Accurate Sentence Matching with Hybrid Siamese Networks

Massimo Nicosia[◇] and Alessandro Moschitti
[◇]DISI, University of Trento, 38123 Povo (TN), Italy
Qatar Computing Research Institute, HBKU, 5825, Doha, Qatar
{m.nicosia,amoschitti}@gmail.com

ABSTRACT

Recent neural network approaches to sentence matching compute the probability of two sentences being similar by minimizing a logistic loss. In this paper, we learn sentence representations by means of a siamese network, which: (i) uses encoders that share parameters; and (ii) enables the comparison between two sentences in terms of their euclidean distance, by minimizing a contrastive loss. Moreover, we add a multilayer perceptron in the architecture to simultaneously optimize the contrastive and the logistic losses. This way, our network can exploit a more informative feedback, given by the logistic loss, which is also quantified by the distance that the two sentences have according to their representation in the euclidean space. We show that jointly minimizing the two losses yields higher accuracy than minimizing them independently. We verify this finding by evaluating several baseline architectures in two sentence matching tasks: question paraphrasing and textual entailment recognition. Our network approaches the state of the art, while being much simpler and faster to train, and with less parameters than its competitors.

CCS CONCEPTS

• **Computing methodologies → Natural language processing**; • **Information systems →** *Similarity measures*; • **Computer systems organization →** *Neural networks*;

1 INTRODUCTION

Sentence matching is an important problem in Information Retrieval (IR), typically tackled in question answering (QA), e.g., [18], and more recently in community QA (cQA), e.g., [12, 21] as also shown by a recent workshop on semantic matching [8]. More in general, the sentence matching technology can be useful for many tasks such as semantic textual similarity [1], textual entailment [2], and answer sentence selection [17], to name a few.

Recently, the problem has been approached with neural network models, e.g., convolutional neural networks (CNN) [20] or Long-Short Memory Networks (LSTM) [6], for passage reranking. Such methods firstly build a representation for each sentence, and then model their similarity with a feed-forward network. It should be noted that the sentence matching constraint is only enforced in the last layer, where the network output is compared with the gold label for computing the logistic loss. However, this clearly happens too late in the network, causing some of the properties of the sentence representation being neglected for learning the overall similarity.

In contrast, Siamese Networks [3, 4] learn a distance metric between two sentences by mapping them into an interpretable geometric space. For this purpose, they apply the so-called contrastive loss, based on the distance between two sentences, to optimize the matching task. The main drawback is that their encoder produces representations that are independent from each other, i.e., the generation of one representation is not conditioned on the other.

In this paper, we explore the use of siamese mechanisms in a neural network with additional layers that model the interaction between two sentence representations. More specifically, we use a siamese formulation to learn sentence encoders with shared parameters, and we enable the comparison between two encoded sentences in terms of their euclidean distance. In addition, we add a multilayer perceptron in the architecture such that both the contrastive and logistic losses are simultaneously optimized. Therefore, our network uses (i) the contrastive loss to learn suitable sentence representation based on their euclidean distance; and (ii) the informative feedback given by the logistic loss in the multilayer perceptron to capture interdependencies between the representations of the two sentences.

We carried out comparative experiments against several state-of-the-art networks on the Quora [1] dataset, for the question duplication task, and on the Stanford Natural Language Inference (SNLI) dataset [2], for the entailment task. The results show that our network, also thanks to the optimization of our joint contrastive and logistic loss, approaches the the state of the art. It should be noted that the latter is achieved by much more complex architectures, which are more expensive in terms of the number of parameters, and in the computational time for training and testing.

2 RELATED WORK

A basic network for sentence matching encodes each sentence into a vector, concatenates and passes them trough a number of hidden layers to make a prediction. Apart from this basic architecture, we can define three more sophisticated network types: Siamese, Attentive, and Compare-Aggregate networks.

