STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Extrinsic Evaluation of Document Embedding Techniques

Classifying Subreddit comments using semantic similarity to measure the quality of embedding techniques.

Brenden Brusberg
CS-584: Natural Language Processing
May 17th, 2021

Introduction

Brenden Brusberg

- Last Semester Undergraduate – Computer Science Major
- First Semester Graduate – MS in Machine Learning
- Data Scientist at a Consulting Firm
  - Primarily Machine Learning
  - Products and Service deployments
  - Ad-Hoc Analyses
  - Mostly People Analytics

Framing the Problem

Cosine Similarity Solutions

- Let's say you have a large corpus of text that you can’t possibly look through and need to find a certain document.
  - You could search via looking at key words.
    - What happens if documents you are looking for don’t have the exact key words?
    - What happens if you don’t know what key words you are looking for and just have a general idea?
  - You have a survey with a open response question with hundreds of thousands of responses and you want to find representative quotes.
  - You are making a tool that is able to search documents in your company that stretch past multiple years and projects.
- The answer is to encode your corpus into vector space and use co-sine similarity to find documents. ***<u>However, how do you measure how well your corpus is represented?</u>***

Evaluation of Word Embeddings

Overview

There is no one way to evaluate the quality of word embeddings. Many papers break it down by the task the embeddings are created for. However, embeddings in transfer learning can be applied to tasks they weren't originally trained on.

Common problems in evaluating word embeddings.

1. Obscureness of the notion of semantics.
2. Lack of proper training data.
3. Absence of correlation between intrinsic and extrinsic methods.
4. Lack of significance tests.
5. The hubness problem. "It is unclear how to deal with so-called hubs which are word vectors representing very frequent words."

- A sentence embedding is like a word embedding in which you can represent a sentence instead of a word using embeddings.
- A document embedding is the same principle applied to documents (a collection of words, sentences, or even paragraphs, etc.)
- There is a long history of document embedding methods stemming from using LDA (Latent Dirichlet allocation) to averaging word embeddings to doc2vec and more.
- A great history if you want to learn more can be found here.

Evaluation of Document Embeddings

Intrinsic vs Extrinsic

Intrinsic Evaluators: “Methods of intrinsic evaluation are experiments in which word embeddings are compared with human judgments on words relations.”

- **Word semantic similarity**
- Word analogy
- Thematic fit
- Concept categorization
- Synonym detection
- Outlier word detection
- Dictionary definition graph
- **Semantic difference**
- Thesaurus Intrinsic Evaluation (QVEC score)
- Linguistic-driven methods
- And more.

Extrinsic Evaluators: ”based on the ability of word embeddings to be used as the feature vectors of supervised machine learning algorithms (like Maximum Entropy Model) used in one of various downstream NLP task.”

- Noun Phrase Chunking
- Named Entity Recognition
- Sentiment Analysis
- Shallow Syntax Parsing
- Semantic Role Labeling
- Negation scope
- Part-of-Speech Tagging
- **Text Classification**
- Metaphor Detection
- Paraphrase Detection
- Textual Entailment Detection

Data Overview

Reddit Subreddit Comments

- I sampled comments randomly from November 2019 by Subreddit from a open sourced data repository.
- I tried selecting 50,000 comments from 11 Subreddits that were greater than 6 tokens and less than 500 tokens.
- In the end, I then randomly selected 5,000 comments from each class to be a train and test set. And used the remaining comments for fine tuning BERT and S-BERT.
- I used a 80-20 train test split.
    - Train size: 44,000
    - Test set: 11,000 documents
- I selected subreddits that are large, popular, and active that express different possible topics for conversation, while still being broad topics.

Subreddits selected:

r/WritingPrompts, r/history, r/Jokes, r/Fitness, r/legaladvice, r/Music, r/space, r/food, r/gaming, r/wallstreetbets, r/politics

I did not want to have overlapping subreddits like r/bento, r/greasy_food , r/fried_food, r/today_I_ate, r/breakfast, r/dinner, and so on.

