Reversing a hacker’s attack

By: Mahdi Shadkam-Farrokhi

Problem Statement: Hacker’s Note

> “
>
> When an anime character can simply kill opponents in one punch or by writing a name in a notebook, the plot quickly becomes stale and boring!
>
> Such shows become indistinguishable from each other. As proof, I've randomly swapped posts between **r/OnePunchMan** and **r/deathnote** - not that anyone could tell the difference...
>
> \- *Boop*

Plan of attack!

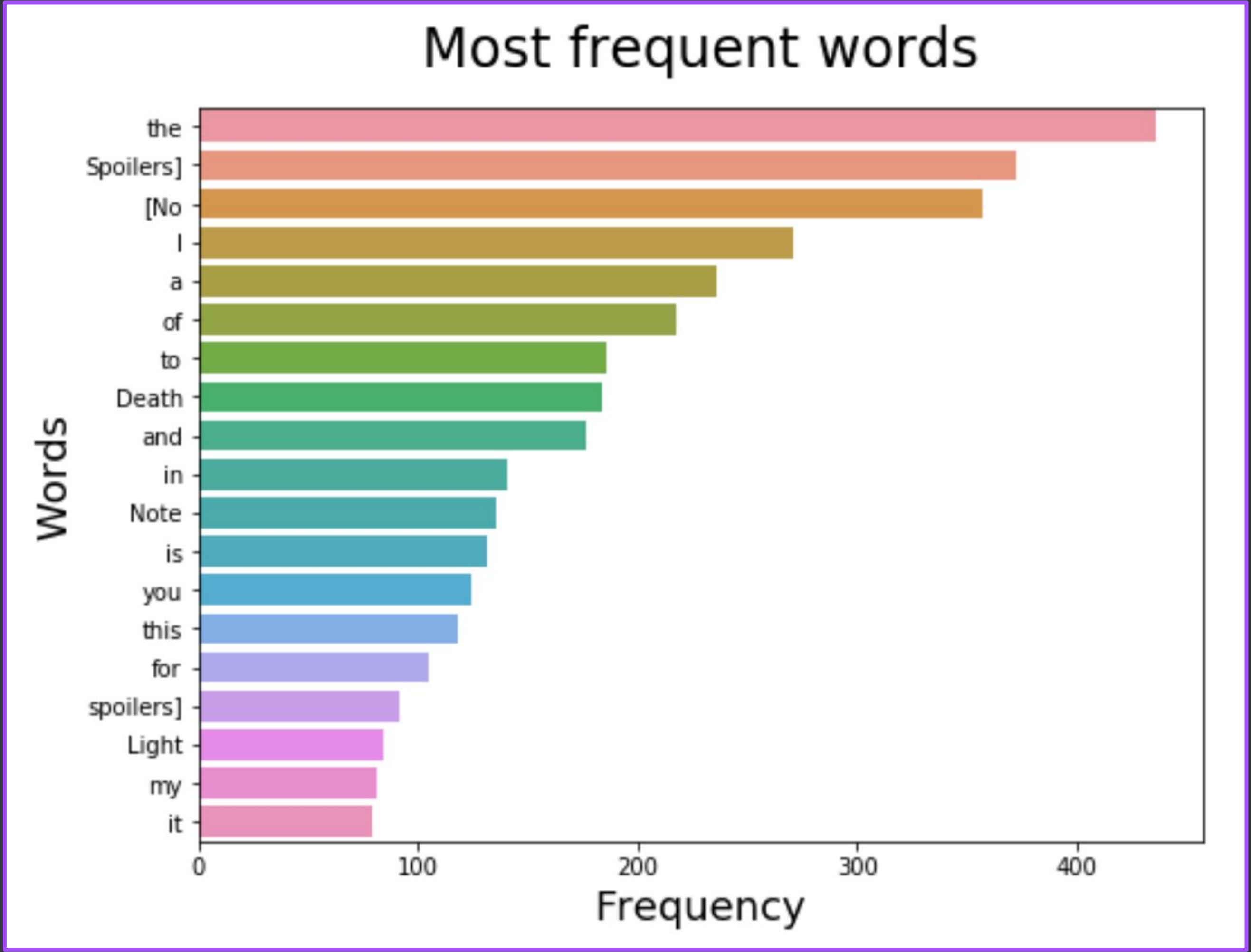
1. Use data from **r/OnePunchMan** and **r/deathnote**
2. Explore data and assess if they're distinguishable
3. Use NLP to build models to accurately distinguish subreddits
4. Evaluate models and select best
5. ***Implement and thwart the dastardly hacker!***
6. Observations & Conclusions

