Generating Texts for Natural Language Inference

Arianna Yuan
Dept. of Psychology
xfyuan@stanford.edu

Te-Lin Wu
Dept. of Electrical Engineering
telin@stanford.edu

Ting-Po Lee
Dept. of Statistics
tingpo@stanford.edu

Abstract

In the current study, we address the question if deep recurrent neural network can make logical inference without explicit knowledge of logic, grammar and semantics. Particularly, we train Long Short-Term Memory based recurrent neural networks to generate sentences that are either entailed or contradicted by the given sentences. Using the similar model architecture, we build a multi-tasking model to generate two sentences among which one is entailed and the other is contradicted by the source sentence. The models learns the task even when no training example has the target sentences of both tasks. We analyze the hidden-layer representations of generated sentences and find that the network learns to project the source sentences to orthogonal sentence spaces in order to get the contradictory and entailed sentences. Our study demonstrates the great power of sequence-to-sequence learning neural network models in making logical inference with natural language. It also shows the plausibility and the mechanisms of performing multiply logical inference tasks simultaneously using one model.

1 Introduction

In the current study, we aim at using neural networks to generate sentences that have logical relationships with given sentences. Particularly, we want to generate sentences that are either entailed or contradicted by the source sentences. The challenge of this task is that the model needs to output not only interpretable, sensible sentences but also sentences having specific logical relationships with the given sentences. In addition, no explicit syntactical and semantic knowledge about language and logic is built into the system before the training, which makes the task more challenging but also more exciting.

We are not aware of any previous research that directly uses neural network to generate sentences that are entailed or contradicted by the given premises. However, there are several lines of research relevant to the current work. The first line of research is automatic theorem proving using symbolic approaches. Logical provers have been used to generate logical consequences of given statements in logical forms for many years. When premises are given in logical forms, it is very easy to apply logical inference rules, such as resolution, to obtain its logical consequence. However, transforming natural language to formal logic is itself a non-trivial problem, since natural language contains idioms, pragmatic expressions, etc. A system that first transform natural language to logical forms and then compute logical inference based on logical forms often ends up being very brittle (Akhmatova, 2005). Therefore, researchers have proposed *natural logic*, a logic whose vehicle of inference is natural language (MacCartney and Manning, 2007). This system identifies valid inferences based on their lexical and syntactic features, without fully interpreting their semantics (MacCartney and Manning, 2008). With natural logic, inference can be viewed as a finite state machine, and you can use it to generate valid or invalid statements (Angeli and Manning, 2014). Although it is very reliable, it requires prior knowledge of logic.

The logical systems using *natural logic* is much more robust. However, it still requires prior knowledge about logic to be built into the system. The second line of research that is relevant to the current study is neural network models in logical inference. Compared with the two kinds of models mentioned above, neural network models require less knowledge about logic itself, although they do require some knowledge about natural language, such as syntactic structures. Bowman and colleagues proposed a tree-structured recursive neural network (TreeRNN) to learn logical semantics (Bowman, 2013; Bowman et al., 2014). Their models are powerful in extracting the logic relationship between two sentences. It demonstrates the ability of neural network models to make logical inference without explicit knowledge of logic. However, it is hard to apply these models to the task of generating sentences that are entailed or contradicted by given sentences.

Although no previous research has applied neural network models to generate sentences that are entailed or contradicted by given premises, many researchers have used them to generate other texts, e.g., in text compression and machine translation (Cho et al., 2014). The most relevant one to the current research is the sequence-to-sequence learning model proposed in those studies. For instance, Graves (2013) demonstrates that word-level recurrent neural networks (RNNs) with long-short-term-memory (LSTM) cells are able to generate text with Wikipedia data as training examples (Graves, 2013). Also, Sutskever et al. propose a Long Short-Term Memory (LSTM) based neural network to learn machine translation (Sutskever et al., 2014). The model takes sentences with varying lengths as inputs. Word embeddings are sequentially fed into the LSTM cells until EOS is detected, then the translated sentence is returned word-by-word by LSTM cells.

There are other variations of RNN model in solving the text generation task. For example, the RNN model proposed by Cho et al. uses a cell design simpler than LSTM but pertains the ability to remember and forget. The cell architecture is known as Gated Recurrent Unit (GRU), where a reset gate r and an update gate z are used to determine whether to ignore previous hidden states and whether to update the remembered hidden states with new hidden states, which is a function of the new input. GRU has the ability to adaptively carry over input states to construct long-term dependencies and its simple design gives it computational and implementation advantages.

