Survey of Textual Entailment Approaches

Rakesh Khobragade, Heaven Patel and Pushpak Bhattacharyya
Indian Institute of Technology, Bombay
{rkhobrag, heaven, pb}@cse.iitb.ac.in

Abstract

Natural Language Inference, or Textual Entailment, has applications in many NLP tasks such as question-answering, text summarization, paraphrase detection, and machine translation evaluation. In this survey, we perform an analysis of developments in the NLI tasks. We discuss various approaches that have been proposed and datasets available to train deep learning models. We also describe a few applications of NLI.

1 Introduction

Textual Entailment, or Natural Language Inference, is considered as one of the major tasks in Natural Language Processing(NLP) that requires deep semantic understanding. For a pair of sentences, textual entailment can be defined as the task of finding if one of them can be inferred from the other. Consider Example 1 where we can infer hypothesis "*John knows how to drive*" from the premise "*John cleared driving test*". Entailment task needs a semantic understanding, models trained on the entailment data can be applied to many other NLP tasks such as text summarization, paraphrase detection, and machine translation. We can note that the inference relation in Entailment is a unidirectional relation only from premise to hypothesis. If we also have Entailment from hypothesis to premise, we can say that both sentences convey the meaning.

Example 1 *John took driving test today and cleared it* $\implies$ *John knows how to drive*

2 RTE Challenges

RTE(Recognizing Textual Entailment) challenge, started in 2005 (Dagan et al., 2005), is an annual competition where researchers from the NLP community showcase entailment models. In this competition, two text fragments are given and the task is to determine if one can be inferred from the other. There was a total of 7 RTE challenges from 2005-2011. RTE 1-3 focused on 2-way classification including only Entailment and Non-entailment classes. From RTE 4, the neutral class was also introduced making RTE as a 3-way classification task. This addition of neutral class hardened the RTE task but also made it possible for systems to be able to distinguish between contradictory and unknown premise and hypothesis pairs. RTE challenges also included Summarization and novelty detection tasks apart from Entailment task to increase the difficulty and make challenges more realistic. RTE challenges have resulted in the development of benchmarks for NLI task and have uncovered the challenges faced in recognizing textual entailment.

3 Classical Methods

The basic solution to Recognizing Entailment is to compare the re-represented version of premise and hypothesis and determine if the hypothesis can be derived from the premise based on the comparison. We can re-represent the sentences using either lexical or syntactical methods. The lexical method works directly on the input surface strings by operating solely on a string comparison between the text and the hypothesis. Lexical approach ignores the semantic relationship between premise and hypothesis and determines the entailment based on only lexical concepts. Most of the common approaches in lexical methods are word overlapping, subsequence matching, etc. Syntax-based or syntactic approaches convert premise and hypothesis into directed graphs. These graphs can be created using a parse tree and then compared with each other. Entailment or Non-Entailment is determined using the comparison between the

graphs of premise and hypothesis. Entailment can be recognized using different rules and tasks. In Example 2, we need to know following things to identify entailment; "*Cisco Systems Inc.*" is same as "*Cisco*" (Entity Matching), "*Filed a lawsuit*" is same as "*accused*" (Paraphrasing), Eliminate "*last year*" from the text (Alignment task), "*Cisco accused Apple*" is different from "*Apple accused Cisco*" (Semantic Role Labeling).

Example 2 *"Cisco Systems Inc. filed a lawsuit against Apple for a patent violation last year."* $\implies$ *"Cisco accused Apple for patent violation."*

4 Datasets

RTE challenges provided initial datasets to test and train NLI models. But these were very limited in quantity, comprising of only a few thousand of sentence pairs. As neural networks architectures became more powerful, attempts were made to create a large corpus that can be used for training of deep neural networks. The accuracy achieved on these datasets is considered as the benchmark to compare different NLI models. In this section, we look at some of the datasets that enabled models to learn to predict correctly by looking at many examples.

