A SURVEY OF CONTRADICTION DETECTION

A PREPRINT

Shravan Singh*
Department of Computer Science
IIT Kharagpur, India
singh.acad.shravan@gmail.com

September 21, 2021

ABSTRACT

This is a survey of different methodologies of contradiction-detection in text. This survey covers a decade of work in the field, beginning with lexical features, and leading up to the modern Transformer based architectures.

***K*eywords** contradiction detection · neural network · Antonym · relation extraction · BERT · word embedding and Neural Attention

1 Introduction

Contradictions occur when two sentences are extremely unlikely to be true simultaneously(De Marneffe et al., 2008). De Marneffe et al.(2008) identified two primary categories of contradictions: (1) those occurring via antonym, negation, and date/number mismatch, which are relatively simple to detect because of the presence of negation or antonym, and (2) contradictions arising from the use of factive or modal words, structural and subtle lexical contrasts, as well as world knowledge (WK). Contradictions in category (2) are more difficult to detect automatically because it is not dependent on the presence of some word. For example, to detect contradictions between sentence A: Narendra Modi was born in Vadnagar, and sentence B:Narendra Modi was born in Gujarat, we should have the WK that Vadnagar is a town in Gujarat. De Marneffe, et al., provide examples of several different types of contradictions in Table 1. The columns, Text and Hypothesis follows the PASCAL Recognizing Textual Entailment (RTE) Challenge's convention, where Text is a single paragraph and the Hypothesis is one sentence, and task is to classify them into three categories(Entailment, Neutral, and Contradiction).

The *Factive* type denotes what we assume to be true, e.g., "He knows sun rises from east." The *Structural* type deals with the structure of a sentence like in example 6, Text is 'Santer succeeded Delors' whereas, in hypothesis 'Delors succeeded Santer', which is clearly a contradiction. In example 8, it is important to learn that 'did nothing wrong' and 'accuses' are incompatible to each other. In example 9, it is even harder to find the relationship between 'withdrawn from peackeeping mission' and 'stay on course'. These are categorised as Lexical types.

In this paper we will be traversing through different methodologies used for detecting contradiction between two sentences.

Contradiction detection can be used for information verification, document summarizing, and question and answering system. It is being used to detect rumours claims in social media(Lendvai et al. 2016). Earlier methods were based on hand crafted lexical, syntactical, and semantic features of two sentences whereas, in recent methods deep learning methods are used to extract features that are significant for contradiction detection. We will explore these methods chronologically as it also fits into the two categories of methodologies that we described above.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

ID	TYPE	Text	Hypothesis
1	Antonym	Capital punishment is a catalyst for more crime.	Capital punishment is a deterrent to crime.
2	Negation	A closely divided Supreme Court said that juries and not judges must impose a death sentence.	The Supreme Court decided that only judges can impose the death sentence.
3	Numeric	The tragedy of the explosion in Qana that killed more than 50 civilians has presented Israel with a dilemma.	An investigation into the strike in Qana found 28 confirmed dead thus far.
4	Factive	Prime Minister John Howard says he will not be swayed by a warning that Australia faces more terrorism attacks unless it withdraws its troops from Iraq.	Australia withdraws from Iraq.
5	Factive	The bombers had not managed to enter the embassy.	The bombers entered the embassy.
6	Structure	Jacques Santer succeeded Jacques Delors as president of the European Commission in 1995.	Delors succeeded Santer in the presidency of the European Commission.
7	Structure	The Channel Tunnel stretches from England to France. It is the second-longest rail tunnel in the world, the longest being a tunnel in Japan.	The Channel Tunnel connects France and Japan.
8	Lexical	The Canadian parliament's Ethics Commission said former immigration minister, Judy Sgro, did nothing wrong and her staff had put her into a conflict of interest.	The Canadian parliament's Ethics Commission accuses Judy Sgro.
9	Lexical	In the election, Bush called for U.S. troops to be withdrawn from the peacekeeping mission in the Balkans.	He cites such missions as an example of how America must "stay the course."
10	WK	Microsoft Israel, one of the first Microsoft branches outside the USA, was founded in 1989.	Microsoft was established in 1989.

Table 1: Examples of contradiction types.