Document Encoding Methods

Using Contextualized Embeddings

- I will be using 4 document encoding methods.
- Baseline:
  - BERT
  - Sentence-BERT
- Experimental Models:
  - Reddit-BERT
    - Add Subreddit specific terms to the BERT vocabulary and tokenizer.
    - Fine-tune weights on a sequence classification task.
  - Reddit-S-BERT
    - Use a fine-tuned Reddit-BERT model as input to S-BERT.
    - Continue training on manually labeled data set for S-BERT.

BERT

Encoding Documents with Naïve-BERT

- Each document is a whole comment. All punctuation is stripped from the comment.
- Using ‘bert-base-uncased’ tokenizer and model. The tokenizer is zero padded at 500 tokens. The BERT tokenizer as previously discussed by other projects does its best to breakdown vocabulary it does not know.
- This model is to be used a baseline to represent how well pre-trained contextualized embeddings can represent Reddit data with a model straight out of the box.

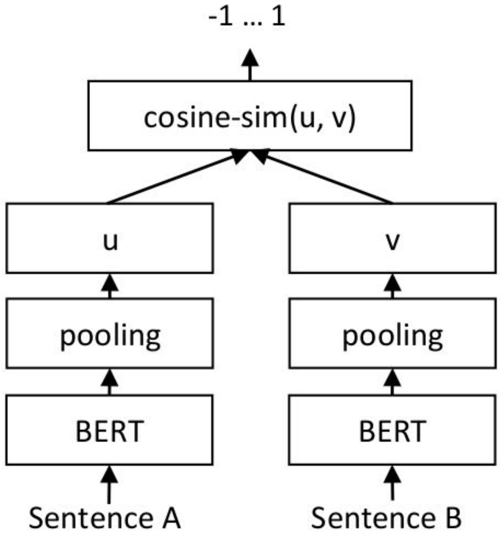
- ’bert-base-uncased’ consists of 12 hidden layers each with 768 features.
- For a document embedding, since we are passing the entire document encoded with BERT-ids we can sue the last hidden layer as a document embedding.
- However we can use different variations of hidden layers to represent the document embedding.
    - Last hidden layer, first hidden layer, sums of different hidden layers.
    - In the end, concatenating the last 4 hidden layers to create an embedding with 3072 features was the best representation for comment documents.

Sentence-BERT

Encoding Documents with S-BERT

- “Sentence-BERT (SBERT), a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.”
- S-BERT is evaluated on STS tasks and transfer learning tasks.
- I am using ‘roberta-large-nli-stsb-mean-tokens’ which produces embeddings with 1024 dimensions. I picked this model because it has the best performance of co-sine similarity for similar sentences.
- This model is trained with CosineSimilarityLoss.

- For each comment, I clean the comment normally except for removing punctuation.
- For each document/comment, I separate them by sentences using NLTK’s punkt’s sentence tokenizer. I encode each sentence for a document individually.
- To create a document embedding I average all sentence embeddings for each document.



