Siamese Networks:

Learning a Neural Representation of Semantic Similarity

Trevor McInroe

MSDS-458

August 9, 2020

Abstract

Measuring the semantic similarity between documents is a difficult problem in the domain of natural language that has many useful, practical applications such as information retrieval in search engines. In the past, closed-form solutions have been applied to the problem with some success. The rise of deep learning methods shines new light on this problem by providing the ability to develop learned neural representations of similarity. In this work, we explore the Siamese network architecture's capability of learning the semantic similarity between sentence pairs. We test this architecture with a variety of data augmentation techniques with the hope of encouraging greater generalization on unseen data. We find that Siamese networks are a natural fit for the problem of semantic similarity but that data augmentation techniques on text data may actually harm the performance of the model. These results encourage the exploration of learned neural representations of semantic similarity.

Background and Related Works

The problem of measuring semantic similarity is an important task in natural language processing. It involves measuring the likeness of two or more documents in terms of their meanings, not simply the overlap in terms. Semantic similarity has many use cases, such as in information retrieval engines where queries are compared to documents or in mapping nodes in a graph-like text-based ontology (Hliaoutakis et al. 2006; Gan, Dou, and Jiang 2013).

In general, there is a dichotomy of approaches to measuring semantic similarity: (a) the computation of a static metric or (b) a learned neural representation. In the domain of natural language, (a) is usually performed with a modified version of cosine similarity, $cos(\theta) = 1 - \frac{ab}{||a||_2\,||b||_2}$, that determines if two vectors, a and b, point in a similar direction. (b) involves the iterative training of a neural network that is learning to minimize a given loss. Developing a neural representation of semantic similarity requires a non-standard network architecture. In the vanilla case, a neural network is meant to learn to map a single input to a single output. In the case of semantic similarity, the network must learn to map two paired-inputs to a single output that represents a comparison between the inputs. To approach this problem, we have used the so-called *Siamese network* architecture, proposed by Chopra, Hadsell, and LeCun (2005).

Siamese networks have mainly been used in the domains of natural language or computer vision (Zhu et al. 2018b; Li, Bilodeau, and Bouachir 2018; Zhu et al. 2018a; Kamineni et al. 2018). A Siamese network is made of two neural networks with mirrored architectures that share weights. The networks learn to output encoded versions of their inputs which are ultimately passed through a distance measure that concatenates them into a single value. The most common distance measure for Siamese networks is Manhattan distance, or the ℓ_1 norm of the difference between the two vectors (Mueller and Thyagarajan 2016). To constrain the output of the network, the result of the ℓ_1 norm is made negative and then passed through an exponential function:

$$\exp(-||x_1 - x_2||_1)$$

where x_1 and x_2 are the outputs from the left and right network, respectively. The intuition here is that the mirrored networks will learn to output vectors whose difference produces a very small ℓ_1 norm for semantically similar sentences. This small ℓ_1 norm, when made negative and exponentiated, will produce a number very close to one.

In the domain of natural language, Siamese networks are usually either made of Long Short Term Memory (LSTM) cells (Hochreiter and Schmidhuber 1997) or Gated Recurrent Unit (GRU) cells (Cho et al. 2014). These types of cells are used in recurrent neural networks and have the ability to learn long- and short-term structure in data that have a serially-dependent nature, such as time-series or text data. GRU cells are a simplified version of the

LSTM cell with fewer learnable parameters:

$$z_t = \sigma(W_{xz}^{\intercal} x_t + W_{hz}^{\intercal} h_{t-1} + b_z)$$

$$r_t = \sigma(W_{xr}^{\intercal} x_t + W_{hr}^{\intercal} h_{t-1} + b_r)$$

$$g_t = \phi(W_{xg}^{\intercal} x_t + W_{hg}^{\intercal} (r_t \odot h_{t-1}) + b_g)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot g_z$$

where x_t is the input vector for step t, h_t is the output of the GRU cell at step t, σ is the sigmoid activation function, ϕ is the tanh activation function, $\odot$ is the element-wise product, W_{ij} is the weight matrix connecting inputs i and j, and b_i is the bias of unit i in the GRU cell. For a graphical depiction, see Figure 1, below.

