NTCIR-11 Conference

NUL System at RITE-VAL tasks

Ai Ishii, Hiroshi Miyashita,
Mio Kobayashi, Chikara Hoshino
Technology Research & Innovation
Nihon Unisys, Ltd.
2014/12/10
Background

- Text Mining
- Common Sense
- Feature words extraction twitter bot
- UnNatural Language Processing

aimed at high-precision semantic analysis
joined RITE-VAL as bench mark
Agenda

• Introduction
• Shallow approach for RTE
• Search Strategy at FV
• Experimental Results & Discussion
• Future Efforts & Conclusion
Introduction

• Textual entailment recognition (RTE) system for two Japanese subtasks:
  – System Validation (SV)
  – Fact Validation (FV)

• Simple, but robust approach
Overview of our approach

• Shallow approach for RTE
  – linear classifier mainly based on
    • word overlap feature
    • named-entity feature

  – RITE-2’s “A strong shallow system” of team BnO as base system
    • improved named entity extraction
    • transformed some variables

• Apache Solr for FV
Shallow approach for RTE
System Architecture

Feature extraction, learning and classification
Feature Extraction Steps

For a pair of text T and hypothesis H:
1. Chunking
2. Named entity extraction
3. Number expressions extraction
4. Synonym finding
5. Features calculation
Feature Extraction - Chunking

• Chunking
  – divide T and H into word chunks using
    • KNP (for SV)
    • Cabocho (mainly for FV)

  word chunk is a independence word and subsequent attached word (KNP’s basic clause)

– identify a word chunk as a Content Word
  • eliminating some stop words
    – "する", "ある", "こと", "もの" : : : etc.
• Named entity extraction
  – max-length matching from the left
    • for each word chunk in H
    • with concatenating word chunks

  – The knowledge of named entity:
    • Wikipedia titles
      except
        – Nihongo goi taikei ‘s common noun
        – Some exclusion pattern from Wikipedia title list
    • Nihongo goi taikei’s proper nouns
Feature Extraction - Number expression extraction

• Number expression extraction
  – normalizedNumexp
    • extracting number expression
    • converting them into number or date range

with some hard cording
  example:
    – remove “一つ”(one) from number expression
      » It is often used, as “one of XX” than as number
Feature Extraction - Synonym finding

• Synonym finding
  For each word in H, find synonym in T
  – The knowledge of synonyms:
    • Wikipedia redirect, Japanese WordNet
    • Nihongo goi taikei
      – for orthographic variation
    • Wikipedia hypernym dictionary
      – by Hyponymy extraction tool
    • Levenshtein Distance
      – for orthographic variation

For each number expression in H
  – Numerical and temporal entailment recognition
    • number expression of H ⊃ T
Feature Extraction - Features calculation

- Features calculation
  features are defined by:
  
  – f1: Number expression correspondence
    
    • f1 = 1 if every number expression in H ⊃ T
    • Otherwise f1 = 0.1

  – f2: Named entity correspondence
    
    • f2 = 1 if every named entity in H as a synonym in T.
    • Otherwise f1 = 0.1

  – f3: Content word correspondence rate
    
    • f3 = D_H / L_H
      
      – L_H: number of words in H
      – D_H: number of words in H that have found their synonyms in T
Feature Extraction - Features calculation

Features calculation

Features are defined by:

– Other features
  
  • f4: Content word first character correspondence rate
  • f5: Word2vec cosign distance
  • f6: Exclusive word
  • f7: Non match content word rate
Learning & Classification

• Input
  – Features and answer labels (Y/N)

• Learning by Logistic Regression

• Classification
  – $y = 0$ or $1$
  – Threshold classifier at 0.5
Search Strategy at FV
Features for FV

Features are defined by:

- **f1**: Named entity correspondence
  - $f1 = 1$ if every named entity in $H$ has a synonym in $T$.
  - Otherwise $f1 = 0.1$
  - Named entity includes number expression

- **f2**: Content word correspondence rate
  - $f2 = \frac{\log(D_H + 1)}{\log(L_H + 1)}$
    - $L_H$: number of words in $H$
    - $D_H$: number of words in $H$ that have found their synonyms in $T$