Reddit-BERT

Fine Tuning BERT

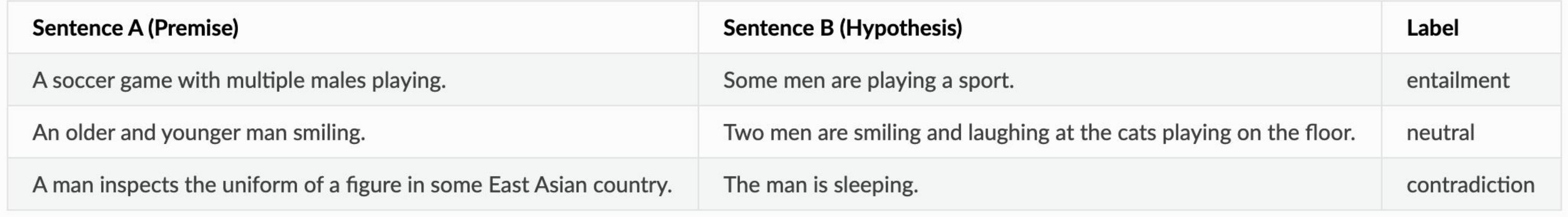
- To fine tune BERT to extend domain knowledge like BERT I followed a few papers that wanted to extend domain specific knowledge like reading medical papers.
- I want to take the ‘bert-base-uncased’ tokenizer and add new vocabulary. This model has roughly 980 unused tokens that are able to replace with out compromising the existing vocab. I also changed max length to 200 tokens per document (which encapsulates 97% of comments).
- I went to each class, using TF-IDF calculated the the top 2,000 terms by TF-IDF that at least show up 60 times in a class. I made sure the existing vocab didn’t already contain the new vocab. In total, added around 960 new terms.
- I then added an additional layer to the ‘bert-base-uncased’ model for sequence classification and continued training the model on 15,000 to 45,000 more documents/comments per class.

- **r/music:** audioslave, grunge, metallica, soundgarden, spotify, synthpop,
- **r/wsb:** bearish, bezos, brokerage, bruh, drizzle, payout, reimbursement, stonks, tendies, robinhood, aapl
- **r/gaming:** bethesda, cutscene, ftl, fortnite, gta, n64, overwatch, pve, pvp, quests, rdr2, remaster, tf2,
- **r/politics:** biden, bipartisan, boomer, buttigieg, dems, potus,
- **r/fitness:** caffeine, calisthenics, overtraining, pullup, pulldown, underweight,
- **r/food:** caramel, carrots, celery, cilantro, broccoli, mcdonalds, meats, overcooked, pickle, sushi, taco,
- **r/space:** spacex, spaceflight, supermassive, supernova,
- **r/legaladvice:** suing
- **/writingprompts:** acronyms, abbreviations

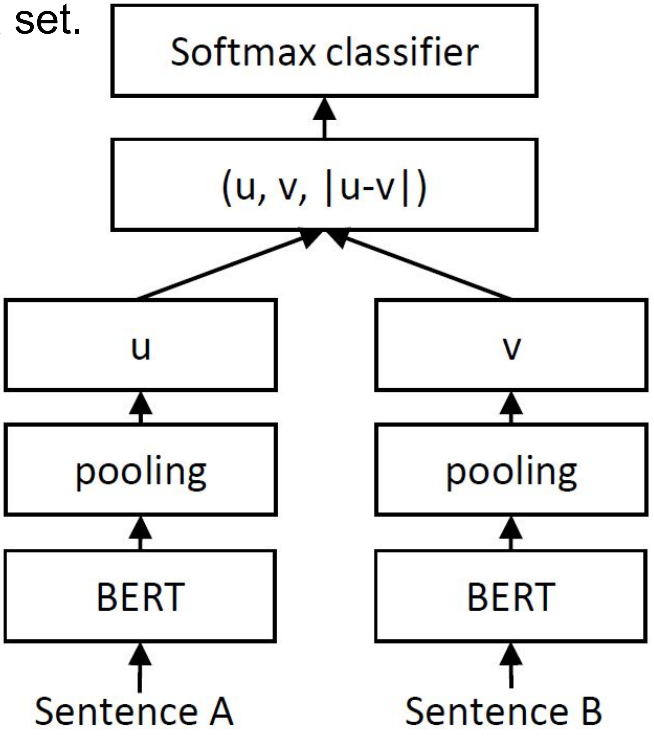
Reddit-S-BERT

Fine Tuning S-BERT using Reddit-BERT

- I am using the newly created 'Reddit-BERT' from the previous slide as a tokenizer and contextualized embedding input into S-BERT.
- I then manually labeled 500 sentence pairs on a scale from 0 as contradiction, 1 as neutral, and 2 as entailment.
- I then standardized this to be between 0-1 to match cosine similarity scores.
- I then fine tuned S-BERT via additional layer seen on the right.
- The ideal goal of this model is to create a gold data set than is manually labeled that can create a Bi-Encoder.
- Then you use this bi-encoder to create a silver data set on the rest of your training data to create more training data for S-BERT. However, my manually labeled dataset overfitted on certain labels for the silver set. In the end, I only used the Gold data set.



