Learning Semantic Textual Similarity from Conversations

Yinfei Yang[a], Steve Yuan[c], Daniel Cer[a], Sheng-yi Kong[a], Noah Constant[a], Petr Pilar[c], Heming Ge[a], Yun-Hsuan Sung[a], Brian Strope[a], Ray Kurzweil[a]

[a]Google AI
Mountain View, CA, USA

[b]Google
Cambridge, MA, USA

[c]Google
Zurich, Switzerland

Abstract

We present a novel approach to learn representations for sentence-level semantic similarity using conversational data. Our method trains an unsupervised model to predict conversational responses. The resulting sentence embeddings perform well on the Semantic Textual Similarity (STS) Benchmark and SemEval 2017's Community Question Answering (CQA) question similarity subtask. Performance is further improved by introducing multi-task training, combining conversational response prediction and natural language inference. Extensive experiments show the proposed model achieves the best performance among all neural models on the STS Benchmark and is competitive with the state-of-the-art feature engineered and mixed systems for both tasks.

使用对话的数据，为判别句子层次的语义相似度学习表达.

无监督

评估：
STS
相似问题检测

引入多任务学习
(response prediction + NLI)

1 Introduction

We propose a novel approach to sentence-level semantic similarity based on unsupervised learning from conversational data. We observe that semantically similar sentences have a similar distribution of potential conversational responses, and that a model trained to predict conversational responses should implicitly learn useful semantic representations. As illustrated in Figure 1, "How old are you?" and "What is your age?" are both questions about age, which can be answered by similar responses such as "I am 20 years old". In contrast, "How are you?" and "How old are you?" use similar words but have different meanings and lead to different responses.

语义相似的句子有相似的response分布

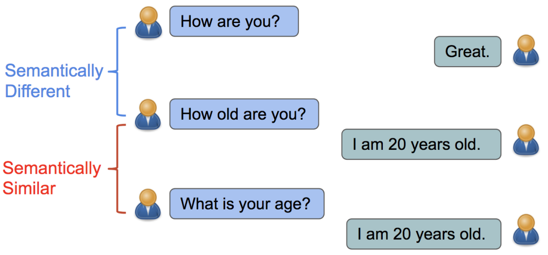


Figure 1: Sentences have similar meanings if they can be answered by a similar distribution of conversational responses.

Deep learning models have been shown to predict conversational responses with increasingly good accuracy (Henderson et al., 2017; Kannan et al., 2016). The internal representations of such models resolve the semantics necessary to predict the correct response across a broad selection of input messages. Meaning similarity between sentences then can be obtained by comparing the sentence-level representations learned by such models. We follow this approach, and assess the quality of the resulting similarity scores on the Semantic Textual Similarity (STS) Benchmark (Cer et al., 2017) and a question similarity subtask from SemEval 2017's Community Question Answering (CQA) evaluation. The STS benchmark scores sentence pairs based on their degree of meaning similarity. The Community Question Answering (CQA) subtask B (Nakov et al., 2017) ranks questions based on their similarity with a target question.

We first assess representations learned from unsupervised conversational input-response pairs. We then explore augmenting our model with multi-task training over a combination of unsupervised conversational response prediction and supervised training on Natural Language Inference (NLI) data, as training to NLI has been shown to independently yield useful general purpose representations (Conneau et al., 2017). Unsupervised training over conversational data yields represen-

无监督

多任务学习
无监督
+
有监督 NLI

Proceedings of the 3rd Workshop on Representation Learning for NLP, pages 164–174
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

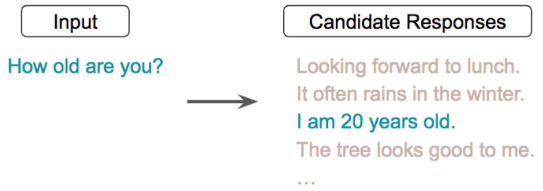

Input
How old are you?
Candidate Responses
Looking forward to lunch.
It often rains in the winter.
I am 20 years old.
The tree looks good to me.
…

Figure 2: The conversational response selection problem attempts to identify the correct response from a collection of candidate responses. We train using batch negatives with each candidate response serving as a positive example for one input and a negative sample for the remaining inputs.

tations that perform well on STS and CQA question similarity. The addition of supervised SNLI data leads to further improvements and reaches state-of-the-art performance for neural STS models, surpassing training on NLI data alone.

2 Approach

This section describes the conversational learning task and our architecture for predicting conversational responses. We detail two encoding methods for converting sentences into sentence embeddings and describe multitask learning over conversational and NLI data.