The final line of relevant research is the multi-tasking neural network models. It would be fantastic if we could generate the entailed sentence and the contradictory sentences using a single model. This is analogous to the multi-tasking problem in machine translation. Training a separate model each time for a different language translation is very expensive. To solve this problem, Luong and colleagues have developed multitasking MT model (Luong et al., 2015). There are three types of multi-task learning for sequence-to-sequence models: (1) one-to-many: the encoder is shared between tasks, like in multilingual machine translation or semantic parsing. (2) many-to-one: one decoder, like in image captioning system. (3) many-to-many: encoders and decoders are shared. This work proposed an interesting idea, i.e., models for translation can be fine-tuned after training on image captioning and parsing. The loss functions of the two tasks are different and the errors can be back propagated to perform simultaneous training. Multi-task training paradigm can be applied to many other applications, as long as the decoders are well-defined and trained end-to-end. The "one-to-many" framework is particularly relevant to our work. In this paper, we also attempt to understand how these models work in the domain of natural language inference.

2 Current Approach

We are inspired by the above-mentioned machine translation neural network models. Particularly, we are curious to know if they can solve the logical inference problem as well. Therefore, we plan to use an approach similar to the sequence-to-sequence learning model in machine translation. At each time step, the encoder LSTM RNN takes one word of the premise. Its hidden state at last time step is then fed into a decoder LSTM RNN to predict a sentence that is entailed by the premise word by word. Basically, LSTM has three gates, i.e., input gates, forget gates and output gates. Input gates control the extent to which the current words should be encoded, the forget gates control how much information from the previous time step is forgotten, and output gates determine how much the activation of the current LSTM cell contribute to the current prediction. The memory cells endow the networks with great power to capture long-term dependencies. This is critical for encoding and generating texts that have many long-range regularities. It also helps alleviate the problem of vanishing gradients, since the back-propagation of error signals are gated and the information of long-term dependencies are maintained by the memory cell. We have 2 hidden layers, and each of them has 256 units. Figure 1 illustrates the current model's architecture. Details of LSTM RNN can be found in Sutskever et al. (2014). For comparison, we also use GRU RNN to perform these tasks.

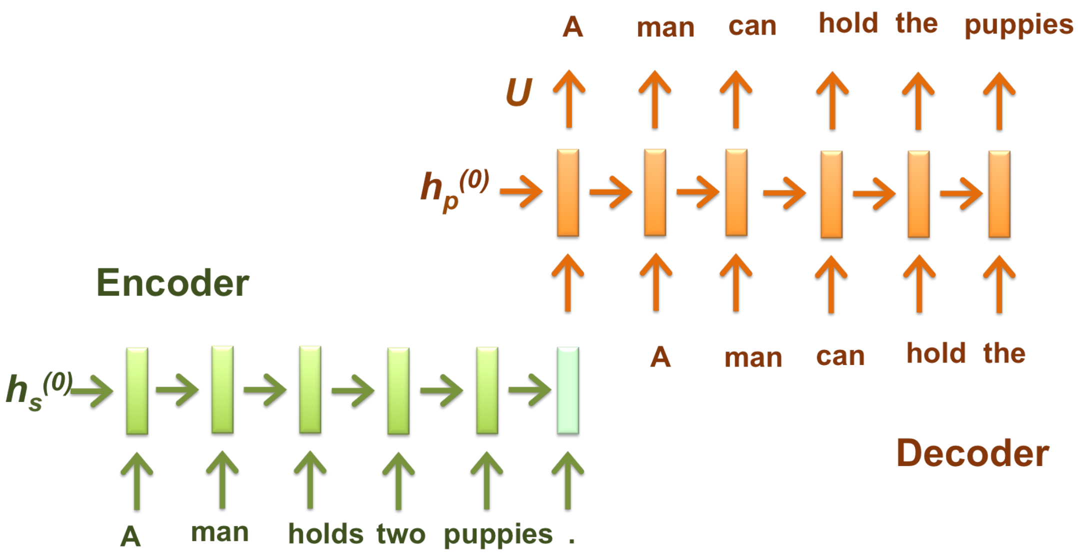

Figure 1: Model Architecture for Entailment Task.

