Detection of Constructive Comments in Online Discussions

NLP

Labbé Ugo, Bernal Nicolas, Karbeutz Gerhard

Overview

- Introduction
- Challenges
- Insights
- Preprocessing
- Classification
- Further reflection
- Conclusion

Introduction

- **Goal of the Project**
  - Classify constructive comments in online discussions.
  - Understand what makes a constructive comment
- **Dataset Used**
  - SFU Opinion and Comments Corpus (C3).
  - 12,000 online news comments annotated by crowdworkers

Definition

- **What is a constructive comment?**
  - Offer actionable suggestions
  - Provide well-reasoned arguments
  - Encourage meaningful dialogue or collaboration

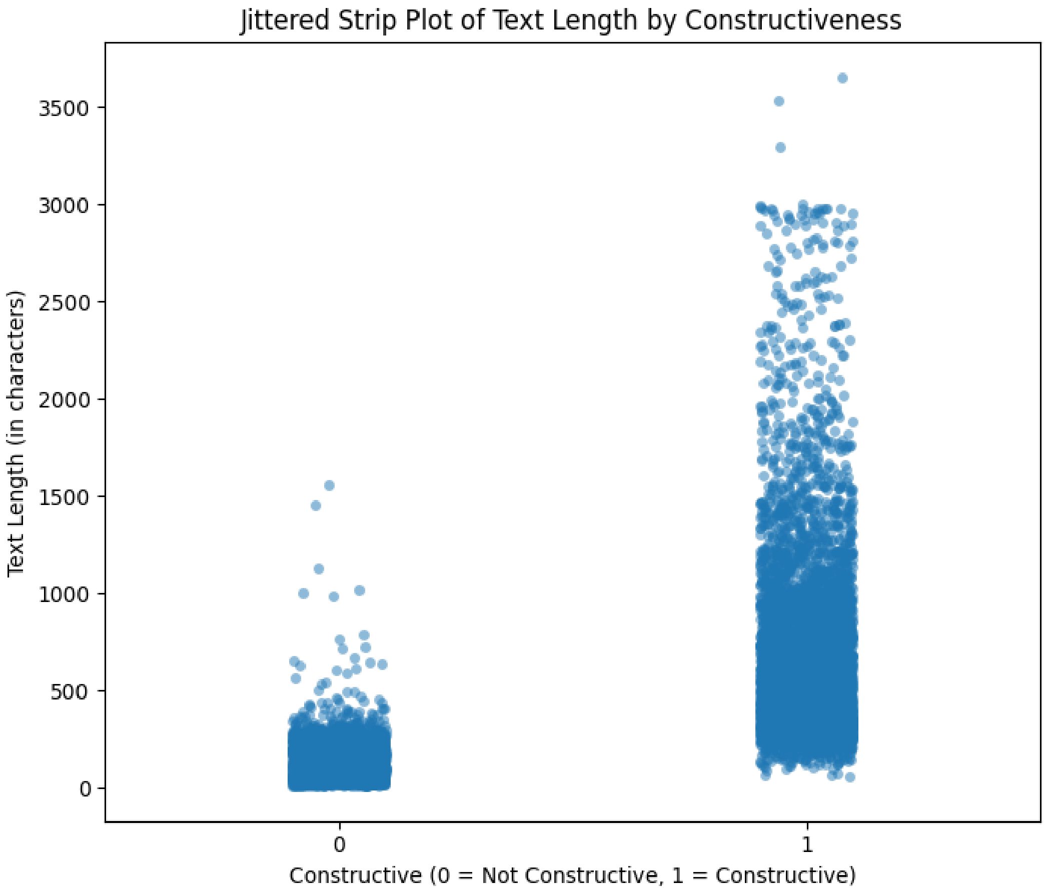
Comment example

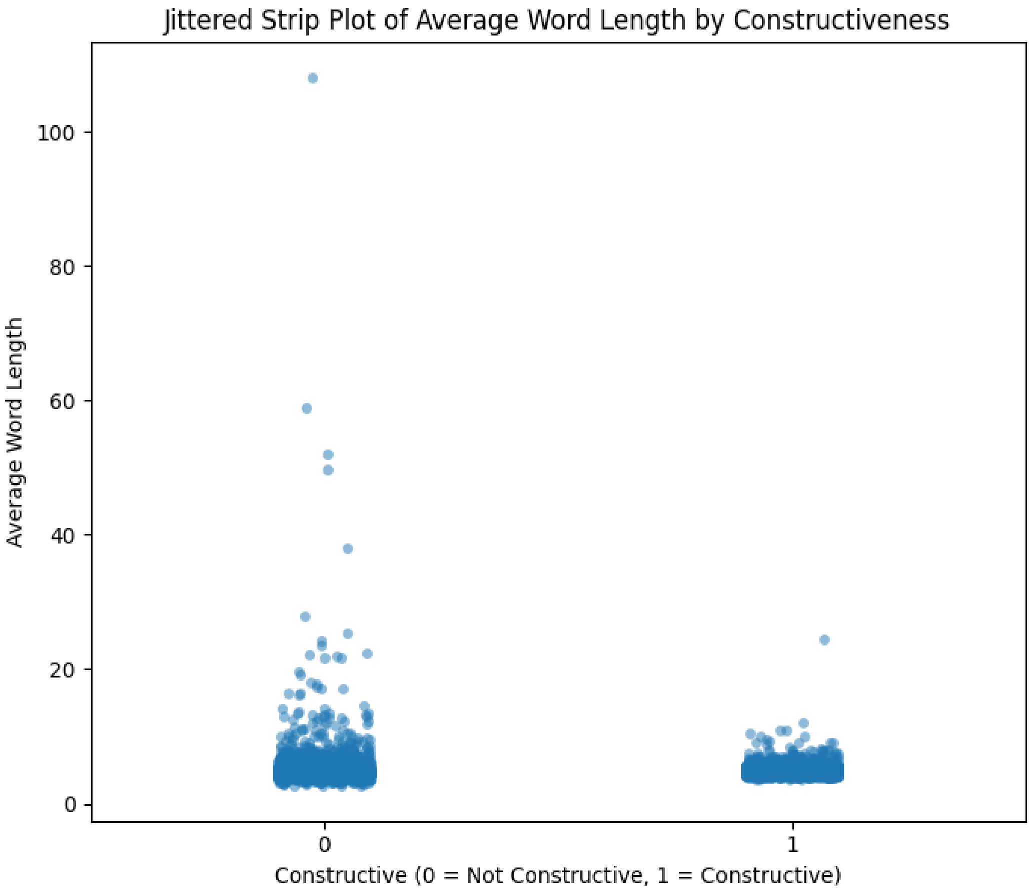
- **Non-constructive**: *“He takes a great selfie, which endears him to the social media crowd. But hypocrisy is an unattractive trait in any leader and this is a blatant example of just that.”*

- **Constructive**: *"Do you think your paper would ever endorse a New Democrat leader anywhere in Canada? If we all recall your editorial board endorsed Stephen Harper and look what that got us…a country that is viewed negatively worldwide."*

Insights

- Prevalence of certain words in the constructive class (e.g. "but", "should")





Challenges

- **Subjectivity**
  - Constructiveness is not universally defined and depends on the context and audience.
  - Different annotators might interpret the same comment differently.
- **Noisy Online Text**
  - Includes slang, emojis, typos, abbreviations, and incomplete sentences.
  - Example: “gunna”, ”gonna”, “going to”, links…

Dataset Preprocessing

- **Stanza Pipleline:**
  - Tokenization
  - Lemmatization
  - POS tagging

- **Tests**:
  - Removing stopwords
  - Removing punctuation

Baseline Models

- **Naive Bayesian Classifier** (using TF-IDF) and **Feature Based models** as baselines