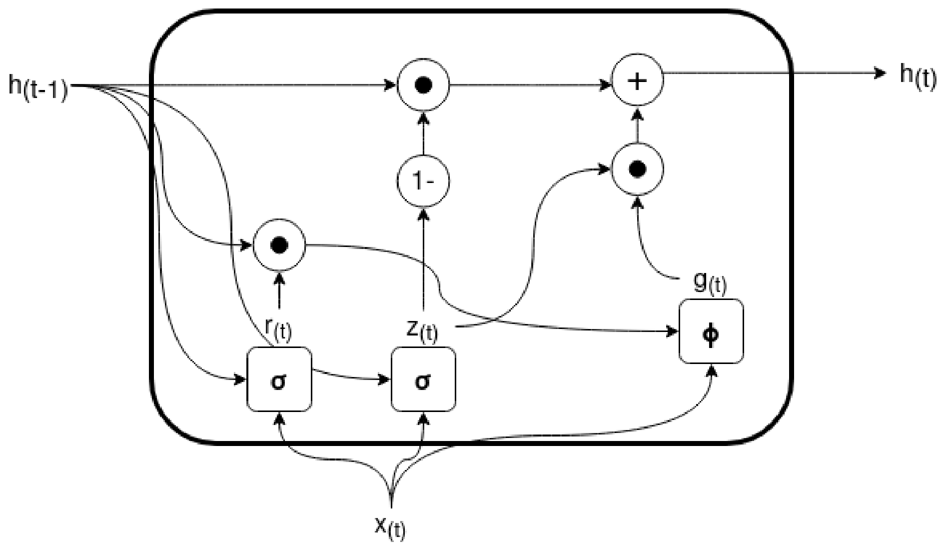

Figure 1. Depiction of GRU cell. Circle with dot depicts element-wise product.

In addition to having the ability to process sequential data, these types of networks can also be configured to see data backwards in time. This framework, called *bidirectional* recurrent networks (Schuster and Paliwal 1997), was originally developed to overcome the "near-sightedness" of uni-directional networks. That is, uni-directional networks are not able to use information beyond x_t, such as x_{t+2}, to help predict y_t. In terms of learning on text data, a bidirectional network contains separate sets of cells for the forward and backwards versions of the input vectors. This ultimately means that the network will be learning from both the forward-sentence and the backward-sentence (e.g., "Charles likes ice cream" $\rightarrow$ "cream ice likes Charles"). Bidirectional recurrent networks have proven to be superior to uni-directional networks in certain cases (Ogawa and Hori 2017; Sun, Zhang, and Akashi 2019).

Generally, neural networks have a large number of learnable parameters and therefore require a sizeable dataset to train successfully. Data augmentation can be used to synthetically inflate the size of a dataset by randomly applying a small amount of noise to the training data. In addition, some research suggests that this practice can act as an implicit regularizer of neural networks (Hernandez-Garcia and Konig 2019). While augmenting training datasets is commonplace in vision tasks, it is much less prevalent in the domain of natural language.

The work of Wei and Zou (2019) explores four methods for augmenting text datasets. The first is synonym replacement, which involves choosing n random non-stopword terms and replacing them with their synonym via a thesaurus lookup, or with a term whose embedding has a strong cosine similarity to the original word. The second is random insertion, which uses a synonym lookup, but instead of replacement, the synonym is inserted into a random spot in the text. The third is random swap, which chooses n random term pairs and swaps their positions. The fourth is random deletion, which removes n random terms with probability p. The authors find modest gains from these augmentation practices in terms of generalization of the model to the test dataset. Also, they find that their impact increases as the size of the entire dataset decreases. In addition to these four methods, natural language models may benefit from the practice of back-translation, which is the process of translating a sentence into a foreign language and then re-translating it back into the original language (Sennrich, Haddow, and Birch 2016). This method relies on a transition between languages that yields an imperfect translation, thus causing the sentence to change its structure and terms while retaining semantic meaning.

The practice of data augmentation goes beyond the training set. Using data augmentation techniques during inference, called test time augmentation (TTA), has been shown to help the performance of a model in some cases (Wang et al. 2019; Nalepa, Myller, and Kawulok 2020; Moshkov et al. 2020). This practice is meant to present the model with data during inference that "looks similar" to the data is was trained with. In general, the augmentations used during TTA will be the same augmentations and augmentation probabilities as were using on the training dataset.