- **f3**: Length of $H$
  - $f3 = \log(L_H + 1)$
Search Strategy

• “Distributed Search” of Solr
  – Search across multiple indexes
    • Wikipedia index
    • textbook index
  – Merge each search results

• Highest-scoring entry as T
Search Strategy

• Unit of search index
  – Search keywords should be near each other regardless of the word order
  – chose paragraphs as unit of search
  – separate by a newline

• Search query
  – weight named entity 5 times
Experimental Results & Discussion
# Experimental Results

## SV

<table>
<thead>
<tr>
<th>id</th>
<th>accuracy</th>
<th>Macro F1</th>
<th>Y-F1</th>
<th>N-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL-JA-SV-04</td>
<td>77.81</td>
<td>69.59</td>
<td>53.78</td>
<td>85.40</td>
</tr>
</tbody>
</table>

## FV

<table>
<thead>
<tr>
<th>id</th>
<th>accuracy</th>
<th>Macro F1</th>
<th>Y-F1</th>
<th>N-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL-JA-FV-03</td>
<td>63.23</td>
<td>61.93</td>
<td>54.89</td>
<td>68.97</td>
</tr>
</tbody>
</table>
Discussion for Effective features

As a result of replication study

• Effective features are follows:
  – f1: Number Expression Correspondence
  – f2: Named Entity Correspondence
  – f3: Content Word Correspondence Rate
Discussion for Search Strategy

• Unit of Search Index
  – Paragraph was very effective

• Search Query
  – Weighting named entity 5 times was not good for test datasets
Future Works

• Deep approach
  – based on syntactic parsing and inference

• Keywords specialized in domain
  – Word to be a singleton is different by the domain
    example:
    • "稲作" (rice crop)
    • "貿易" (trade)
Conclusion

• Effective features are follows:
  – Number Expression Correspondence
  – Named Entity Correspondence
  – Content Word Correspondence Rate

• Effective search unit is
  – paragraph by a new line
Ex-post Experimental Results of the search method

Effective method as follows:

- Paragraph as unit of search index
- Default weighting of Solr or avoiding length norm
- Sentence as it is for search query

<table>
<thead>
<tr>
<th>pattern</th>
<th>f1 CC</th>
<th>f2 CC</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
<td>test</td>
<td>dev</td>
</tr>
<tr>
<td>Base line</td>
<td>0.2510</td>
<td>0.2116</td>
<td>0.5212</td>
</tr>
<tr>
<td>paragraph</td>
<td>0.3279</td>
<td><strong>0.2951</strong></td>
<td>0.4611</td>
</tr>
<tr>
<td>paragraph + NE^5</td>
<td>0.3056</td>
<td>0.2711</td>
<td>0.2176</td>
</tr>
<tr>
<td>paragraph - TF</td>
<td>0.3086</td>
<td>0.2871</td>
<td>0.4569</td>
</tr>
<tr>
<td>paragraph - LN</td>
<td><strong>0.3117</strong></td>
<td>0.2945</td>
<td><strong>0.5408</strong></td>
</tr>
<tr>
<td>paragraph + synonym dic</td>
<td>0.3005</td>
<td>0.2771</td>
<td>0.4480</td>
</tr>
</tbody>
</table>

Base line: page unit index, default weighting, Sentence as it is for search query
CC = correlation coefficient, NE = Named Entity, TF = Term Frequency, LN = Length Norm
Experimental Results of variables

- good variable
  - f1: all named entity has synonym
  - f2: correspondence rate of content word
  - f3: number of words of H

<table>
<thead>
<tr>
<th>Variables</th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>Accuracy</th>
<th>MacroF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BnO (RITE-2)</td>
<td>1 or 0.1</td>
<td>f1 * log (D_H + 1)</td>
<td>f1 * log (L_H + 1)</td>
<td>64.13</td>
<td>61.87</td>
</tr>
<tr>
<td>Test-1</td>
<td>1 or 0.1</td>
<td>log (D_H + 1)</td>
<td>log (L_H + 1)</td>
<td>65.06</td>
<td>64.2</td>
</tr>
<tr>
<td>Test-2</td>
<td>1 or 0.1</td>
<td>log (D_H + 1) / log (L_H + 1)</td>
<td>log (L_H + 1)</td>
<td>65.52</td>
<td>64.59</td>
</tr>
</tbody>
</table>
Definition of Solr

