Supervised Mover's Distance: A Simple Model for Sentence Comparison

Muktabh Mayank Srivastava(✉)

ParallelDots, Inc., Gurugram, India
muktabh@paralleldots.com
https://paralleldots.xyz

Abstract. We propose a simple neural network model which can learn relation between sentences by passing their representations obtained from Long Short Term Memory (LSTM) through a Relation Network. The Relation Network module tries to extract similarity between multiple contextual representations obtained from LSTM. The aim is to build a model which is simple to implement, light in terms of parameters and works across multiple supervised sentence comparison tasks. We show good results for the model on two sentence comparison datasets.

Keywords: Supervised Mover's Distance · Sentence comparison
Paraphrase detection · Natural language inference

1 Introduction

Sentence Comparison is a common NLP task which comes up in multiple domains. Sentence comparison measure might be needed to check redundant data [6] or check sentences for being paraphrases [3]. We propose a new method to compare sentences for both these tasks, which uses Relation Networks (RN) module [11] in combination with a Long Short Term Memory (LSTM) [4]. To compare two sentences, all possible pairs of dense vectors, one from each sentence in a pair, are passed through a Relation Network module to decipher relationship information between sentences. To make sure the dense vectors passed to Relation Network have contextual information, sentences are individually passed through a LSTM and the hidden units obtained for each word of a sentence are used as dense vectors. The inspiration of the model comes from Earth Mover's Distance (EMD) [10] which can be used to calculate distances between two distributions of points represented by vectors by optimal weighted comparison of points pairwise. The assumption is that LSTM can generate contextual vectors which can be then fed pairwise to RN to determine similarity. This is the reason for referring the algorithm as Supervised Mover's Distance, however, the algorithm does not solve optimal transport like EMD.

D. Ustalov et al. (Eds.): AINL 2018, CCIS 930, pp. 61–66, 2018.
https://doi.org/10.1007/978-3-030-01204-5_6

2 Previous Work

In our experiments, we focus on two sentence comparison tasks: 1. Duplication detection between questions [6] and 2. Paraphrase detection [3]. Duplication detection task aims to check whether two questions intend to ask about the same topic. Paraphrase Detection task aims to classify sentences according to whether they have a paraphrase/semantic equivalence relationship. Deep Neural Networks have shown state of the art performance in sentence comparison tasks. Most top methods for paraphrase detection are based on Deep Neural Networks [1,2]. BiMPM model [12] combines a custom matching layer with LSTMs [4] for question duplication detection.

Relation Networks (RN) [11] was introduced as a simple module for relational reasoning. The module has been used for spatial relational reasoning in images earlier, but we try to use it for deciphering relationships in text by combining it with an LSTM. RNs operate on a set of objects without regard to the objects' order, so we use LSTMs to extract out temporal information containing word importances and use RNs on top for reasoning. RN module has a g-layer which models relation between all possible pairs of objects and a f-layer which models the final output looking at the relation between objects.

Another set of models which use pairwise relationships to model document similarity are Word Mover's Distance (WMD) [8] and its supervised variant (SWMD) [5]. They both are methods to calculate Earth Mover's Distance (EMD) [10] between documents for document calculation. Both these methods calculate flows (weightages) to be given distances between each possible pair of words to calculate document distance. WMD is an unsupervised distance measure between documents. The SWMD architecture works on longer documents (with more than 40 words) and uses a complex optimization procedure to optimize EMD. SWMD uses a cascaded loss where the inner loss optimizes word importance and outer loss optimizes EMD flow. Our method is inspired from WMD and SWMD algorithms as it takes pairwise modelling of words into account but tries to achieve it using a single RN module. However, our method is not trying to solve optimal transport problem, but is trying to use contextual vectors derived from LSTM in pairs as input to RN module to model similarity.

3 Method

As Supervised Mover's Distance, we propose a baseline that generalizes well across different tasks. Our network combines LSTM layers [4] with a RN [11] module modeling semantic relationship between the sentences. The neural network architecture we propose is trained on pair of sentences to predict one of various classes the pair might fall into. For redundancy detection and paraphrase detection the labels are positive or negative, but might be different for any other tasks. The architecture has two basic parts: 1. LSTM layers and 2. RN layer. The LSTM layers can have depth of one or higher which take both sentences as input individually and produce hidden layers as output for each of the words in the sentences. This would yield two series of output hidden states, one hidden state for each time step of each sentence. To clarify again, there is one common LSTM which runs on both sentences separately to produce respective hidden states. In the RN, all possible pairs of hidden states across both sentences are taken as concatenated vectors and passed through a fully connected (or Dense) layer. Aforementioned fully connected layer is the g-layer of the RN. This yields an embedding for each possible pair of hidden state outputs from the LSTM. These embeddings are averaged and passed through another fully connected layer to predict the output. This fully connected layer is the f-layer of the RN. By taking all pairs of hidden states and using them to model sentence comparison task, we hypothesize that the LSTM would be able to make contextual vectors and RN can model pairwise differences to understand relationships between two sentences.