Methods

Data

For the task of semantic similarity, we used the "Sentences Involving Compositional Knowledge" (SICK) dataset from the SemEval-2014 natural language competition[1]. The SICK dataset was introduced by Bentivogli et al. (2014) and contains just under 10,000 sentence pairs, using a designated 50/50 split for the training and testing datasets. The sentence pairs were manually tagged with a relatedness score, $s \in [1, 5]$, that captures the degree of semantic similarity between the sentences in the pair. For examples of sentence pairs and scores, see Table 1, below. To make this relatedness score workable for our experiments, we standardized it to be $s \in [0, 1]$. To compare the performance of our models, we used the official metric of the competition, Pearson's correlation between the ground truth labels and the model predictions on the test dataset.

Relatedness score	Sentence Pair
1.6	A: "A man is jumping into an empty pool" B: "There is no biker jumping in the air"
2.9	A: "Two children are lying in the snow and are making snow angels" B: "Two angels are making snow on the lying children"
3.6	A: "The young boys are playing outdoors and the man is smiling nearby" B: "There is no boy playing outdoors and there is no man smiling"
4.9	A: "A person in a black jacket is doing tricks on a motorbike" B: "A man in a black jacket is doing tricks on a motorbike"

Table 1. Examples of sentence pairs and their relatedness score from Bentivogli et al. (2014)

1. Retrieved from alt.qcri.org/semeval2014/task1/

Augmentation

For our research, we tested three sets of text augmentation. First (Aug 1), we tested synonym augmentation using WordNet, which is a large lexical database that is meant to map words together via lexical meaning (Princeton 2010). Our second method (Aug 2), takes in a sentence and produces four versions of it using random deletion, synonym substitution, random swapping, and random insertion. The third method (Aug 3), was back-translation. For this, we translated our sentences from English to Korean and then back to English[2]. Once augmented, the new sentences were added to the datasets, ultimately increasing their length. During experiments with TTA, final predictions for a sentence were made by averaging predictions for all versions (augmented and original) of the sentence. For an example of the result of each augmentation, see Table 2, below:

Method	Input	Output
Synonym	A person is not making a bed	A mortal is not making a bed
Deletion	A dog is pushing a toddler into a rain puddle	A dog is pushing a toddler into a rain
Swapping	A man with tattoos is lounging on a couch and holding a pencil	A man with tattoos is lounging on a couch and pencil a holding
Insertion	There is no dog chasing a ball in the grass	There lump is no dog chasing a ball in the grass
Back-translation	Philippines, Canada pledge to further boost relations	Philippines, Canada Strengthen Relations

Table 2. Examples of text-data augmentation

2. We note that we do not speak Korean, but instead used automated translation packages.

Model

For our Siamese networks, we employed GRU cells and Manhattan distance as the concatenation mechanism. Each of the mirrored networks were made of a word2vec embedding layer followed by a single set of GRU cells. The embedding layer made use of a word2vec model that was pre-trained on the Google News corpus[3] that embeds each word in the input text into a 300-dimensional vector. The GRU cells then map this into a vector of length 100. During training, the weights within the embedding layer were frozen. For a graphical depiction of our Siamese network, see Figure 2. After training our Siamese networks on the

