Fake Sentence Detection as a Training Task for Sentence Encoding

Anonymous EMNLP submission

Abstract

Sentence encoders are typically trained on language modeling tasks which enable them to use large unlabeled datasets. While these models achieve state-of-the-art results on many sentence-level tasks, they are difficult to train with long training cycles. We introduce fake sentence detection as a new training task for learning sentence encodings. We automatically generate fake sentences by corrupting some original sentence and train the encoders to produce representations that are effective at detecting fake sentences. This binary classification task allows for efficient training and forces the encoder to learn the distinctions introduced by a small edit to sentences. We train a basic BiLSTM encoder to produce sentence representations and find that it outperforms a strong sentence encoding model trained on language modeling tasks, while also training much faster on smaller amount of data (20 hours instead of weeks). Further analysis shows the learned representations capture many syntactic and semantic properties expected from good sentence representations.

1 Introduction

Unsupervised sentence encoders are trained on language modeling tasks where the encoded sentence representations are used to either reconstruct the input sentence (Hill et al., 2016) or generate neighboring sentences (Kiros et al., 2015; Hill et al., 2016). This enables encoders to create representations such that sentences that similar in meaning or topic are closer in the embedded space. The trained representations achieve the best performance on many sentence-level prediction tasks (Hill et al., 2016).

However, this language modeling based training is problematic in two respects: 1) Training a language model to predict over larger contexts (neighboring sentences) requires large amounts of training data and time. Predicting neighboring sentences is a difficult and under-constrained task as there can be many valid possibilities for nearby sentences for any particular input sentence. 2) There is nothing explicit in the training task to force the encoder to learn fine grained distinctions between sentences that are mostly similar, a requirement often needed in downstream applications such as natural language inference (NLI).

In this paper we introduce an unsupervised discriminative training task, *fake sentence detection*, which is aimed at learning representations that distinguish sentences that are mostly similar in their words but may differ significantly in meaning or structure. The main idea is to generate fake sentences by corrupting an original sentence. We use two methods to generate fake sentences: *word shuffling* where we swap the positions of two words at random and *word dropping*, where we drop a word at random from the original sentence.

This training task formulation has two key advantages: (i) Corrupting a sentence can lead to break in syntactic coherence (e.g. missing a verb) leading to a malformed sentence or can cause a big change in the semantics (e.g., swapping subjects with object can be relevant for NLI) or a minor but meaningful distinction (e.g., dropping an adjective can be relevant for sentiment). In extremely rare cases the meaning may not change at all. Given that the sentences are going to be mostly similar (every pair is within a edit distance of two), for the encoder to be successful it must learn to tease apart the compositional aspects of meaning and explicitly learn to detect these small but meaningful shifts. (ii) This binary classification task can be modeled with fewer parameters in the output layer and can be trained more efficiently compared to the language modeling training tasks where the output layer has many parameters depending on the vocabulary size.

Given a large unlabeled corpus, for every original sentence, we add multiple fake sentences. The training task is then to take any given sentence as input and predict whether it is a real or fake sentence. We train a bidirectional long short term memory network (BiLSTM) encoder that produces a representation of the input sentence, which is then used by a three-layer feed-forward network for prediction. We then evaluate this trained encoder *without any further tuning* on multiple sentence-level tasks and test for syntactic and semantic properties which demonstrate the benefits of fake sentence training.

In summary, this paper makes the following contributions: 1) Introduces fake sentence detection as an unsupervised training task for learning sentence encoders that can distinguish between small changes in mostly similar sentences. 2) An empirical evaluation on multiple sentence-level tasks showing representations trained on the fake sentence tasks outperform a strong baseline model trained on language modeling tasks, even when training on small amounts of data (1M vs. 64M sentences) reducing training time from weeks to within 20 hours.

2 Related Work

Previous sentence encoding approaches can be broadly classified as supervised (Conneau et al., 2017; Cer et al., 2018; Marcheggiani and Titov, 2017; Wieting et al., 2015), unsupervised (Kiros et al., 2015; Hill et al., 2016) or semi-supervised approaches (**?**Peters et al., 2018; Dai and Le, 2015; Socher et al., 2011). The supervised approaches train the encoders on tasks such as NLI and use transfer learning to adapt the learned encoders to different downstream tasks. The unsupervised approaches extend the skip-gram (Mikolov et al., 2013) to the sentence level, and use the sentence embedding to predict the adjacent sentences. Skipthought (Kiros et al., 2015) uses a BiLSTM encoder to obtain a fixed length embedding for a sentence, and uses a BiLSTM decoder to predict adjacent sentences. Training Skipthought model is expensive, and one epoch of training on the Toronto BookCorpus (Zhu et al., 2015) dataset takes more than two weeks (Hill et al., 2016) on a single GPU. FastSent (Hill et al., 2016) uses embeddings of a sentence to predict words from the adjacent sentences. A sentence is represented by simply summing up the word representation of all the words in the sentence. FastSent requires less training time than Skipthought, but FastSent has worse performance. Semi-supervised approaches train sentence encoders on large unlabeled datasets, and do a task specific adaptation using labeled data.