2 Machine learning models based on Lexical, Semantic, and Syntactic features

2.1 Harabagiu et al. (2006)

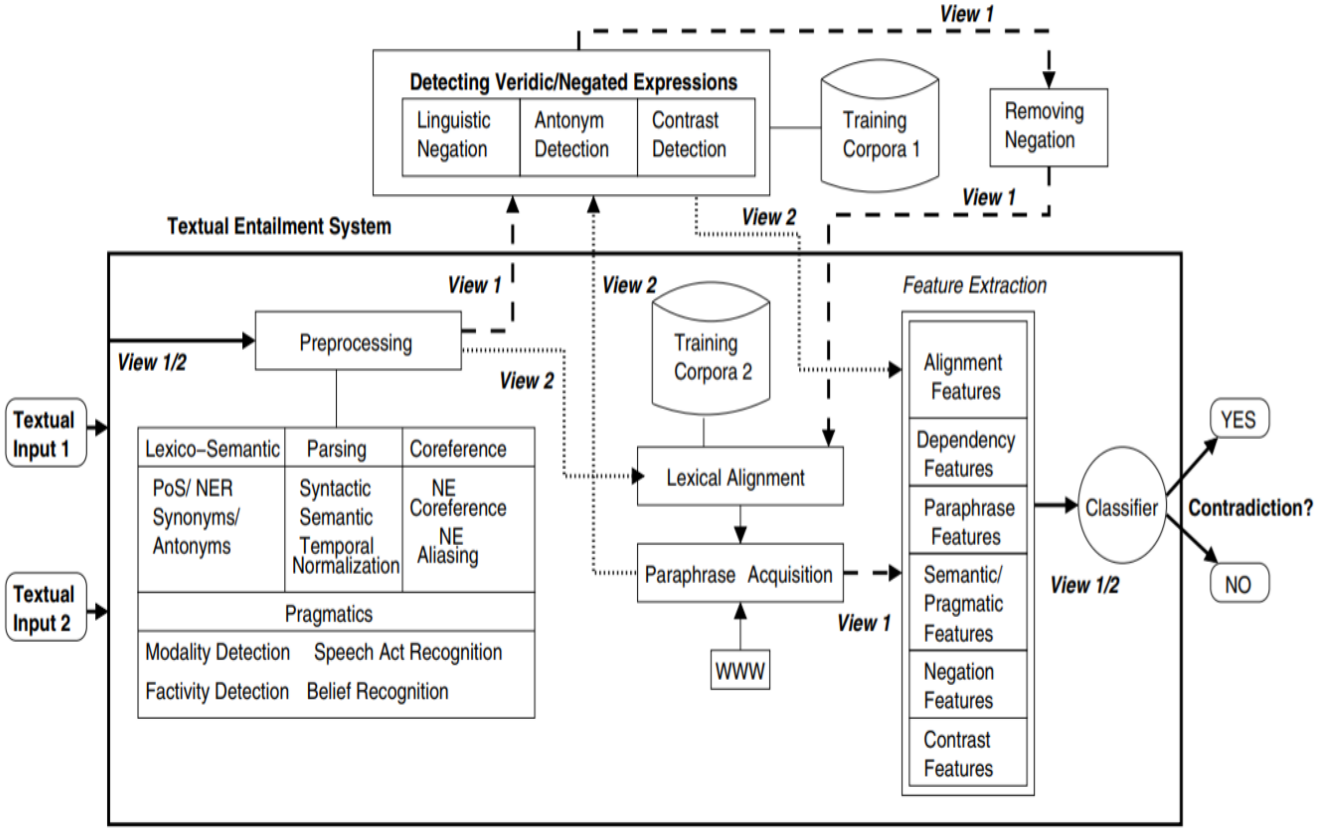

Figure 3: The Architecture Used for Recognizing Contradictions with the Help of Textual Entailment.