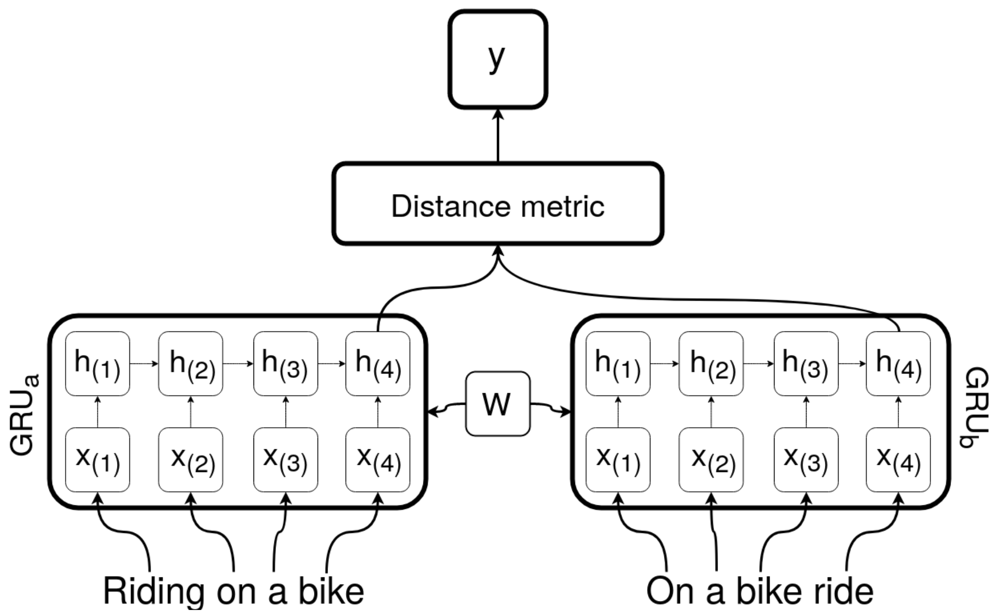

Figure 2. Depiction of Siamese network

training dataset and tuning them on the test dataset, we ended with the following hyperparameters: biases initialized to 2.5, mini-batch size of 64, Adadelta optimizer (Zeiler 2012) with decay of 0.985, gradient clipping for values beyond 2.0. All networks were trained until overfitting.

---

3. Retrieved from https://code.google.com/archive/p/word2vec/!

Analysis of Results

Observing Table 3, below, we note several findings. First, TTA was not helpful for any of the three augmentation sets tested. Second, in all but one case (Aug 3), using augmentation harmed the performance of the model. We hypothesize that the cause of these two results could be that we did not carefully select or tune the augmentations. Perhaps with more domain-specific augmentations, this strategy could help. In addition, the close performance of the back-translation to the non-augmentation models could be explained by non-lossy translations. It is possible that our attempts to capture non-perfect reproductions of the original sentences were in vain and what mainly returned were the original sentences.

Thirdly, we find that the bidirectional GRU models performed worse than the unidirectional GRU models in almost every case. We hypothesize that this may be due to the overall shortness of the sentences in our dataset or perhaps even the task itself not being fit for bidirectional models.

	No aug	Aug 1	Aug 1 TTA	Aug 2	Aug 2 TTA	Aug 3	Aug 3 TTA
GRU	0.841	0.776	0.718	0.829	0.804	0.838	0.813
BiGru	0.838	0.775	0.703	0.828	0.801	0.837	0.832

Table 3. Pearson's correlation between model predictions and ground truth labels of SICK dataset

Conclusion

In this work, we tested the Siamese network architecture with GRU cells for the task of estimating the semantic similarity of sentence pairs. We performed a thorough evaluation of the architecture by using both bidirectional and unidirectional cells in combination with three different sets of data augmentation. We evaluated these combinations on the SICK dataset with Pearson's correlation between the ground truth labels and the model predictions. We found that, in almost all cases, the unidirectional cells performed better than the bidirectional cells. In addition, we found that none of the augmentation sets aided in performance of the model on the test datasets.

Future Work

We should investigate the outputs of the augmentation sets more closely to understand why the back-translation method performed the most similar to the non-augmentation models. In addition to manual checking, this should be tested with languages other than Korean. In addition, it may be worthwhile to explore deeper models of stacked recurrent layers and different distance metrics.

Bibliography

Bentivogli, Luisa, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. 2014. "SICK Through the SemEval Glasses. Lesson Learned from the Evaluation of Compositional Distributional Semantic Models on Full Sentences Through Semantic Relatedness and Textual Entailment". In *9th International Conference on Language Resources and Evaluation*, 216–223. Reykjavik, Iceland: ELRA.

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation". arXiv: `1406.1078[cs.CL]`.

Chopra, Sumit, Raia Hadsell, and Yan LeCun. 2005. "Learning a Similarity Metric Discriminatively, with Application to Face Verification". In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference*, 1:539–546. San Diego, CA: IEEE.