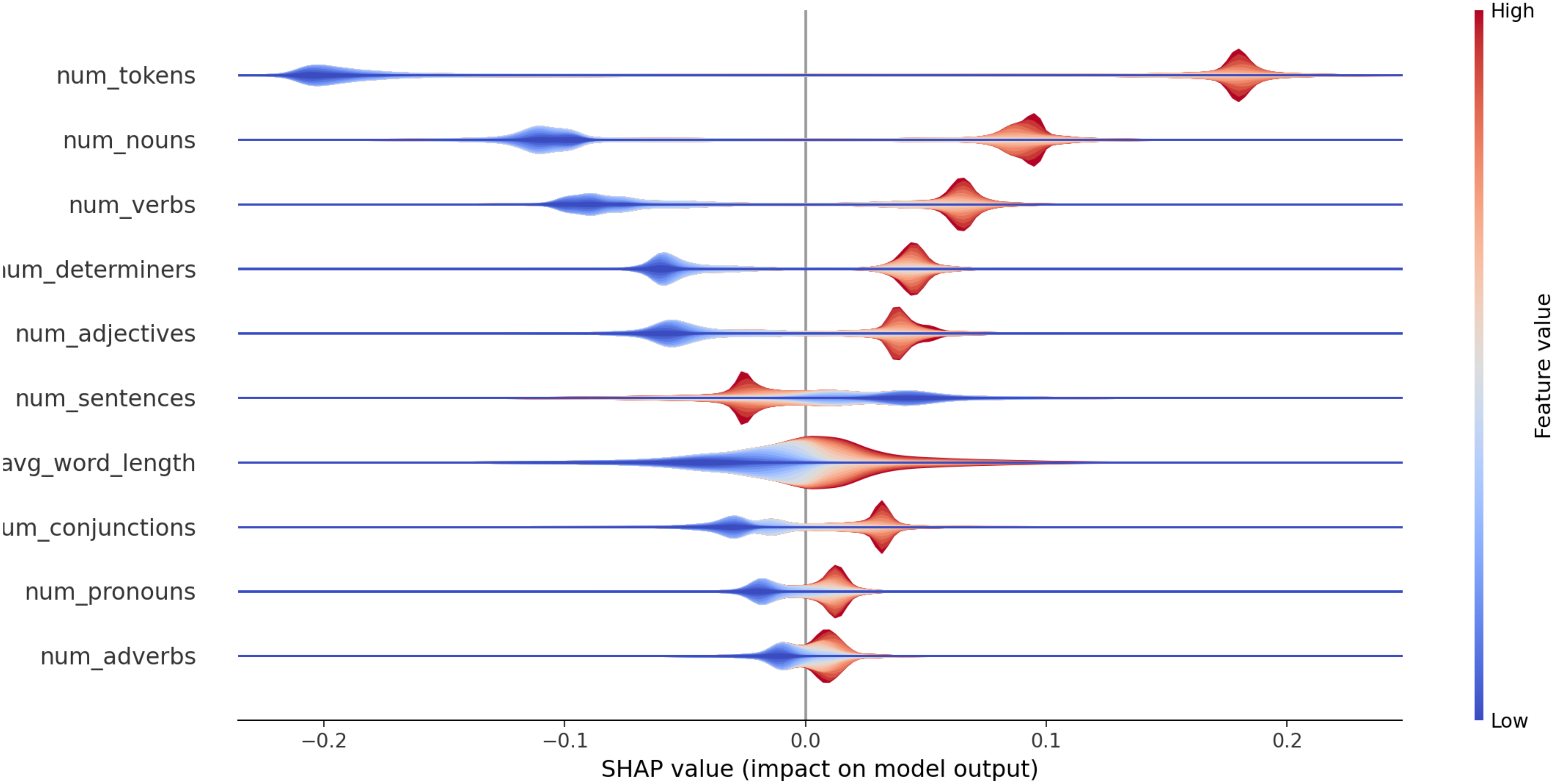
Classifier	Accuracy	Precision (0*)	Precision (1**)	Recall (0)	Recall (1)	F1-score (0)	F1-score(1)
NB original text	0.69	0.74	0.67	0.48	0.86	0.58	0.75
NB preprocessed text	0.68	0.73	0.66	0.47	0.85	0.57	0.75
K-NN	0.91	0.90	0.91	0.89	0.92	0.90	0.92
Logistic Regression	0.91	0.87	**0.95**	**0.94**	0.89	0.90	0.92
Random Forest	**0.92**	**0.91**	0.93	0.92	**0.93**	**0.92**	**0.93**
Stacking (NB+RF)	0.91	**0.91**	0.92	0.90	**0.93**	0.90	0.92

* Non Constructive
** Constructive
Precision: TP/(TP+FP)
Recall: TP/(TP+FN)
F1-score: 2*((Precision*Recall)/(Precision+Recall))

Qualitative analysis takeaways

- Features based model don't take into accounts what's written
- Example: '*that small symbolic acts have great power*'. *Yeah, symbology like the fact that some of the strongest proponents for massive lifestyle change just happen to be amongst the biggest carbon gluttons on the planet. You listening Dave?*
    - Comment classified as constructive, but not by our model
    - Is it really constructive?
    - What is the context?