2.1 Conversational Response Prediction

We formulate the conversational learning task as response prediction given an input (Kannan et al., 2016; Henderson et al., 2017). Following prior work, the prediction task is cast as a response selection problem. As shown in Figure 2, the model $P(y|x)$ attempts to identify the correct response y from $K-1$ randomly sampled alternatives.

response selection

2.2 Model Architecture

Our model architecture encodes input and response sentences into fixed-length vectors u and v, respectively. The preference of an input described by u for a response described by v is scored by the dot product of the two vectors. The dot product scores are converted into probabilities using a softmax over the scores from all other candidate responses. Model parameters are trained to maximize the log-likelihood of the correct responses.

编码：

相似度计算：

loss函数：

Figure 3 illustrates the input-response scoring model architecture. Tied parameters are used for the input and response encoders. In order to model the mapping between inputs and their expected responses, the response embeddings are passed through an additional feed-forward network to get the final response vector v' before computing the dot product with the input sentence embedding.[1]

两边的encoder共享参数

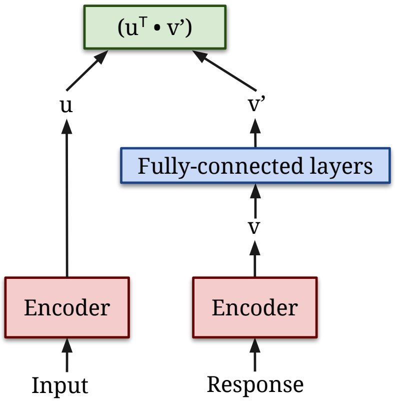


Figure 3: Conversational response prediction model. The sentence encoders are in red and use shared parameters. Fully connected DNN layers perform the mapping between the semantics of the input sentence and the candidate response.

Training is performed using batches of K randomly shuffled input-response pairs. Within a batch, each response serves as the correct answer to its corresponding input and the incorrect response to the remaining $K-1$ inputs in the batch. In the remaining sections, this architecture is referred to as the *input-response model*.

训练方式

2.3 Encoders

Figure 4 illustrates the encoders we explore for obtaining sentence embeddings: DANs (Iyyer et al., 2015) and Transformer (Vaswani et al., 2017).[2]

2.3.1 DAN

Deep averaging networks (DAN) compute sentence-level embeddings by first averaging word-level embeddings and then feeding the averaged representation to a deep neural network (DNN) (Iyyer et al., 2015). We provide our encoder with input embeddings for both words and bigrams in the sentence being encoded. This simple architecture has been found to outperform LSTMs on email response prediction (Henderson et al., 2017). The embeddings for words and

word embedding

$$\frac{\sum_{i=1}^{n} h_i}{\sqrt{n}}$$

bigram embedding

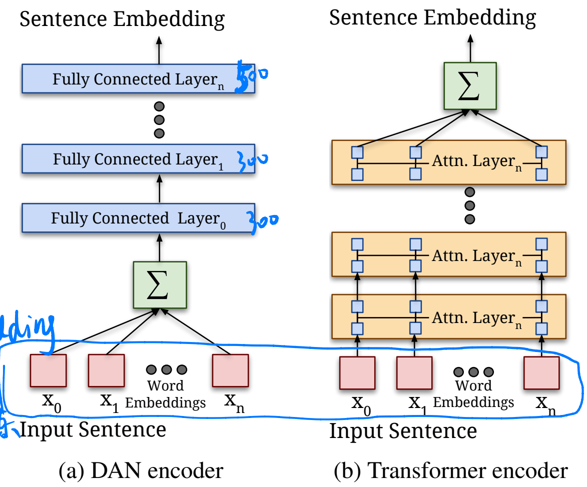
$$\frac{\sum_{i=1}^{n} h_i'}{\sqrt{n}}$$

[1] While feed-forward layers could have been added to the input encoder as well, early experiments suggested it was sufficient to add additional layers to only one of the encoders.

[2] We tried other encoder architectures, notably LSTM (Hochreiter and Schmidhuber, 1997) and Bi-LSTM (Graves and Schmidhuber, 2005), but found they performed worse than transformer in preliminary experiments.

论文未明确说明这两种embedding如何融合.

bigrams are learned during training of the input-response model. Our implementation sums the input embeddings and then divides by $sqrt(n)$, where n is the sentence length.[3] The resulting vector is passed as input to the DNN.



(a) DAN encoder

(b) Transformer encoder

word embedding 是从0开始学，不使用预训练的.

Figure 4: Model architectures for the DAN and Transformer sentence encoders.

DANs perform well in practice on sentence-level prediction and encoding tasks (Iyyer et al., 2015; Henderson et al., 2017). However, they lack any explicit network structure for encoding long range relationships between words.