We illustrate the architecture part-by-part in upcoming Figs. 1, 2 and 3. Lets say two sentences s1 and s2 are to be compared. The sentence s1 is processed by a LSTM as shown in Fig. 1. The same LSTM processes s2 to get its hidden states as in Fig. 2. Now hidden states obtained from both s1 and s2 through LSTM are grouped into pairs (each pair has one hidden state from s1 and another from s2) and classified into possible classes using RN. The RN is potrayed in Fig. 3. Please note that although LSTM and RN are depicted in different diagrams for explanation, both the modules are part of the same neural network architecture and are backpropogated together. Our model is light in terms of parameters as it has only a LSTM layer and two dense (fully connected) layers in RN. A limiting case of the architecture can be when the number of LSTM layers is zero, and word embeddings are passed as inputs directly to RN.

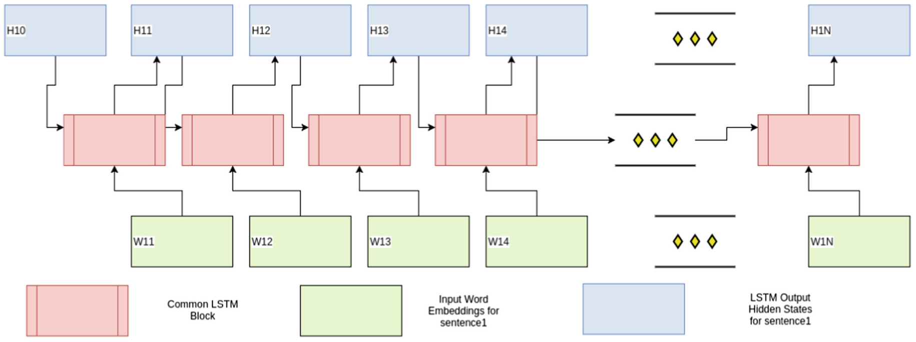
H10
H11
H12
H13
H14
H1N
W11
W12
W13
W14
W1N
Common LSTM Block
Input Word Embeddings for sentence1
LSTM Output Hidden States for sentence1

Fig. 1. Embeddings from sentence 1 are passed through an LSTM and its hidden states are taken corresponding to each input word

The network is trained with common hyperparameters for both the tasks. Pretrained word embeddings are used to initialize the word embedding layer which are finetuned by backpropogation. We use the publicly available 6 Billion token 100 dimensional version of GloVe embeddings [9]. The hidden state output from the LSTM is 100 dimensions and the size of embedding generated in the relational layers is 100 dimensions too. The network is trained with simple Stochastic Gradient Descent with momentum (common values for training across both datasets, learning rate = 0.001, momentum = 0.9).

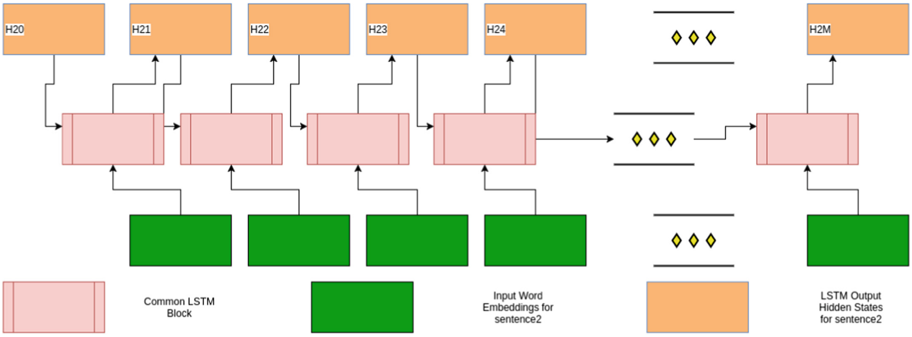
H20
H21
H22
H23
H24
H2M
Common LSTM Block
Input Word Embeddings for sentence2
LSTM Output Hidden States for sentence2

Fig. 2. The same LSTM is used to process sentence 2 and obtain each of its hidden states corresponding to its input words