SHAP Values


num_tokens
num_nouns
num_verbs
ıum_determiners
num_adjectives
num_sentences
avg_word_length
um_conjunctions
num_pronouns
num_adverbs
−0.2
−0.1
0.0
0.1
0.2
SHAP value (impact on model output)
High
Feature value
Low

Neural Network

- **Good Baseline Performance**:
- Achieved **85.25% accuracy** and **86.71% F1-Score** using TF-IDF features.
- Effective for simpler tasks without deep contextual needs.
- **Straightforward and Computationally Efficient**:
- Easier to implement compared to complex models like BERT.
- Less resource-intensive, making it suitable for limited hardware.
- **Lacks Contextual Understanding**:
- Relies on manually engineered features (TF-IDF), missing nuances like tone or semantic relationships.

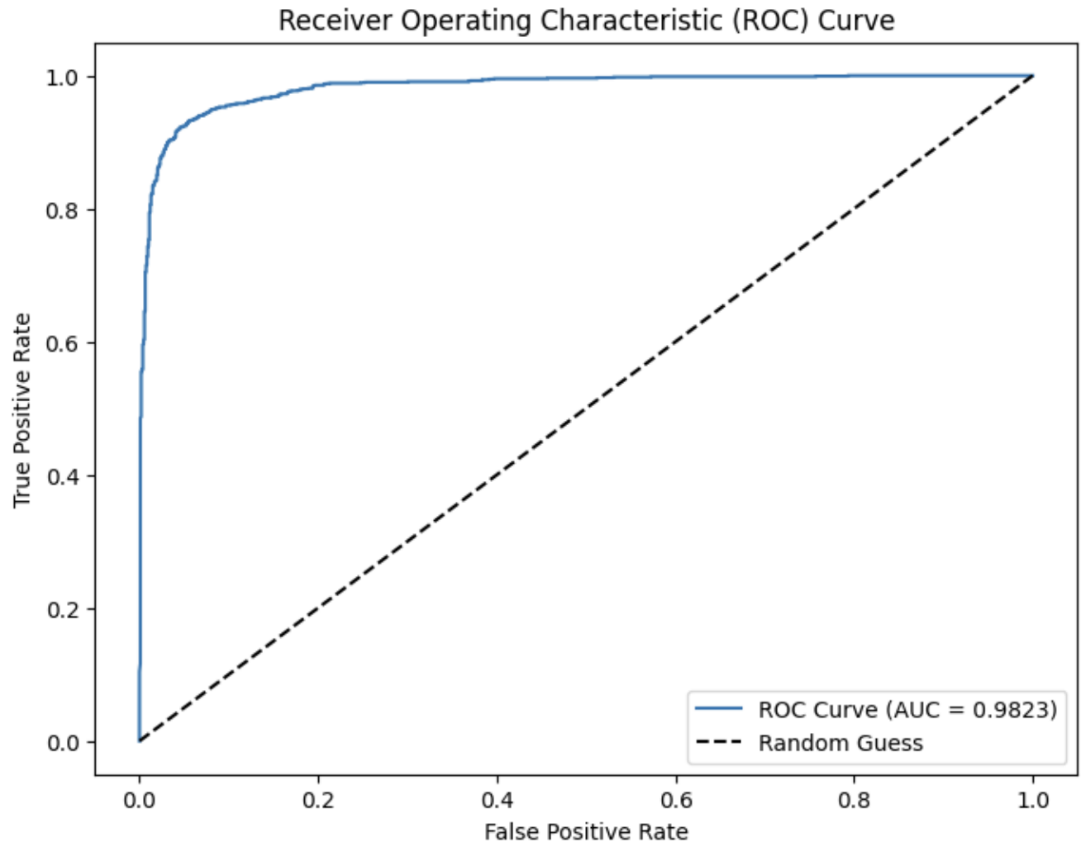
Deep Learning Takeaways

- **Insights**: NN performs well for simple tasks but lacks semantic understanding.
- BERT significantly outperforms NN in all metrics.
- Computational cost of BERT is a trade-off.