In this work, we propose an unsupervised sentence encoder that takes around 20 hours to train on a single GPU, and outperforms Skipthought and FastSent encoders on multiple downstream tasks. Unlike the previous unsupervised approaches, we use the binary task of real versus fake sentence classification to train a BiLSTM based sentence encoder.

3 Training Tasks for Encoders

We propose a discriminative task for training sentence encoders. The key bottleneck in training sentence encoders is the need for large amounts of labeled data. Prior work use language modeling as a training task leveraging unlabeled text data. Encoders are trained to produce sentence representations which are effective at either generating neighboring sentences (e.g., Skipthought (Kiros et al., 2015) or at least effective at predict the words in the neighboring sentences (Hill et al., 2016).The challenge becomes one of balance between model coverage (i.e. the number of output words it can predict) and model complexity (i.e. the number of parameters need for prediction).

Rather address the language modeling challenges, we instead propose a simpler training task that requires making a single prediction over an input sentence. In particular, we propose to learn a sentence encoder by training a sequential model to solve the binary classification task of detecting whether a given input sentence is fake or real. This real-fake sentence classification task would perhaps be trivial if the fake sentences look very different from the real sentences. We propose two simple methods to generate noisy sentences which look *mostly* similar to real sentences. We describe the noisy sentence generation strategies in Section 3.1. Thus, we create a labeled dataset of real and fake sentences, and train a sequential model to distinguish between real and fake sentences, which results in a model whose classification layer has far fewer parameters than previous language model based encoders. Our model architecture is described in Section 3.2.

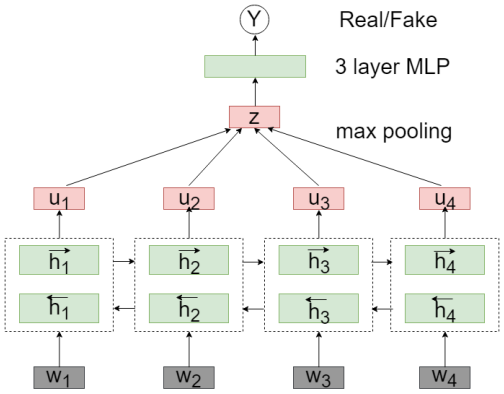

Figure 1: Figure shows the block diagram of the encoder and fully connected layers. Encoder consists of a bidirectional LSTM followed by a max pooling layer. For classification, we use a MLP with two hidden layers.

3.1 Fake Sentence Generation

For a sentence $X = w_1, w_2, \ldots, w_n$ comprising of n words, we consider two strategies to generate a noisy version of the sentence: **1) WordShuffle**: randomly sample two indices i and j corresponding to words w_i and w_j in X, and shuffle the words to obtain the noisy sentence $\hat{X}$. Noisy sentence $\hat{X}$ would be of the same length as the original sentence X. **2) WordDrop**: randomly pick one index i corresponding to word w_i and drop the word from the sentence to obtain $\hat{X}$. Note there can be many variants for this strategy but here we experiment with this basic choice.

3.2 Real Versus Fake Sentence Classification

Figure 1 shows the proposed architecture of our fake sentence classifier with an encoder and a Multi-layer Perceptron(MLP) with 2 hidden layers. The encoder consists of a bidirectional LSTM followed by a max pooling layer. At each time step we concatenate the forward and backward hidden states to get $u_i = (\overrightarrow{h_i}, \overleftarrow{h_i})$. We apply max-pooling to these concatenated hidden states to get a fixed length representation (z), which we then use as input to a MLP for classifying into real/fake classes.

4 Evaluation Setup

Downstream Tasks: We compare the sentence encoders trained on a large collection (BookCorpus (Zhu et al., 2015)) by testing them on multiple sentence level classification tasks (MR, CR, SUBJ, MPQA, TREC, SST) and one NLI task defined over sentence-pairs (SICK). We also evaluate the sentence representations for image and caption retrieval tasks on the COCO dataset (Lin et al., 2014). We use the same evaluation protocol and dataset split as (Karpathy and Fei-Fei, 2015; Conneau et al., 2017). Table 1 lists the classification tasks and the datasets. We also compare the sentence representations for how well they capture important syntactic and semantic properties using probing classification tasks (Conneau et al., 2018). For all downstream and probing tasks, we use the encoders to obtain representation for all the sentences, and train logistic regression classifiers on the training split. We tune the L_2-norm regularizer using the validation split, and report the results on the test split.

Name	Size	Task	Classes
MR	11K	Sentiment	2
CR	4K	Product Review	2
TREC	11K	Question type	6
SST	70K	Sentiment	2
MPQA	11K	Opinion Polarity	2
SUBJ	10K	Subjectivity	2
SICK	10K	NLI	3
COCO	123K	Retrieval	-

Table 1: Downstream tasks and datasets.

Training Corpus: The FastSent and Skipthought encoders are trained on the full Toronto BookCorpus of 64M sentences (Zhu et al., 2015). Our models, however, train on a much smaller subset of *only* 1M sentences.

