mathbf{e}^{co}_j$ represent the hidden states of the two LSTMs for row and column contexts in $\mathbf{X}^{MGM}$ at the time i and j, respectively. $\mathbf{o}^{ro}_i \in \mathbb{R}^t$ denotes the i-th output of the LSTM for context in rows which encodes contextual information surrounding $\left[x^{MGM}_{i,*}\right]^{\mathrm{T}}$ to some extent, and $\mathbf{o}^{co}_j \in \mathbb{R}^{t'}$ contains the information on contexts in column vectors. We use $\mathbf{O}^{ro} = \left[\mathbf{o}^{ro}_1, \mathbf{o}^{ro}_2, ..., \mathbf{o}^{ro}_{m-h+1}\right]$ and $\mathbf{O}^{co} = \left[\mathbf{o}^{co}_1, \mathbf{o}^{co}_2, ..., \mathbf{o}^{co}_{n-h+1}\right]$ to record all contextual information. The attention vectors for rows ($\mathbf{a}^{ro} \in \mathbb{R}^{m-h+1}$) and columns ($\mathbf{a}^{co} \in \mathbb{R}^{n-h+1}$) are dependent upon $\mathbf{O}^{ro} \in \mathbb{R}^{(m-h+1)\times t}$ and $\mathbf{O}^{co} \in \mathbb{R}^{(n-h+1)\times t'}$ respectively:

$$\mathbf{a}^{ro} = \mathrm{softmax}(\mathbf{O}^{ro}\mathbf{w}^{ro}_a + \mathbf{b}^{ro}_a) \tag{13}$$

and

$$\mathbf{a}^{co} = \mathrm{softmax}(\mathbf{O}^{co}\mathbf{w}^{co}_a + \mathbf{b}^{co}_a), \tag{14}$$

in which $\mathbf{w}^{ro}_a \in \mathbb{R}^t$ and $\mathbf{w}^{co}_a \in \mathbb{R}^{t'}$ are weights for learning attentions. $\mathbf{b}^{ro}_a \in \mathbb{R}^{m-h+1}$ and $\mathbf{b}^{co}_a \in \mathbb{R}^{n-h+1}$ are bias terms.

The attention matrix $\mathbf{A}^{MGA}$ for MGANN is defined as

$$\mathbf{A}^{MGA} = \mathbf{a}^{ro}(\mathbf{a}^{co})^{\mathrm{T}}, \tag{15}$$

and $\mathbf{X}^{MGA}$ is the re-weighted semantic matching matrix after applying attention mechanism:

$$\mathbf{X}^{MGA} = \mathbf{X}^{MGM} \odot \mathbf{A}^{MGA}, \tag{16}$$

where $\odot$ denotes element-wise multiplication of two matrices.

If the 2D matrix $\mathbf{X}^{MGA}$ is mapped into an 1D feature vector $\mathbf{x}^{MGA}$, MGANN can directly acquire the weights θ in Eq. (8) for PI.

3.2.2 Sentence-level Matching. The sentence-level embeddings $\mathbf{s}^P$ and $\mathbf{s}^Q$ do not explicitly encode multi-granular semantics of the sentences P and Q. However, $\mathbf{s}^P$ and $\mathbf{s}^Q$ hold important information of the two sentences from different perspectives by means of applying multiple filters. An intuitive way on exploiting the embeddings is to concatenate them and to feed them into Equation 8 as features for paraphrase identification. We name the method *sentence-level concatenation neural network* abbreviated as SLCNN which has the similar structure with ARC-I [11]. The feature vector $\mathbf{x}^{SLC}$ taking place of $\mathbf{x}$ is defined as

$$(\mathbf{x}^{SLC})^{\mathrm{T}} = (\mathbf{s}^P)^{\mathrm{T}} \oplus (\mathbf{s}^Q)^{\mathrm{T}}, \tag{17}$$

in which $\mathbf{x}^{SLC} \in \mathbb{R}^{2kh}$ is a column vector to denote the concatenated sentence-level embedding.

For the purpose of capturing information on semantic matching at the sentence level, we propose SLMNN (*sentence-level matching*

Table 1: Statistics of Quora-QP corpus for paraphrase identification.