Siamese networks have been applied in a subset of NLP tasks such as textual similarity, paraphrase identification and mention normalization. MaLSTM [13] learns the Manhattan distance between two sentences encoded with an LSTM. The network is trained on a similarity task, then the encoder parameters are fixed such that it is used as feature extractor. Encoded sentences are fed into

CIKM'17 , November 6–10, 2017, Singapore, Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133156

[1] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

an SVM classifier to learn a textual entailment task, i.e., the siamese representations is not learned end-to-end for the second task.

The SCQA network [7] exploits question and answer pairs in a cQA archive to learn question-question similarity. Questions and answers are mapped into the same space by a convolutional siamese network. Those pairs increase the available training data with the assumption that answers share similarity with the questions. The network performs better on the question-question similarity task, where input sentences are represented as bags of trigrams. In addition, the output of the network is combined with the BM25 score of the questions in the pair in order to boost the textual similarity.

A two-layer bidirectional LSTM network on characters was used in [14] to normalize job titles. This siamese network was able to capture semantic differences while being invariant to non-semantic string differences. The aformentioned models establish a match between sentences in terms of their distance. In our case, we also model the interactions between the sentence representations learned by the siamese encoder, drawing some analogies with **?**].

Attentive Networks [15**?**] build a representation of a sentence in a pair by also considering the other sentence, weighting the contribution of its parts with the so-called attention mechanism. The computation has a relatively high cost, i.e., quadratic complexity.

Compare-Aggregate Networks [22] decompose two input sentences in parts, which are matched through several similarity functions. The results are then aggregated to quantify the final match. We compare our work with the state-of-the-art Bilateral Multi Perspective Matching (BiMPM) model [23].

The use of auxiliary losses is at the core of multitask learning. A network can be trained to perform multiple tasks, by stacking multiple softmax layers on top of the last hidden layer. For example, a part-of-speech tagger can be trained to predict the tag and the binned log frequency of the next token [16]. Alternatively, a network can perform a task at a lower architectural level, and a different task at a higher level. Stack-propagation [25] uses this method to learn part-of-speech tags at a lower level, and dependecy relations at higher level. In our network, we do not learn different tasks, but we employ different losses to extract different aspects of the input: the semantic traits of the sentence, which are captured by the distance of their representations in a geometric space, and the interactions between those representations modeled by a classifier.

3 SENTENCE MATCHING USING OUR HYBRID SIAMESE NETWORK

In this section, we present our approach to sentence matching. We define the siamese network encoder for representing sentences, and the multilayer perceptron that models their interaction.

3.1 Model architecture

The first module of our deep learning model is the sentence encoder. A sentence of length n is a sequence of words $(w_1, ..., w_n)$, which are drawn from a vocabulary V. Each word is represented as a vector, $\mathbf{w} \in \mathbb{R}^d$, looked up into an embedding matrix, $\mathbf{E} \in \mathbb{R}^{d \times |V|}$. The sentence encoder f takes a sequence of words in input, embeds and transforms them into a fixed-sized vector. The function f is used to encode both sentences in a pair, by sharing the same set of weights. This is typical of a siamese setting, where the same network maps the two objects of a pair into a low dimensional space, where their distance is small if they are similar. We compute the euclidean distance of two sentences s_1 and s_2 as:

$$d(s_1, s_2) = \sqrt{\sum_{i=1}^{n} (f(s_1)_i - f(s_2)_i)^2} \tag{1}$$

Our full hybrid siamese network furtherly models the interaction between the two sentences by feeding the output of the siamese encoder to a multilayer perceptron (MLP). The MLP takes the concatenation of the sentence representations and their distance, $c = [f(s_1); f(s_2); d(s_1, s_2)]$, in input, and outputs the probability of a match between the two sentences.

