CS 522
Advanced Data Mining

Team Meeting

April 16, 2020

Projects

- Bert for Geo Mapping of Entities
- Covid-19 Data, Bert pre-training
- Bert for Classification, native language of the writer

Bert for Geo Mapping of Entities

- Goal of the project
  - Wikiwords are entities that are detected in the text
    - E.g. scientific articles discuss some geographic locations or institutions
  - Discover topics or clusters of Wikiwords
  - Map the geo distribution of the topics
    - Extract the geolocations of the Wikiwords in a particular topic
    - Map using visualization tool
- Required output
  - For each document, Wikiword pair show the topics/cluster assignment

Space science is closely associated to NASA, California Institute of Technology, the Royal Astronomical Society, and has a geographical footprint that stretches in a straight line from Hawaii through California and New Mexico, through the East Coast, to several locations in Europe

Africa does is little involved in the Study Abroad topic above, but it catalyzes a topic on native culture and human history.

Currently, the visual interfaces take a data structure as below:

DocumentName1.txt,WordSurfaceForm1,Dimension117\n
DocumentName1.txt,WordSurfaceForm2,Dimension5\n
DocumentName1.txt,WordSurfaceForm1,Dimension1\n
DocumentName1.txt,WordSurfaceForm3,Dimension3\n
DocumentName2.txt,WordSurfaceForm2,Dimension249\n

- Documents are sorted alphabetically
- Columns are separated by commas
- No commas elsewhere
- No quote marks
- Dimension numbers start with 0 (and end with 299)

This will look like this in the real example:

12344Reddit_body_plural_humanities_123123.txt,Santa,117
12344Reddit_body_plural_humanities_123123.txt,Barbara,115
12344Reddit_body_plural_humanities_123123.txt,officials,118
12344Reddit_body_plural_humanities_123123.txt,say,123
12344Reddit_body_plural_humanities_123123.txt,everyone,115
12344Reddit_body_plural_humanities_123123.txt,is,0
12344Reddit_body_plural_humanities_123123.txt,safe,123
12344Reddit_body_plural_humanities_123124.txt,Chicago,98
12344Reddit_body_plural_humanities_123124.txt,officials,118
12344Reddit_body_plural_humanities_123124.txt,say,123
12344Reddit_body_plural_humanities_123124.txt,everyone,299
12344Reddit_body_plural_humanities_123124.txt,is,90
12344Reddit_body_plural_humanities_123124.txt,safe,7

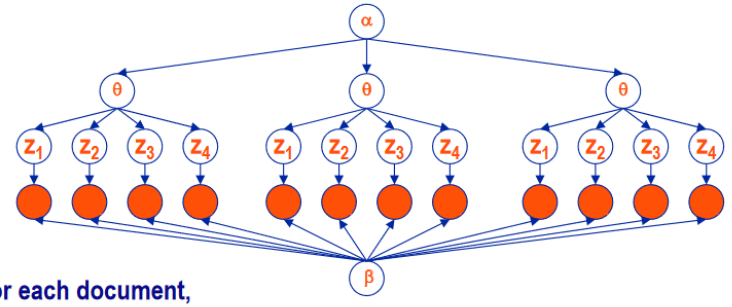
If all fine-tuned embedding dimensions are recorded, the example could look as follows:

DocumentName1.txt,WordSurfaceForm1,WeigthDim0,WeigthDim1,WeigthDim2,WeigthDim3,…,WeigthDim299\n
DocumentName1.txt,WordSurfaceForm2,WeigthDim0,WeigthDim1,WeigthDim2,WeigthDim3,…,WeigthDim299\n
DocumentName1.txt,WordSurfaceForm1,WeigthDim0,WeigthDim1,WeigthDim2,WeigthDim3,…,WeigthDim299\n
DocumentName1.txt,WordSurfaceForm3,WeigthDim0,WeigthDim1,WeigthDim2,WeigthDim3,…,WeigthDim299\n
DocumentName2.txt,WordSurfaceForm2,WeigthDim0,WeigthDim1,WeigthDim2,WeigthDim3,…,WeigthDim299\n

How is this done in LDA

- LDA generative model assumes that for each word position in the document, a topic is selected first, that the actual word is selected from that topic based on the topic's word distribution.
- This means that there is only one topic for each document/word pair.

