Detection of contradictions in text

Sachin Kumar Yadav
Akashdeep Patel

Guided by:
Prof. Amitabha Mukerjee
Abstract

Detecting contradiction is a subtask of recognizing textual entailment between texts. We have used a methodology based on lexical mismatch and structural similarity of the texts. Our method assumes that lexical contradictions propagate to the contradictions of whole text. Sometimes contradiction is embedded into the semantics of the texts and it can not be identified by using lexical and structural analysis.

Introduction

In natural language processing (NLP), the term textual entailment (TE) refers to a directional relation between two text segments, usually sentences. TE can be said to exist when the truth of one text segment follows from that of another. The entailing segment is termed t while the entailed text (hypothesis) is termed h. Textual entailment has a more relaxed definition than pure logical entailment: “t entails h” (t ⇒ h) if, normally, a person reading t would infer that h is probably true [Monz and Rijke 2001]. TE relations are considered directional because even if “t entails h”, the reverse “h entails t” may not be true [Marneffe et al. 2008].

Many NLP applications, such as question answering, information extraction, and machine translation evaluation, could make use of a model that would allow them to recognize whether the same meaning can be inferred from different sentence variants.

RTE has become one the most popular research topic in NLP.

Textual entailment problem can be categorized into three categories: entailment, contradiction and irrelevant. Analysis of RITE results show that there is scope of improvement in the direction of contradiction detection. And this can result in filtering the text entailment more appropriately.

According to Marneffe et al. [2008], the prominent types of contradiction can be defined as: (1) those occurring via overt negations such as antonyms, negations, and time/date/number mismatches; and (2) contradictions resulting from factive/modal words, text structure, certain lexical contrasts, and word knowledge. Unlike the former type, which may only need a little external information for detection, conflict cases of the second type cannot be solved without large amounts of presupposition, background knowledge, and even specific know-how.

For example:
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.

Examples of contradiction pair belonging to first type:

**Antonym:**
- Capital punishment is a **catalyst** for more crime.
- Capital punishment is a **deterrent** to crime.

**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.

**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.

**Factive:**
- The bombers had **not managed to enter** the embassy.
- The bombers **entered** the embassy.

Real life applications of detecting contradicting texts can be in political candidate debates, intelligence reports and in bioinformatics.

**Past work**
This is a very new field and the amount of work in this field has been very limited. Marneffe et al. [2008] described corpus annotation guidelines and several types of contradiction but also proposed some salient features (such as number/date/time, negations, antonyms, factivity, modality, relations, and structures) useful in identifying incompatible information in texts.

In RTE-4 and RTE-5, more than 20 participants in the three-type RTE subtask presented strategies for dealing with contradictions. Most of these approaches looked for the presence of negations and antonyms. Other cues, such as time, date, number, location name, quantifiers were widely used to improve the accuracy of contradiction detection.

Recently Andrade et al. [2013] presented an approach based on lexical and structural features for contradicting detection. We have taken inspiration from them and in addition to it incorporated a feature to detect coreference from Marneffe et. al(2008).
Processing Steps:
Following diagram shows the three state architecture.

Extracting Simple Sentences from complex sentence: A simple sentence is a sentence that contains only one predicate. The splitting in simple sentences is done in such a way that a simple sentence contains exactly one predicate and all predicate arguments. Here is an example:
“Ram went to the supermarket and bought Apples.”
=> “Ram goes to supermarket”, “Ram buys apple.”
This splitting breaks difficult task of detecting entailment or contradiction in complex sentences simpler as for small sentences lexical contradiction detection and alignment of the two is simpler.
Calculating minimum alignment costs between two simple sentences: Structural similarity of the two sentences T1 and T2, leads lexical contradiction to decide for the sentence contradiction. As example, "Mohit buys a car from Tanmay"(T1) entails "Tanmay sells a car to Mohit"(T2). Using the predicate relation database, the system might be able to align "buy" and "sell" and detect the lexical contradiction. Structural similarity feature, which is low here, is decided by alignment costs of T1 and T2.

<table>
<thead>
<tr>
<th>Alignment Cost (range)</th>
<th>Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>if both the words are the same.</td>
</tr>
<tr>
<td>0.0 - 0.2</td>
<td>if both the words are synonyms or antonyms according to WordNet or Wikipedia redirect</td>
</tr>
<tr>
<td>0.2 - 1.0</td>
<td>if the words belong to similar categories according to Wikipedia.</td>
</tr>
<tr>
<td>3.0</td>
<td>otherwise.</td>
</tr>
</tbody>
</table>

Table 1

The costs for aligning two chunks are determined by the similarity of the head content words as shown in Table 1. This alignment focuses on the words that have a high degree of selectional preference in common. These words should be aligned. Simple Sentence from T1: Ram buys apple. Simple Sentence from T2: Ram sells apple.