4.1 SNLI

Stanford Natural Language Inference(SNLI) (Bowman et al., 2015) corpus is the first one to have presented the research community a platform to test their neural network models on. SNLI is created using crowdsourcing, using Amazon Mechanical Turks platform. In this process, human volunteers were presented with captions of images and were asked to construct three sentences, one in favor of subject of the image, one in contradiction to it and one unrelated to the subject of the image. This created the hypothesis belonging to three classes *entailment*, *contradiction* and *neutral*. The premise was constructed by another crowdsourced task in which volunteers were asked to caption images from Flickr40k (Young et al., 2014). It has about 30k images and 160k of total captions were acquired. Overall there are total 550,152 training pairs, and 10k each in development and test set.

4.2 MNLI

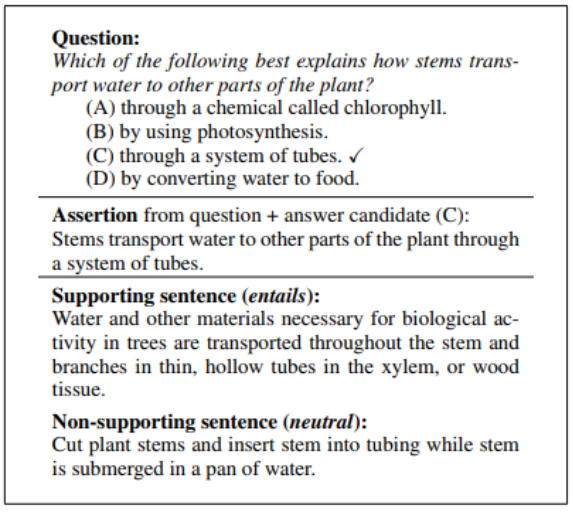
Multi-genre Natural Language Inference(MNLI) (Williams et al., 2018) is an improvement over the SNLI corpus as it tries to add more diversity in the types of sentences. MNLI is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information. It contains sentence pairs from 10 different genres, out of which only 5 appears in the training set while all 10 appear in the test set. This makes the learning NLI more challenging and generic.

4.3 XNLI

Cross-lingual NLI(XNLI) (Conneau et al., 2018) corpus is created intended to encourage research in cross-linguality. It is derived from the MNLI corpus for 15 languages using crowd-sourcing. Dev and test sets of MNLI are manually translated to 15 languages: English, French, Spanish, German, Greek, Bulgarian, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, Hindi, Swahili, and Urdu. Out of these Urdu and Swahili are low-resource languages.

4.4 SciTail

Although SNLI has proved to be beneficial for the advancement of techniques for recognizing textual entailment and has provided researchers to run many deep neural models on but it has been observed the dataset is not very effective for training a model for a particular end task like question-answering. SCITAIL (Khot et al., 2018) is created with end-task of question-answering in view.


Question:
Which of the following best explains how stems transport water to other parts of the plant?
(A) through a chemical called chlorophyll.
(B) by using photosynthesis.
(C) through a system of tubes. ✓
(D) by converting water to food.

Assertion from question + answer candidate (C):
Stems transport water to other parts of the plant through a system of tubes.

Supporting sentence (*entails*):
Water and other materials necessary for biological activity in trees are transported throughout the stem and branches in thin, hollow tubes in the xylem, or wood tissue.

Non-supporting sentence (*neutral*):
Cut plant stems and insert stem into tubing while stem is submerged in a pan of water.

Figure 1: An example from Scitail dataset

The premise-hypothesis pairs are created from

a corpus of high school level science related multiple choice questions. The hypothesis is created by combining the question and the correct answer. The premise is obtained independently from a corpus of web text such that it entails the hypothesis. Similarly, premises are created for contradiction and neutral pairs. Since the premise and hypothesis are created independently they are very different from each other in terms of the syntactic and lexical structure. This makes the task of entailment more challenging since there can be sentences that do not have a huge overlap of words but are similar in meaning and at the same time there can sentences with many words overlapping but are not related. The annotation of such premise-hypothesis pair as supports (entails) or not (neutral), is crowdsourced in order to create the SciTail dataset. The dataset contains 27,026 examples with 10,101 examples with entails label and 16,925 examples with a neutral label.