DATASETS	# (UNIQUE SENT)	# (SENT PAIR)	# (PARAPHRASE)	# (NON-PARAPHRASE)
Train	336,904	227,406	83,987	143,419
Dev	129,874	75,803	28,044	47,759
Test	167,401	101,070	37,232	63,838
Total	537,349	404,279	149,263	255,016

neural network) to further check each pair of values ($\mathbf{s}_i^P$ and $\mathbf{s}_i^Q$) which co-occur at the same position in $\mathbf{s}^P$ and $\mathbf{s}^Q$. The sentence-level matching feature $\mathbf{x}^{SLM}$ is represented as

$$(\mathbf{x}^{SLM})^{\mathrm{T}} = (\mathbf{s}^P - \mathbf{s}^Q)^{\mathrm{T}} \oplus (\mathbf{s}^P \odot \mathbf{s}^Q)^{\mathrm{T}}, \tag{18}$$

where $\mathbf{x}^{SLM} \in \mathbb{R}^{2kh}$ captures the sentence-level matching information from two perspectives: element-wise difference ($\mathbf{s}^P - \mathbf{s}^Q$) and element-wise multiplication ($\mathbf{s}^P \odot \mathbf{s}^Q$). The reason why we use element-wise operations to match sentence-level embeddings, is that each pair of values ($\mathbf{s}_i^P$ and $\mathbf{s}_i^Q$) is produced by a shared filter which explores the sentence-level semantics from the same perspective. The element-wise multiplication amplifies the co-occurred values at the same index for paraphrastic semantics, while the element-wise difference tends to be significant if two sentences do not have identical meaning.

3.2.3 Globalization-semantic Matching. The comprehensive model GSMNN (*globalization-semantic matching neural network*) is composed by the *multi-granular attention neural network* (MGANN) and the *sentence-level matching neural network* (SLMNN) to cover all the paraphrasing phenomena including *word-to-word*, *word-to-phrase*, *phrase-to-phrase*, and *sentence-to-sentence*. Therefore, the feature vector of GSMNN is defined as

$$(\mathbf{x}^{GSM})^{\mathrm{T}} = (\mathbf{x}^{MGA})^{\mathrm{T}} \oplus (\mathbf{x}^{SLM})^{\mathrm{T}}. \tag{19}$$

The features within $\mathbf{x}^{GSM}$ are then directly optimized by a binary classifier (which in this paper is the logistic regression with l_2-regularization) for paraphrase identification.

4 EXPERIMENTS

In order to offer fair evaluations and comparisons to all the competitive models mentioned in this paper, we carry out experiments on the same large-scale dataset and use the standard metrics of paraphrase identification to evaluate the performance of each model.

4.1 Datasets

The data science team in Quora.com has recently released a large-scale dataset for the purpose of training intelligent models to discover duplicated English questions[6]. We believe that it is an ideal dataset for the task of paraphrase identification, given its scale (the dataset contains more than 400 thousand pairs of paraphrasing and no-paraphrasing English questions) and quality (the dataset is also published as the standard training data for a Kaggle competition[7]).

The original dataset does not provide a standard partition for validation and testing. To evaluate the performance of our approaches along with the baseline and the state-of-the-art methods, we randomly split the dataset into three subsets, i.e., *train*, *dev* and *test*, with the proportion of 3:1:1.5[8]. The processed dataset is named Quora-QP (Quora Question Pairs) corpus. Table 1 records the statistics of Quora-QP corpus, and we find out that it is unbalanced. For each subset, nearly two-thirds of the question pairs are not paraphrasing. This fact suggests that we should carefully choose suitable metrics for binary classification with unbalanced data.

4.2 Metrics & Setups

F1-score and accuracy are the standard metrics to evaluate algorithms for the PI task. Compared with accuracy which generally measures the proportion of correct predictions, F1-score mainly evaluates the model capability of finding paraphrasing (positive) sentence pairs, especially with unbalanced datasets.

We conduct experiments on performance comparison among the modern approaches ARC-I [11], Bi-CNN-MI [28], ABCNN [29], HSCNN [24], and several advanced models we proposed in this paper (MGMNN, MGANN, SLMNN, GSMNN). All the models start with the same embeddings of vocabulary $\mathbf{V} \in \mathbb{R}^{d \times l}$ initialized by the normal distribution. In the training phase, word embeddings along with the specific parameters of each model learn from the *training* set of large-scale Quora-QP corpus simultaneously.