- Ram <=> Ram alignment costs = 0.0 (same base form)
- Apple <=> Apple alignment costs = 0.0 (same base form)
- Buys <=> sells alignment costs = 0.2 (can be antonyms)

The calculation of minimum alignment cost between T1 and T2 using cost matrix of previous steps and finding best alignments of simple sentences in T1 and T2 globally. This computation is done using Tree edit distance algorithm.

Calculating degree of lexical contradiction: For each aligned chunks, in each aligned simple sentence pair, we calculate the degree of lexical contradiction. The degree of lexical contradiction is defined as follows:
<table>
<thead>
<tr>
<th>Degree of Lexical Contradiction</th>
<th>Case Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>if both the words are place names but their surface forms are different (place names are extracted from Wikipedia).</td>
</tr>
<tr>
<td>1.0</td>
<td>if both the words contain antonyms according to Wordnet.</td>
</tr>
<tr>
<td>0.5</td>
<td>if the head content word is the same but all other words (morphemes) are different and are not listed as synonyms (in any of the used resources).</td>
</tr>
<tr>
<td>0.0</td>
<td>otherwise.</td>
</tr>
</tbody>
</table>

Simple Sentence from T1: Ram buys apple.
Simple Sentence from T2: Ram sells apple.
Lexical Contradiction between these two: 1.0

**Calculate the degree of contradiction between T1 and T2:** Finally, the degree of lexical contradiction for each chunk in each simple sentence pair is added and Thus we get the total degree of lexical contradiction $c_{total}$. Lexical contradiction does not necessarily imply contradiction between the whole texts T1 and T2. Therefore the total degree of lexical contradiction is adjusted to get the degree of contradiction between T1 and T2 $c_{text}$.

$$c_{text} = \frac{c_{total}}{a_{total} + d}$$

where $d > 0$ is set using the training data. For all of our experiments $d$ was set to 10. $a_{total}$ is the total minimum alignment cost. The intuition for the above Formula is that high structural similarity between text T1 and T2, in combination with (high) lexical contradiction, implies, in general, that T1 and T2 are contradicting.

**Feature Extraction:**
In the final stage, we extract contradiction features on which we apply decision tree to classify the pair as contradictory or not. The feature weights are hand-set, guided by linguistic intuition.

Number, date and time features. Numeric mismatches can indicate contradiction. The numeric features recognize (mis-)matches between numbers, dates, and times. We represent the date and time expressions in term of time range. Aligned numbers are considered to be mismatch when the surrounding terms match and numbers don’t.
Antonymy features. Aligned antonyms can be considered as hint for contradiction. We use Wordnet for the antonym list. We look at the aligned pairs for contradiction by checking the list and also check for common antonym prefixes (un, anti).

Structural features. These features aim to determine whether the syntactic structures of the text and hypothesis create contradictory statements. For example, we compare the subjects and objects for each aligned verb. If the subject in the text overlaps with the object in the hypothesis, we find evidence for a contradiction. Consider example 6 in table 1. In the text, the subject of succeed is Jacques Santer while in the hypothesis, Santer is the object of succeed, suggesting that the two sentences are incompatible.

**Results and Conclusions**

We were not able to get results as good as done by Andrade and Marneffe. We were not able to incorporate the subtask of simplifying the complex sentences i.e. those sentences which have more than one predicate into simpler. Due to which our model performed poor on the complex sentences.

Adding to it our methodology was more focused on detection of contradiction from the lexical mismatches which did not take care of the contradictions embedded in the semantics of the sentences and was not recognizable by only looking into it.

Much better results can be achieved by introducing features/methodology which incorporated semantics of the complete sentence.

<table>
<thead>
<tr>
<th></th>
<th>Marneffe[Stanford]</th>
<th>Andrade[NEC Corp.]</th>
<th>Our Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>62.7</td>
<td>64.51</td>
<td>44.07</td>
</tr>
<tr>
<td>Recall</td>
<td>62.9</td>
<td>57.0</td>
<td>34.6</td>
</tr>
<tr>
<td>Precision</td>
<td>62.5</td>
<td>32.95</td>
<td>32.5</td>
</tr>
</tbody>
</table>

**Future Work**

We plan to modify this model by adding a component implementing framenet technique to capture better semantic properties and try to implement similar model for Hindi.

**Acknowledgment**

We thank our professor Dr. Amitabha Mukerjee for his guidance. We appreciate the help provided by Daniel Andrade by giving more information about their work. And finally, the course teaching assistants Mr. M S Ram and Mr. Ravi Teja for providing help whenever needed.
References


http://doi.acm.org/10.1145/2382593.2382599