The data

- Pulled ~2000 total posts from **r/OnePunchMan** and **r/deathnote** from the Reddit API on 7/10/19
- **Only** title and subreddit from post used
- Issues:
  - Removing duplicates (~1600 posts left)
  - Preprocessing considerations (stemming, “L”, 😜, stop words)
- Final count:
  - 61% **r/deathnote**
  - 39% **r/OnePunchMan**

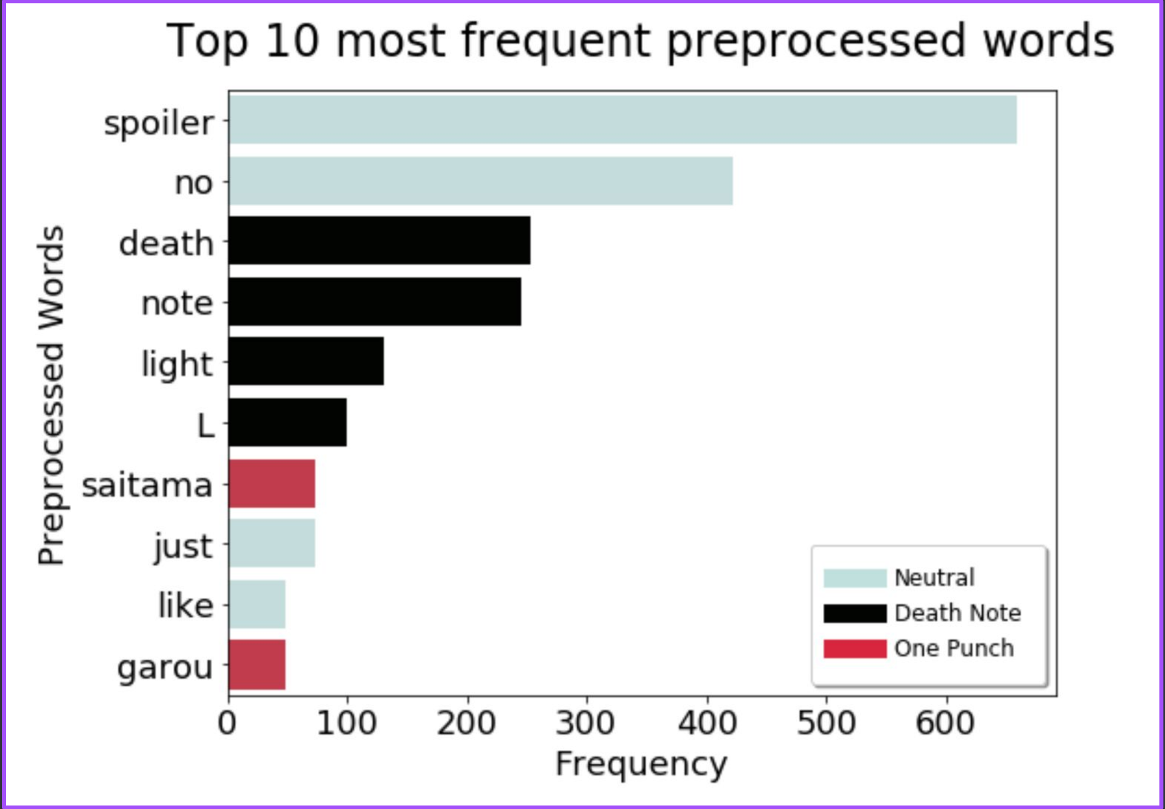
Exploratory Data Analysis

- Stop words
- Special characters
- Word casing
- Some keywords!

