Pun Classification and Location

Shantanu Karnwal shantanu.karnwal@colorado.edu

Amir Kashipazha amirhossein.kashipazha@colorado.edu

Brandon Boylan-Peck brbo9266@colorado.edu

Cathlyn Stone cathlyn.stone@colorado.edu

Kenneth Hunter Wapman kennethwapman@colorado.edu

Abstract

Pun identification and location are challenging natural language processing tasks. We implemented several algorithms for both, with results which were comparable to those outlined in the SemEval conference which originally defined the tasks.

1 Overview

1.1 Pun Structure

here we will talk about what puns are

- homographic
- heterographic
- other types

here we will talk about the problems we worked on.

homographic pun example

heterographic pun example

Please read carefully the instructions below and follow them faithfully.

1.2 Motivation

1.3 Tasks

section references 2, 3, and 5 below.

2 Pun Detection

here we will talk about pun detection and our algorithms.

University of Colorado, Boulder Machine Learning Project (2017), Boulder, CO.

Table 1: List of Features and Their Indexes in Code

Index	Feaute Name
0	Lesk Algorithm
1	Pos
2	tfidf
3	Embeddings
4	Unigram
5	Number of Homophone in Pun
6	Number of Each Homophone in Pun
7	Homophone is in Pun or Not
8	Idiom is in Pun or Not
9	Antonyms is in Pun or Not
10	Homonym is in Pun or Not
11	Bigram
12	Trigram
13	Positives
14	Negatives
15	All or First Caps

2.1 Baseline

2.2 Algorithms

2.3 Feature Comparison

3 Pun Location

3.1 Baseline

3.2 Algorithms

3.3 Feature Comparison

We introduced various features to improve accuracy Table 1. Indexes in this table represent the index of each feature in the code and figures we made for feature analysis. Features were added to the code one by one to find whether they improve accuracy. In general, we saw accuracy improved with adding features. Accuracy change behave differently against each feature.
Next step was to find what feature combination led to the largest accuracy. We designed a code to find various combinations of features. The code generated list of indexes of all possible number of feature a code can have. Length of the list was between one to number of feature, 1 to 16. With 16 feature we could have 65535 possible combination. The pun detection code took about 60s to be done. If we wanted to test all these combination it took about 1092 hours. In this regards, we did trail and error run to see what feature improve accuracy. Finally, we used 11 features that indexed from 0 to 10 in the table 1 for feature comparison.
With using 11 feature we were able to create 2047 features combination. Then we calculated accuracy for detection of homographic and heterographic puns. This let to 4089 runs. We found 100 percent accuracy anytime unigaram appears in any combination which is sign on over fitting. So we eliminated this feature form analysis.
Figures 1 and 2 shows the 10 largest accuracy of feature combination for training set of homographic and heterographic pun. Figures 3 and 4 shows the 10 largest accuracy of testing set for homographic and heterographic.