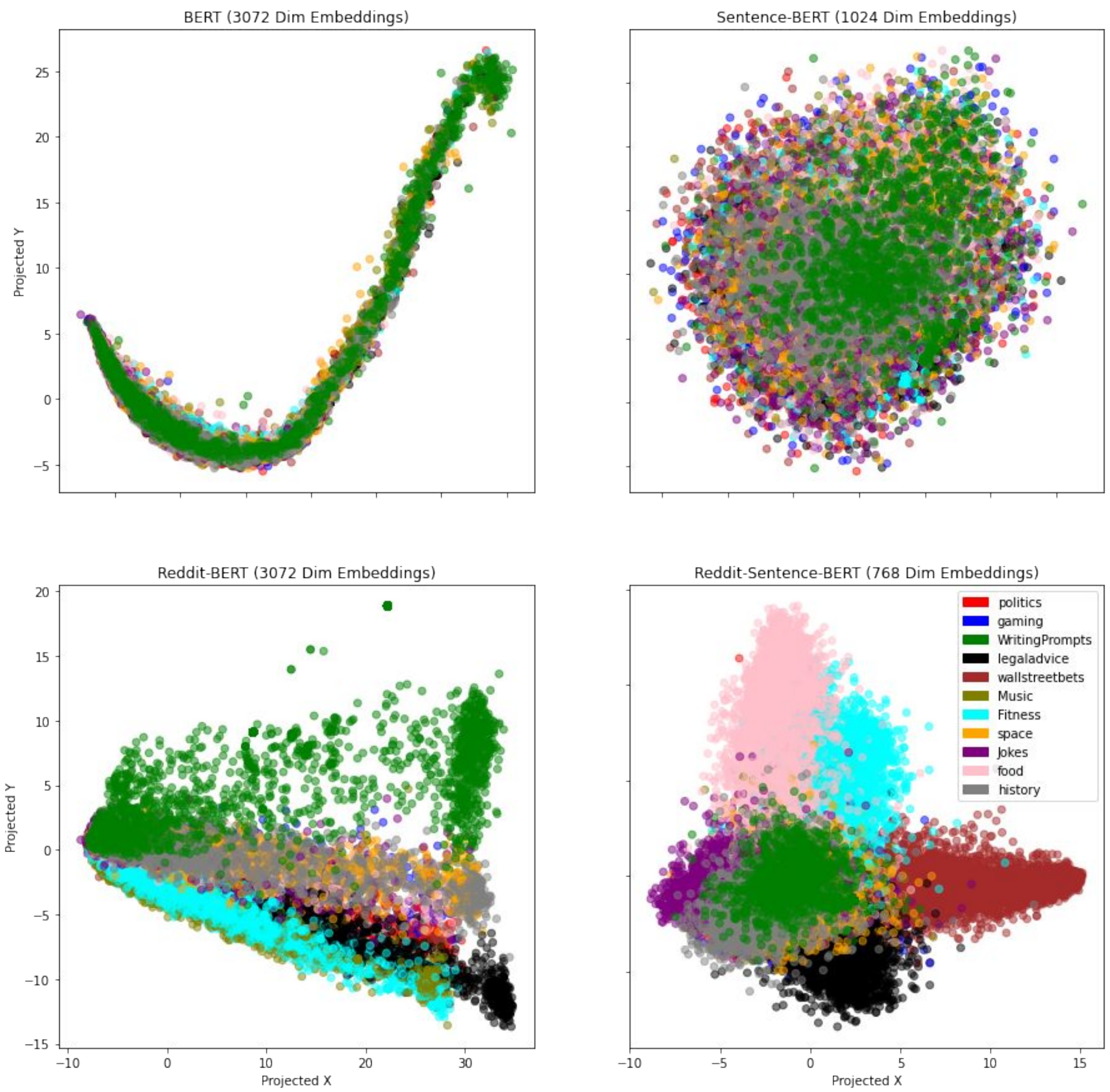
Sentence A (Premise)	Sentence B (Hypothesis)	Label
A soccer game with multiple males playing.	Some men are playing a sport.	entailment
An older and younger man smiling.	Two men are smiling and laughing at the cats playing on the floor.	neutral
A man inspects the uniform of a figure in some East Asian country.	The man is sleeping.	contradiction