We further apply the multi-tasking machine translation model to perform multiple logical inference tasks simultaneously. Specifically, the encoder of the premises are shared, but the decoders for different logical inference tasks are separated. Using different decoders, the network can either generate a sentence that is entailed by the source sentence or contradicted by the source sentence. The model architecture is illustrate in Figure 2. The challenge for the multi-tasking logical inference is that we do not have the data for one-to-many task, i.e., for any given source sentence, we only have either the sentence it entails or the sentence it contradicts, but not both of them. This is different from Luong and colleague's study, in which both the German translation and the parsing of the same English sentence is available (Luong et al., 2015). To tackle this challenge, we create a dataset by combining the Entailment Dataset and the Contradiction Dataset. We set the target sentence to be the source sentence if the desired target is not available. For instance, a training example created from the Entailment Dataset looks like (X: source sentence, Y: {entailment: entailed sentence, contradiction: source sentence}), whereas a training example created from the Contradiction Dataset looks like (X: source sentence, Y: {entailment: source sentence, contradiction: contradicted sentence}). In the training phase, we feed the model with Entailment and Contradiction data alternately. The model updates the weights of the encoder and the weights of the decoder whose target sentence are available in the current batch of data .

We use cross-entropy errors between the predicted sentences and the target sentences as our loss function, which we try to minimize. We also calculate the BLEU score of the generated sentences, an evaluation metrics widely used in machine translation (Papineni et al., 2002). It indicates the extent to which the output sentences deviate from the target sentences.

3 Experiments and Results

3.1 Experiment 1 on the Entailment Dataset

We tailor Tensorflow codes of sequence-to-sequence learning model to suit our goals, and we use the Stanford Natural Language Inference (SNLI) Corpus dataset (Bowman et al., 2015). It contains manually created logically meaningful sentence pairs. Although the parse trees of the sentences are provided, they are not used. All training examples are labeled with one of 3 logical relations, i.e., entailment, contradiction, and neutral. In Experiment 1-2, we use the SNLI sample dataset, which contains 6000 entailment examples (5000 for training and 1000 for development), 6000 contradiction examples (5000 for training and 1000 for development) and some neutral examples. Neutral examples are excluded

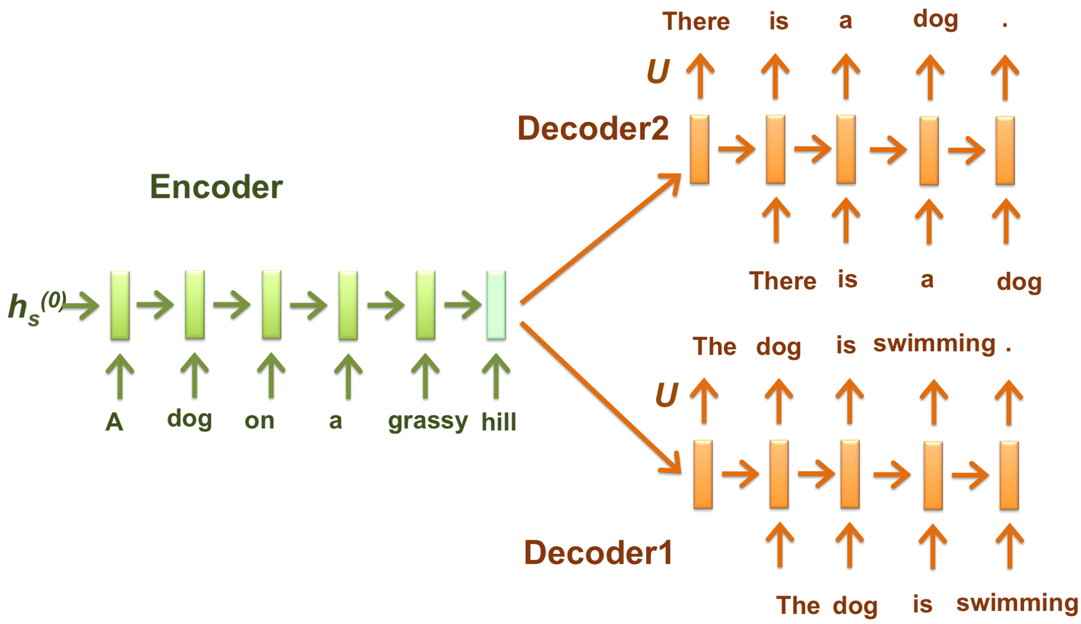

Figure 2: Model Architecture for Multi-task.