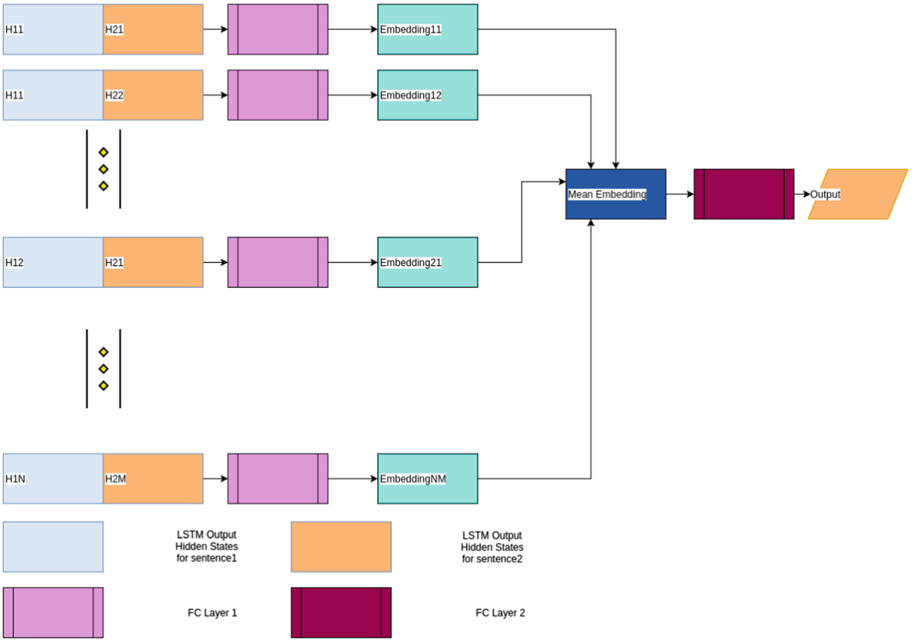

Fig. 3. A Relation Network (RN) is used to process all possible pairs of hidden states (one from both sentences which need to be compared)

4 Results

As stated we test our model on two datasets. Model is compared to state of the art methods and baselines for each dataset in this section.

Microsoft Research Paraphrase Corpus. Microsoft paraphrase corpus [3] is a corpus of sentence pairs classified as paraphrases or non-paraphrases. The dataset has 4076 sentences in training set and 1725 sentences in test set. Our model was trained on the training set with the standard set of hyper parameters mentioned above and evaluated on the test set. The accuracy numbers of different models were taken from this url[1]. Our model gets an accuracy of 80.2% on the dataset as compared to state of the art accuracy of 80.4% [7].

Quora Questions' Pair Dataset. Quora Questions' Pair Dataset contains question pairs from the Q&A website[2] tagged as similar or not. A random 90%–10% train-test split is performed as is customary for other methods and the model is trained on the train set and evaluated on the test set. As in case of other datasets, the hyperparameters are fixed as the standard values specified earlier while training. Our model gets an accuracy of 81.2% on the dataset. List of state of the art models on the dataset is available on this url[3]. The best accuracy a model gets on the dataset is 88% [12]. Although our model doesn't get results as good as the state of the art, it is competitive to baselines like siamese Convolutional Neural Networks (79.6%) and siamese LSTMs (82.58%).

It should be noted that in both models, dataset specific hyperparameter tuning was not performed.

5 Conclusion and Future Work

We propose a new method which uses a new and simple neural network model to compare sentences. Our method combines Long Short Term Memory (LSTM) and Relation Network (RN) module to model relationship between the sentences. LSTMs generate contextual hidden state vectors and RN module models sentence relationship. Models performance is calculated on two sentence comparison datasets. In future work, we will incorporate trainable components in our method to determine importance of each component as the current RN takes a mean over all vectors, treating each of the components with equal weightage.

[1] https://aclweb.org/aclwiki/Paraphrase_Identification_(State_of_the_art).

[2] quora.com.

[3] https://github.com/bradleypallen/keras-quora-question-pairs.

References

1. Cheng, J., Kartsaklis, D.: Syntax-aware multi-sense word embeddings for deep compositional models of meaning. arXiv preprint arXiv:1508.02354 (2015)
2. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1034–1046. Association for Computational Linguistics (2011)
3. Dolan, B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Third International Workshop on Paraphrasing (IWP2005). Asia Federation of Natural Language Processing, January 2005. https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
5. Huang, G., Guo, C., Kusner, M.J., Sun, Y., Sha, F., Weinberger, K.Q.: Supervised word mover's distance. In: Advances in Neural Information Processing Systems, pp. 4862–4870 (2016)
6. Iyar, S., Dandekar, N., Csernai, K.: First quora dataset release: question pairs, January 2016. https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs
7. Ji, Y., Eisenstein, J.: Discriminative improvements to distributional sentence similarity. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 891–896 (2013)
8. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML 2015, vol. 37, pp. 957–966. JMLR.org (2015). http://dl.acm.org/citation.cfm?id=3045118.3045221
9. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). http://www.aclweb.org/anthology/D14-1162
10. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision, pp. 59–66. IEEE (1998)
11. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Advances in Neural Information Processing Systems, pp. 4974–4983 (2017)
12. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. arXiv preprint arXiv:1702.03814 (2017)