This framework combines the processing and removal of negation, the derivation of antonymy with the detection of CONTRAST relations. Antonymy is discovered by mining WordNet paths that extend an encoded antonymy. These paths are also used for recognizing CONTRAST relations, as they belong to six features designed specifically for capturing opposing information. Another novelty of this paper is the casting of the recognition of the CONTRAST discourse relation as a classification problem that operates on the results of textual alignment. Following two ways are used for this: View 1: Contradictions are recognized by identifying and removing negations of propositions and then testing for textual entailment; View 2: Contradictions are recognized by deriving linguistic information from the text inputs, including information that identifies (a) negations, (b) contrasts, or (c) oppositions (antonyms) and by training a classifier based on examples, similarly the way a classifier for entailment is trained. These are the steps that are being followed(not in order): (a) a PREPROCESSING MODULE, which derives linguistic knowledge from the text pair; (b) an ALIGNMENT MODULE, which takes advantage of the notions of lexical alignment and textual paraphrases; (c)a module that enables us to detect veridic and negated expressions in the textual inputs as well as a module that eliminates the detected negations;and (d) a CLASSIFICATION MODULE, which contains a feature extractor and a classifier to classify whether it is a contradiction or not.

The preprocessing module annotates both text inputs with four forms of linguistic information: (1) lexico-semantic information; (2) parsing information; (3) coreference information; and (4) pragmatic information. The lexico-semantic information consists of (i) part-of-speech information; (ii) synonymy and antonymy information derived from WordNet (Fellbaum 1998); and (iii) named entity classes provided by our named entity recognition (NER) system capable of distinguishing more than 150 different classes. The parsing information consists of syntactic and semantic parses.The parses contain also temporal and spatial normalizations of temporal expressions (including dates and time intervals) and spatial expressions (including names of politic and geographic locations). The coreference information is based on (a) the recognition of the name aliases; and (b) the recognition of named entity coreference. The pragmatic linguistic

information that is derived is based on the recognition of modal auxiliaries (e.g. “would”,“could”), factive verbs (e.g. “manage”), belief verbs (e.g. “consider”, “believe”), or lexicons associated with speech acts (e.g. “say”, “announced”). After preprocessing the text units, the process flow associated with View 1 leads to the detection of veridic and negated expressions, whereas the process flow associated with view 2 leads to the alignment module. We process negation by considering (i) overt (e.g. “don’t”) negation, and (ii) indirectly licensed negation(e.g. “deny”, “fail”, “refuse”). We detect three types of negated constituents in texts: (1) negated events, (2) negated entities, and (3) negated states using the above overt and indirect negations. The contrasts are detected based on a large training corpus. Maximum Entropy classifier is used on the paraphrase features, dependency features, contrast features and semantic / pragmatic features to detect contrast. For Antonym detection, Antonymy chains are used. It is a lexico-semantic chains that are based on relations encoded in WordNet in which one relation is the antonymy relation. Negation removal is performed only in the processing corresponding to View 1. Then both processing flows complete the feature extraction that also depends on dependency features and semantic/pragmatic features derived from the knowledge produced by the preprocessing module.All these features are used by the inference classifier, which is trained on the Training Corpora.

Data prepared for the 2006 PASCAL Recognizing Textual Entailment (RTE) Challenge has been used in order to create three new types(overtly negated, paraphrased, mixed) of training and evaluation corpora for textual contradiction system. Around 64% accuracy has been achieved with this.

2.2 De Marneffe et al., 2008

This system is based on the stage architecture of the Stanford RTE system (MacCartney et al., 2006), but adds a stage for event coreference decision. Three datasets are used to check the performance of system 1) RTE dataset 2) Dataset similar to harabaigu2006, based on negation and paraphrses 3) Contradictions that cover all the different types of contradictions as mentioned above.

2.2.1 Linguistic analysis

The goal in this stage is to compute linguistic representations of the text and hypothesis that contain as much information as possible about their semantic content. Typed dependency graphs are used, which contain a node for each word and labeled edges representing the grammatical relations between words.

2.2.2 Alignment

The purpose of the second phase is to find a good partial alignment between the typed dependency graphs representing the hypothesis and the text. An alignment consists of a mapping from each node (word) in the hypothesis graph to a single node in the text graph, or to null. Measure of alignment quality is a score, and a procedure for identifying high scoring alignments. Score of an alignment is the sum of the local node and edge alignment scores.

2.2.3 Filtering non-coreferent events

Contradiction features are extracted based on mismatches between the text and hypothesis. Therefore, we must first remove pairs of sentences which do not describe the same event, and thus cannot be contradictory to one another. Assuming two sentences of comparable complexity, we hypothesize that modeling topicality could be used to assess whether the sentences describe the same event. There is a continuum of topicality from the start to the end of a sentence. Topicality score of a sentence is calculated as a normalized score across all aligned NPs.The text and hypothesis are topically related if either sentence score is above a tuned threshold.