source	target	predicted
Two feet wearing yellow ski boots on ski.	Two boots are touching a ski.	People with boots.
Three little girls smiling.	Three smiling little girls.	The girls are smiling.
a beautiful sailboat out in the ocean.	The sailboat is out on the water.	People are in the water.
A girl picking dandelions in a grassy meadow.	A girl is picking flowers.	a girls picks flowers.
Several people standing in the street.	people were outside.	people are standing in the street .

Table 1: Test examples of LSTM RNN on Entailment Dataset.

because they are not related to our goal. We create the Entailment Dataset consisting of all examples labeled entailment, the Contradiction Dataset consisting of all examples labeled contradiction, and the Combined Dataset that is the combination of the two. To facilitate batch training, we use several buckets and pad each sentence to the length of the bucket above it, which is a standard technique in natural language processing. We evaluate sentences belonging to 3 buckets: bucket 0 (source sentence: 5 words, target sentence: 10 words), bucket 1 (source sentence: 10 words, target sentence: 15 words) and bucket 2 (source sentence: 20 words, target sentence: 25 words).

In the first experiment, we train LSTM RNN and GRU RNN to generate sentences that are entailed by the source sentences. Table 1 shows a couple of predictions by our LSTM RNN trained on Entailment dataset.

We can see that the model not only learns to generate sensible sentences, but also sentences that are logically entailed by the source sentences. Due to the limit of space, we do not list the test examples of GRU RNN, but they yield qualitatively similar results. We then look at the loss history of our models to examine its learning curve (Figure 3, left).

The first thing we notice is that, while the training perplexity is constantly decreasing, the validation error has a U-shaped curve, indicating over-fitting as the number of training steps becomes large. However, the large iteration number have different effects on different sentence lengths. Particularly, performance for short source sentences would benefit from more iterations (bucket 0 in the figure) but performance for long source sentences are greatly harmed by more training (bucket 2). Therefore, we need to define our optimal stopping time depending on the average length of the source sentences.

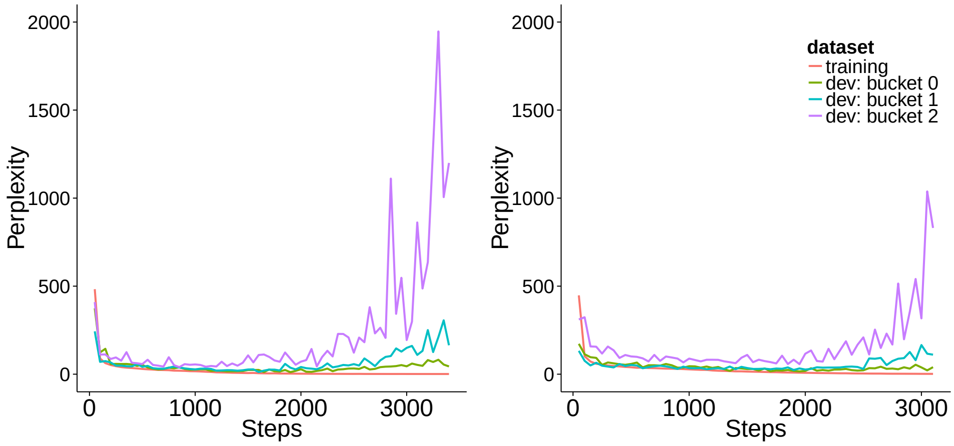

Figure 3: Left: Loss History of LSTM RNN on Entailment Dataset. Right: Loss History of LSTM RNN on Contradiction Dataset.

source	**target**	**predicted**
People in line for plates of rice.	People are sitting down eating.	People are asleep.
girl posing on tree.	A boy and a bush.	A man is sitting in the bench.
A person in an airplane.	A dog in an airplane.	The person is riding a roller.
a young woman is playing a trombone.	The instrument is a trumpet.	a young man is playing the violin.
A person riding a rodeo bull.	A person is driving to work.	A woman riding a horse.

Table 2: Test examples of LSTM RNN on Contradiction Dataset.