Evaluating By Eye

Using PCA

- A great indication of quality embeddings representing your documents is to use PCA to make a quick inferential evaluation.
- I used PCA to do dimensionality reduction on each set of document embeddings.
- BERT: You can clearly see that ‘bert-base-uncased’ is struggling with the additional unseen vocab. Moreover, due to the naïve approach, the tokenization was not optimized for niche domain vocab.
- S-BERT: It is relatively better than BERT, but not great. Classes are clustered but there is no clear separation.
- Reddit-BERT: This looks great! There is clear separation, but there is room for improvement.
- Reddit-Sentence-BERT: Takes the cake between these four models, however, it can be better.



Evaluating Empirically

Classifier KNN

- The idea of the project is to measure the quality of embeddings for semantic search. The proposed method to evaluate these embeddings is to try to classify the original documents solely based on cosine similarity.
- For the KNN, I standardized cosine similarity by calculating 1 – cosine similarity score.

- I parameter tuned on the BERT model to get the best performance I could with k=25. I then kept this parameter for all four models to standardized the experiment so only the quality of the embeddings are evaluated.

$$1 - \frac{u \cdot v}{||u||_2 ||v||_2}$$

	Accuracy	Roc Auc Score	Precision Score	Recall Score	F1 Score
BERT	0.3615	0.6488	0.4021	0.3615	0.3688
S-BERT	0.5798	0.7689	0.6055	0.5798	0.5771
Reddit-BERT	0.7426	0.8585	0.7758	0.7426	0.7513
Reddit-S-BERT	0.8244	0.9034	0.8459	0.8244	0.8282

Conclusion and Future Work

Evaluating Document Embeddings

$$\text{angular distance} = \frac{\cos^{-1}(\text{cosine similarity})}{\pi}$$

$$\text{angular similarity} = 1 - \text{angular distance}$$

Conclusions

- It is clear that fine tuning the baseline models improved models tremendously.
- PCA is a great and a fast way to identify if your embeddings are performing well.
- A KNN is great at evaluating the quality of service or tool that uses cosine similarity to semantically search for similar documents.
- Fine tuning the 'bert-base-uncased' model was the most beneficial jump in performance for the amount of work. It takes more training time, yet, will produce better embeddings.
- Reddit-S-BERT was the cherry on top that really allowed the fine tuned Reddit-BERT excel on creating sentence embeddings.

Future Work

- Develop Q-Vec for sentence/document embeddings that will enable a score that correlate to performance across different embedding spaces.
  - For example, BERT uncased has hidden layer sized of 768, we created an embedding space was 3096, S-BERT was 1024. It is hard to directly compare them.
- Possibly transform embedding spaces onto each other for a direct comparison?
- Cleaner data. Reddit data is messy.
- More labeled data for S-BERT to train a silver dataset. Is there a better way to encode S-BERT?
- Possibly using angular distance/angular similarity for the KNN. For better standardization of emebddings.

1870

STEVENS

INSTITUTE of TECHNOLOGY

THE INNOVATION UNIVERSITY®

stevens.edu

Thank you for listening and the great semester!

Any questions?