3.2 Joint loss optimization

The siamese network encoder is learned by optimizing a contrastive loss, computed over the pairs of sentences in the dataset. This loss compares the distance between two representations produced by the siamese encoder with the true label. Given the dataset $X = \left\{ \langle s_1^i, s_2^i, y^i \rangle \right\}$, where y is 1 if the two sentences match, and 0 otherwise, the total loss is:

$$\mathcal{L}_c = \sum_{i=1}^{N} L_c^i(s_1^i, s_2^i, y^i), \tag{2}$$

where L_c^i is the contrastive loss for an instance defined as:

$$L_c^i(s_1^i, s_2^i, y^i) = y^i d(s_1^i, s_2^i)^2 + (1 - y^i) max(M - d(s_1^i, s_2^i), 0)^2 \tag{3}$$

The loss for matching sentences is the square of their euclidean distance, while there is a loss in the opposite case, only when the distance between sentences that do not match is smaller than the margin M. In that case, the loss is equal to the square difference between the margin and the distance.

Our final network optimizes the logistic loss computed on the output of the MLP with respect to the true labels:

$$\mathcal{L}_l = \sum_{i=1}^{N} y^i log(\tilde{y}^i) + (1 - y^i) log(1 - \tilde{y}^i) \tag{4}$$

The global loss of our full network is thus the sum of the two losses:

$$\mathcal{L}_{tot} = \lambda_c \mathcal{L}_c + \mathcal{L}_l, \tag{5}$$

where λ_c is an hyperparameter to tune the effect of the contrastive loss during parameter learning.

4 EXPERIMENTS

We evaluate our model on two tasks: identification of question paraphrases in a cQA setting and textual entailment recognition.

4.1 Sentence encoders and the MLP

We experiment with different network architectures for our sentence encoding function f: (i) an LSTM network [9], with 200 units; (ii) a Gated Recurrent Unit (GRU) network [5], with 200 units; (iii) a CNN network [11], with 3 groups of 100 convolutional filters of size 1, 3 and 5; (iv) a stack of 2 Bidirectional GRU (BiGRU) networks [19], with GRUs of 200 units. The LSTM and GRU consume the input from left to right, and their last state is used to encode the sentence.

The stacked BiGRU network has higher capacity and should better capture longer dependencies in a sentence. The BiGRU consumes token or states in both directions, producing a state for each input. Those states are fed to the upper BiGRU layer. The maximum values across the dimensions of the output states are selected, producing a sentence vector with 400 dimensions. We tuned the number of stacked layers on the development set.

The MLP is composed of two hidden layers of size 200 and takes the sentences encoded by the siamese network together with their euclidean distance in input.

We carry out three different experiments for each architecture, minimizing: (i) the contrastive loss only, (ii) the logistic loss only, (iii) the sum of the contrastive and logistic loss.

4.2 Network training

We summarize here the network settings common to all the experiments. We initialize word embeddings with pretrained GloVe vectors of size 300. Sentences are truncated or padded to 50 words. Words without a pretrained vector are initialized with a random vector sampled from the uniform distribution $U[-0.25, 0.25]$. Word embeddings are fine-tuned for the task by updating them together with the network parameters.

The network is trained with the Adam optimizer [10], setting learning rate to .001. The contrastive loss margin M is set to 1. λ_c is tuned on the validation set: it is set to 1.0 for the paraphrase identification models, and to 0.01 for the textual entailment (TE) models. The selection of smaller λ_c value for TE may be due to the fact that task is not strictly symmetric. Training examples are fed to the network in shuffled mini-batches of size 128. All the models are trained for 20 epochs and the reported test accuracy corresponds to the best accuracy obtained on the validation set. In the contrastive only setting, labels are obtained by thresholding the distances at 0.5 (value selected from the dev. set).

4.3 Paraphrase identification

Dataset. The Quora dataset contains pairs of questions from the Quora web site. Each pair is labelled as positive if the two questions ask for the same thing, and negative otherwise. For the evaluation, we use the dataset splits and word embeddings [2] provided by Wang et al. [23]. Their training split contains 384,348 pairs, and the balanced development and test sets contain 10,000 pairs each. The embeddings are a subset of the 300-dimensional GloVe word vectors pretrained on the Common Crawl corpus, [3] covering the Quora dataset vocabulary. Regarding data preprocessing, the sentences are already tokenized; we only lowercase the questions before splitting on whitespaces.