• Default Weight

\[ w_{t,d} = tf_{t,d} \cdot idf_t^2 \cdot boost_t \cdot norm_d \]

\[ = frequency_{t,d}^{1/2} \cdot \left( 1 + \log \frac{N}{df_t + 1} \right)^2 \cdot boost_t \cdot norm_d \]

• Length Norm

\[ lengthNorm_d = \frac{1}{\sqrt{numTerms_d}} \]
Discussion for Synonym Finding

- Idea: matching of content words loosely
- It is effective, but remaining challenges are a lot

<table>
<thead>
<tr>
<th>Type</th>
<th>Good Example</th>
<th>Bad Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wikipedia Hypernym Dictionary</td>
<td>ウジェーヌ・ドラクロワ⊂画家 著作権⊂知的財産権</td>
<td>衆議院⊂国会議員 銀行⊂各国 搾取⊂企業</td>
</tr>
<tr>
<td>Levenshtein Distance</td>
<td>ユーゴスラビア&amp;ユーゴスラヴィア(dist:3) ゴードン内閣&amp;ゴードン改造内閣(dist:2)</td>
<td>アメリカ&amp;アフリカ(dist:1) ルイ15世&amp;ルイ14世(dist:1)</td>
</tr>
</tbody>
</table>
Comparison with other team

- **RITE-2 Data Set**

<table>
<thead>
<tr>
<th>Team</th>
<th>Accuracy</th>
<th>Macro F1</th>
<th>Y-F1</th>
<th>N-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest of RITE-2</td>
<td>64.51</td>
<td>58.12</td>
<td>41.76</td>
<td>74.48</td>
</tr>
<tr>
<td>Our System</td>
<td>65.42</td>
<td>65.12</td>
<td>61.03</td>
<td>69.00</td>
</tr>
</tbody>
</table>

- **RITE-VAL Data Set**

<table>
<thead>
<tr>
<th>Team</th>
<th>Accuracy</th>
<th>Macro F1</th>
<th>Y-F1</th>
<th>N-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Highest of the other team</td>
<td>57.20</td>
<td>56.57</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Our System</td>
<td>64.59</td>
<td>64.28</td>
<td>60.34</td>
<td>67.38</td>
</tr>
</tbody>
</table>
### Exclusion pattern list by Sekiguchi removed some entry

<table>
<thead>
<tr>
<th>一覧</th>
<th>うち</th>
<th>\u301aFHAIYU:</th>
<th>の旗$</th>
<th>行政区画$</th>
<th>¥d+</th>
<th>¥d+世紀$</th>
<th>¥d+年</th>
<th>明治.+年</th>
<th>大正.+年</th>
<th>昭和.+年</th>
<th>平成.+年</th>
<th>¥d+年代$</th>
<th>¥d+月¥d+日$</th>
<th>決議¥d+</th>
<th>¥d+条</th>
<th>^第¥d+</th>
<th>第.+回</th>
<th>第.+期</th>
<th>第.+次</th>
<th>道.+号.*線</th>
<th>^オリンピック.+選手団$</th>
<th>^全国高等学校野球選手権.+大</th>
</tr>
</thead>
<tbody>
<tr>
<td>における</td>
<td>おいて</td>
<td>日本の</td>
<td></td>
<td>県$</td>
<td>市$</td>
<td>区$</td>
<td>町$</td>
<td>村$</td>
<td>郡$</td>
<td>州$</td>
<td>出入口$</td>
<td>の統領$</td>
<td>の首相$</td>
<td>の国王$</td>
<td>形電車$</td>
<td>系電車$</td>
<td>駅$</td>
<td>高等学校$</td>
<td>中学校$</td>
<td>小学校$</td>
<td>幼稚園$</td>
<td>方法$</td>
</tr>
</tbody>
</table>