A Hybrid Recognition System for Check-Worthy Claims Using Heuristics and Supervised Learning

Team: Prise de Fer
Chaoyuan Zuo¹, Ayla Ida Karakas², Ritwik Banerjee¹
¹Department of Computer Science
²Department of Linguistic

Presented by
Chaoyuan Zuo
PhD Candidate
Introduction

A Hybrid Recognition System using Heuristics and Supervised Learning

Data Preprocessing
- Speaker Name Normalization
- Sub-Datasets Creation

Feature Design
- Feature Extraction
- Feature Selection

Model
- Imbalanced Learning
- Supervised Learning Methods
- Heuristics
Data Preprocessing

Speaker Name Normalization
- Hillary Clinton (D-NY)
- Former Secretary of State, Presidential Candidate
- Clinton

Sub-Datasets Creation
- Training Data: Debate
- Test Data: Debate & Speech

Hillary Clinton
Two Classifiers
Feature Extraction

- **Lexical Features**: Remove stopwords and stem the words
- **Semantic Features**: Named entity
- **Word Embedding**: Word Vector
- **Stylometric Features**: POS tags, tense, negations, selected tags of constituency parse trees
- **Affective Features**: Sentiment analysis, subjectivity, bias...
- **Metadata Features**: Binary non-linguistic features
- **Discourse Features**: Segment features
Feature Selection

**Part 1**
Univariate feature selection
Select 2000 best lexical features based on Chi-Square test

**Part 2**
Embedded feature selection

---

**Imbalanced Dataset!**

<table>
<thead>
<tr>
<th>Training Data</th>
<th># Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label 0</td>
<td>3895</td>
</tr>
<tr>
<td>Label 1</td>
<td>94 (2.36%)</td>
</tr>
</tbody>
</table>

---

- Under-Sampling
  - Create balanced subsets

- Feature Selection
  - SVM with linear kernel with L1 norm

- Features Combination
  - Over 3500 features → 2600
Model

System

Imbalanced Learning
- Under-sampling
- Over-sampling

Classifier
- SVM
- MLP

Heuristic
- Rule
- Strict Mode
Imbalanced Learning

From random over-sampling to ADASYN

\[ x_{new} = x_i + \lambda \cdot (x_{zi} - x_i) \]

\[ \lambda \in [0,1], x_{zi} \in k \text{ nearest-neighbors} \]

**ADASYN**: the number of samples generated for each \( x_i \) is proportional to the number of samples which are not from the same class than \( x_i \) in a given neighborhood

Classifier

SVM
- Linear kernel with L2 loss
- L2 Regularization
- Activation Function: Tanh
- Optimization: Adam

MLP
- 2 hidden layer: 100 and 8 units
- L2 Regularization
- Activation Function: Tanh
- Optimization: Adam
Heuristics

Motivation

False Positive Instances:

• “The USA, the USA, the USA…”
• “Can you imagine the people that are, frankly, doing so well against us with ISIS?…”

---

### Algorithm 1

Heuristics for assigning the check-worthiness score $w(\cdot)$ to sentences.

```plaintext
Require: category $\in \{\text{SPEECH, DEBATE}\}$,
\hspace{1em} strict_mode $\in \{\text{true, false}\}$, sentence $S$.
\hspace{1em} MIN_TOKEN_COUNT $\leftarrow 0$
\hspace{1em} if category is SPEECH then
\hspace{2em} if strict_mode then
\hspace{3em} MIN_TOKEN_COUNT $\leftarrow 10$
\hspace{2em} else
\hspace{3em} MIN_TOKEN_COUNT $\leftarrow 8$
\hspace{2em} end if
\hspace{1em} else
\hspace{2em} if strict_mode then
\hspace{3em} MIN_TOKEN_COUNT $\leftarrow 7$
\hspace{2em} else
\hspace{3em} MIN_TOKEN_COUNT $\leftarrow 5$
\hspace{2em} end if
\hspace{1em} end if
\hspace{1em} if $S_{\text{SPEAKER}}$ is SYSTEM then
\hspace{2em} $w(s) \leftarrow 10^{-8}$
\hspace{1em} end if
\hspace{1em} if $S_{\text{NUMBER OF TOKENS}} < \text{MIN_TOKEN_COUNT}$ then
\hspace{2em} $w(s) \leftarrow 10^{-8}$
\hspace{1em} end if
\hspace{1em} if $S$ contains “thank you” then
\hspace{2em} $w(s) \leftarrow 10^{-8}$
\hspace{1em} end if
\hspace{1em} if $S_{\text{NUMBER OF SUBJECTS}} < 1$ then
\hspace{2em} if category is SPEECH then
\hspace{3em} $w(s) \leftarrow 10^{-8}$
\hspace{2em} else if $S$ contains “?” then
\hspace{3em} $w(s) \leftarrow 10^{-8}$
\hspace{2em} end if
\hspace{1em} end if
```