2.2.4 Classifier

In the final stage, contradiction features are extracted on which logistic regression is applied to classify the pair as contradictory or not. The feature weights are hand-set, guided by linguistic intuition. Polarity features: Polarity difference between the text and hypothesis is often a good indicator of contradiction, provided there is a good alignment. The polarity features capture the presence (or absence) of linguistic markers of negative polarity contexts. Number, date and time features: Numeric mismatches can indicate contradiction. Antonymy features: Aligned antonyms are a very good cue for contradiction. List of antonyms and contrasting words comes from WordNet, from which words are extracted with direct antonymy links and expand the list by adding words from the same synset as the antonyms. Structural features: These features aim to determine whether the syntactic structures of the text and hypothesis create contradictory statements. Factivity features: The context in which a verb phrase is embedded may give rise to contradiction. Modality features: Simple patterns of modal reasoning are captured by mapping the text and hypothesis to one of six modalities ((not)possible, (not)actual, (not)necessary), according to the presence of

predefined modality markers such as can or maybe. A feature is produced if the text/hypothesis modality pair gives rise to a contradiction. Relational features: A large proportion of the RTE data is derived from information extraction tasks where the hypothesis captures a relation between elements in the text. Using Semgrex, a pattern matching language for dependency graphs, such relations are found and ensured that the arguments between the text and the hypothesis match.

System performed well in LCC negation data(harabaigu2006) because of overtly negative features of sentences but unable to perform well in RTE3 data because category 2 types of contradiction is hard to identify. Even in RTE3 data for category 1 features(except the numeric features) it performed well.Harabagiu2006 et al.'s performance demonstrates that further improvement on category 1 is possible; indeed, they use more sophisticated techniques to extract oppositional terms and detect polarity differences.

2.3 Contradiction Detection for Rumorous Claims (Lendvai et al., 2016)

This work identifies two different contexts in which contradiction emerges in twitter: its broader form can be observed across independently posted tweets and its more specific form in threaded conversations.

The two datasets corresponding to two tasks are drawn from a freely available, annotated social media corpus(PHEME) that was collected from the Twitter platform. English tweets are related to four events: the Ottawa shooting , the Sydney Siege , the Germanwings crash , and the Charlie Hebdo shooting For each tweet pair vocabulary overlap and local text alignment features are extracted. Vocabulary overlap was calculated for content word stem types in terms of the Cosine similarity and the F1 score. The Cosine similarity of two tweets is defined as C(X, Y) = $\frac{|X \cap Y|}{\sqrt{|X|.|Y|}}$, where X and Y denote the sets of content word stems in the tweet pair. The F1 score is defined as the harmonic mean of precision and recall. Precision and recall here refer to covering the vocabulary X of one tweet by the vocabulary Y of another tweet (or vice versa). These two metrics are additionally applied to the content word POS label inventories within the tweet pair, which gives the four features cosine, cosine pos, f score, and f score pos, respectively. The amount of stemmed word token overlap was measured by applying local alignment of the token sequences using the Smith-Waterman algorithm.

The similarity features reach highest values for the ENT class, followed by CON and UNK. In order to predict the RTE classes based on the features introduced above, two classifiers are used: Nearest (shrunken) centroids (NC) and Random forest(RF). It turns out generally that classifying CON is more difficult than classifying ENT or UNK. F1 score is .35 and .37 for threads and independent posts respectively. General performance across all three classes was better in independent posts than in conversational threads.

2.4 It's a Contradiction—No, it's Not: A Case Study using Functional Relations(Ritter et al., 2008)

In this work first, a simple logical foundation for the CD task is established, which suggests that extensive world knowledge is essential for building a domain-independent CD system. Second, it shows that most of the apparent contradictions are actually consistent statements due to meronyms (Alan Turing was born in London and in England), synonyms (George Bush is married to both Mrs. Bush and Laura Bush), hypernyms (Mozart died of both renal failure and kidney disease), and reference ambiguity. Next, they show how background knowledge enables a CD system to discard seeming contradictions and focus on genuine ones.

A contradiction between T and H arises only in the context of K. That is: $((K \wedge T) \models \neg H) \vee ((K \wedge H) \models \neg T)$ In these cases, T and H alone are mutually consistent and a contradiction between T and H arises only in the context of K, where K is the knowledge base.