4.5 MRPC

MRPC paraphrase corpus (Dolan and Brockett, 2005) consists of 5800 paraphrase sentence pairs with a label(Paraphrase or Non-paraphrase). This corpus is collected from the web news sources in which 67% of the examples in the corpus are positive whereas only 33% examples are negative. In MRPC dataset, around 4100 pairs are training examples and 1700 pairs are test examples.

4.6 Glockner corpus

Glockner dataset (Glockner et al., 2018) was created using SNLI dataset for the sole purpose of measuring the lexical similarity of a model. The premise is the sentence that is extracted from the SNLI training set whereas hypothesis is made by replacing a single term in the premise with a related term i.e synonym or antonym. Some of the examples from the dataset are shown below:

"The man is holding a **saxophone**" $\implies$ "The man is holding **an electric guitar**"
"A little girl is very **sad**" $\implies$ "A little girl is very **unhappy**"
"A couple drinking **wine**" $\implies$ "A couple drinking **champagne**"

4.7 Quora Question-Pair Dataset

Quora dataset was introduced by the Quora platform to promote research in identifying question similarity task. It includes around 400,000 questions duplicate examples. Each line has ID for the Question pair, individual ids for question pair, two separate questions, and label of whether they are duplicate or not.

Training data includes 255027 non-duplicates and 149263 duplicates whereas Testing data on the Kaggle platform includes 2345796 question pairs.

5 Approaches

A lot about the meaning of a sentence can be inferred from its lexical and syntactic composition. Features like the presence of negation or synonyms can be used to compare sentences. Dependency graph can also be used to understand how different entities interact with each other. But because of the variability and ambiguity of natural language, semantics cannot be ignored. In this section, we first look at an approach using machine learning and then move on to more recent neural network models.

With the advent of SNLI and MNLI corpus, it became possible to run deep learning models for NLI. Large data which comprises of many examples of differing composition allows discovery of features that would otherwise be difficult to identify. Prior to deep learning, most of the approaches made use of hand-crafted features, mostly distance metrics, to train the NLI models. This lead to a very restricted set of features that could be used while training. Also, since different languages have a different composition of sentences, manually extracting various language features is not feasible. As a result, most of the recent developments have relied upon learned features using deep neural networks.

5.1 SVM

Support Vector Machine(SVM) (Cortes and Vapnik, 1995) is one of the popular ways of classifying a set of features into one of the target classes. It tries to find a hyperplane that separates the input into distinct classes. (Malakasiotis and Androutsopoulos, 2007) applies SVM on 10 lexical and shallow syntactic features, namely Levenshtein distance, Jaro-Winkler distance, Soundex, Manhattan distance, Euclidean distance, Cosine similarity, N-gram distance, Matching coefficient, Dice coefficient, and Jaccard coefficient. (Castillo and i Alemany, 2008) uses lexical and seman-

tic features along with some hand-crafted rules. (Castillo, 2010) uses 32 lexical and semantic features to perform the classification. It makes use of WordNet to relate similar meaning words. RTE challenges attracted many SVM based solutions but even the best performing system had accuracy less than 65%.

5.2 LSTM

Long Short Term Memory(LSTM) (Hochreiter and Schmidhuber, 1997) is created in an attempt to get rid of the vanishing gradient problem that was existing in RNN and to remember contexts for a longer period. In simple RNN as more and more words are input, previously learned network weights start fading as they are replaced by more recently seen words. This is because the learning of weights happens based on the gradient of loss and with time gradient goes on reducing since it has to be back-propagated through time.

(Bowman et al., 2015) proposed the first deep learning approach to learn the NLI task. The architecture used is shown in figure **??**. The input premise and hypothesis is encoded using an LSTM into two 100 dimension vectors. Concatenation of these two vectors is then used for learning the 3-class classification. The network consists of a stack of three tanh layers followed by a softmax layer.

5.3 Attention

LSTMs processes sentences sequentially in order to learn a succinct representation. This seems like a natural approach since humans also learn to read text sequentially, word-by-word. But as we get better at reading, we tend to focus more on certain words in the sentence which are more important than the others. This behavior can be learned by focusing on important words. Attention (Bahdanau et al., 2014) is a mechanism to learn what words to focus on. It combines the overall representation generated by LSTM with individual word vectors. Based on the loss calculated, each words vector gets its weight adjusted which determines how much the word contributes to the final representation.