---
### Results & Analysis

<table>
<thead>
<tr>
<th></th>
<th>MAP</th>
<th>MRR</th>
<th>MRP</th>
<th>MP@1</th>
<th>MP@3</th>
<th>MP@5</th>
<th>MP@10</th>
<th>MP@20</th>
<th>MP@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP*</td>
<td>0.1332</td>
<td>0.4965</td>
<td>0.1352</td>
<td>0.4286</td>
<td>0.2857</td>
<td>0.2000</td>
<td>0.1429</td>
<td>0.1571</td>
<td>0.1200</td>
</tr>
<tr>
<td>MLP_str</td>
<td>0.1366</td>
<td>0.5246</td>
<td>0.1475</td>
<td>0.4286</td>
<td>0.2857</td>
<td>0.2286</td>
<td>0.1571</td>
<td>0.1714</td>
<td>0.1229</td>
</tr>
<tr>
<td>ENS</td>
<td>0.1317</td>
<td>0.4139</td>
<td>0.1523</td>
<td>0.2857</td>
<td>0.1905</td>
<td>0.1714</td>
<td>0.1571</td>
<td>0.1571</td>
<td>0.1429</td>
</tr>
<tr>
<td>MLP_none</td>
<td>0.1086</td>
<td>0.4767</td>
<td>0.1037</td>
<td>0.2857</td>
<td>0.2857</td>
<td>0.2000</td>
<td>0.1286</td>
<td>0.1071</td>
<td>0.1000</td>
</tr>
</tbody>
</table>

**Table 1:** Results for the Check-Worthiness task of our submitted models: MLP* was the primary submission, along with two contrastive runs, MLP_str and ENS (MLP with strict heuristics and the ensemble model, respectively). MLP_none shows the results of the MLP without any heuristics being applied. The primary evaluation metric was mean avg. precision (MAP). The mean reciprocal rank (MRR), mean R-precision (MRP), and mean precision at k (MP@k) are also shown.
Results & Analysis

01. Tense: e.g. “We’re cutting taxes.”
02. Anecdotal stories
03. Rhetorical figures of speech
04. Duplicate sentences
05. Sentence Fragments:
   e.g. “Ambassador Stevens – Ambassador Stevens sent 600 requests for help”
Conclusion

- Feature Design
- Imbalanced Learning
- Heuristics

Future Work

- Deep syntactic features
- Automated name normalization
- Complex neural network
Oversampling with SMOTE

The SMOTE algorithm is parameterized with k_neighbors (the number of nearest neighbors it will consider) and the number of new points you wish to create. Each step of the algorithm will:

• Randomly select a minority point.
• Randomly select any of its k_neighbors nearest neighbors belonging to the same class.
• Randomly specify a lambda value in the range [0, 1].
• Generate and place a new point on the vector between the two points, located lambda percent of the way from the original point.

\[ x_{new} = x_i + \lambda \cdot (x_{zi} - x_i) \]
Oversampling with ADASYN

ADASYN is similar to SMOTE, and derived from it, featuring just one important difference. It will bias the sample space (that is, the likelihood that any particular point will be chosen for duping) towards points which are located not in homogenous neighborhoods.

\[ x_{new} = x_i + \lambda \cdot (x_{zi} - x_i) \]