2.3.2 Transformer

Transformer (Vaswani et al., 2017) is a recent network architecture that makes use of attention mechanisms to explicitly capture relationships between words appearing at any position in a sentence. The architecture is able to achieve state-of-the-art performance on translation tasks and is available as open-source.[4]

While the original transformer architecture contains an encoder and decoder, we only need the encoder component in our training procedure. The encoder is constructed as a series of attention layers consisting of a multi-headed self-attention operation over all input positions followed by a feed-forward layer that processes each position independently (see figure 4b). Positional information is captured by injecting a "timing signal" into the input embeddings based on sine/cosine functions at different frequencies.

多层transformer堆叠

The transformer encoder output is a variable-length sequence. We reduce it to fixed length by averaging across all sequence positions. Intuitively, this is similar to building a bag-of-words representation, except that the words have had a chance to interact with their contexts through the attention layers. In practice, we see that the learned attention masks focus largely on nearby words in the first layer, and attend to progressively more distant context in the higher layers.

2.4 Multitask Encoder

We anticipate that learning good semantic representations 期望 may benefit from the inclusion of multiple distinct tasks during training. Multiple tasks should improve the coverage of semantic phenomenon that are critical to one task but less essential to another. We explore multitask models that use a shared encoder for learning conversational response prediction and natural language inference (NLI). The NLI data are from the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) corpus. The sentences are mostly non-conversational, providing a complementary learning signal.

多任务可以帮助学习不同的语义现象.

多任务的训练使用共享的Encoder.

Figure 5 illustrates the multitask model with SNLI. We keep the input-response model the same, and build another two encoders for SNLI pairs, sharing parameters with the input-response encoders. Following Conneau et al. (2017), we encode a sentence pair into vectors u_1, u_2 and construct a feature vector $(u_1, u_2, |u_1 - u_2|, u_1 * u_2)$. The feature vector is fed into a 3-way classifier consisting of a feedforward network culminating in a softmax layer. Following prior work, we use a single 512 unit hidden layer for our experiments.

3 Conversational Data

Our unsupervised model relies on structured conversational data. The data for our experiments are drawn from Reddit conversations spanning 2007 to 2016, extracted by Al-Rfou et al. (2016). This corpus contains 133 million posts and a total of 2.4 billion comments. The comments are mostly conversational and well structured, making it a good resource for training conversational models.

Figure 6 provides an example of a Reddit comment chain. Comment B is a child of comment A if comment B is a reply to comment A. We extract

[3] `sqrtn` is one of TensorFlow's built-in embedding combiners. The intuition behind dividing by $sqrt(n)$ is as follows: We want our input embeddings to be sensitive to length. However, we also want to ensure that for short sequences the relative differences in the representations are not dominated by sentence length effects.

[4] https://github.com/tensorflow/tensor2tensor

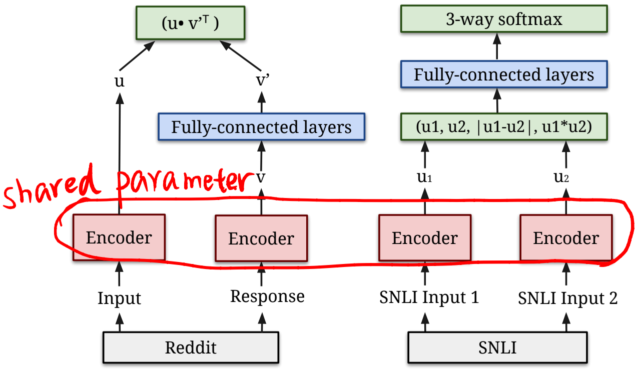
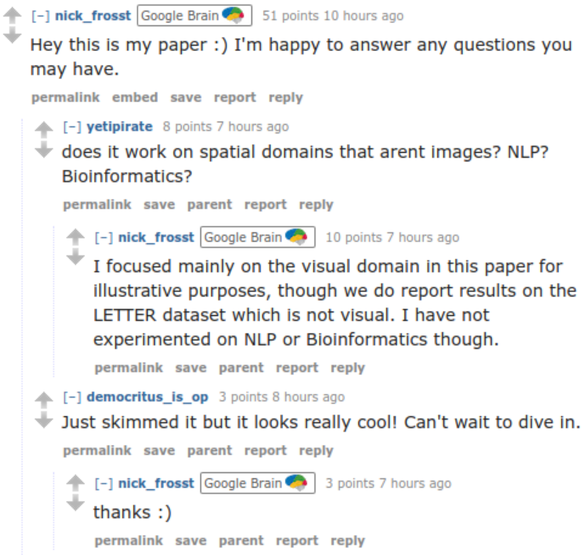


Figure 5: Architecture of the multitask model. Sentence encoders are in red and share parameters.