(Rocktäschel et al., 2015) applies attention mechanism to identify words in the premise that can be influential in deciding the overall classification. (Rocktäschel et al., 2015) proposes three models for comparison. The first one uses two LSTMs, one for the premise and one for the hypothesis, arranged such that the initial state of second LSTM that processes hypothesis is set to the final state of the first LSTM. This is termed as a conditional encoding since the hypothesis is encoded conditioned on the premise. Figure 2 shows the architecture. (Zhao et al., 2016) tries to mimic the same for tree representation of sentences. The intuition here is that natural language sentences are inherently recursive and can be represented as a tree. Representing premise and hypothesis as a binary tree and then using the attention mechanism to find alignment between nodes of these trees gives a way to find entailment relation recursively. Neural Tree Indexer(NTI) (Munkhdalai and Yu, 2017), is a way to capture compositionality of a sentence. Compared to other tree-based methods, here we are not creating two trees for premise and hypothesis but instead, we create only one tree based on the words in hypothesis and combination of these words become the nodes of the tree.

5.4 WordNet

Knowledge-based Inference Model(KIM) (Chen et al., 2018) involves the utilization of external knowledge in neural networks. Generally, neural networks for Entailment includes encoder, attention, local entailment, and sentence level entailment. This model does the same but with the use of External Knowledge in the form of WordNet. This addition of WordNet is clearly beneficial to help recognize word or phrase level relations. It also helps when the training data is limited and the model is not able to learn much from the given data.

At first, sentences are encoded by encoders as context-dependent representations. Second, we calculate co-attention between premise and hypothesis to obtain word-level alignment. After that, we collect local inference information for Entailment/Non-entailment prediction. Finally, the composition component aggregates all sentences and predicts the label.

WordNet relations need to be converted to a numerical representation. Semantic relations among the words are determined using various relations like synonymy, antonymy, hypernymy, hyponymy, etc. All these relations or features are converted to real numbers. Positive relations like synonymy, hypernymy, and hyponymy help us in cap-

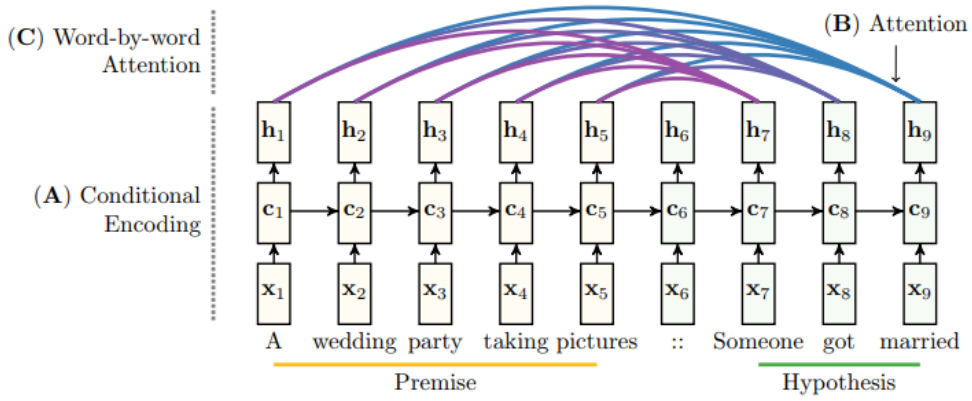


Figure 2: Attention model for NLI. *x1, x2..* are the input word representation. *c1, c2..* are the cell state and *h1, h2..* are the output vector representations

turing the Entailment whereas negative relations antonymy, co-hyponymy(words with the same hypernymy) helps in determining Non-entailment.

Co-Attention component use co-attention matrix and softly aligns word pairs between the premise and hypothesis with the help of external knowledge. Inference collection component computes local entailment between words or phrases by comparing premise and hypothesis alignment vectors. Inference composition uses Bi-LSTM layer and determines sentence level entailment between a premise and a hypothesis.