Results. Table 1 shows the results of our models for the Quora paraphrase identification task. Each model is trained using (i) the contrastive loss on the euclidean distance of the sentence representations from the siamese encoder; (ii) the logistic loss on the prediction of an MLP applied on the concatenation of the sentence representations; (iii) and both losses, considering the MLP prediction as the final matching score. The last setup yields consistently higher accuracy with respect to separate loss minimization.

Siamese Encoder	Contrastive	Logistic	Joint loss
LSTM	85.26	81.98	85.71
GRU	86.72	83.00	**86.82**
CNN	83.1	83.04	83.95
Stacked BiGRU	85.74	84.38	86.06

Table 1: Test accuracies of our models for the paraphrase identification task (Quora), trained with the different losses.

Siamese Encoder	Contrastive	Logistic	Joint loss
LSTM	84.58	88.25	89.08
GRU	84.96	88.53	90.58
CNN	80.78	90.99	**92.00**
Stacked BiGRU	85.58	91.40	91.97

Table 2: Test accuracies of our models for the textual entailment task (SNLI), trained with the different losses.

4.4 Recognizing textual entailment

Dataset. For this task, we use the SNLI dataset [2]. It contains 570,000 premise/hypothesis pairs, which are labelled by humans. The labels are *entailment*, *contradiction* and *neutral*. We use the official training, development and test partitions to respectively train, validate and test our models. We keep only examples from the *entailment* and *contradiction* classes, experimenting with the binary task of understanding if the meaning of the hypothesis can be inferred from the premise or not. The resulting training, development and test partitions have respectively 366,603/9,842/9,824 examples, respectively. We use GloVe vectors again, but we extract them from the distribution available on the GloVe project web site. Our processing applied to the sentence is minimal, we just tokenize them using SpaCy [4] and apply lowercasing.

Results. Table 2 shows the performance of our models for the binary textual entailment task. We evaluated the models on this additional semantic task to corroborate the advantage of using the joint loss: the improvement is consistent. The lower accuracy in the contrastive loss setting may be due to the asymmetry of the task.

4.5 Discussion

The results in Table 1 for paraphrase identification shows that minimizing the joint loss is the better strategy for all the architectures used in the siamese sentence encoder.

Table 3 offers a comparison of the models above along with BiMPM [23]. The authors implement siamese and multi perspective CNN and LSTM networks. Our models outperform their corresponding siamese baselines even when we do not use the joint losses, and, when we do, the improvement becomes more substantial. All the models use the same data and word embeddings, thus our improvement on siamese models just comes from a better hyperparameter tuning and some architectural differences: we use 200 recurrent units instead of 100, we learn the euclidean distance between representations instead of the cosine distance, and we do not use character embeddings to learn word vectors.

Except for the CNN model, our other architectures are comparable or outperform the L.D.C. model [24], even in the single

[2] https://github.com/zhiguowang/BiMPM

[3] https://nlp.stanford.edu/projects/glove/

[4] https://spacy.io/

Model	Accuracy
Siamese-CNN [23]	79.60
Multi-Perspective-CNN [23]	81.38
Siamese-LSTM [23]	82.58
Multi-Perspective-LSTM [23]	83.21
L.D.C. [23]	85.55
BiMPM [23]	88.17
Hybrid Siamese LSTM	85.71
Hybrid Siamese GRU	86.82
Hybrid Siamese CNN	83.95
Hybrid Siamese Stacked BiGRU	86.06

Table 3: Comparison between the accuracies of the models in Wang et al. [23] and our models (trained with the joint loss), for the paraphrase identification task (Quora).

contrastive loss optimization case. L.D.C. belongs to the compare-aggregate approach class, and is very accurate on the answer selection task. It is possible that the CNN architecture can reach higher accuracy with further hyper-parameter tuning. Our best model, the hybrid siamese GRU network achieves a test accuracy of 86.82%, i.e., only 1.35% less than the state-of-the-art BiMPM model. We stress the fact that the latter is much more complex: the two sentences in a pair are encoded with a bidirectional recurrent network and every pair of encoded words from the two sentences are matched using multiple functions. One of the latter applies an expensive attention mechanism. For all these reasons, this model (i) is one order of magnitude slower to train than ours: 100 vs. 10 seconds per 100 training steps on a Tesla K40m GPU; (ii) it has many more trainable parameters: 6.5M vs. 300k, without considering word embeddings.