[–] nick_frosst Google Brain 51 points 10 hours ago

Hey this is my paper :) I'm happy to answer any questions you may have.

permalink embed save report reply

[–] yetipirate 8 points 7 hours ago

does it work on spatial domains that arent images? NLP? Bioinformatics?

permalink save parent report reply

[–] nick_frosst Google Brain 10 points 7 hours ago

I focused mainly on the visual domain in this paper for illustrative purposes, though we do report results on the LETTER dataset which is not visual. I have not experimented on NLP or Bioinformatics though.

permalink save parent report reply

[–] democritus_is_op 3 points 8 hours ago

Just skimmed it but it looks really cool! Can't wait to dive in.

permalink save parent report reply

[–] nick_frosst Google Brain 3 points 7 hours ago

thanks :)

permalink save parent report reply

Figure 6: Reddit comment chain.

comments and their children to form the input-response pairs described above. Several rules are applied to filter out the noisy data. A comment is removed if any of the following conditions holds: number of characters $\geq$ 350, percentage of alphabetic characters $\leq$ 70%, starts with "https", "/r/" or "@", author's name contains "bot". The total number of extracted pairs is around 600 million.

3.1 Model Configuration

Model configuration and hyperparameters are set based on prior experiments on Reddit response prediction and performance of the multi-task model on SNLI. All inputs are tokenized and normalized before being fed into model. For all experiments, we use SGD with a batch size of 128 and a learning rate of 0.01. The total training steps are 40 million steps for the Reddit model and 30 million steps for the Reddit+SNLI model. We adjust the batch size to 256 and learning rate to 0.001 after 30 million and 20 million steps for the Reddit and the Reddit+SNLI models, respectively. When training the multitask model, we initialize the shared parameters with a pretrained Reddit model. We employ a distributed training system with multiple workers, where 95% of workers are used to continue training the Reddit task and 5% of workers are used to train the SNLI task. We use a sentence embedding size of 500 in all experiments, and normalize sentence embeddings prior to use in subsequent network layers. The parameters were only lightly tuned to prevent overfitting on the SNLI task.

	P@1	P@3	P@10
Transformer	**65.7**	**78.7**	**89.8**
DAN	56.1	70.2	83.6

Table 1: Precision at N (P@N) results on the Reddit response predication test set for models built using the DAN and Transformer encoders. Models attempt to select the true response for an input against 99 randomly selected negatives.

The encoder configurations are taken from the default parameters from previous work. For DAN, we employ a 3-layer DNN with layers containing 300, 300, and 500 hidden units. For the transformer encoder, our experiments make use of 6 attention layers (num_hidden_layers) and 8 attentions heads (num_heads). Within each attention layer, the feedforward network applied to each head has an input and output size of 512 (hidden_size) and makes use of a 2048 unit inner-layer (filter_size).

4 Experiments

We first evaluate the different encoders on the ① 评估 response prediction task. For the multitask models, we then examine their performance ② on SNLI. Finally, we evaluate the encoders on the STS Benchmark ③ (Cer et al., 2017) and on SemEval 2017 Community Question Answering (CQA) subtask B ④ (Nakov et al., 2017). We refer to the model trained over Reddit input-response pairs as *Reddit* and the multitask model as *Reddit+SNLI*.

4.1 Response Prediction

Following Henderson et al. (2017), we use precision at N (P@N) as an evaluation metric for the conversational response prediction task. Given an

	Accuracy
Reddit+SNLI	84.1
InferSent	**84.5**
KIM Ensemble	**<u>89.0</u>**
Gumbel TreeLSTM	86.0

Table 2: SNLI classification performance for the Reddit+SNLI model using the transformer encoder with reference evaluation numbers from prior work. We note that similar to InferSent, our goal is to use SNLI to obtain better sentence representations rather than achieving state-of-the-art performance on the SNLI task itself.

input, the task is to select the true response (positive) from 99 randomly selected responses (negatives). We rank all 100 candidate responses by their dot-product scores from the input-response model. The P@N score evaluates if the true response (positive) appears in the top N responses. For the evaluation, the Reddit data is randomly split into train (90%) and test (10%) sets.

Table 1 shows the P@N results of Reddit models trained with different encoders, for N=1, 3, 10. The DAN encoder (with n-grams), as investigated by Henderson et al. (2017), provides a strong baseline. We observe the transformer encoder outperforms DAN for all values of N. The transformer encoder achieves a P@1 metric of 65.7% while DAN achieves only 56.1%. Given its greater performance, we use a transformer encoder for the remainder of the experiments reported in this work.