We fine tune the hyperparameters of those models with the *dev* set of Quora-QP corpus, and report their results on the *test* set of Quora-QP corpus. For the core hyperparameters for the models we proposed, such as the dimension of word embeddings d, the number of filters k, and h kinds of grams (from unigram to h-gram), we give several trials: $d \in \{20, 50, 100, 150\}$, $k \in \{32, 64, 128\}$ and $h \in \{1, 2, 3\}$. The best performance of MGMNN and SLMNN on the *dev* set of Quora-QP occurs when $d = 100$, $k = 128$ and $h = 3$; and the best performance of MGANN and GSMNN on the *dev* set of Quora-QP occurs when $d = 50$, $k = 32$ and $h = 3$.

4.3 Results

Table 2 reports the experimental performance comparisons among all the neural network models for PI mentioned in this paper. The competitive approaches include ARC-I [11], HSCNN [24], Bi-CNN-MI [28], ABCNN [29], MGMNN, MGANN, SLMNN and GSMNN, as evaluated by the standard metrics (precision, recall, F1-score and accuracy) on the test set of Quora-QP. The results illustrate that GSMNN outperforms the state-of-the-art model ABCNN [29] in view of increasing the accuracy by **4.82%** and the F1-score by **4.27%** on the Quora-QP test set. Significant testing (t-test) on the improvements of GSMNN over ABCNN is also conducted on the

[6] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
[7] https://www.kaggle.com/c/quora-question-pairs
[8] Readers can download the Quora-QP corpus from https://drive.google.com/drive/folders/0BwgT_2kGBl2IVkRhLW96clExUnc?usp=sharing.

Table 2: Performance comparison of modern neural network models for paraphrase identification on Quora-QP corpus.

MODELS	PRECISION	RECALL	F1-SCORE	ACCURACY
ARC-I [11]	62.82%	70.96%	66.64%	72.28%
HSCNN [24]	63.71%	71.34%	67.31%	73.95%
Bi-CNN-MI [28]	61.62%	75.25%	67.76%	73.62%
ABCNN [29]	62.38%	73.97%	67.68%	73.98%
MGMNN	63.43%	73.88%	68.26%	74.69%
MGANN	67.47%	69.17%	68.31%	76.36%
SLMNN	68.05%	70.32%	69.16%	77.62%
GSMNN	70.17%	73.81%	**71.95**%*	**78.80**%*

Table 3: Performance comparison between concatenation-based (MGCNN and SLCNN) and matching-based neural network models (MGMNN, MGANN and SLMNN) on Quora-QP corpus.

MODELS	PRECISION	RECALL	F1-SCORE	ACCURACY
MGCNN	62.32%	73.27%	67.35%	73.62%
MGMNN	63.43%	73.88%	68.26%	74.69%
MGANN	67.47%	69.17%	**68.31**%*	**76.36**%*
SLCNN / ARC-I [11]	62.82%	70.96%	66.64%	72.28%
SLMNN	68.05%	70.32%	**69.16**%*	**77.62**%*

main metrics including F1-score and accuracy. We mark the entries with * if the improvements are significant (p-value < 0.05).

Additionally, our experimental results demonstrate that multi-granular matching sub-models, i.e., MGMNN and MGANN, exhibit better performance than ARC-I, HSCNN, Bi-CNN-MI and ABCNN. The sentence-level matching neural network (SLMNN) brings a leap forward as a component of GSMNN.

5 DISCUSSIONS

To further explore the essence of GSMNN and fully explain the reason why GSMNN has the potential to outperform competitive models, we set up extensive discussions in this section to answer the three questions as follows:

(1) Why do we prefer matching semantics rather than concatenating embeddings?
(2) Why do we apply the attention mechanism to the multi-granular matching layer?
(3) Why does GSMNN outperform ABCNN?

Each question above indicates the topic of its corresponding subsection. Besides more comparison results we are going to show in each subsection for quantitative analysis, qualitative analysis is conducted by visualizing those core neural layers of matching/concatenation for the pair of paraphrastic sentences, i.e., **P5** & **Q5**, which is randomly picked up from the test set of Quora-QP corpus. Both quantitative and qualitative analysis will help us better understand inner behaviors of all competitive neural models for PI.