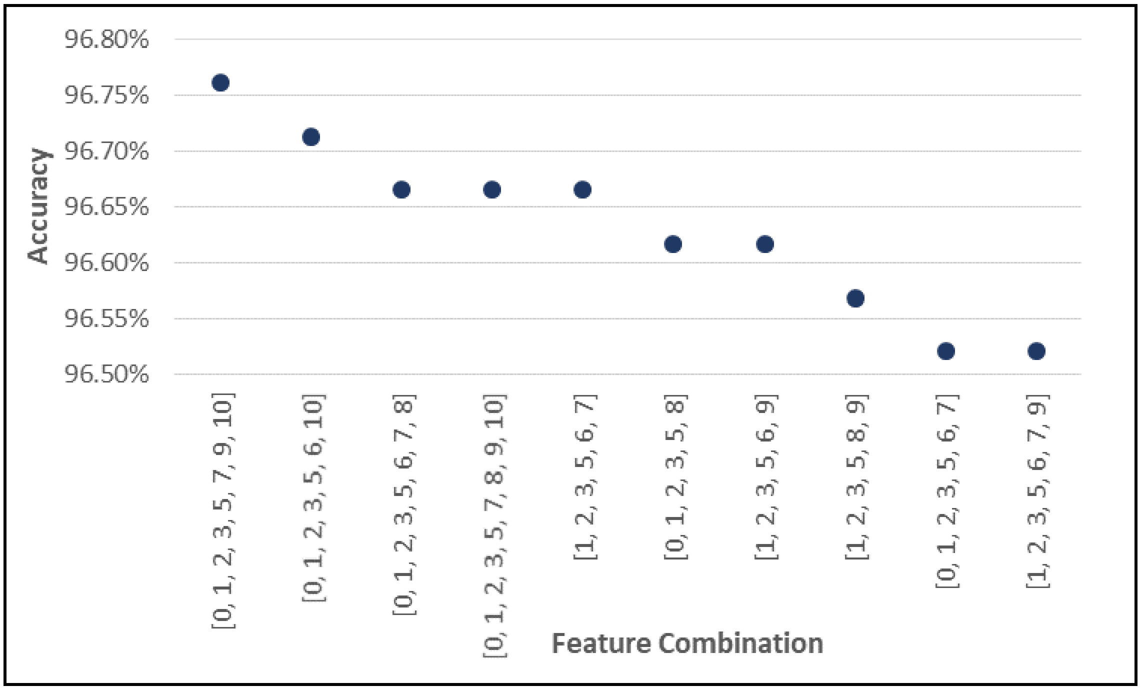

Figure 1: Accuracy on Training Set for Homographic Pun

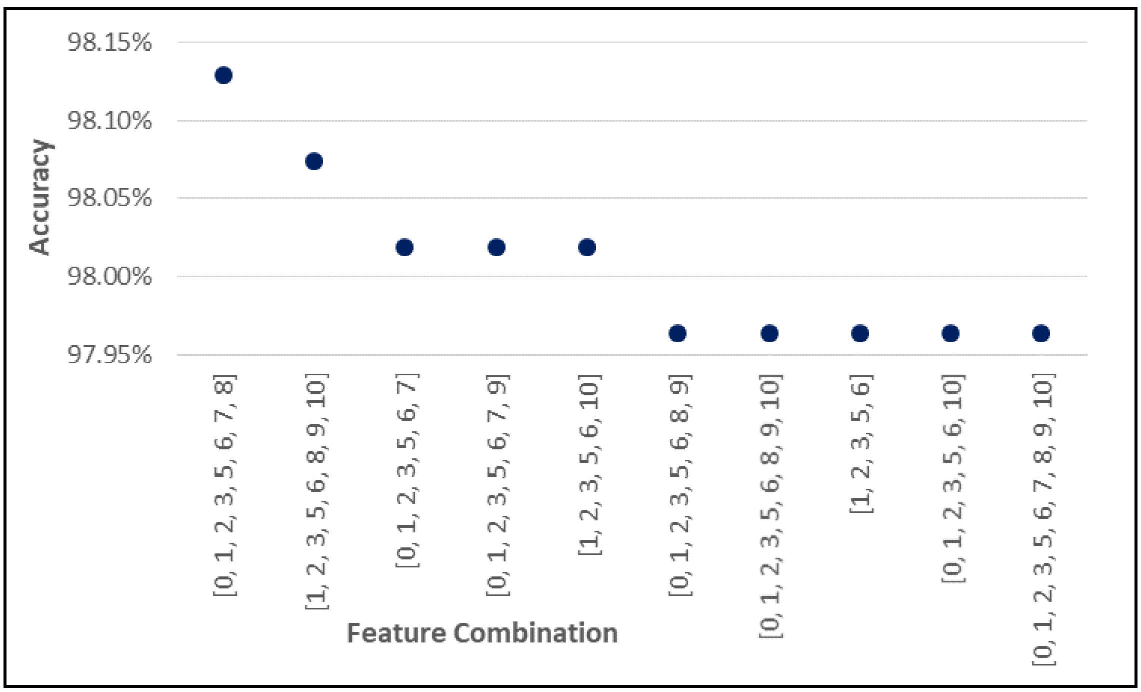

Figure 2: Accuracy on Training Set for Heterographic Pun

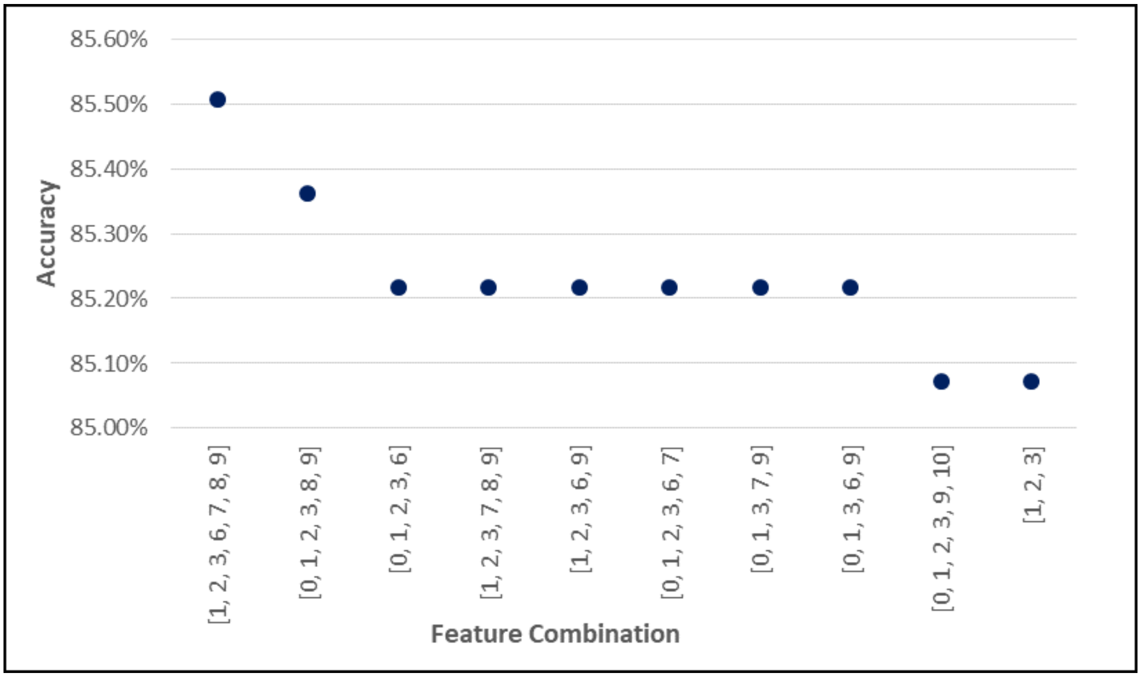

Figure 3: Accuracy on Testing Set for Homographic Pun

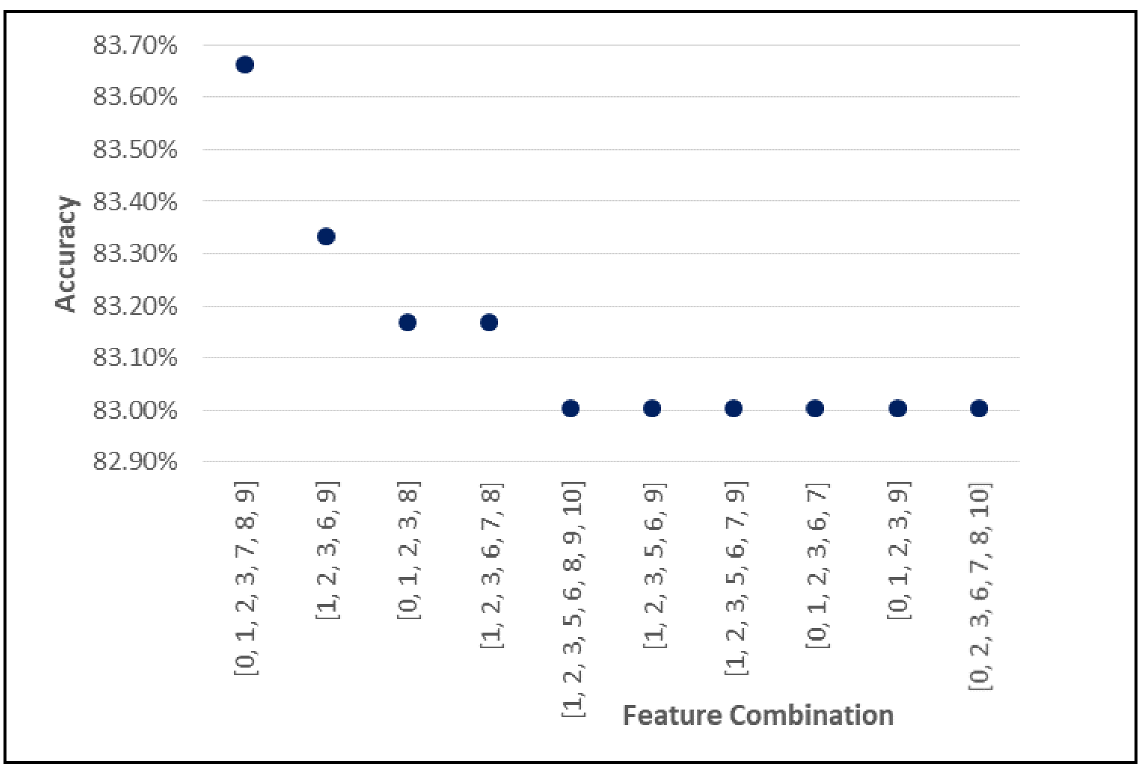

Figure 4: Accuracy on Testing Set for Heterographic Pun

4 Results

4.1 Pun Detection

Here's how we think we did.

4.1.1 Evaluation

4.1.2 Results

here are our results. here's the baseline. here's semeval's

4.1.3 Error Analysis

4.2 Pun Location

Here's how we think we did.

4.2.1 Evaluation

f score etc etc

4.2.2 Results

here are our results. here's the baseline. here's semeval's

4.2.3 Error Analysis

5 Conclusion

5.1 who did what

5.2 what went well

5.3 what we could have done better

Acknowledgments

here's where we acknowledge stuff

References

[1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

6 Style Stuff

`this is code-ish looking`

here's a percent: $\sim$15% **boldboldbold** *italics*

here's a fraction: ¼

Table 2: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

This is paragraphed

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated…

ref number: [4]

footnotes.[1]

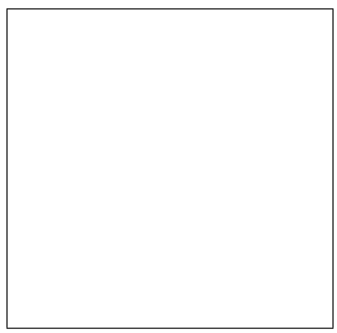

Figure 5: Sample figure caption.

Table 2.

use `booktabs` package for tables

`https://www.ctan.org/pkg/booktabs`

[1]they go after the period