3.2 Experiment 2 on the Contradiction Dataset

In the second experiment, we train LSTM RNN and GRU RNN to generate sentences that contradicts the source sentences. Table 2 shows some predictions by our LSTM RNN on the Contradiction dataset.

Examining those predictions, we find that the model not only generate sentences that contradict the source sentences, but also generate them creatively. In other words, most of the time it is not trying to simulate the target sentences, but inventing its own way to generate contradictions. This exploits the nature of the task that contradiction is not a one-to-one mapping. One sentence can have many sentences that contradict it. The learning curve for contradiction is similar to the one for entailment, as shown in Figure 3, right. We also evaluate our model performance using BLEU points, and the BLEU points reach 33 for "entailment task" and 27 for "contradiction task".

3.3 Experiment 3 on the full data Dataset

Although the model learns to perform the task, the performance is not perfect. Particularly, the model trained on sample data is likely to generate sentences with repetitive words, e.g., "two people sitting sitting". This might due to the small vocabulary size in sample data. Therefore, we further train our model on SNLI full dataset, which has the following statistics: {entailment: {train:183416, dev: 3329, test: 3368}, contradiction: {train: 183187, dev: 3278, test: 3237}}. After the training, the BLEU points of model prediction increase from 33 to 39 for "entailment task" and from 27 to 30 for "contradiction task". We also found that the model is more likely to product sensible sentences. In Table 3, we randomly pick up 10 sentences the models predicted that are entailed by the source sentence, and we find that the sentences produced by the model trained by the full data is more likely to be grammatically correct. Also, the "repetitive word" is largely resolved when full dataset is used.

Interestingly, we find that the model not only learns to generate sentences that are superficially entailed by the source sentences, i.e., paraphrase, but also sometimes exhibits the ability to reason deeply (Table 4). For instance, it learns to reason about intention of a person, which is not explicitly said in the

source	model (sample dataset)	model (full dataset)
A local band performing at a local venue.	A band of musicians music	People are performing .
Boy riding a horse.	A horse riding a horse .	a boy rides a horse
Man fixing a train as it stops.	A man is inside a building	A man is fixing a train .
Two men standing on a street.	Two men standing standing .	There were two men standing on a street
A dog runs in the woods.	A dog is running in the snow .	A dog runs outside .
Two men rest near a mountain range.	two people sitting sitting .	Two men are outside .
Two feet wearing yellow ski boots on ski	People with boots	Two feet are in ski ski .
A window washer is cleaning the bookstore windows.	A skateboarder cleaning cleaning .	A person wants windows to be shiny .
Two soccer players on a soccer field.	Two soccer are soccer soccer soccer .	There are people on a soccer field .
A classroom of students watching a presentation.	A students of students .	A room full of people

Table 3: Comparison of Model Trained on Sample Data and Full Dataset for Entailment Task

property	source	model (full dataset)
intention	A window washer is cleaning the bookstore windows.	A person wants windows to be shiny .
intention	people walking through a lane in supermarket.	Some customers at a supermarket are checking out of the mall .
summarization	A woman grilling chicken and veggies.	A woman is cooking .
common sense	A man is serving drinks to people.	There are people receiving beverages .
common sense	Three women holding hands on a beach.	They are friends .

Table 4: Non-trivial Reasoning by the Model

source sentence. It also learns to summarize a scenario, i.e., to drop the concrete information and returns the gist of a sentence. In addition, it learns to do common sense reasoning, such as inferring one's attitude from behavior, inferring receivers' states based on actors' states. We are excited to see that the model is able to perform relatively complicated reasoning without any hand-crafted features.

For the Contradiction task, we find that our model sometimes generated sentences that are "more contradictory" to the source sentence than the target sentence in the data. For instance, given the source sentence "A lady in jeans and high heels", our model predicts "A lady without shoes", which directly contradicts the source, whereas the target sentence in the dataset, "A woman getting ready to perform surgery", only vaguely contradicts the source.

Human Evaluation To further evaluate the model performance, we randomly select 300 sentences from the Entailment validation set and ask human judges to evaluate the model predictions. The predictions, i.e., the generated sentences, are evaluated in two ways: whether they are grammatically correct and whether they are semantically entailed by the source sentences. We allow the case in which the sentence's grammar is incorrect but the semantics is correct. This is mainly for the sentences with repetitive words. The model trained on sample dataset generates 55% grammatically correct sentences, and 37% sentences that are entailed by the source sentences, whereas the model trained on full dataset generates 94% grammatically correct sentences, and 87% sentences that are entailed by the source sentences. In other words, the model trained on full dataset predicts more grammatically correct sentences, and also more sentences that are entailed by the source sentences.