P5: *What do Americans think about Donald Trump?*
Q5: *What do you think about Donald Trump pick?*

5.1 Why Matching? Concatenation vs. Matching

The single-layer CNN with multiple filters produces two kinds of semantics for a pair of sentences P and Q: multi-granular representations ($\mathbf{g}^P$ and $\mathbf{g}^Q$) and sentence-level embeddings ($\mathbf{s}^P$ and $\mathbf{s}^Q$). We prefer matching each pair of semantic features rather than concatenating them. The reason is explained by subsequent analysis.

Quantitative Analysis: To explore the differences in performance between matching and concatenation, we extensively conduct two groups of experiments on the Quora-QP corpus: MGCNN vs. MGANN/MGMNN, and SLCNN/ARC-I [11] vs. SLMNN. Table 3 reports the results of performance comparison, and it demonstrates that the approaches based on matching (MGANN/MGMNN and SLMNN) consistently outperform the methods based on concatenation (MGCNN and SLCNN/ARC-I). On average, the F1-score is increased by 1.74% and the accuracy is improved by 4.04% if the matching mechanism is adopted.

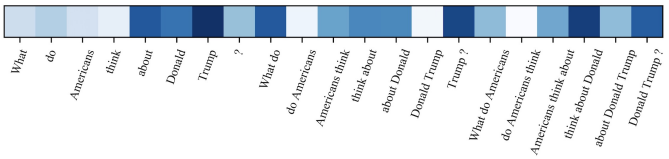
Qualitative Analysis: Fig. 2 visualizes the concatenation layer $\mathbf{x}^{MGC}$ of MGCNN for the pair of sentences **P5** & **Q5**, and Fig. 3 illustrates the semantic matching matrix $\mathbf{X}^{MGM}$ of MGMNN given the pair of sentences **P5** & **Q5**. An entry painted by dark blue indicates a significant feature. Both MGCNN and MGMNN concern the semantic granularities from unigrams to trigrams. However, MGCNN cannot capture synergistic features of paraphrasing. For example, Fig. 2(a) shows that the trigram *do Americans think* in **P5** almost does not support paraphrasing as a feature, but the trigram *do you think* in **Q5** is a strong indicator shown by Fig. 2(b). On the contrary, the features of semantic alignments within the semantic matching matrix $\mathbf{X}^{MGM}$ can be directly acquired by MGMNN. Fig. 3 shows that the matching feature aligned by *do Americans think* and *do you think* contributes much on identifying the paraphrase **P5** &

Table 4: Performance comparison between multi-granular matching neural network with attention (MGANN) and without attention (MGMNN) processed by different sizes of filters on Quora-QP corpus.

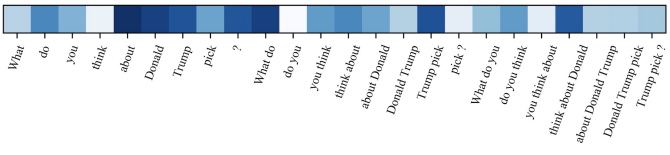
MODELS	PRECISION	RECALL	F1-SCORE	ACCURACY
MGMNN *(Unigrams)*	61.52%	71.85%	66.29%	73.08%
MGANN *(Unigrams)*	62.60%	73.29%	67.52%	74.03%
MGMNN *(Unigrams & Bigrams)*	62.92%	73.86%	67.95%	74.34%
MGANN *(Unigrams & Bigrams)*	63.87%	72.20%	67.78%	74.71%
MGMNN *(Unigrams & Bigrams & Trigrams)*	63.43%	73.88%	68.26%	74.69%
MGANN *(Unigrams& Bigrams & Trigrams)*	67.47%	69.17%	**68.31%***	**76.36%***

Table 5: Performance comparison between multi-granular matching neural network (MGMNN) and ABCNN processed by different sizes of filters on Quora-QP corpus.

MODELS	PRECISION	RECALL	F1-SCORE	ACCURACY
ABCNN *(Unigrams)*	60.79%	71.41%	65.67%	72.61%
MGMNN *(Unigrams)*	61.52%	71.85%	66.29%	73.08%
ABCNN *(Unigrams; Bigrams)*	62.13%	73.99%	67.54%	73.81%
MGMNN *(Unigrams & Bigrams)*	62.92%	73.86%	67.95%	74.34%
ABCNN *(Unigrams; Bigrams; Trigrams)*	62.38%	73.97%	67.68%	73.98%
MGMNN *(Unigrams & Bigrams & Trigrams)*	63.43%	73.88%	**68.26%***	**74.69%***



(a) Visualization of multi-granular embeddings for sentence **P5**.