The LDA Model



- **For each document,**
  - Choose the topic distribution $\theta \sim \text{Dirichlet}(\alpha)$
  - For each of the N words w_n:
    - Choose a topic $z \sim \text{Multinomial}(\theta)$
    - Then choose a word $w_n \sim \text{Multinomial}(\beta_z)$
      - Where each topic has a different parameter vector β for the words

How is it done with BERT

- Bert is not a topic model per se
- If we analyze the similarities between the embeddings for word in Bert, we are likely to see “semantic fields”, i.e. groups of semantically related words to have higher similarities.
- Since Bert is not a generative topic model, we don’t have a notion of topics per document the same way we have them in LDA.

How is it done with BERT

- 2 Approaches to approximate the topic assignment for each document/word pair with Bert
  - For each occurrence of a Wikiword in a document (document/word pair) get the Bert embedding for the Wikiword. Use the dimension in the Bert embedding vector that has the highest absolute value as the “topic”.
  - Extract Bert embedding vectors for the Wikiwords, cluster them. Use the cluster id as the topic assignment for each document/Wikiword pair.

WikiWords and Bert

- Issue 1
  - Bert has a fixed vocabulary
    - Wiki_iit_asif is NOT in Bert's vocabulary
- Solution 1
  - Bert has place holders for unknown words. Can use those to represent the wikified words
- Issue 2
  - Bert tokenizers will split up multiword phrases, it uses underscores as word separators
    - Wiki__CS_Department__xhgt will be split up into >=4 tokens
- Solution 2
  - Replace underscores with something the tokenizer does not recognize as word boundary
- Issue 3
  - Over 78K Wikiwords

Observations from Data Analysis

- Bert might have some of the Wikiwords in its vocabulary already
  - (without the Wikifier tags. E.g. IIT is in the vocabulary, and its Wikiword is Wiki__IIT__sdfg)
- Not all Wikiwords in the Reddit data are frequent
- 3 Approaches
  - Focus only on the frequent words, try to add new words into Bert, pretrain Bert so that is gains knowledge about the new words
  - Focus on the words that are already in Bert, use the embeddings from the original Bert
  - Focus on the words that are already in Bert. Pretrain the Bert with the Reddit data. Use the updated embeddings.

How to evaluate the quality of Bert Embeddings

- Compute cosine similarities between pairs of tokens and manually analyze if they make sense
- Contextual embeddings
  - Use phrases, or token+2,3 words context for embeddings
  - Cosine similarity

Next Steps

- For each occurrence of a Wikiword in a document (document/word pair) get the Bert embedding for the Wikiword. Use the dimension in the Bert embedding vector that has the highest absolute value as the “topic”.
- Extract Bert embedding vectors for the Wikiwords, cluster them. Use the cluster id as the topic assignment for each document/Wikiword pair.

Bert for Geo Mapping of Entities: Teams

- Team 1
  - Analyzed ways to add Wikifieds words (as they are in Reddit, with the Wikifier tags to Bert)
  - Pretrained
  - Extracted Embeddings
- Team 2
  - Used the Wikified words that are already in Bert.
  - Removed the Wikifier tags, used the actual token.
  - Analyzed Bert similarities
  - Analyzed Bert similarities in context
- Team 3
  - Analyzed ways to add Wikifieds words (as they are in Reddit, with the Wikifier tags to Bert)
  - Pretrained
  - Tried to extract Embeddings
- Team 4
  - Analyzed ways to add Wikifieds words (as they are in Reddit, with the Wikifier tags to Bert)
  - Pretrained Bert with Reddit data

- Team 1

Approach:

Update BERT vocabulary and use the concept of embedding on the data provided to us and the pre-trained data.

Corpus:

- 118059 comments from Reddit.
- # Distinct Wiki words – 78,597.

BERT Vocabulary:

- Uncased Base
- Unused Tokens – 993

Problem:

Wiki Words > # Unused Tokens

Solution:

- We use the most frequent Wiki Words.
- 4500 Wiki Words – 983 Unused Tokens.

