

Authorship Attribution

Aparna Budhavarapu
1767790

Aashrit Mathur
1611541



Contents

- Introduction
- Data Used
- Libraries
- Method Implemented
- Features
- Performance comparison
- References

Introduction

Time frame: 1 month

Given a set of authors, Authorship Attribution(AA) is the task of figuring out who, if any of them is the actual author of a piece of text.

Data Used

- Amazon Dataset: ‘reviews’ table in amazon_db_software database[1]
- 1000 authors, 50 reviews per author
- **Preprocessing:** removing the URLs and converting the text to lower case

Libraries

- mysql.connector
- sklearn
- nltk
- numpy
- pandas
- textstat
- re

Method Implemented

- **n-Verifier model**
 - 2 class classification output
- For each author
 - Positive class: review belongs to that author
 - Negative class: review doesn't belong to that author
 - Training the model on equal number of positive and negative class examples

Features from baseline paper

- **Lexical :**
 - word unigrams
 - character unigrams, bigrams and trigrams
- **Syntactic:**
 - POS unigrams, bigrams and trigrams
- **Writing density:**
 - average number of characters per word
 - average number of syllables per word
 - average number of word per sentence

Features from baseline paper

- **Readability:**
 - Flesch-Kincaid grade level
 - Gunning fog index
- **POS trigram diversity:** number of unique POS trigrams normalized by the total number of POS trigrams
- **Stopword frequency:** total number of stopwords in the text divided by the overall count of the words in the text

Additional features

- **Punctuation:** count of punctuations used in the text. (‘.’, ‘,’, ‘:’, ‘?’, ‘:’, ‘!’)
- **Readability:**
 - Flesch reading ease
 - Smog index
 - Automated readability index
 - Coleman-Liau index
 - Linsear Write formula
 - Dale-Chall readability score
 - Difficult words