4.2 SNLI

SNLI (Bowman et al., 2015) annotates the inferential relationship between paired sentences as entailment, contradiction, or neural. One sentence is entailed by another sentence if its meaning can be inferred from the other. Sentences contradict each other if the meaning of one implies that the other is not true. The sentence pairs in the dataset are partitioned into train (550,152), dev (10,000), and test (10,000). Model performance is evaluated based on classification accuracy.

Our multitask model learns a shared encoder for the conversational response prediction and SNLI tasks. We report evaluation results on the SNLI task in order to facilitate better comparison with InferSent (Conneau et al., 2017), which served as the inspiration for the inclusion of the SNLI task within a multitask model. For reference, we provide the results of Gumbel TreeLSTM (Williams et al., 2017), which is the best sentence encoder based model, and KIM Ensemble (Chen et al., 2017), which is the current state-of-the-art.

Sentence encoder based models first encode the two sentences in an SNLI input pair separately, and then feed the encodings into a classifier. By comparison, other models explicitly consider word-level interactions between the paired sentences (e.g., using cross-attention). We note that our model is sentence encoder based.

Table 2 shows the accuracy on the test set of the joint model and baselines. The multitask model achieves 84.1% accuracy and is close to the performance of InferSent. There are two significant differences between our model and prior work. First, the proposed model learns all model parameters from scratch, including the word embeddings. Due in part to the size of the SNLI training set, InferSent uses a large pre-trained word embedding model fit via GloVe (Pennington et al., 2014) on 840 billion tokens of web crawl data, which results in fewer out-of-vocabulary words. For our multitask model, the Reddit dataset is large enough that we do not necessarily require pre-trained word embeddings. However, it is possible the pre-trained GloVe embeddings provide slightly better performance on the SNLI task.[5] Secondly, our multi-task model learns two tasks simultaneously, balancing performance between them, while InferSent only optimizes performance on SNLI. As will be presented below, our multi-task model performs better on STS. We suspect multi-task training both increases coverage of different language phenomenon and acts as a regularizer across tasks that prevents the resulting sentence embeddings from overfitting any particular task, thus improving transfer performance to new tasks.[6]

4.3 STS Benchmark

The proposed models encode text into a sentence-level embedding space. We evaluate the extent to which the embeddings accurately encode sentence-level meaning using the Semantic Tex-

[5] Preliminary experiments with pre-trained embeddings on a P@N Reddit response prediction evaluation revealed no performance advantage over embeddings learned directly from the data.

[6] We note that, if our model is reduced to just training on SNLI without multitask training on Reddit, it would be equivalent to InferSent but without the use of pretrained sentence embeddings. We do not provide results for this configuration as preliminary experiments suggested it performed poorly.

使用STS的数据进行微调，对输出的编码进行一次转化.

不微调

	dev	test
Reddit+SNLI tuned	**0.835**	**0.808**
Reddit+SNLI	0.814	0.782
Reddit tuned	0.809	0.781
Reddit	0.762	0.731
Neural representation models		
CNN (HCTI)	**0.834**	**0.784**
InferSent	0.801	0.758
Sent2Vec	0.787	0.755
SIF	0.801	0.720
PV-DBOW	0.722	0.649
C-PHRASE	0.743	0.639
Feature engineered and mixed systems		
ECNU	**0.847**	**0.810**
BIT	0.829	0.809

Table 3: Pearson's r on the STS Benchmark.

tual Similarity (STS) Benchmark. The benchmark includes English datasets from the SemEval/*SEM STS shared tasks between 2012 and 2017 (Cer et al., 2017; Agirre et al., 2016, 2015, 2014, 2013, 2012). The data include 8,628 sentence pairs from three categories: *captions*, *news* and *forums*. Each pair is annotated with a human-labeled degree of meaning similarity, ranging from 0 to 5. The dataset is divided into train (5,749), dev (1,500) and test (1,379).

We report results using two configurations for the evaluation of the Reddit and Reddit+SNLI models. The first configuration is "out-of-the-box" with no adaptation for the STS task. Rather, we take the original sentence embeddings u, v and directly score the sentence pair similarity based on the angular distance between the two vectors, $-\arccos\left(\frac{uv}{||u||\ ||v||}\right)$.[7] We suspect the original sentence embeddings from the Reddit and Reddit+SNLI models will not necessary weight all semantic distinctions in a way that is consistent with the annotations for STS. The second configuration for evaluating the two models uses a single transformation matrix to fine-tune the sentence embedding representations for the STS task. The matrix, which is parameterized using the STS training data, transforms the original sentence embedding vectors u, v to u^*, v^*.