Gan, Mingxin, Xue Dou, and Rui Jiang. 2013. "From Ontology to Semantic Similarity: Calculation of Ontology-Based Semantic Similarity". *The Scientific World* 2013:793091–11.

Hernandez-Garcia, Alex, and Peter Konig. 2019. "Further Advantages of Data Augmentation on Convolutional Neural Networks". In *International Conference on Artificial Neural Networks (ICANN)*. Rhodes, Greece: Springer.

Hliaoutakis, Angelos, Giannis Varelas, Epimenidis Voutsakis, Euripides G.M Petrakis, and Evangelos Milios. 2006. "Information Retrieval by Semantic Similarity". *International Journal on Semantic Web and Information Systems* 2 (3): 55–73.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory". *Neural Computing* 9:1735–1780.

Kamineni, Avinash, Manish Shrivastava, Harish Yenala, and Manoj Chinnakotla. 2018. "Siamese LSTM with Convolutional Similarity for Similar Question Retrieval". In *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, 1–7. Pattaya, Thailand: IEEE.

Li, Zhenxi, Guillaume-Alexandre Bilodeau, and Wassim Bouachir. 2018. "Multibranch Siamese Networks with Online Selection for Object Tracking". arXiv: `1808.07349[cs.CV]`.

Moshkov, Nikita, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi, and Peter Horvath. 2020. "Test-Time Augmentation for Deep Learning-Based Cell Segmentation on Microscopy Images". *Scientific reports* 10 (1): 5068–7.

Mueller, Jonas, and Aditya Thyagarajan. 2016. "Siamese Recurrent Architectures for Learning Sentence Similarity". In *Thirtieth AAAI Conference on Artificial Intelligence.* Phoenix, Arizona: AAAI Press.

Nalepa, J., M. Myller, and M. Kawulok. 2020. "Training- and Test-Time Data Augmentation for Hyperspectral Image Segmentation". *IEEE Geoscience and Remote Sensing Letters* 17 (2): 292–296.

Ogawa, Atsunori, and Takaaki Hori. 2017. "Error Detection and Accuracy Estimation in Automatic Speech Recognition Using Deep Bidirectional Recurrent Neural Networks". *Speech Communication* 89:70–83.

Princeton, University. 2010. "About WordNet". `https://wordnet.princeton.edu/`.

Schuster, Mike, and Kuldip Paliwal. 1997. "Bidirectional recurrent neural networks". *Signal Processing, IEEE Transactions on* 45 (): 2673–2681. doi:`10.1109/78.650093`.

Sennrich, Rick, Barry Haddow, and Alexandra Birch. 2016. "Improveing Neural Machine Translation Mofels with Monolingual Data". In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers*, 1715–1725. Berlin, Germany: ACL.

Sun, Haitian, Chao Zhang, and Takuya Akashi. 2019. "Recurrent bidirectional visual human pose retrieval". *IEEJ Transactions on Electrical and Electronic Engineering* 14 (7): 1074–1081.

Wang, Guotai, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. 2019. "Aleatoric Uncertainty Estimation with Test-Time Augmentation for Medical Image Segmentation with Convolutional Neural Networks". *Neurocomputing* 338:34–45.

Wei, Jason, and Kai Zou. 2019. "EDA: Easy Data Augmentation Techniques for Booksting Performance on Text Classification Tasks". In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Join Conference on Naturual Language Processing.* Hong Kong: ACL.

Zeiler, Matthew. 2012. "ADADELTA: An Adaptive Learning Rate Method". arXiv: `1212.5701[cs.LG]`.

Zhu, Wenhao, Tengjun Yao, Jianyue Ni, Baogang Wei, Zhiguo Lu, and Xuchu Weng. 2018a. "Dependency-Based Siamese Long Short-Term Memory Network for Learning Sentence Representations". *PLoS ONE* 13 (3).

Zhu, Zheng, Qiang Wang, Bo Li, Wei Wu, Junji Yan, and Weiming Hu. 2018b. "Distractor-Aware Siamese Networks for Visual Object Tracking". In *ECCV 2018*, 11213:103–119. Munich, Germany: Springer Verlag.