Performance Metrics

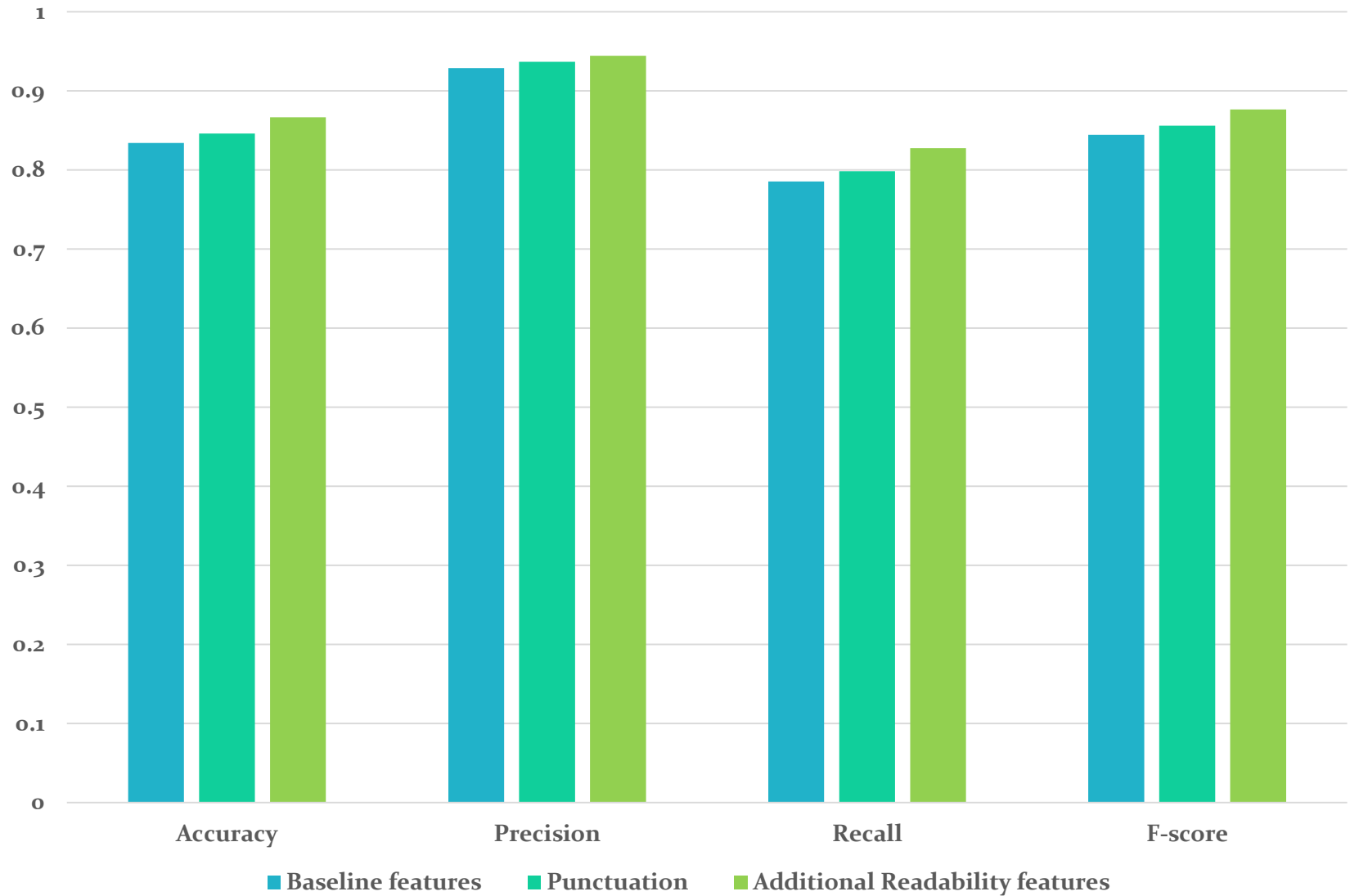
Baseline paper

Dataset	Method	Positive Class			Negative Class			Accuracy
		Precision	Recall	F-score	Precision	Recall	F-score	
Amazon Reviews	NOS	0.8674	0.9165	0.8846	0.9193	0.8423	0.8696	87.94
Amazon Reviews	NRS	0.8600	0.9162	0.8806	0.9187	0.8331	0.8639	87.47
Yelp Hotel	NOS	0.8517	0.8921	0.8678	0.8915	0.8358	0.8579	86.39
Yelp Hotel	NRS	0.8636	0.8916	0.8732	0.8927	0.8495	0.8656	87.05
Yelp Restaurant	NOS	0.8595	0.8757	0.8617	0.8804	0.8449	0.8557	86.03
Yelp Restaurant	NRS	0.8567	0.8799	0.8628	0.8825	0.8401	0.854	86.00

50 reviews per author (1000 authors)

Performance metrics	Baseline features	Punctuation	Additional Readability features
Accuracy	0.83419	0.84599	0.8667
Precision	0.9289	0.9368	0.9444
Recall	0.7853	0.7983	0.8275
F-score	0.8445	0.8559	0.8765