(b) Visualization of multi-granular embeddings for sentence **Q5**.

Figure 2: Visualizations of multi-granular (unigrams & bigrams & trigrams) embeddings (to be concatenated) in MGCNN for P5 & Q5.

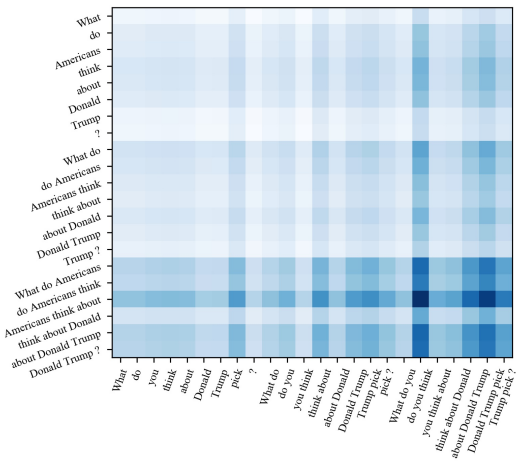


Figure 3: Visualization of the multi-granular matching matrix in MGMNN for paraphrastic sentences P5 & Q5.

Q5, as the entry which represents the matching feature shows the darkest color in the matrix $\mathbf{X}^{MGM}$.

Generally speaking, semantic matching approaches outperform concatenation-based methods regardless of the fed embeddings (multi-granular or sentence-level representations).

5.2 Why Attention? MGMNN vs. MGANN

ABCNN [29] is a pioneering study that applies the attention mechanism [1] to Bi-CNN-MI [28] and achieves the state-of-the-art performance on PI. Table 2 also demonstrates that ABCNN is comparable with Bi-CNN-MI measured by the F1-score and outperforms Bi-CNN-MI evaluated by accuracy on Quora-QP corpus. Though Yin et al. [29] conduct quantitative analysis that proves the effectiveness of the attention mechanism, we believe that qualitative analysis is also required by means of visualizing neural cells before and after applying the attention mechanism.

Quantitative Analysis: We conduct extensive experiments on comparing performance between MGMNN and MGANN (MGMNN with attention) on Quora-QP corpus. Table 4 shows that MGANN consistently outperforms MGMNN regardless of the kinds of granularities, i.e., unigrams, unigrams & bigrams, and unigrams & bigrams & trigrams. MGANN (unigrams & bigrams & trigrams) achieves the best performance compared with other competitive models. On average, the F1-score is increased by 0.37% and the accuracy is improved by 1.0% if MGANN takes place of MGMNN.

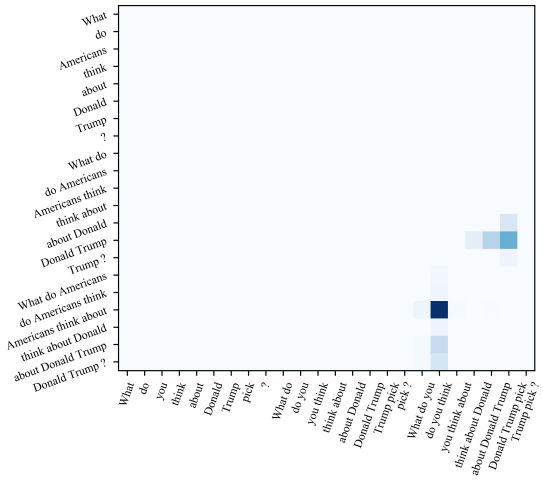


Figure 4: Visualization of the multi-granular matching matrix with attention in MGANN for sentences P5 & Q5.