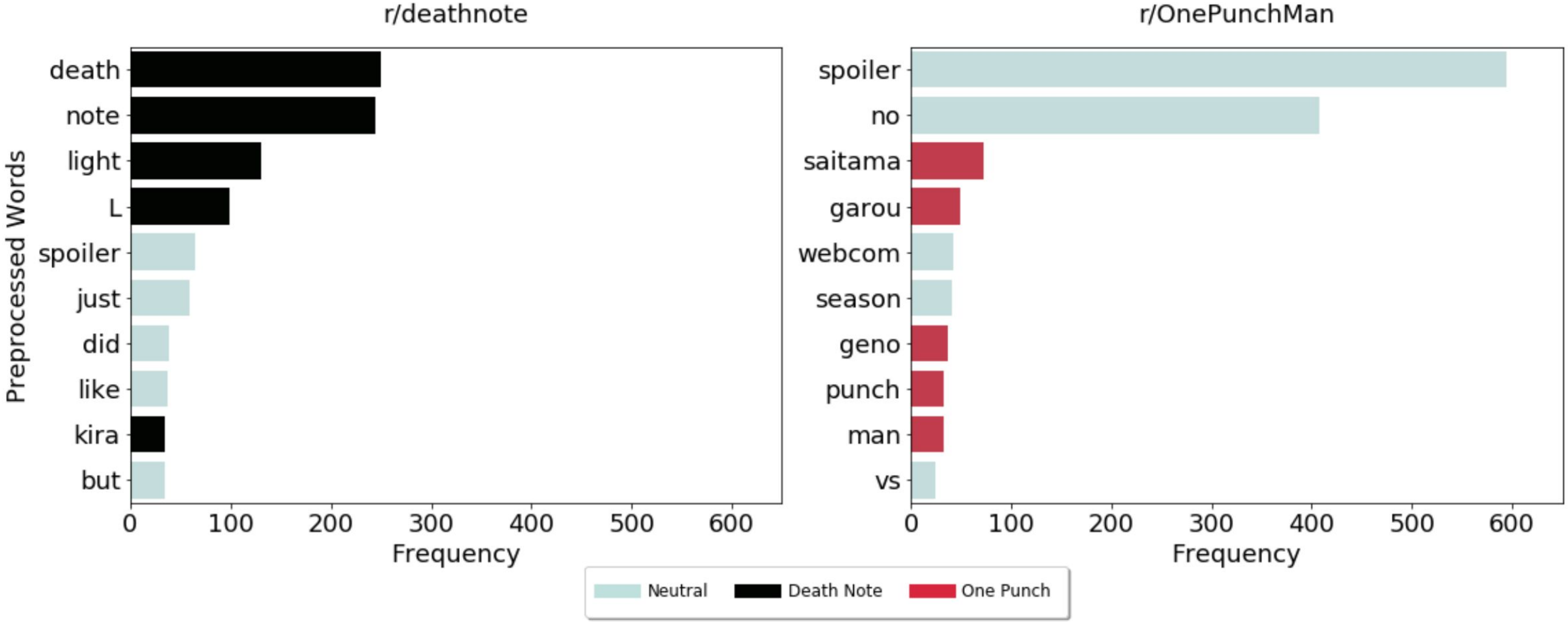


Preprocessed words

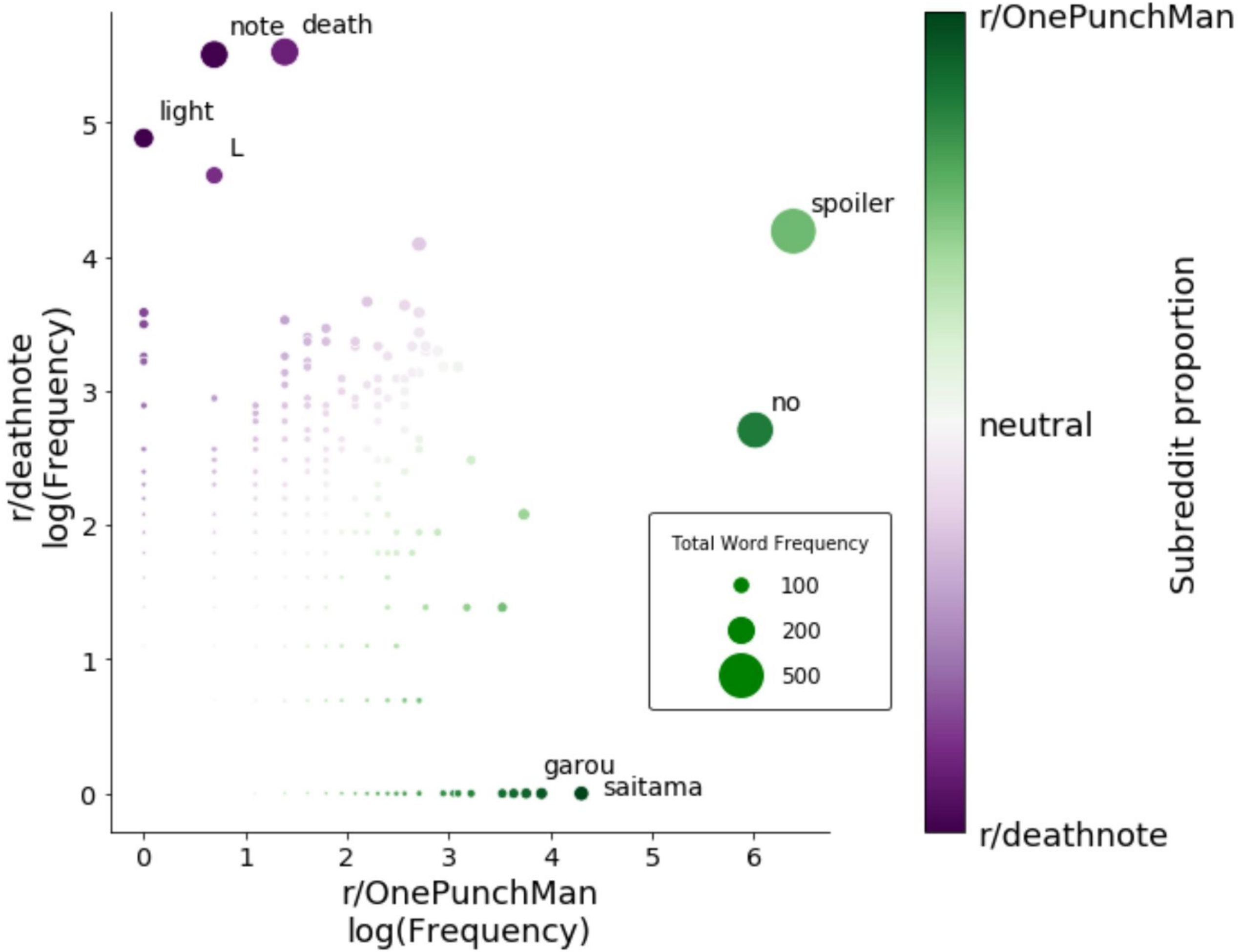
- Clear keywords found!
- Some “neutral” words
- Are these two subreddits distinct?



Top 10 most frequent words from...



Word Frequency: r/OnePunchMan vs. r/deathnote



Dummy Model

- Recall:
  - 61% **r/deathnote**
  - 39% **r/OnePunchMan**
- Blindly guessing a post is from **r/deathnote** will by right more often than not
- Any model with >61% accuracy will be better than blind guess!

