Deep Learning Techniques in Textual Entailment

Anish Mishra
Computer Science and Eng.
IIT Bombay
anish@cse.iitb.ac.in

Pushpak Bhattacharyya
Computer Science and Eng.
IIT Bombay
pb@cse.iitb.ac.in

Abstract

Textual Entailment relationship between two piece of text deals with deriving whether one implies the meaning of another. Textual Entailment is applied in various fields of NLP like Machine Translation evaluation, Text Summarization, Question Answering. Therefore good research in Textual Entailment will help other areas. Deep Learning Techniques are being successfully applied in many areas in NLP such as Machine Translation, Sentiment Analysis, etc. It has also started to be applied in Textual Entailment(TE) where various neural network based systems are being recently designed. These research has been possible due to introduction of high quality datasets such as SNLI and MNLI. These are manually annotated datasets consisting large number of sentences essential for training deep learning models. In this paper we present some of the recent works building neural network based Entailment system using LSTM, CNN and attention.

1 Introduction

Text Entailment or **Natural Language Inference** is a field of study in NLP which deals with understanding the meaning/semantics of sentences/text-pieces. Understanding the meaning of a given piece of text is an open problem in NLP. Given two general English sentences the Inference/Entailment system can conclude whether meaning of one implies the other.

S1: "A girl playing a violin along with a group of people."

S2: "A girl is playing an instrument."

In the above two sentence the meaning of S1 implies that S2 is also true. Many applications in NLP, such as Machine Translation Evaluation, Information Retrieval(IR), Question Answering(QA), Information Extraction(IE) and Text Summarization perform language inference task. For example, Machine Translation evaluation systems assign translation scores based on entailment between translated text and the reference text. In Information Retrieval systems, the search results should entail the search query. In QA systems, the answer provided by the system should be entailed by the reference answers provided.

All the above systems perform the task of inferencing independently. The study of Textual Entailment, tries to bring all the task involved in deriving the Entailment relationship under one umbrella. In the process of designing approach for Textual Entailment we try to come up with better approach towards solving this problem. The various techniques available for Recognizing Textual Entailment can be broadly classified into three categories, viz. Classical Rule Based approach, Machine Learning based approach and Deep Learning approach.

Given a pair of sentences, the goal of recognizing textual entailment(RTE) is to determine whether the hypothesis can be reasonably inferred from the premise. There are three relations defined for RTE - Entailment(inferred to be true), Contradiction(inferred to be false) and Neutral(truth unknown). An example of each case is shown in below table.

We have different approaches developed for building Textual Entailment models viz. Rule-based models, Machine Learning models and Deep Learning models. In this paper we will discuss about the deep learning models that are being developed for building Entailment systems.

Premise (P)	Hypothesis (H)	Label
A girl playing a violin along with a group of people.	A girl is playing an instrument .	Entailment
A girl playing a violin along with a group of people	girl is washing a load of laundry .	Contradiction
A girl playing a violin along with a group of people	A group of people are playing in a symphony .	Neutral

2 General Deep Learning Concepts

2.1 Word Embedding

A word embedding $W : words \rightarrow \mathbb{R}^n$ is a parameterized function mapping words in some language to high-dimensional vectors (perhaps 200 to 500 dimensions) (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017). For example, we might find:

$$W(\text{"}boy\text{"}) = (0.2, 0.1, 0.7, ...)$$

$$W(\text{"}toy\text{"}) = (0.4, 0.0, 0.1, ...)$$

Similarly, if we see the words that are closest in the embedding to a given word shown in figure **??** taken from (), we find that it makes lot of sense.

Similar words being close together allows us to generalize from one sentence to a class of similar sentences. This doesn't just mean switching a word for a synonym, but also switching a word for a word in a similar class (eg. "the wall is blue" $\rightarrow$ "the wall is red"). Further, we can change multiple words (eg. "the wall is blue" $\rightarrow$ "the ceiling is red").

Word embeddings exhibit an even more remarkable property: analogies between words seem to be encoded in the difference vectors between words. For example, there seems to be a constant male-female difference vector:

$$\begin{aligned} &W(\text{“}woman\text{”}) - W(\text{“}man\text{”}) \\ &\approx W(\text{“}aunt\text{”}) - W(\text{“}uncle\text{”}) \end{aligned} \tag{1}$$

$$\begin{aligned} &W(\text{“}woman\text{”}) - W(\text{“}man\text{”}) \\ &\approx W(\text{“}queen\text{”}) - W(\text{“}king\text{”}) \end{aligned} \tag{2}$$

2.2 Recurrent Neural Network(RNN)

The idea behind RNNs is to make use of sequential information. In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But for many tasks that's a very bad idea. If you want to predict the next word in a sentence you better know which words came before it. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being dependent on the previous computations.

One major drawback on RNNs is that they are limited to looking back only a few steps because of vanishing gradient problem. LSTMs as described in the next section overcome this problem.

