Stony Brook University

AVIGNON CLEF 2018

Conference and Labs for the Evaluation Forum

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

Team: **Prise de Fer**

Chaoyuan Zuo[1], Ayla Ida Karakas[2], Ritwik Banerjee[1]

[1] Department of Computer Science

[2] Department of Linguistic

Presented by

Chaoyuan Zuo

PhD Candidate

# Content

# 01 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

02

# Model and System

# Data Preprocessing

**Speaker Name Normalization**

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

→ **Hillary Clinton**

**Sub-Datasets Creation**

- Training Data: Debate
- Test Data: Debate & Speech

→ **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

**Imbalanced Dataset!**

| Training Data | # Sentences |
|---|---|
| Label 0 | 3895 |
| Label 1 | 94 (2.36%) |

Part 1 Univariate feature selection

Select 2000 best lexical features based on Chi-Square test

Part 2 Embedded feature selection

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

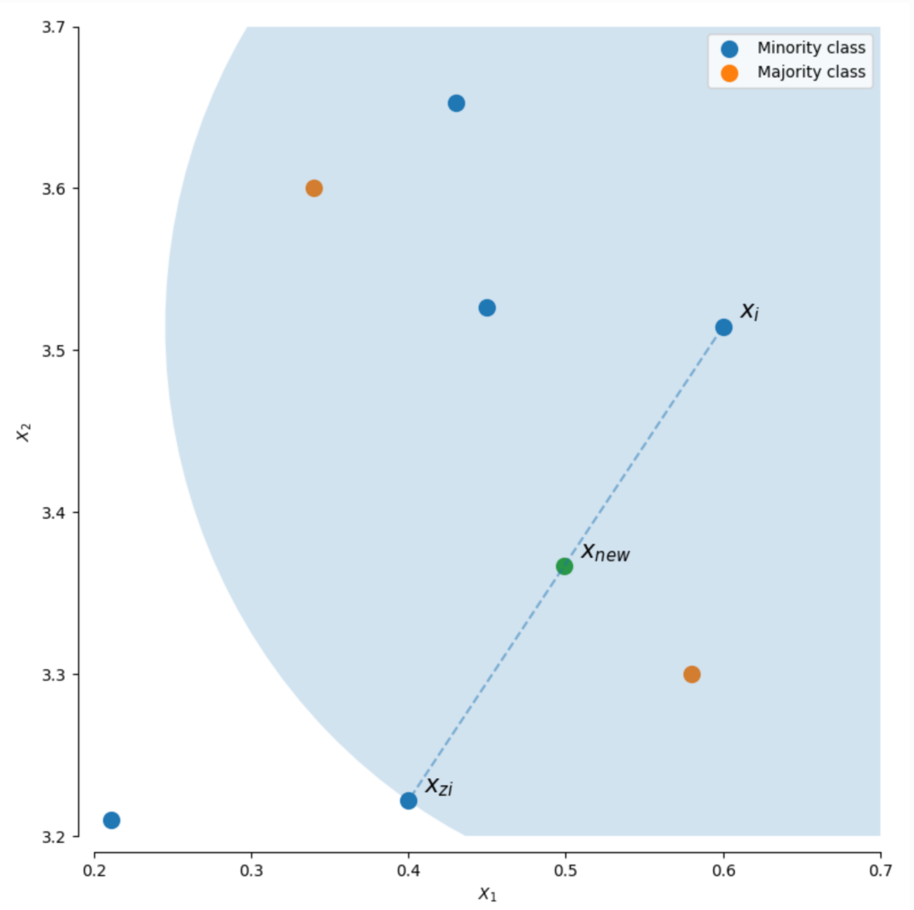
# Imbalanced Learning

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




http://contrib.scikit-learn.org/imbalanced-learn/stable/over_sampling.html#from-random-over-sampling-to-smote-and-adasyn

# Classifier

## SVM

- Linear kernel with L2 loss
- L2 Regularization

## 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.

**Require:** category $\in$ {SPEECH, DEBATE}, strict_mode $\in$ {**true**, **false**}, sentence $S$.

- MIN_TOKEN_COUNT $\leftarrow 0$
- **if** category is SPEECH **then**
  - **if** strict_mode **then**
    - MIN_TOKEN_COUNT $\leftarrow 10$
  - **else**
    - MIN_TOKEN_COUNT $\leftarrow 8$
  - **end if**
- **else**
  - **if** strict_mode **then**
    - MIN_TOKEN_COUNT $\leftarrow 7$
  - **else**
    - MIN_TOKEN_COUNT $\leftarrow 5$
  - **end if**
- **end if**
- **if** $S_{\text{SPEAKER}}$ is SYSTEM **then**
  - $w(s) \leftarrow 10^{-8}$
- **end if**
- **if** $S_{\text{NUMBER OF TOKENS}} <$ MIN_TOKEN_COUNT **then**
  - $w(s) \leftarrow 10^{-8}$
- **end if**
- **if** $S$ contains "thank you" **then**
  - $w(s) \leftarrow 10^{-8}$
- **end if**
- **if** $S_{\text{NUMBER OF SUBJECTS}} < 1$ **then**
  - **if** category is SPEECH **then**
    - $w(s) \leftarrow 10^{-8}$
  - **else if** $S$ contains "?" **then**
    - $w(s) \leftarrow 10^{-8}$
  - **end if**
- **end if**