5.5 Transformer Network

(Vaswani et al., 2017) proposed to process the input in a non-sequential manner so that the computations could be parallelized. The proposed model is called a Transformer network. It follows a similar approach to other sequence-to-sequence models. It has an encoder layer which converts the input into an intermediate representation, which is then decoded to output value by the decoder layer. But it differs from recurrent architectures in that it does not need the encoder layer to process the input sequentially before the decoder can start generating the output. The encoder layer produces the output for each word in parallel. The decoder learns to attend to output from the encoder.

Figure 3 shows the encoder and decoder architecture of transformer. The input to the model is a combination of word embeddings and position vector. Position vector allows making use of features related to the position of words in the sentence. There are multiple layers of encoder and

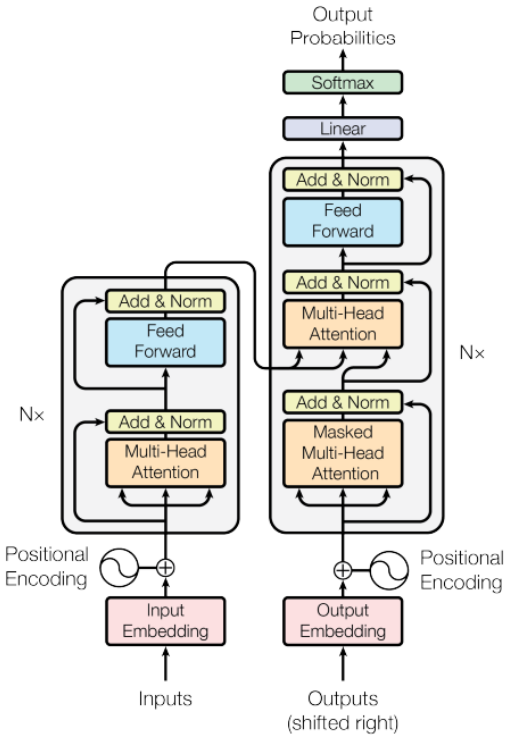


Figure 3: Transformer encoder(left) and decoder(right) architecture

decoder and each layer has multiple sub-layers. In the case of an encoder, there are two sub-layers: first is the multi-head attention layer and second is a fully connected neural network layer. The attention layer learns a weighted combination of inputs that helps in focusing on the relevant words. Multi-head attention learns these attention weights in multiple representation subspaces and then combines them. The decoder has a similar

structure except that it has a third layer of multi-head attention that attends to the outputs from the encoder layer.

6 Applications

Entailment and Paraphrase are closely related to each other. We can say that paraphrase is a special type of entailment in which the entailment relation is existing in both directions. In other words, bi-directional entailment can be used for detecting paraphrases. This has many applications in NLP like text summarization, question answering, machine translation evaluation. We demonstrate the application of bi-directional entailment in machine translation evaluation.

In Machine translation evaluation, a system translated sentence is to be evaluated to judge the quality of the translation system. A human translated reference translation is provided to compare the machine-generated translation with. We can use bi-directional entailment to determine if the candidate and reference are a paraphrase of each other.

7 Conclusion

In conclusion, the classical models that rely on lexical feature extraction do not result in very high accuracy since NLI requires a deeper understanding of the semantics. LSTM and attention mechanism gives good results. The most recent work is on transformer networks, which is state-of-the-art for most of the NLP tasks, including NLI.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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*, pages 632–642. Association for Computational Linguistics.

Julio Javier Castillo. 2010. An approach to recognizing textual entailment and te search task using svm.

Julio Javier Castillo and Laura Alonso i Alemany. 2008. An approach using named entities for recognizing textual entailment. In *Notebook Papers of the Text Analysis Conference, TAC 2008 Workshop*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

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

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Prodromos Malakasiotis and Ion Androutsopoulos. 2007. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 42–47.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural tree indexers for text understanding. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, page 11. NIH Public Access.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *CoRR*, abs/1509.06664.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Kai Zhao, Liang Huang, and Mingbo Ma. 2016. Textual entailment with structured attentions and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2248–2258. The COLING 2016 Organizing Committee.