First a formal model for computing the probability that a phrase denotes a function based on a set of extracted tuples is to be calculated. An extracted tuple takes the form R(x, y) where (roughly) x is the subject of a sentence, y is the object, and R is a phrase denoting the relationship between them. The main evidence that a relation R(x, y) is functional comes from the distribution of y values for a given x value. If R denotes a function and x is unambiguous(like Mozart), then we expect the extractions to be predominantly a single y value, with a few outliers due to noise. TEXTRUNNER is used for extraction purpose. Then they use an EM style algorithm that alternately estimates the probability that R is functional and the probability that x is ambiguous.

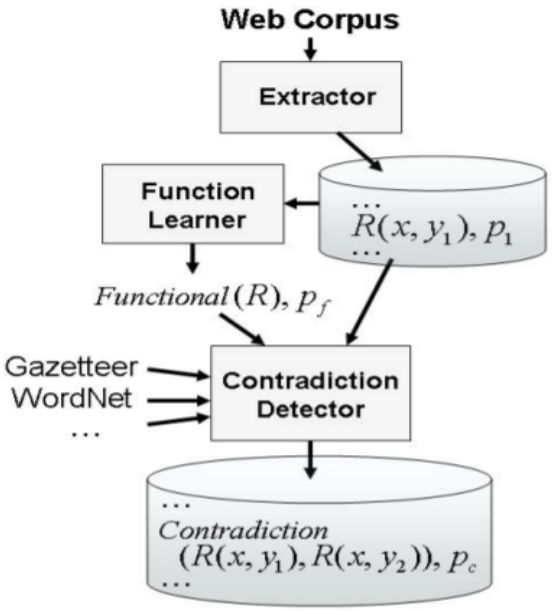

Figure 2: AUCONTRAIRE architecture

For each relation R that AUCONTRAIRE has judged to be functional, we identify contradiction sets R(x, ·), where a relation R and domain argument x have multiple range arguments y. For a variety of reasons, a pair of extractions R(x, y1) and R(x, y2) may not be actually contradictory. The following is a list of the major sources of false positives—pairs of extractions that are not genuine contradictions, and how they are handled by AUCONTRAIRE. Synonyms: The set of potential contradictions died from(Mozart,·) may contain assertions that Mozart died from renal failure and that he died from kidney failure. These are distinct values of y, but do not contradict each other. Same applies to Meronyms. Argument Typing: Two y values are not contradictory if they are of different argument types. For example, the relation born in can take a date or a location for the y value. These features are combined with other features such as string similarity and arugment type to be used in Logistic Regression, in order to estimate the probability that a given pair, R(x, y1), R(x, y2) is a genuine contradiction.

3 Deep Learning based Models

3.1 Contradiction Detection with Contradiction-Specific Word Embedding(Luyang Li, 2017)

For the contradiction detection task, an effective feature learning approach is to learn the semantics relation from input texts using representation learning rather than professional knowledge. Despite the effectiveness of traditional context-based word embedding learning algorithms in many natural language processing tasks, such algorithms are not powerful enough for contradiction detection. Contrasting words such as “overfull” and “empty” are mostly mapped into close vectors in such embedding space because of the distributional semantics hypothesis. To solve this problem, a tailored neural network is made to learn contradiction-specific word embedding (CWE). The method can separate antonyms in the opposite ends of a spectrum. CWE is learned from a large-scale corpus of contrasting pairs which are generated from PPDB(Ganitkevitch 2013) and WordNet automatically. Glove vectors are used as initial embeddings and then updated at each iteration.

In detecting contradiction, the global semantic features and local semantic features are both important. The global features refer to the semantic meaning of the sentence which is relevant in terms of the word order. Two sentences which consist of similar words in different word orders may have different meanings. We use the representation of sentence-level semantic relation to capture global semantic features. The local features explore the semantic relation between unaligned phrases from the pair of sentences. The unaligned phrases always contain contrasting meanings in the contradictory sentences. Along with these two features, some shallow features such as, number of negation words, number of unaligned words, word orders are also used. The output has three classes: Entailment, Contradiction, and Neutral. The performance test is being carried out on SemEval 2014 task 1 benchmark dataset. Experimental results show that CWE outperforms traditional context-based word embedding in contradiction detection and general RTE task.