Error Analysis The first source of error is from repetitive words, e.g. the model sometimes produce the same word successively. The second source of error comes from polysemants. For instance, given the source: "A basketball player is shooting a basketball", the model predicts "A basketball player is taking a shot". Apparently it learns the association between shoot and shot, but it fails to consider the other meaning of the word "shot".

3.4 Experiment 4 Multi-tasking logical inference

We use SNLI full dataset for the multi-tasking model. To facilitate the training, we only include training examples that fit into one bucket, i.e., both the source sentence and the target sentence are less than 15 words. As mentioned in the **Current Approach** we combine the dataset for entailment and contradiction to get a new dataset (183416 training examples from Entailment Dataset, 183187 training examples from Contradiction Dataset). The vocabulary size is 40000. As can be seen in Table 5, the model is able to generate sentences that are entailed or contradicted by novel source sentences after training.

We then examine the loss history of our multi-tasking model (Figure 4, left). We find several intriguing

source	model (entailed)	model (contradicted)
A man reads the paper in a bar with green lighting.	A man is reading .	A man is sitting on a table .
A middle eastern marketplace.	A group of people .	The street is empty .
This child is on the library steps.	The child is on the steps .	The child is in a car .
Some dogs are running on a deserted beach.	There are dogs on the beach .	The dogs are sleeping .
A newlywed couple laughing and talking amongst themselves.	A couple is laughing .	The couple are fighting each other .
A guy and a girl look down a mountain range.	A couple are looking at the mountain .	A guy and a girl are watching TV .

Table 5: Test Examples of LSTM RNN on Multi-tasking Inference.

properties. First, while the training perplexity constantly decreases, the change of validation perplexity clearly demonstrates a U-shaped curve, indicating overfitting as the number of training steps becomes larger. Critically, the optimal stopping time for Entailment Task (step 135×10^3) is different from the optimal stopping time for Contradiction Task (step 145×10^3), meaning that when the model starts to overfit the entailment data, the decoder for the contradiction task is still benefitting from the training. This creates a hard problem, i.e., how to define the optimal stopping time. We could simply stop updating individual decoder depending on the validation perplexity of their corresponding task, but we still need to decide when to stop updating the encoder, which is shared across different tasks. One solution is to assign weights to each task based on which task we care more, and use these weights and the validation perplexities of both tasks to determine when to stop the training.

In addition, we find that although the training loss curves of the Entailment Task and the Contradiction Task largely overlap (solid green line and dashed green line), the validation loss of Contradiction Task is consistently higher than the one of Entailment Task throughout the training. This might because there are presumably more ways to generate contradicted sentences than to generate entailed sentences. Note that here contradiction is not negation, thus it might has less constraint on output sentences than entailment.

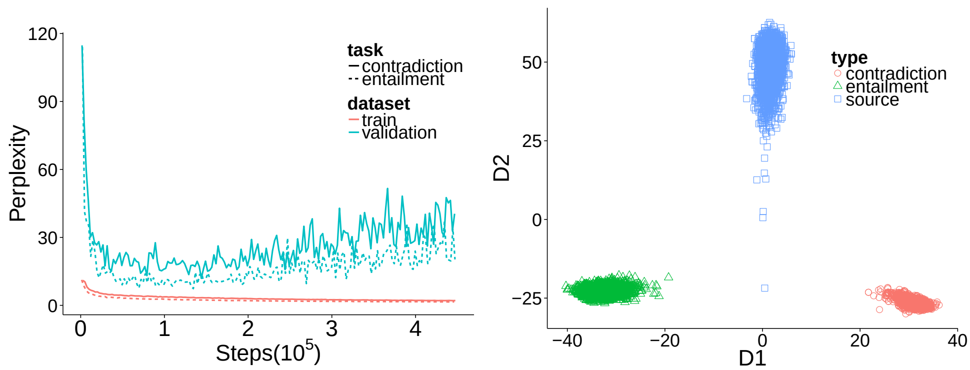

Figure 4: Left: Loss History of LSTM RNN for Multi-task. Right: Principle Components of learned sentence embeddings.