Table 3 presents results on the dev and test sets of the STS Benchmark. For model comparisons, we include the state-of-the-art neural STS model CNN (HCTI) (Shao, 2017) and other systems in Cer et al. (2017).[8] The untuned Reddit model is competitive with many of the other neural representation models, demonstrating that the sentence embeddings learned on Reddit conversations do keep text with similar semantics close in embedding space. The "out-of-the-box" multitask model, Reddit+SNLI, achieves an r of 0.814 on the dev set and 0.782 on test. Using a transformation matrix to adapt the Reddit model trained without SNLI to STS, we achieve Pearson's r of 0.809 on dev and 0.781 on test. This surpasses InferSent and is close to the performance of the best neural representation approach, CNN (HCTI).[9]

The adapted multitask model achieves the best performance among all neural models, with an r of 0.835 on the dev data and 0.808 on test. The results are competitive with state-of-the-art feature engineered and mixed systems, e.g. ECNU and BIT. However, our models are simpler and require no feature engineering.[10]

4.4 CQA Subtask B

To further validate the effectiveness of sentence representations learned from conversational data, we assess the proposed models on subtask B of SemEval Community Question Answering (CQA) (Nakov et al., 2017). In this task, given an "original" question Q, and the top ten related questions from a forum $(Q_1, \ldots, Q_{10})$ as retrieved by a search engine, the goal is to rank the related questions according to their similarity with respect

[7] arccos is used to convert the cosine similarity scores into angular distances that obey the triangle inequality.

[8] InferSent (Conneau et al., 2017), Sent2Vec (Pagliardini et al., 2017), SIF (Arora et al., 2017), PV-DBOW (Lau and Baldwin, 2016), C-PHRASE (Kruszewski et al., 2015), ECNU (Tian et al., 2017) and BIT (Wu et al., 2017).

[9] For both the STS shared task and the STS benchmark leaderboard, systems are allowed to use external datasets as long as they do not make use of supervised annotations on data that overlap with the evaluation sets. InferSent introduced the use of SNLI for STS. However, we discovered 4 out of the 1,379 pairs within the STS Benchmark dev set and 5 out of the 1,500 pairs in the STS Benchmark test set overlap with the SNLI training set. We do not believe this minimal overlap had a meaningful impact on the results presented here.

[10] As summarized by Cer et al. (2017), ENCU makes use of a large feature set that includes: n-gram overlap; edit distance; longest common prefix/suffix/substring; tree kernels; word alignment based similarity; summarization and MT evaluation metrics; kernel similarity of bags-of-words and bags-of-dependency triples; and pooled word embeddings. The manually engineered features are combined with scores from DAN and LSTM based deep learning models. BIT relies primarily on a measure of sentence information content (IC) with a non-trivial derivation that is optionally combined with either an alignment based similarity score or the cosine similarity of IDF weighed summed word embeddings.

	dev				test			
	all	captions	forums	news	all	captions	forums	news
Reddit+SNLI	**0.814**	**0.885**	**0.756**	**0.646**	**0.782**	**0.891**	**0.764**	**0.585**
Reddit	0.762	0.815	0.751	0.632	0.731	0.816	0.759	0.578
Reddit+SNLI tuned	**0.835**	**0.888**	**0.759**	**0.731**	**0.808**	**0.894**	**0.767**	0.667
Reddit tuned	0.809	0.843	0.754	0.721	0.781	0.843	0.762	**0.668**

Table 4: Pearson's r of the proposed models on the STS Benchmark with a breakdown by category.

	Score	Label	STS Input Sentences
Good	-0.51	4.2	S1: a small bird sitting on a branch in winter. S2: a small bird perched on an icy branch.
Good	-1.23	0.0	S1: microwave would be your best bet. S2: your best bet is research.
Bad	-0.42	2.2	S1: a little boy is singing and playing a guitar. S2: a man is singing and playing the guitar.
Bad	-0.45	1.0	S1: yes, you have to file a tax return in canada. S2: you are not required to file a tax return in canada if you have no taxable income.

Table 5: Example model and human similarity scores on pairs from the STS Benchmark. System scores are reported as the negative angular distance between the sentence embeddings. The scores can range from 0 to $-\pi$, but in practice are typically between 0 and -$\frac{1}{2}\pi$.

	MAP
Reddit+SNLI	**47.42**
Reddit	47.07
KeLP-contrastive1	**49.00**
SimBow-contrastive2	47.87
SimBow-primary	47.22

Table 6: Mean Average Precision (MAP) on Community Question Answering (CQA) subtask B.