# 03 Results & Analysis

| | MAP | MRR | MRP | MP@1 | MP@3 | MP@5 | MP@10 | MP@20 | MP@50 |
|---|---|---|---|---|---|---|---|---|---|
| **MLP***$^{\star}$ | 0.1332 | 0.4965 | 0.1352 | **0.4286** | **0.2857** | 0.2000 | 0.1429 | 0.1571 | 0.1200 |
| $\text{MLP}_{\text{str}}$ | **0.1366** | **0.5246** | 0.1475 | **0.4286** | **0.2857** | **0.2286** | **0.1571** | **0.1714** | 0.1229 |
| ENS | 0.1317 | 0.4139 | **0.1523** | 0.2857 | 0.1905 | 0.1714 | **0.1571** | 0.1571 | **0.1429** |
| $\text{MLP}_{\text{none}}$ | 0.1086 | 0.4767 | 0.1037 | 0.2857 | **0.2857** | 0.2000 | 0.1286 | 0.1071 | 0.1000 |

TABLE 1: Results for the Check-Worthiness task of our submitted models: **MLP**$^{\star}$ was the primary submission, along with two contrastive runs, $\text{MLP}_{\text{str}}$ and ENS (MLP with strict heuristics and the ensemble model, respectively). $\text{MLP}_{\text{none}}$ 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.

# 03 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*"

# 01 Conclusion

## Conclusion

- Feature Design
- Imbalanced Learning
- Heuristics

## Future Work

- Deep syntactic features
- Automated name normalization
- Complex neural network

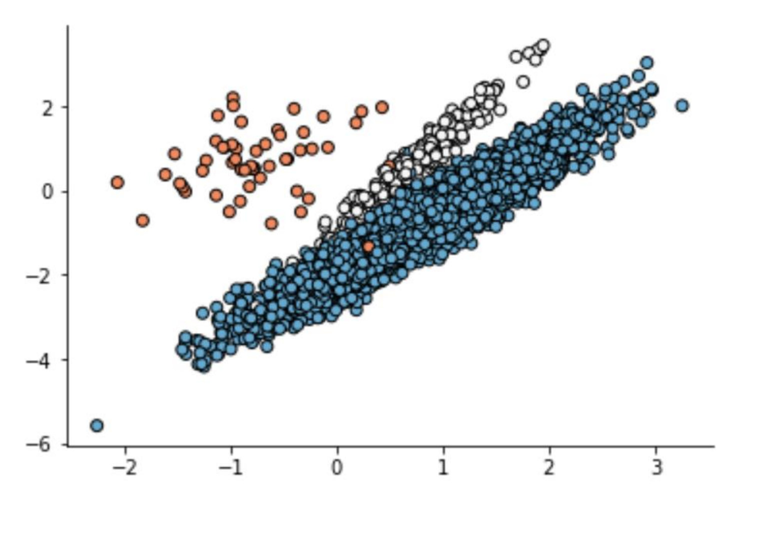
# Q & A

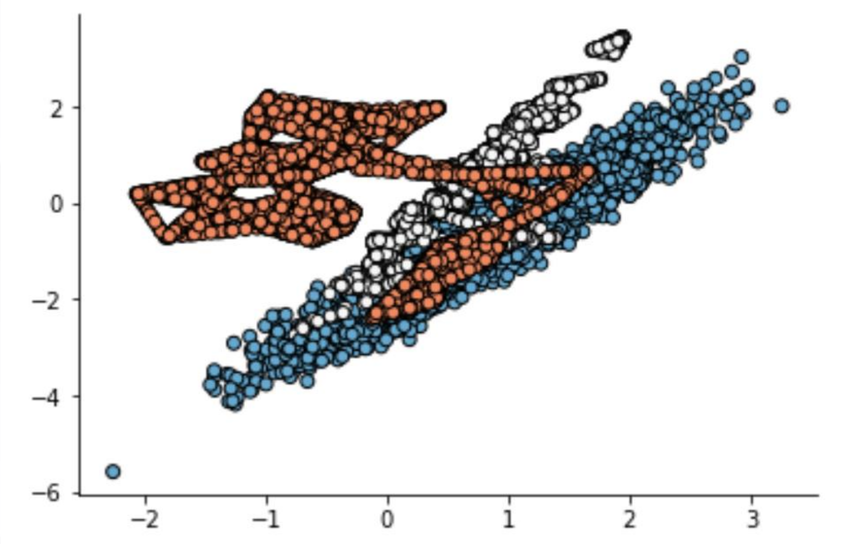
## Oversampling with SMOTE

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

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