- **Future Work**: Experiment with larger datasets.
- Apply model on data set in different domain.
- Test other transformer models (e.g., RoBERTa, DistilBERT).

BERT

- **State-of-the-Art Performance**:
  - Achieved **93.63% accuracy** and **93.58% F1-Score**, significantly outperforming the neural network.
  - Handles complex semantic relationships and contextual nuances effectively.
- **Pretrained Power**:
  - Leveraged **BERT-Base Uncased**, pretrained on massive text corpora, making it highly effective for classification tasks with minimal preprocessing.
- **Handles Raw Text Seamlessly**:
  - No need for manual feature engineering or preprocessing like stopword removal or stemming.
  - Tokenization captures subwords and context (e.g., "running" → "run," "##ning").



Rule-Based Annotation

First approach (feature based)

- A comment its **CONSTRUCTIVE** if it has:
  - Verbs > 10
  - Adjectives > 8
  - Tokens > 300

Rule-Based Annotation

Second approach (feature based + keywords)

Features conditions (70%)

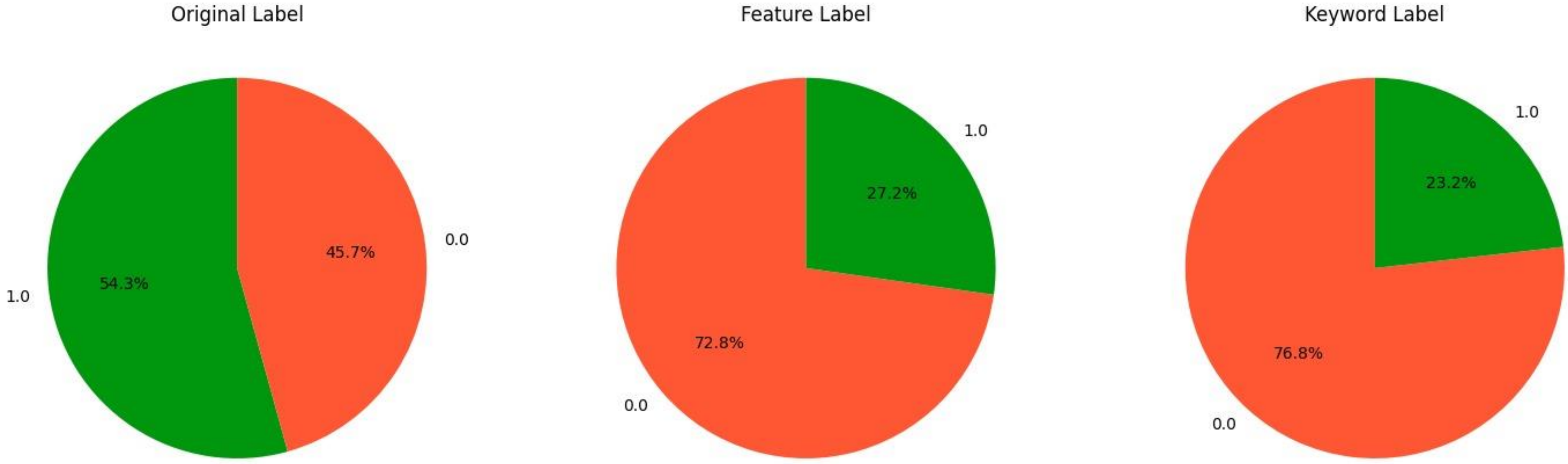
- Return **1** if:
  - Verbs > 10
  - Adjectives > 8
  - Tokens > 300
- Return **0.5** if:
  - Tokens > 300
- Otherwise return **0**

Keywords conditions (30%)

- Dictionary of **constructive** and **non-constructive** words.
- Ratio of "**constructive**" and "**non-constructive**" words.
- Range [-1, 1]

A comment is constructive if the weighted addition is >= 0.5

Rule-Based Annotation



Original Label

Feature Label

Keyword Label

Error Analysis

- The biggest flaw with our initial approach is handling complaint comments and those containing quotes, as they tend to be long and usually challenging to determine if they are constructive or not.

- Dictionary of **constructive** keywords ended up being too specific. Causing keyword-based notations to misclassify **constructive** comments.

Conclusion

- Hard to build robust algorithm with subjective annotations
- Lack of context
- Features importance and imbalance

Questions and Discussion

Any questions?