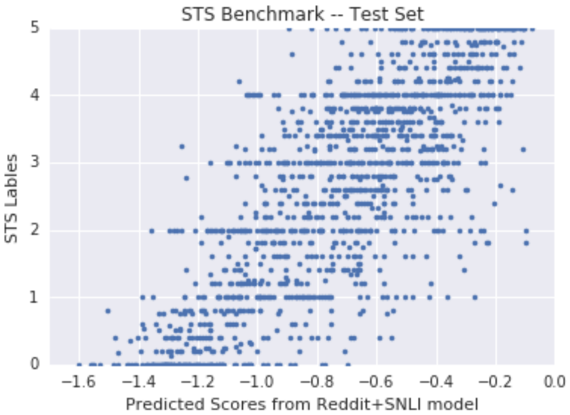


Figure 7: Predicted semantic similarity scores vs. ground truth on the STS Benchmark.

to the original question. Mean average precision (MAP) is used to evaluate candidate models.

Each pairing of an original question and a related question (Q, Q_i) is labeled "PerfectMatch", "Relevant" or "Irrelevant". Both "PerfectMatch" and "Relevant" are considered as *good questions*, which should rank above "Irrelevant" ones.

Similar to the STS experiments, we use cosine similarity between the original question and related questions, without considering any other interaction between the two questions.[11] Given a related question Q_i and its original question Q, we first encode them into vectors u_i and u. Then the related questions are ranked based on the cosine similarity with respect to the original question, $cos(u_i, u)$. Results are shown in table 6. SimBow (Charlet and Damnati, 2017) and KeLP (Filice et al., 2017), which are the best systems on the 2017 task, are used as baselines.[12] Even without tuning on the training data provided by the task, our models show competitive performance. Reddit+SNLI outperforms SimBow-primary, which official ranked first during the 2017 shared task.

[11] Our model also excludes the use of comments and user profiles provided by CQA as optional contextual features.

[12] In the competition, each team can submit one primary run and two contrastive runs. Only the primary run is used for the official ranking.

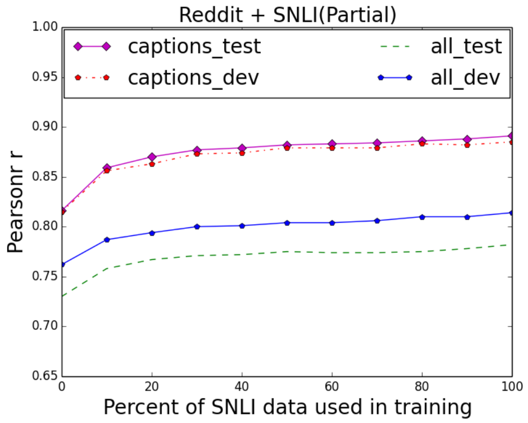


Figure 8: Pearson's r on the STS Benchmark for the multitask model trained with Reddit and varying amounts of SNLI data.

5 Analysis

Model performance on the STS Benchmark can be partition by sentence pair source. The test set contains 625 sentence pairs drawn from *captions*, 500 pairs from *news* data, and 254 from online *forums*.

Table 4 provides results on each sub-group. For the *captions* category, adding the SNLI data improves the baseline Reddit model by about 8% absolute. Even with tuning to STS, mixing in SNLI data still helps dramatically on *captions*, as the STS tuned Reddit+SNLI model is 5% absolute higher than the STS tuned Reddit model on this category. The improvement is likely attributed to the fact that the SNLI sentences are from image captions, while Reddit doesn't contain much caption-style data. Training with the SNLI data has a smaller impact on performance for the other categories, with even a slight decrease for the STS tuned models on *news* test.

增加了SNLI的训练任务之后，在STS的caption类别上有大幅提升可能是因为

We observe that the STS tuned models have only modest performance improvements on the forum data over the untuned models, with much larger improvements for *captions* and *news*. Moreover, for the Reddit+SNLI models, tuning produces a large performance increase for *news* with smaller increases for both *captions* and *forums*. This suggests tuning is impart compensating for domain limitations within the training data.[13] Further improvements on the STS Benchmark could likely be achieved by including additional encoder training data sourced from news data.

Figure 7 plots predicted similarity scores against the ground truth labels within the STS Benchmark test data. The figure shows that while the predicted scores are correlated with human judgment, there is still a sizable range of predicted similarity values for any given gold STS label. We provide examples of good and bad similarity predictions in table 5. For the two good examples, the model correctly has a relatively high similarity score for the first pair, and a relatively low score for the second. For the first bad example, the model fails to penalize its similarity score based on the semantic distinction between "boy" and "man" as much as human raters did. For the second bad example, apparently being on the topic of whether it is necessary to file Canadian tax returns was enough for the model to assign a high similarly score. Human raters correctly assigned a low similarity score since the two sentences are making very different claims.