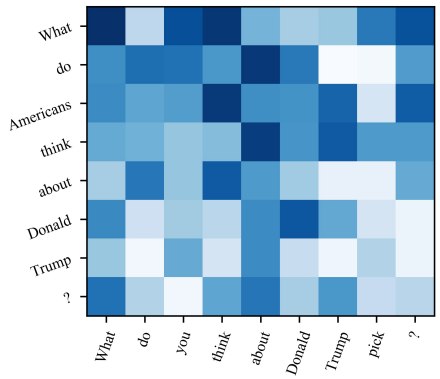
Qualitative Analysis: MGMNN has the capability of discovering multi-granular semantic matchings between a pair of sentences. In the example of **P5** & **Q5**, some significant semantic matchings such as *Americans think about* ↔ *do you think* and *Donald Trump ?* ↔ *Donald Trump pick*, are highlighted in the matching matrix $\mathbf{X}^{MGM}$ illustrated by Fig. 3. Though the features of semantic matching widely spread in the $\mathbf{X}^{MGM}$, most of them cannot be regarded as positive evidence for PI. The observation inspires us that the matching matrix should be sparse and concentrates on much fewer semantic alignments. Fig. 4 illustrates the multi-granular matching matrix after applying the attention mechanism $\mathbf{X}^{MGA}$ in MGANN for paraphrastic sentences **P5** and **Q5**. It demonstrates that the two most important semantic alignments (*Americans think about* ↔ *do you think* and *Donald Trump* ↔ *Donald Trump pick*) draw much more attention/values than the others, which makes binary classifiers easier to identify the pair of sentences **P5** & **Q5**.

In summary, the attention mechanism can help re-weight and focus on the most important semantic matchings, so that insignificant alignments are filtered out and the performance of neural models based on attention mechanism can be further improved.

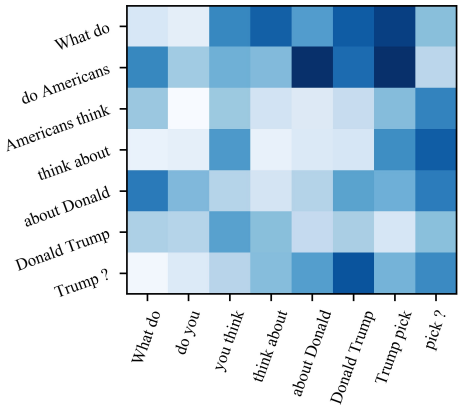
5.3 Why Multi-Granular? ABCNN vs. MGMNN

We argue that ABCNN can only discover the alignments of phrases with the same length. MGMNN, as a core component of GSMNN, has the advantage of capturing semantic matchings among all levels of granularity between a pair of paraphrastic sentences, i.e., *word-to-word*, *word-to-phrase*, *phrase-to-phrase*. Elaborate analyses are conducted as follows.

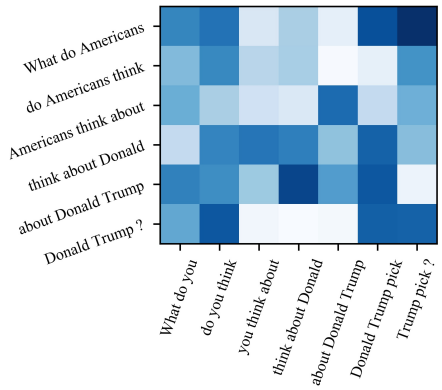
Quantitative Analysis: Extensive experiments are carried out to compare performance between MGCNN and ABCNN on Quora-QP corpus. Table 5 shows that MGMNN consistently outperforms ABCNN regardless of the kinds of granularities such as unigrams, bigrams and trigrams. MGMNN (Unigrams & Bigrams & Trigrams) achieves the best performance as it concerns the most semantic features. On average, the F1-score is increased by 0.54% and the accuracy is improved by 0.57%, if we use MGMNN instead of ABCNN.



(a) Attention matrix in ABCNN (unigrams).



(b) Attention matrix in ABCNN (bigrams).



(c) Attention matrix in ABCNN (trigrams).

Figure 5: Visualizations of the attention matrices in a 3-layer ABCNN for the unigrams, bigrams, and trigrams of paraphrastic sentences P5 & Q5.

Qualitative Analysis: Fig. 5 visualizes the attention matrices in the 3-layer ABCNN (unigrams/bigrams/trigrams). In particular, Fig. 5(c) demonstrates that ABCNN can discover the significant alignments of phrases with the same length for, such as *do Americans think* and *do you think*. However, ABCNN cannot capture

multi-granular matchings which MGMNN acquires, e.g., the alignment between *Americans think* and *do you think* as shown in Fig. 3.