To understand how the model performs the task, we further conduct a Principle Component Analysis on the hidden-layer representations of the source sentences and the generated sentences. We plot the first two principal components of the hidden-state representations of the sentences (Figure 4, right) and find surprisingly that the source sentences and the two generated sentences comprise three distinct clusters, with the difference between the source and the entailed sentences orthogonal to the difference between the source and the contradicted sentences. In other words, the model learns to push the sentence embeddings of the source sentences to orthogonal directions to obtain sentences that are either entailed or contradicted by the source. It indicates that the model not only learns to perform each individual task, but also understands the logical relationship between different tasks. In other words, it learns the meta-cognitive aspect of the tasks.

4 Conclusions

In the current study, we build deep neural networks to generate sentences that have logical relationships with the given sentences. Particularly, we use Long Short-Term Memory based recurrent neural networks to generate sentences that are either entailed or contradicted by the given sentences. Using a similar encoder-decoder model architecture, we also build a multi-tasking model in which the encoder is shared across the Entailment task and Contradiction task, but the decoders are different for these tasks. The model learns to generate both the entailed sentence and the contradicted sentence of the same source sentence even when no single training example has both target sentences for both tasks. Our model demonstrates the great power of deep recurrent neural network models in learning logical inference without prior knowledge of logics and natural language. It also shows the plausibility of performing multiple logical inference tasks simultaneously using one model, which would greatly reduce the demand of computation resources. The visualization of learned representation of the sentences reveals the mechanisms of how the model performs multiple logical inference tasks simultaneously: it learns to map the source sentences to orthogonal sentence spaces, which reflects the opposite nature of the two tasks.

Although there are no perfect, automatic way to determine whether a sentence in natural language form is entailed or contradicted by the given premises, we find that the cross-entropy error works well in our models as the cost function. Despite the fact that the model is not equipped with any explicit knowledge of logic or language, it learns the implicit representation of entailment and contradiction relationship by simulating the target sentences during the training phase.

We also find BLEU score useful in capturing the similarity between the predicted sentences and the target sentences. It also quantitatively demonstrate that the model trained on full dataset has better performance than the model trained on sample data. The limitation of cross-entropy error and BLEU score is that there are more than one correct answers for any given premises. If the model makes a correct inference that does not resembles the target sentence, the error would still be large. That is why we need human evaluation. In our human evaluation, we find that the model achieves an accuracy of 87% for Entailment task.

There are many promising future directions. The first is multi-premise reasoning. The current study only deals with examples of single source sentence, but in daily life we frequently encounter the situation where we need to draw inferences from multiple sources. It would be interesting to see if the current model generalize to multi-premises reasoning. Another interesting direction is multi-step reasoning. In the current study, we only ask the model to make inference once. It would be interesting to see if we could ask the model to generate sentences that are entailed by the sentence it generated. In other words, we can explore if the model can perform a sequence of reasoning. Also, the current work only addresses the issue of multi-tasking for two tasks. We can add more tasks like generating sentences that entail the source sentences and examine if the model would create another cluster of hidden states to represent the new category. The results would shed light on how multi-tasking sequence-to-sequence models learn new tasks.

Finally, it would be interesting to see if the model can learn the task in a completely unsupervised manner, rather than learning from the labeled dataset as in the current study. To elaborate on that, we first ask the model to generate a sentence, then use another neural network such as Samuel Bowman and colleague's network to decide whether this sentence is entailed by the source sentence or not. Therefore, the evaluation network provides teaching signals for the decoder networks. This idea is well aligned with the "self-play" method in the reinforcement learning literature (Silver et al., 2016). Using this approach, we literally ask a model to teach another model to generate sentences and fulfill unsupervised learning.

References

Elena Akhmatova. 2005. Textual entailment resolution via atomic propositions. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 150. Citeseer.

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *EMNLP*, pages 534–545.

Samuel R Bowman, Christopher Potts, and Christopher D Manning. 2014. Recursive neural networks for learning logical semantics. *CoRR, abs/1406.1827*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Samuel R Bowman. 2013. Can recursive neural tensor networks learn logical reasoning? *arXiv preprint arXiv:1312.6192*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Bill MacCartney and Christopher D Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.