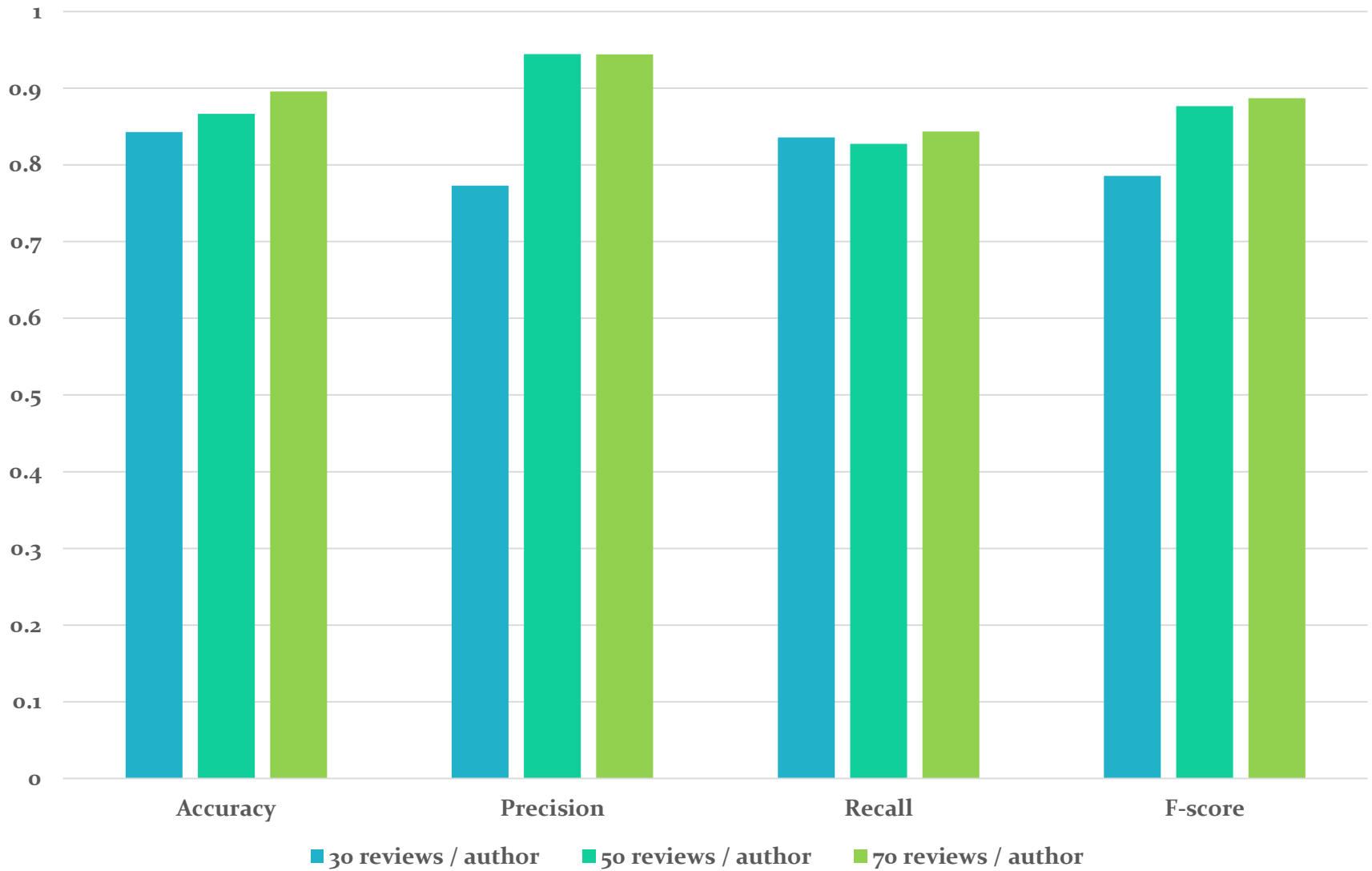
50 Reviews per Author(1000 Authors)



Comparison

Performance metrics	30 reviews / author	50 reviews / author	70 reviews / author
Accuracy	0.84283	0.86674	0.8958
Precision	0.77286	0.9444	0.9442
Recall	0.83575	0.8275	0.8437
F-score	0.78587	0.8765	0.8869

Comparison between 30, 50 and 70 reviews per author



To do

- Average Word Frequency Class
- Experiment on some additional features
- Ensemble learning
- Report

References

1. http://www2.cs.uh.edu/~arjun/papers_new/Shrestha%20et%20al.%20CICLING%2016.pdf
2. <https://blog.michaelckennedy.net/2017/06/21/yelp-reviews-authorship-attribution-with-python-and-scikit-learn/>
3. https://www.researchgate.net/publication/310799885_Generalized_Confusion_Matrix_for_Multiple_Classes
4. <https://pypi.org/project/textstat/>
5. <https://stackoverflow.com/questions/>



Any Questions?

THANK YOU