Persuasion Extraction

Jared Amen - CS 6390 (Information
Extraction From Text)

The Task

- Based off of SEMEVAL-2021 Task 6, Subtask 2, on detection of persuasive techniques in texts and images[1]
- **Inspiration:** Information which is purposefully shaped to foster a predetermined agenda can be considered propaganda.
- This information can contain:
  - Hard-to-detect logical fallacies
  - Incendiary emotional tactics
- In the modern age, memes are a common method to distribute such propaganda
  - Can reinforce/complement any combination of fallacious techniques
- **The goal of the task:** to build a model for extracting sequences that fit any number of 20 common fallacious persuasive techniques in the textual content of a meme.

The Labels

For this task, our desired labels are[2]:

- Appeal to Authority
- Appeal to Fear/Prejudice
- Black-and-White Fallacy/Dictatorship
- Causal Oversimplification
- Doubt
- Exaggeration/Minimization
- Flag-Waving
- Virtue Signaling
- Loaded Language
- Straw Man Arguments
- Name Calling/Labeling
- Intentional Vagueness
- Red Herrings
- Reductio Ad Hitlerum
- Repetition
- Slogans
- Smears
- Thought-Terminating Clichés
- Whataboutism
- Bandwagoning

The Data

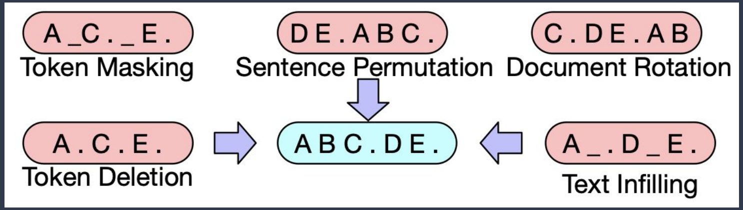
For this task, SEMEVAL provided a total of:[3]

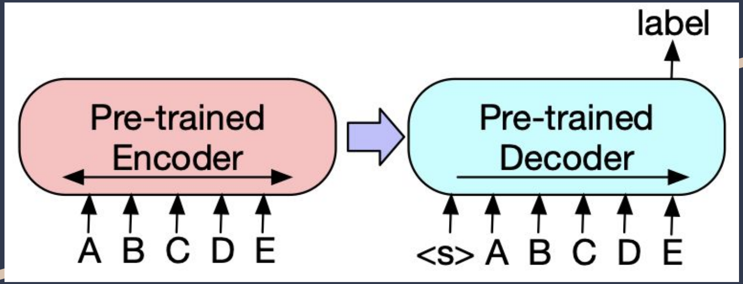
- 687 training entries
- 63 validation entries
- 200 testing entries

All entries were textual content from memes, along with expected gold spans with one of the previous techniques assigned for each entry.

In addition, a corpus of ~20000 sentences from a previous SEMEVAL competition with a smaller set of the same labels was adapted for use in training this model.[4]

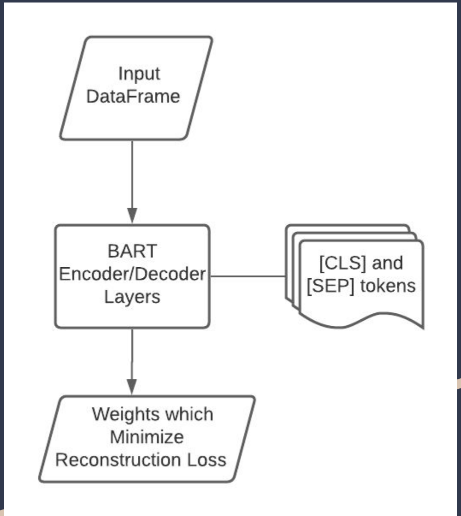
The Model





- **Sequence-to-Sequence Model**, which utilizes BART
- BART is a denoising autoencoder/decoder where each layer utilizes a bidirectional encoder over a “corrupted” text and a left-to-right autoregressive decoder[5]
  - Seeks to optimize reconstruction loss -- the cross-entropy between the decoder’s output and the original text
  - Loss is computed in negative-log-likelihood
  - Activation functions are gaussian linear units instead of rectified linear units[6]
  - The base model utilizes 6 encoder/decoder layers, and the large model utilizes 12
- BART is stronger than alternatives here (BERT, roBERTa, etc.) because of its general nature and its ability to handle sequence classification/tagging tasks[5]