2.3 Long Short Term Memory(LSTM)

Long Short Term Memory networks - usually called "LSTMs", were introduced in 1997 (Hochreiter and Schmidhuber, 1997) are a special kind of RNN, capable of learning long-term dependencies. There usefulness has only recently realized.

Let $X = (x_1, x_2, ..., x_N)$ denote an input sequence. At each position $k(1 \leq k \leq N)$, there is a set of internal vectors, including an input gate i_k, a forget gate f_k, an output gate o_k and a memory cell c_k. All these vectors are used together to generate a d-dimensional hidden state h_k as follows

$$\begin{aligned} i_k &= \sigma(W^i x_k + V^i h_{k-1} + + b^i) \\ f_t &= \sigma(W^f x_k + V^f h_{k-1} + + b^f) \\ o_t &= \sigma(W^o x_k + V^o h_{k-1} + + b^o) \\ \tilde{c_k} &= tanh(W^c x_k + V^c h_{k-1} + b^c) \\ c_k &= f_k * c_{k-1} + i_k * \tilde{c_k} \\ h_k &= o_k * tanh(c_k) \end{aligned}$$

3 Dataset

Recognizing Textual Entailment(RTE) dataset from 2005-11 has been primary dataset being used for building Entailment system. These dataset consist of 1000-5000 sentences. The number of sentences is good for evaluating rule based Entail-

ment models and training simple machine learning based model. However they are very small to train neural network model. require more There are mainly two datasets introduced for the purpose of training neural network based Entailment models.

3.1 SNLI

Stanford Natural Language Inference(SNLI) dataset (Bowman et al., 2015) by NLP group at Stanford was released for the sole purpose of enabling researchers build end to end neural network models for Text Entailment. With 570k pairs of human annotated data, the dataset is largest dataset available publicly. Since its creation, the SNLI corpus has become a vital benchmark for researchers in the field with many models showing better results than machine learning models that depends on many external features.

SNLI is a new, freely available collection of 570K sentence pair, each labelled with one of the following relationship: entailment, neutral and -, where - indicates lack of consensus from the human annotators.

Indeterminant event and entity coreference is a big challenge in annotating corpora.

Eg. "Boat sank in the Pacific Ocean" and "Boat sank in Atlantic Ocean", can be contradiction or neutral depending on if sentences refers to same event or different.

The above problem is solved by showing image caption from Flickr30k to Amazon Mechanical Turk(crowd sourcing platform) workers and asking them to provide three sentence each of entailment, contradiction and neutral. Since all the sentences, the one shown and three sentences provided by workers refers to same image, they are linked by the same event.

Some example pair of sentences taken from the dataset are shown in table 1.

3.2 MNLI

A drawback of SNLI corpus is that all of the sentences were extracted from a single genre - *image captions* and are limited to descriptions of concrete visual scenes, rendering the sentences short and simple, and making the handling of many key phenomena like tense, belief, and modality irrelevant to task performance. Due to these reasons the dataset is not sufficiently demanding to serve as an effective benchmark for NLU with current best model () falling within a few percentage points of human accuracy.

To remedy these limitations of SNLI, (Nangia et al., 2017) have released Multi-Genre NLI (MNLI) corpus. MNLI consists of 433k sentence pairs collected similarly as SNLI, but unlike that corpus MNLI consists of ten distinct genres of written and spoken English, covering most of the complexity of the language. The ten different genres are as listed below -

- face-to-face speech
- telephone transcriptions
- the 9/11 report
- travel guides
- letters
- nonfiction books
- magazines
- news articles
- government documents
- fiction

All of the sources are present in test and development sets, but only five are included in the training set. This is done to evaluate the models both on the quality of their text representation for any of the training genres and also derive good representation outside of those genres.

Some of the examples sentences taken from the dev set of dataset are shown in Table 2.

4 Deep Learning Entailment Models

Application of neural network based models in Textual Entailment has been possible with the introduction of SNLI (Bowman et al., 2015) and MNLI (Nangia et al., 2017). These are manually annotated high quality entailment dataset. It consists of pair of sentences with a label - Entailment, Neutral and Contradiction indicating relationship between the two sentence. The neural network models can be classified two major categories -

- **Sentence encoding based models**: building a sentence representation by encoding the given sentences using sequential encoder such as RNN and LSTM.

Text	Judgment	Hypothesis
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Table 1: Examples from SNLI

Text	Judgment	Hypothesis
He turned and saw Jon sleeping in his half-tent.	FICTION entailment E N E E	He saw Jon was asleep.
8 million in relief in the form of emergency housing.	GOVERNMENT neutral N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
If you need this book, it is probably too late unless you are about to take an SAT or GRE.	VERBATIM contradiction C C C N	Its never too late, unless youre about to take a test.

Table 2: Examples from MNLI

– **Match encoding based models**: directly models the relationship between two given sentences using different types of attention mechanism.