. The proposed model for contradiction detection performs comparably with the top-performing system in accuracy of three-category classification and enhances the accuracy from 75.97% to 82.08% in the contradiction category.

3.2 Scalable Detection of Sentiment-Based Contradictions (Tsytsarau et al., 2011)

Author begins by describing two types of contradiction: 1)Asynchronous : Two set of documents having different sentiments in different time intervals 2) Two set of documents having different sentiments in same time interval. In order to detect contradicting opinions in collections of texts, first they determine all the different topics and then calculate the corresponding sentiments. For identifying topics per sentence, Latent Dirichlet Allocation (LDA) algorithm is used. Then, for each sentence-topic pair a continuous sentiment value is assigned in the range [-1;1] that indicates a polarity of the opinion expressed regarding the topic. In order to be able to identify contradicting opinions a measure of contradiction is defined. Assume that we want to look for contradictions in a shifting time window w(day,week). For a particular topic T, the set of documents D, which we use for calculation, will be restricted to those, that were posted within the window w. Then measure of contradiction is calculated using the variance and mean of sentiments of documents in topic T. Rest of the paper describes how to scale this approach to identify contradictions in large collections of documents.

3.3 REASONING ABOUT ENTAILMENT WITH NEURAL ATTENTION(Rocktaschel et al., 2016)

An end-to-end differentiable solution to RTE is desirable, since it avoids specific assumptions about the underlying language. In particular, there is no need for language features like part-of-speech tags or dependency parses. Furthermore, a generic sequence-to-sequence solution allows to extend the concept of capturing entailment across any sequential data, not only natural language. Following are the contributions of this paper: (i) Neural model based on LSTMs that reads two sentences in one go to determine entailment, as opposed to mapping each sentence independently into a semantic space (ii)This model with a neural word-by-word attention mechanism to encourage reasoning over entailment of pairs of words and phrases, and (iii) a detailed qualitative analysis of neural attention for RTE. As shown in figure, Hypothesis is conditioned on the representation that the first LSTM built for the premise. Finally, for classification a softmax layer over the last output vector into the target space of the three classes (ENTAILMENT,NEUTRAL or CONTRADICTION). Two different kind of attentions are used as shown in figure, where C type outperforms the type B in results.

Author have visually shown how premise and hypothesis are connected using attention. The hypothesis conditioned on the premise instead of encoding each sentence independently gives an improvement of 3.3 percentage points in accuracy over Bowman et al.'s 2015 LSTM. Author argue this is due to information being able to flow from the part of the model that processes the premise to the part that processes the hypothesis. LSTM achieves an accuracy of 80.9% on SNLI(Bowman et al., 2015), outperforming a simple lexical classifier tailored to RTE by 2.7 percentage points. An extension with word-by-word neural attention surpasses benchmark LSTM result by 2.6 percentage points, setting a new state-of-the-art accuracy of 83.5% for recognizing entailment on SNLI.