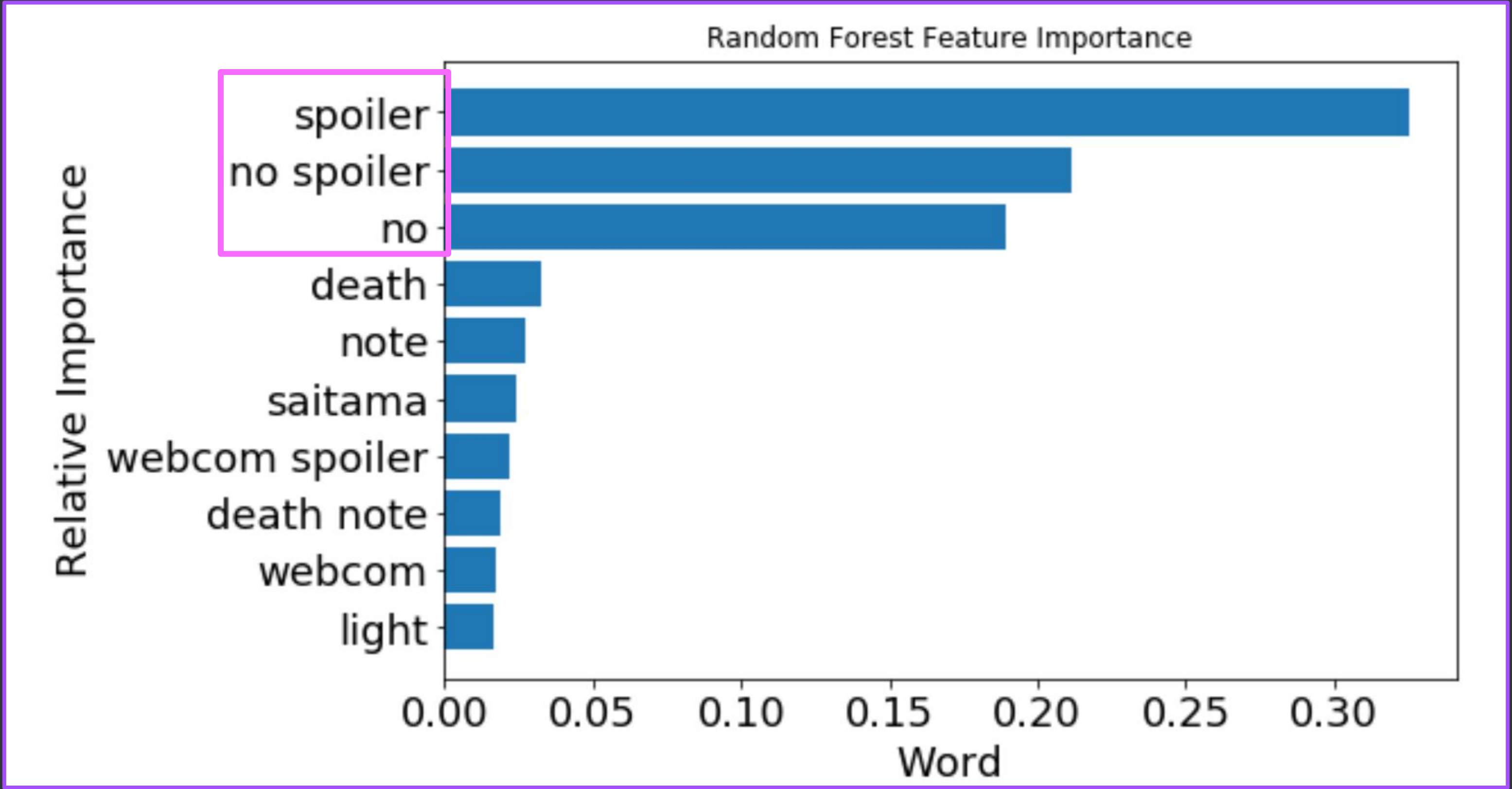
General workflow for models

1. Create pipeline
    a. Transformer (CountVectorizer, TfidfVectorizer*)
    b. Model (LogisticRegression, KNN, etc.)
2. Train/Test split
3. GridSearchCV to discover "best" parameters for model
4. Use ROC AUC to discover optimum classification threshold
5. Score model accuracy using threshold to predict on testing dataset

Models. Lots of model.

MODEL	Train Acc	Test Acc	False Pos	False Neg
TFIDF+LogReg	97.63%	96.69%	11	2
CVEC+LogReg	98.39%	96.69%	9	4
TFIDF+KNN	97.63%	94.66%	9	12
CVEC+KNN	98.64%	96.18%	12	3
CVEC+MULTNB	96.95%	97.20%	9	2
CVEC+GuasNB	94.40%	92.62%	23	6
CVEC+Tree	97.96%	95.93%	11	5
CVEC+BagTree	98.05%	96.18%	10	5
CVEC+RandFor	94.83%	93.13%	1	26
CVEC+GradBoo	98.22%	96.44%	10	4
CVEC+ADA	98.05%	96.95%	9	3
voter	98.47%	96.95%	10	2

CVEC+RandFor : Feature Importance



And the winner is...

- Technically...CVEC+MULTNB at ~97% Accuracy
- BUT other models performed well!
    - CVEC+RandFor best for predicting **r/OnePunchMan**
    - TFIDF+LogReg, CVEC+KNN, CVEC+ADA were great at predicting **r/deathnote**
    - Voter is likely more stable w/new data

CVEC+MULTNB : Missing Data Points

Some representative missing data points:

1. [**spoiler**] could someon explain someth to me?
    - Wrongly Predicted **r/OnePunchMan**
2. biggest plot hole of **opm**
    - Wrongly Predicted **r/deathnote**
3. whi is **fubiki** so perfect?
    - Wrongly Predicted **r/deathnote**

Model implemented!

We’ve restored peace to Reddit!

..well, ~97% peace, but that’s pretty good!

The Hacker is thwarted...for now

If they attack again, we’re ready with a highly effective model and a workflow Reddit developers can use to quickly fix a similar attack

Well...sort of...

Conclusiones y Recomendaciones

Improvements

- Unbalanced classed (bootstrap minority class)
- Use voter model with fine-tuned weights

Assumptions

- Stop words don’t matter
- Stemming converges sentiment
- Numbers, special characters, abd letter casing don’t matter

Portability

- Won’t work for other subreddits!
- “SPOILERS”: Death Note anime is finished - One Punch Man still active, but won’t be forever!