Debate Stance Identification using Unsupervised Techniques

Shashwat Chandra

1 Introduction

This report contains the work done by me towards my thesis topic in the second semester in the academic year 2013-14. The thesis topic is centered around semantic analysis, discourse analysis, information extraction, and textual entailment.

2 Literature Survey

The literature that was surveyed is present in the file ‘chandras.bib’ in this folder.

3 Text-Analysis approach

I have considered two approaches to solve this problem:

3.1 Emotion Detection

This approach aims to find a link between stance and emotion. The data for progressing along this path is scarce, leading the the possibility of conducting a survey. The survey would give a particular situation to the subjects, and ask them to react positively/negatively to that situation. This could lead to some sort of emotionally-driven stance analysis.

3.2 Text Analysis

This approach utilized TF-IDF and pLSA to detect keywords identifying stance. I looked at the difference in results between Word/Document relations and Word/Topic/Document relations.
3.2.1 Results

This House believes that children should be allowed to own and use mobile phones

Keywords returned: ('Yes', ['parent', 'ag', 'child', 'call', 'technolog'])

Mobile phones keep children safer, as it is easier for parents to stay in touch with their children and for children to contact someone in an emergency.

Keywords returned: ('No', ['parent', 'health', 'expens', 'cell', 'mani'])

There are possible potential long-term health risks from using mobile phones. Mobile phones are too expensive for children.

As we can see, the top keywords/topics returned by the algorithm give us the correct topics for the FOR/AGAINST topics.

4 Emotion-Phoneme relationship

A corrolary to the Emotion Detection problem is the following problem: Is there a relationship between the sound a word makes and the emotions attached with it? (Related to Bouba/Kiki effect)

I used SentiWordNet and used Word, Sense, Synset, and score information for 213,702 word senses. Wiktionary was used to extract pronunciation rules.

4.1 Results

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>Meaning</th>
<th>Emotion</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>h</td>
<td>Aspirated h</td>
<td>84.6% positive</td>
<td>52</td>
</tr>
<tr>
<td>v</td>
<td>nut</td>
<td>81.8% positive</td>
<td>22</td>
</tr>
<tr>
<td>Λ</td>
<td>gut</td>
<td>61.7% negative</td>
<td>1118</td>
</tr>
<tr>
<td>ο</td>
<td>not</td>
<td>62.1% negative</td>
<td>1060</td>
</tr>
<tr>
<td>f</td>
<td>fill</td>
<td>60.3% negative</td>
<td>1280</td>
</tr>
</tbody>
</table>

5 Hindi Stemmer

A major requirement to extend this pLSA approach to Hindi is a Hindi Stemmer. Another thing that I do not have at present is a Hindi Debate Database. A possible solution is to get the Lok Sabha/Rajya Sabha Database. The problem is that it’s in PDF format, making it hard to process. Additionally, the topics often go off-topic. Alternatives for a database include news pages and academic organizations.
5.1 Procedure

- Create a list of prefixes and suffixes using a corpus of Hindi Literature.
- Using grounded knowledge, select useful prefixes and suffixes and divide them into two categories on the basis of their prefix/suffix frequency.
  - Calculate the number of times the prefix occurs as a prefix versus the number of times it occurs in a normal word.
- The categories are: Safe Prefix/Suffix List and Unsafe Prefix/Suffix List.
  - Unsafe Prefix/Suffix List:
    - These prefixes/suffixes are those that occur as prefixes/suffixes with a low degree of confidence.
    - Words formed on removal of these prefixes/suffixes are checked using Hindi WordNet before being returned.
  - Safe Prefix/Suffix List:
    - These prefixes/suffixes are those that occur as prefixes with a high percentage.
    - Words formed on removal of these prefixes/suffixes, if they exist in Hindi WordNet, are returned instantly.
    - If the removal of a Safe/Unsafe prefix/suffix does not lead to a word present in Hindi WordNet, the Safe Prefixes/Suffixes are removed from that word, and the result is returned.

5.2 Results

The stemmer appears to have roughly 86% accuracy.

5.3 Next Steps

1. Create a dictionary of small words, since they are likely to give false positives in the process of removing unsafe prefixes and leaving a single letter.
2. Deal with Assimilation.
3. Deal with Infixes.
4. Remove spurious punctuation.
5. Use sentence-structure information.
6 Future Work

1. Improve on the accuracy of the Stemmer.
2. Include POS Tag information in the Stemmer.
3. Perform the unsupervised stance classification algorithm on a dataset in Hindi, or bilingual parallel corpora.
4. Perform survey and get relevant results for use in database creation for emotion/cognition-based approach.