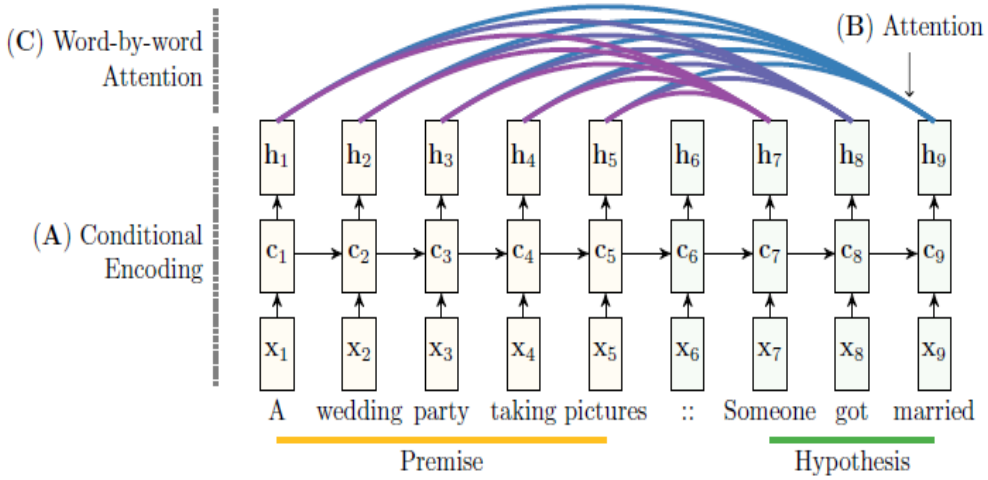

Figure 1: Recognizing textual entailment using (**A**) conditional encoding via two LSTMs, one over the premise and one over the hypothesis conditioned on the representation of the premise (c_5), (**B**) attention only based on the last output vector (h_9) or (**C**) word-by-word attention based on all output vectors of the hypothesis (h_7, h_8 and h_9).

3.4 Learning Natural Language Inference with LSTM(Wang et al., 2016)

There are three fundamental changes done in this model that differs from Rocktaschel et al.(2016) : 1) Hypothesis LSTM's first state is not being initialized by the last cell state of Premise LSTM. 2) Attention weights and weighted representation of Premise r_t do not only depend on r_{t-1} but both on r_{t-1} and hidden state of hypothesis LSTM r_{t-1}. 3) An LSTM is used, instead of RNN, that models the matching between the premise and the hypothesis and its hidden state is r_t. They have tried to match the premise with the hypothesis using the hidden states of the two LSTMs, instead of LSTM of the hypothesis to encode any knowledge about the premise as done in Rocktaschel et al. (2016). The model gives an accuracy of 85.7 when the glove embedding of 150 is used and gives 86.1 when 300 dimension glove vector is used. This is a statistically significant improvement over Rocktaschel(83.5%).

3.5 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (BERT, 2018)

There are two steps in this framework: pre-training and fine-tuning. During pre-training, the model is trained on unlabeled data over 2 different pre-training tasks. For fine tuning, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the downstream tasks. Two pre-training tasks are 1)Masked Language Model 2)Next Sentence Prediction(NSP). Under Masked LM, some percentage of input tokens are masked[Mask] randomly and then it tries to predict these hidden vectors at output layer using softmax. The NSP task is to predict whether sentence B follows sentence A. For training, classes(IsNext, NotNext) are divided equally. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters. At the output layer, tokens are fed into a classification layer that predicts whether the input sentences entail, contradict, or neutral to each other. MNLI and RTE dataset is used for fine-tuning the BERT model and scored accuracy of 86% and 70% respectively. For the pre-training, the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2.5B words) is used.

4 Work to be done

Attention model, contradiction specific work embedding, and BERT based model need to be verified on the PHEME data and other datasets to see their accuracy on all kinds of contradiction. Currently they are only tested on SNLI/MNLI.

References

[1] Sanda Harabagiu, Andrew Hickl, and Finley Lacatusu. Negation, contrast, and contradiction in text processing. In *In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*.

[2] Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. Learning to recognize features of valid textual entailments. In *In Proceedings of the North American Association of Computational Linguistics (NAACL06)*.

[3] Marie-Catherine de Marneffe, Anna Rafferty, and Christopher D. Manning. Finding contradictions in text. In *ACL 2008*

[4] Piroska Lendvai and Uwe D Reichel. Contradiction detection for rumorous claims. In *arXiv preprint arXiv:1611.02588, 2016.*.

[5] Ritter, A., Downey, D., Soderland, S., and Etzioni, O. (2008). It's a contradiction — no, it's not: a case study using functional relations. In *In Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pages 11–20. Association for Computational Linguistics..

[6] Luyang Li, Bing Qin, and Ting Liu. (2017). Contradiction detection with contradiction-specific word embedding. In *Algorithms,* 10(2):59.

[7] Tsytsarau, M., Palpanas, T., and Denecke, K. (2010). Scalable detection of sentiment-based contradictions. In *DiversiWeb, WWW*, 1:9–16.

[8] Tim Rocktaschel, Edward Grefenstette, Karl Moritz Hermann, Toma's Ko, cisk y, and Phil Blunsom. 2016. Reasoning about entailment with neural attention In *Proceedings of the International Conference on Learning Representations 2016*.

[9] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *EMNLP, 2015.*

[10] Wang, S., and Jiang, J. . Learning natural language inference with LSTM. In *In The North American Chapter of the Association for Computational Linguistics.* 1442–1451

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805, 2018.*