Implementation



Libraries used:

- Huggingface Transformers, for BART transformer models and tokenization
- SimpleTransformers, for a simple implementation of the Seq2Seq model using BART as the encoder/decoder

Input:

- A Pandas DataFrame with `input_text` and `target_text` columns, where the `target_text` column contains the input sentences with spans tagged with a start (CLS)/end (SEP) tag that corresponds to the used technique[7]
- Example sentence: `He called them [S-6] “true American heroes.” [E-6]`
  - Corresponds to the “Flag-Waving” technique

Results

As indicated by SEMEVAL, I evaluated my base model based on:

- Micro-F1, Precision and Recall which all account for partial matching between spans
- Micro-F1 for each propaganda technique

```
Please type in the number of sentences you'd like to provide to the model: 2
Accepting 2 sentences...
Type in a sentence you'd like to provide to the model: Make America Great Again!
Type in a sentence you'd like to provide to the model: I like cheese.
Generating outputs: 100% 1/1 [00:00<00:00, 3.04it/s]

======================================================
Sentence:   Make America Great Again!
Found an instance of Slogans in "Make America Great Again!"
Found an instance of Flag-waving in "Make America Great Again! "
======================================================
Sentence: I like cheese.
Could not find persuasive phrases
```

```
F1 = 0.267
Precision = 0.386
Recall = 0.204
F1_Appeal to authority = 0.245
F1_Appeal to fear/prejudice = 0
F1_Black-and-white Fallacy/Dictatorship = 0.25
F1_Causal Oversimplification = 0
F1_Doubt = 0.064
F1_Exaggeration/Minimisation = 0.301
F1_Flag-waving = 0.22
F1_Glittering generalities (Virtue) = 0
F1_Loaded Language = 0.388
F1_Misrepresentation of Someone's Position (Straw Man) = 0
F1_Name calling/Labeling = 0.302
F1_Obfuscation, Intentional vagueness, Confusion = 0
F1_Presenting Irrelevant Data (Red Herring) = 0
F1_Reductio ad hitlerum = 0
F1_Repetition = 0
F1_Slogans = 0.263
F1_Smears = 0.301
F1_Thought-terminating cliché = 0
F1_Whataboutism = 0.128
F1_Bandwagon = 0
```

Insights

```
Appeal to authority: P=0.978 R=0.140 F1=0.245
Black-and-white Fallacy/Dictatorship: P=1.0 R=0.143 F1=0.25
Doubt: P=0.922 R=0.033 F1=0.064
```

Really precise for some labels, just inconclusive in general :(

- For the base model, many labels went completely unclassified
  - This is due to underrepresentation (or complete lack of representation) in the training set, especially when considering the adapted old training set
  - The old training set includes some labels grouped into one “combined” label -- these had to be thrown out entirely
- The effects of a small dataset (especially with so many labels to consider) are felt here
  - As such, the performance of my model is on par with official submissions to this competition, showing that my model is competent given the parameters of the task
- Precision is always much higher than Recall
  - The model generally predicts correctly when it makes a prediction, but is often inconclusive

Ablation Studies/Insights

	Base	Large	MNLI	CNN/DailyMail
F1	0.267	0.297	**0.313**	0.259
Precision	0.386	0.412	**0.407**	0.262
Recall	0.204	0.233	**0.255**	0.257

The **MNLI** (multi-genre natural language inference) model performed the best, which makes sense!