The experiments on textual entailment (Table 2) also show a consistent improvement when using the joint loss. The contrastive loss incentives the sentence encoder to produce representations that can be semantically separated in a geometric space. We can think the loss as a regularization mechanism imposing a structure on the intermediate network representations. Then, the MLP can refine the final network output both in terms of the interaction between the sentences and their distance. This way, it may be able to make a better choice for sentences that are mistakenly put close or far apart in the geometric space.

5 CONCLUSION

We presented a hybrid siamese and MLP network optimized using a joint loss. This is the sum of (i) the contrastive loss used to optimize the sentence representations produced by a siamese sentence encoder; and (ii) the logistic loss used to optimize an MLP that models the interaction between the sentence representations. The contrastive loss introduces an additional level of supervision when learning the sentence encodings, mapping them into an euclidean space. The joint loss consistently benefits the performance of our models across the different network architectures selected for the sentence encoder. This positive effect allows us to keep the last part of our network, i.e., the MLP, simple, and avoid computationally expensive matching and attention mechanisms. The resulting networks are competitive with state-of-the-art models, with lower complexity and number of parameters.

ACKNOWLEDGMENTS

This work has been partially supported by the EC project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action). The first author was supported by the Google Europe Doctoral Fellowship Award 2015. Many thanks to the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proc. of SemEval-2016.*

[2] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proc. of EMNLP.*

[3] Jane Bromley, Isabelle Guyon, Yann Lecun, Eduard SÃďckinger, and Roopak Shah. 1994. Signature Verification using a "Siamese" Time Delay Neural Network. In *Proc. of NIPS.*

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of CVPR*, Vol. 1. IEEE.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] Daniel Cohen and W. Bruce Croft. 2016. End to End Long Short Term Memory Networks for Non-Factoid Question Answering. In *Proc. of ICTIR.*

[7] Arpita Das, Harish Yenala, Manoj Chinnakotla, and Manish Shrivastava. 2016. Together we stand: Siamese Networks for Similar Question Retrieval. In *Proc. of ACL.*

[8] Julio Gonzalo, Hang Li, Alessandro Moschitti, and Jun Xu. 2014. SIGIR 2014 Workshop on Semantic Matching in Information Retrieval. In *Proc. of SIGIR.*

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[10] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR.*

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems.*

[12] Yandong Liu and Eugene Agichtein. 2008. On the Evolution of the Yahoo! Answers QA Community. In *Proc. of SIGIR.*

[13] Jonas Mueller and Aditya Thyagarajan. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity. In *Proc. of AAAI.*

[14] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP.*

[15] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proc. of EMNLP.*

[16] Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In *Proc. of ACL.*

[17] Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-Contrastive Estimation for Answer Selection with Deep Neural Networks. In *Proc. of CIKM.*

[18] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a Question Answering System. In *Proc. of ACL.*

[19] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997).

[20] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proc. of SIGIR.*

[21] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. 2009. A Syntactic Tree Matching Approach to Finding Similar Questions in Community-based QA Services. In *Proc. of SIGIR.*

[22] Shuohang Wang and Jing Jiang. 2017. A Compare-Aggregate Model for Matching Text Sequences. In *Proc. of ICLR.*

[23] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral Multi-Perspective Matching for Natural Language Sentences. In *Proceedings of IJCAI.*

[24] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Sentence Similarity Learning by Lexical Decomposition and Composition. In *Proc. of COLING.*

[25] Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved Representation Learning for Syntax. In *Proc. of ACL.*