We will describe these models in below sections.

4.1 Sentence Encoder Based Model

The paper by (Bowman et al., 2015) that introduces us to SNLI corpus also describes a simple baseline neural network models. In their model they take the distributed word representation(also known as word vector) of individual words in a sentence and combines them to build sentence vector representation that captures the meaning of the sentences. The sentence representation for both premise and hypothesis is passed on to classifier for predicting the correct relationship - entailment, contradiction and neutral. An outline of the model is shown in the figure 1.

A simple sentence vectors can be built from individual word vector using three methods described below -

– Bag of Words - Word vectors of individual

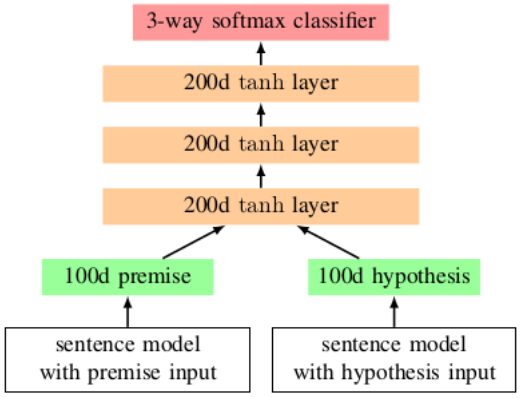

Figure 1: (Bowman et al., 2015) Neural Network model

words in the sentence are summed and averaged. While this preserves the semantic properties of words but word ordering in the sentence is lost.

– RNN - The sentence is considered as sequence of words. Therefore, to capture the word order in a sentence, the word vectors are passed to RNN starting from the first word sequentially till the last. This is done separately for both premise and hypothesis.

– LSTM - Due to the problem of exploding and vanishing gradients in RNNs, they cannot remember dependency among words for long distances. LSTMs with their memory cells are able to remember word dependency for long sentences. They are used in place of RNN to build better sentence encoding in same way as described in the above point for RNN.

The output of the sentence encoder is further passed on to the classifier layer. The neural network classifier can be simply stack of 'n' $200d$ $tanh$ layers. The accuracy obtained by using above models is 77.6% on SNLI.

4.2 Match Encoding Based Model

While sentence encoding methods learn to derive good representation for sentences embedding, there has been significant research to the represent the matching information between given sentences for Inference. In matching between two sentences using encoding method we give equal weight to all the words in the sentence. But all the words are not equally important for Inference as shown in below example.

P : "I am playing football today"

H : "I am not playing any sports"

The word "not" in sentence *H* implies the relationship between above sentences to be contradiction and therefore be given more weight during creating sentence encoding.

The above idea was first implemented by (Rocktschel et al., 2015) using **attention mechanism** that significantly improved classification accuracy by 5-6 percentage points. An outline of the (Rocktschel et al., 2015) model is shown in figure 2.

4.3 Other Models

- Mou et al. (2016) used tree-based CNN encoder to obtain sentence encoding and achieved an accuracy of 82.1%.

- Liu et al. (2016) proposed a hybrid model. It used a BiLSTM to create sentence representation and a technique called "Inner Attention" to give more importance to functional or content words in a sentence.

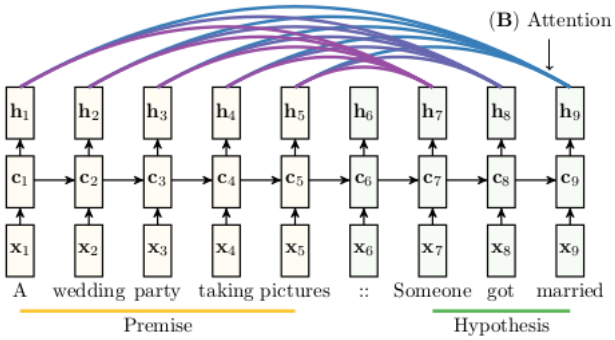

Figure 2: (Rocktschel et al., 2015) Attention model

5 Conclusion

Neural Network models have shown promising results for Textual Entailment Recognition. We see current deep learning techniques like Word Embeddings, RNNs and Attention, Memory Nets being used in this area. Use of better techniques pushes the accuracy scores on NLI datasets like SNLI and MNLI. On the other hand better quality dataset like MNLI has been released to further develop better quality neural network Entailment models.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

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*. Association for Computational Linguistics, pages 632–642. https://doi.org/10.18653/v1/D15-1075.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 160–167.

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

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090* .

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *ACL (2)*. The Association for Computer Linguistics.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R. Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, RepEval@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*. pages 1–10. https://aclanthology.info/papers/W17-5301/w17-5301.

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. http://www.aclweb.org/anthology/D14-1162.

Tim Rocktschel, Edward Grefenstette, Karl Moritz Hermann, Toms Kocisk, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR* abs/1509.06664.