- The MNLI model consists of the Large Model with an additional two-layer sequence classification head finetuned on the multi-genre natural language inference corpus[8]

```
F1 = 0.313
Precision = 0.407
Recall = 0.255
F1_Appeal to authority = 0.352
F1_Appeal to fear/prejudice = 0
F1_Black-and-white Fallacy/Dictatorship = 0.25
F1_Causal Oversimplification = 0.385
F1_Doubt = 0.195
F1_Exaggeration/Minimisation = 0.313
F1_Flag-waving = 0.286
F1_Glittering generalities (Virtue) = 0
F1_Loaded Language = 0.444
F1_Misrepresentation of Someone's Position (Straw Man) = 0
F1_Name calling/Labeling = 0.303
F1_Obfuscation, Intentional vagueness, Confusion = 0
F1_Presenting Irrelevant Data (Red Herring) = 0
F1_Reductio ad hitlerum = 0
F1_Repetition = 0
F1_Slogans = 0.294
F1_Smears = 0.29
F1_Thought-terminating cliché = 0
F1_Whataboutism = 0
F1_Bandwagon = 0
```

Phase 2

- **In Phase 2**, I attempted to utilize a bidirectional LSTM with a conditional random field layer for token-based transitional predictions using a BILOU labeling scheme applied to the provided datasets for the intended task
    - This was not effective in any measure
    - Even after phase 2's submission, accounting for 'O' tag filtering during training, performance was not good (roughly a 6.1% micro-F1 score).
    - Phase 2's submission also did not account for 'O' tag filtering during training, which explained good accuracy despite poor predictions

Phase 3

- **For Phase 3,** I switched to the SimpleTransformers-implemented sequence-to-sequence model utilizing BART which was adapted to sequence tagging (explained above)
  - Used a CLS-SEP labeling scheme applied to the provided datasets for the intended task
  - Also utilized an adapted dataset from a previous competition under the same organization which had the same label set
  - This yielded usable results and was surprisingly easier to implement than Phase 2
- Whereas Phase 2 was focused on a single-label sequence tagging problem, the implementations in Phase 3 fit a multi-label sequence tagging problem.

Lessons Learned

- Don’t be afraid to utilize cutting-edge technology! In the case of transformers, that technology is actually quite simple and effective!
- Although model architectures might be built initially for a certain purpose, those which are general enough can be adapted to fit other requirements (with the right amount of research!)
- Do your absolute best to expand your dataset if at all possible/applicable!

Works/Research Cited

All other work is done by Jared Amen, this team's sole group member.

[1] SemEval 2021 task 6 on "Detection of persuasion techniques in texts and images". (n.d.). Retrieved April 27, 2021, from https://propaganda.math.unipd.it/semeval2021task6/

[2] Propaganda Technique Definitions. (n.d.). Retrieved April 27, 2021, from https://propaganda.math.unipd.it/semeval2021task6/definitions.html

[3] Di-Dimitrov. (n.d.). SEMEVAL Task6 Corpus. Retrieved April 27, 2021, from https://github.com/di-dimitrov/SEMEVAL-2021-task6-corpus

[4] Semeval 2020 task 11 "detection of propaganda techniques in news articles". (n.d.). Retrieved April 27, 2021, from https://propaganda.qcri.org/semeval2020-task11/

[5] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., . . . Zettlemoyer, L. (2020). BART: Denoising Sequence-to-sequence Pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. doi:10.18653/v1/2020.acl-main.703

[6] Hendrycks, D., & Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). arXiv:1606.08415.

[7] Yeung, A. (2020, June 18). Albert Yeung. Retrieved April 27, 2021, from https://albertauyeung.github.io/2020/06/19/bert-tokenization.html

[8] MultiNLI Corpus. (n.d.). Retrieved April 27, 2021, from https://cims.nyu.edu/~sbowman/multinli/

A website to test out this model will soon be up @ propdetector.com!

Thank you!