The multi-granular matching neural network (MGMNN) generally shows higher performance than the state-of-the-art ABCNN especially when more *n*-grams are considered, because linguistic alignments of var-length phrases could be captured by MGMNN.

6 CONCLUSIONS

In this paper, we introduce the globalization-semantic matching neural network (**GSMNN**) for the task of paraphrase identification (PI). Compared to the state-of-the-art model ABCNN [29] which could discover linguistic alignments of phrases with the same length, GSMNN captures essentially all the phenomena of semantic matching such as *word-to-word*, *word-to-phrase*, *phrase-to-phrase*, and even *sentence-to-sentence*, to identify a pair of paraphrastic sentences. GSMNN accomplishes the goal with the help of multi-granular matching neural network with attention mechanism (MGANN) and sentence-level matching neural network (SLMNN). Extensive evaluations are conducted on a newly released large-scale dataset Quora-QP which contains more than 400 thousand pairs of paraphrastic and no-paraphrastic questions from Quora.com. Experimental results show that MGANN, SLMNN and GSMNN respectively increase the accuracy by **2.38%**, **4.64%**, **4.82%**, and F1-score by **0.63%**, **3.97%**, **4.27%**, compared with the state-of-the-art ABCNN model (accuracy: 73.98%, F1-score: 67.68%). It is worth mentioning that, although MGANN does not perform well as SLMNN, it makes semantic matching explainable.

Moreover, the paper has extensively discussed three questions, to help elaborate the advantages of our models, quantitatively and qualitatively. We conclude that: 1) semantic matching models outperform concatenation methods regardless of the type of fed embeddings (multi-granular or sentence-level representations); 2) the attention mechanism can help re-weight and focus on the most significant features for semantic matching, which makes binary classifiers much easier to identify paraphrases; 3) the multi-granular matching neural network (MGMNN) generally achieves better performance than the state-of-the-art ABCNN especially when more *n*-grams (unigrams, bigrams, and trigrams) are considered.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Alberto Barrón-Cedeño, Marta Vila, M Antònia Martí, and Paolo Rosso. 2013. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics* 39, 4 (2013), 917–947.

[3] Regina Barzilay and Kathleen R McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 50–57.

[4] Yoshua Bengio et al. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.

[5] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*. 737–744.

[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.

[7] Jianfeng Gao, Patrick Pantel, Michael Gamon, Xiaodong He, and Li Deng. 2014. Modeling Interestingness with Deep Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*. 2–13.

[8] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[9] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28*. JMLR. org, III–1319.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780.

[11] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *NIPS*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 2042–2050.

[12] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 655–665.

[13] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751.

[14] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, 177–180.

[15] Zornitsa Kozareva and Andrés Montoyo. 2006. Paraphrase identification on the basis of supervised machine learning techniques. In *Advances in natural language processing*. Springer, 524–533.

[16] Yann LeCun and Yoshua Bengio. 1998. The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA, USA, Chapter Convolutional Networks for Images, Speech, and Time Series, 255–258.

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[18] Simone Magnolini. 2014. A Survey on Paraphrase Recognition.. In *DWAI@ AI* IA*. 33–41.

[19] Inderjeet Mani. 1999. *Advances in automatic text summarization*. MIT press.

[20] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, Vol. 6. 775–780.

[21] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 Task 3: Community Question Answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 27–48.

[22] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree. 2015. SemEval-2015 Task 3: Answer Selection in Community Question Answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, 269–281.

[23] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. SemEval-2016 Task 3: Community Question Answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, 525–545.

[24] Massimo Nicosia and Alessandro Moschitti. 2017. Accurate Sentence Matching with Hybrid Siamese Networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 2235–2238.

[25] Alan Parker and James O Hamblen. 1989. Computer algorithms for plagiarism detection. *IEEE Transactions on Education* 32, 2 (1989), 94–99.

[26] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*. 801–809.

[27] Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the âĂIJpara-farceâĂİ out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop*, Vol. 2006.

[28] Wenpeng Yin and Hinrich Schütze. 2015. Convolutional Neural Network for Paraphrase Identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, 901–911.

[29] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.

[30] Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820* (2015).

[31] Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating Machine Translation Results with Paraphrase Support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 77–84.