Sentence Encoder Implementation: Our sentence encoder architecture is the same as the BiLSTM-max model (Conneau et al., 2017). We represent words using 300-d pretrained Glove embeddings (Pennington et al., 2014). We use a single layer BiLSTM model, with 2048-d hidden states. The MLP classifier we use for fake sentence detection has two hidden layers with 1024 and 512 neurons. We train separate models for word drop and word shuffle. The models are trained for 15 epochs with a batch size of 64 using SGD algorithm, when training converges with a validation set accuracy of 87.2 for word shuffle. The entire training completes in less than 20 hours on a single GPU machine.

Baseline Approaches: We compare our results with previous unsupervised sentences encoders, Skipthought (Kiros et al., 2015) and FastSent (Hill et al., 2016). We use the FastSent and Skipthought results trained on the full BookCorpus as mentioned in (Conneau et al., 2017).

5 Results

Classification and NLI: Results are shown in Table 2. Both fake sentence training tasks yield

Model	MR	CR	TREC	SST	MPQA	SUBJ	SICK	COCO-Cap	COCO-Img
FastSent	70.8	78.4	80.6	-	80.6	88.7	-	-	-
Skipthought (full)	76.5	80.1	**92.2**	82.0	87.1	**93.6**	**82.3**	72.2	66.2
Skipthought (1M)	65.2	70.9	79.2	66.9	81.6	86.1	75.6	-	-
WordDrop	78.8	82.2	86.6	**82.9**	**89.8**	92.7	**83.2**	**73.8**	**67.3**
WordShuffle	**79.8**	**82.4**	88.4	**82.4**	**89.8**	92.6	**82.3**	**74.2**	**67.3**

Table 2: Results on downstream tasks: Bold face indicates best result and underlined results show when fake sentence training is better than Skipthought (full). COCO-Cap and COCO-Img are caption and image retrieval tasks on COCO. We report Recall@5 for the COCO retrieval tasks.

Model	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
Skipthought (full)	85.4	79.6	41.1	**82.5**	69.6	**90.4**	85.6	**83.6**	53.9	69.1
Skipthought (1M)	54.7	33.9	30.0	60.7	58.9	85.3	76.4	70.9	51.9	61.4
WordDrop	**86.7**	90.1	48.0	81.9	73.2	87.7	**87.3**	82.7	59.2	70.6
WordShuffle	84.9	**91.2**	**48.8**	82.3	**79.9**	88.2	86.7	83.3	**59.8**	**70.7**

Table 3: Probing task accuracies. Tasks: SentLen: predict sentence length, WC: is word in sentence, TreeDepth: depth of syntactic tree, TopConst: predict top-level constituent, BShift: is bigram in flipped in sentence, Tense: predict tense of word, Subj(Obj)Num: singular or plural subject, SOMO: semantic odd man out, CoordInv: is co-ordination is inverted.

better performance on five out of the seven language tasks when compared to Skipthought (full), i.e., even when it is trained on the full BookCorpus. Word drop and word shuffle performances are mostly comparable. The Skipthought (1M) row shows that training on a sentence-level language modeling task can fare substantially worse when trained on a smaller subset of data. FastSent, while easier to train and has faster training cycles, is better than Skipthought (1M) but is worse than the full Skipthought model.

Image-Caption Retrieval: On both caption and image retrieval tasks (last 2 columns of Table 2), fake sentence training with word dropping and word shuffle are better than the published Skipthought results.

Probing Tasks: Table 3 compares sentence encoders using the recently proposed probing tasks (Conneau et al., 2018). The goal of each task is to use the input sentence encoding to predict a particular syntactic or semantic property of the original sentence it encodes (e.g., predict if the sentence contains a specific word). Encodings from fake sentence training score higher in six out of the ten tasks. WordShuffle encodings are significantly better than Skipthought in some semantic properties: tracking word content (WC), bigram shuffles (BShift), semantic odd man out (SOMO). Skipthought and WordShuffle are comparable on syntactic properties: agreement (SubjNum, ObjNum, Tense, and CoordInv). The only exception is TreeDepth, where WordShuffle is substantially better. Table 4 shows examples of the BShift task and cases where the word shuffle and Skipthought models fail. In general we find that word shuffle works better when shifted bigrams involve prepositions, articles, or conjunctions.

Shuffled Sentence	WS	ST
It shone the in light .	✓	×
I seized the and sword leapt to the window .	✓	×
Once again Amadeus held out arm his .	✓	✓
When we get inside , I know that I have to leave and Marceline find .	×	✓

Table 4: Word shuffle (WS) and Skipthought (ST) performance on BShift. Underlined positions are swapped.

6 Conclusions

Using language modeling tasks to learn sentence representations is challenging because learning to generate nearby sentences is a difficult under-constrained task. This work introduced an unsupervised training task, fake sentence detection, where the sentence encoders are trained to produce representations which are effective at detecting if a given sentence is an original or a fake. This leads to better performance on downstream tasks and is able to represent semantic and syntactic properties, while also reducing the amount of training needed. More generally the results suggest that tasks which test for different syntactic and semantic properties in altered sentences can be useful for learning effective representations.

References

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, et al. 2018. Deep contextualized word representations.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the conference on empirical methods in natural language processing*, pages 151–161. Association for Computational Linguistics.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.