5.1 Quantity of SNLI data and Performance

The experiments in the previous section show that supervised in-domain data, SNLI's image captions, can be used to improve the semantic representations of in-domain (*caption*) sentences. However, supervised data is difficult to obtain, especially on the order of SNLI's 570,000 sentence pairs. In order to learn how much supervised data is needed, we train multitask models with Reddit and varying amounts of SNLI data, ranging from 10% to 90% of the full dataset.

Figure 8 shows the STS Benchmark results for all data and for caption data only, on both dev and test sets. When first adding the SNLI data into the training task, Pearson's r increases rapidly across all measures. Even with only 10% of the SNLI data, r reaches around 0.85 for captions data on both dev and test. The curves mostly flatten out after using 40% of the data, with performance only improving slightly past this point. This suggests encoders trained primarily on Reddit data can be efficiently adapted to perform well on other domains using a small sample of in-domain data.

使用一部分SNLI有监督数据即可.

6 Related Work

The STS task was first introduced by Agirre et al. (2012). Early methods focused on lexical semantics, surface form matching and basic syntactic similarity (Bär et al., 2012; Jimenez et al., 2012). More recently, deep learning based methods became competitive (Shao, 2017; Tai et al., 2015).

[13] e.g., the Reddit+SNLI model is trained on image caption and discussion forum data but not news.

One approach to this task is to encode sentences into sentence-level embeddings and then calculate the cosine similarity between the encoded representations of the sentence pair. The encoding model can be directly trained on the STS task (Shao, 2017) or it can be trained on an alternative supervised (Conneau et al., 2017) or unsupervised (Pagliardini et al., 2017) task. The primary contribution of the work described in this paper falls into the latter category, introducing a new unsupervised task based on conversational data that achieves good performance on predicting semantic similarity scores. Training on input-response data has been previously shown to be effective at email response prediction (Kannan et al., 2016; Henderson et al., 2017). We extend prior work by exploring the effectiveness of representations learned from conversations in capturing general-purpose semantic information. The approach is similar to Skip-Thought (Kiros et al., 2015), which learns sentence-level representations through prior and next sentence prediction within a document. However, within our work, the adjacent sentences are pulled from turns in a conversation.

使用对话数据进行无监督的训练，然后在语义相似度上取得不错的效果.

7 Conclusion

In this paper, we propose using conversational response prediction models to obtain sentence-level embeddings. We find that encodings learned for conversational response prediction perform well on sentence-level semantic similarity. Sentence embeddings extracted from a model trained on conversational data can be used to obtain results on the STS Benchmark that are competitive with well performing models based on sentence-level encoders. A multitask model trained on response prediction and SNLI achieves state-of-the-art performance for sentence encoding based models on the STS Benchmark, and surpasses prior work that trained on SNLI alone (InferSet). Finally, even without any task-specific training, the sentence embeddings obtained from both the conversational response prediction model and the multitask model that includes SNLI are competitive on CQA subtask B.

Acknowledgments

We thank the anonymous reviewers and our teammates from Descartes and other Google groups for their feedback and suggestions, particularly Dan Gillick and Raphael Hoffman.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

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 *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

Rami Al-Rfou, Marc Pickett, Javier Snaider, Yunhsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv preprint arXiv:1606.00372*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations (ICLR)*.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main* *conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 435–440. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and cross-lingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Delphine Charlet and Geraldine Damnati. 2017. Simbow at semeval-2017 task 3: Soft-cosine semantic similarity between questions for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 315–319.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2017. Natural language inference with external knowledge. *arXiv preprint arXiv:1711.04289*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 681–691, Copenhagen, Denmark. Association for Computational Linguistics.

Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. 2017. Kelp at semeval-2017 task 3: Learning pairwise patterns in community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 326–333.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602 – 610. IJCNN 2005.

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012. Soft cardinality: A parameterized similarity function for text comparison. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 449–453. Association for Computational Linguistics.

Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufman, Balint Miklos, Greg Corrado, Andrew Tomkins, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2016).*

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Germán Kruszewski, Angeliki Lazaridou, Marco Baroni, et al. 2015. Jointly optimizing word representations for lexical and sentential tasks with the c-phrase model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 971–981.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Processings of ACL Workshop on Representation Learning for NLP*, page 78.

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 '17, Vancouver, Canada. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Junfeng Tian, Zhiheng Zhou, Man Lan, and Yuanbin Wu. 2017. Ecnu at semeval-2017 task 1: Leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Adina Williams, Andrew Drozdov, and Samuel R Bowman. 2017. Learning to parse from a semantic objective: It works. is it syntax? *arXiv preprint arXiv:1709.01121*.

Hao Wu, Heyan Huang, Ping Jian, Yuhang Guo, and Chao Su. 2017. Bit at semeval-2017 task 1: Using semantic information space to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84.