Team - Agnes Gaspard, Kanyakumari Kashyap, Kavya Ravella , Shweta Metkar

Pre- Training:

- Pre-train the top layer of the model.
- Environment setup in Colab.
- Wiki words are frequent all over the file.
- Example - Wiki__University__eilqf repeated 2574 times
- We will collect all sentences with this word.
- Pre-train the model to learn the context of the word in each sentence.

Embedding:

- Used bert on sentences without wikification.
- Got vectors of 12*768 for each token and concatenate last 4 hidden layers.
- So each token has vector of dimension 3072.

Team - Agnes Gaspard, Kanyakumari Kashyap, Kavya Ravella , Shweta Metkar

TEAM 2 - ENV PROJECT

Team Members :

Akash Tanwani(A20448831)
Disha Sharma (A20443440)
Ninad Parikh (A20427382)
Sathyaveer Karmarkar

ILLINOIS INSTITUTE OF TECHNOLOGY

ILLINOIS INSTITUTE OF TECHNOLOGY

Tasks achieved and steps followed

Step 1: Preprocessing

We created a text file ‘raw_data’ which is a combination of all the text files provided to us in the data set. This combined text file is used in all further computations.

Step 2 : Wiki words extractor

We extracted the wikified words in the data set and calculated the frequency of each word using the .count function of python. This data was combined into a csv file with the words arranged in the descending order of their frequencies.

Tasks achieved and steps followed

Step 3 : Separate the actual words from “Wiki_” and “_xxxx”

Regular Expression was used to remove “wiki” and “__” from the wikified words so that they can be used in computation.

Step 4 : Comparing the actual words from the words in the vocabulary

We tested the similarity for words that could be grouped in a category.

We identified the maximum and minimum cosine similarity value for the words belonging to the same category. The value would tell us Bert embeddings represent the words belonging to the same category.

We also checked using the same words in different sentences and calculated the vector value for the word. This was to check for the vector values changing for words used in different context.

Results and Observations

In [38]:
```
vectors = bc.encode(['harvard','yale'])
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

```
0.94169194
```

In [39]:
```
vectors = bc.encode(['i studied at harvard','i graduated from yale'])
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

```
0.9279686
```

In [40]:
```
vectors = bc.encode(['iit','uic'])
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

```
0.6850818
```

In [44]:
```
vectors = bc.encode(['i studied at illinois institute of technology','i graduated from university of illinois chicag
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

```
0.8857086
```

In [46]:
```
vectors = bc.encode(['he graduated from stanford university','stanford university is located in california'])
print(cosine_sim_vectors(vectors[0], vectors[1]))
```

```
0.6295593
```

- Team 3

Extracting the wiki words and their analysis

- Removing wikifier information, finding the frequency, and comparing them with BERT vocab
- 1,321,967 wiki words
- Top 3 wiki_words - Liberal arts - 43,117, Humanities - 42,139, Academic term - 19,185
- After removing duplicates - 3718 single words,14039 matching multiple words with BERT Vocab
- Top 3 NER count of the Single Words – Geographical locations 323, PERSON 177, Organisation 119
- Top 3 NER count of multiple words - PERSON 1168 , 670 Geographical locations, Organisation 404

Word embeddings

- Getting word embeddings of wiki_word matching with bert vocab
- Using sentence encoding service called bert-as-service to extract embeddings
- Getting the vectors values
- Finding cosine similarity between the words

CS 522 Group 4
BERT

Eadhunath Venghatesan
Hiranmayi Raju
Naunidh Singh
Shivam Kulkarni

Agenda

- Data statistics for wikified words
- Approaches tried
- The setup for the experiments
- Results
- Conclusion so far

Data Statistics

- Total number of words in input data (entire Reddit Corpus): ~ **55 Million**
- Total number of wikified words in the corpus: ~ **1.3 Million**
- Number of unique wikified words in the corpus: ~ **79K**
- Wikified words are **2.41%** of total input words
- Unique Wikified words are **23.13%** of total unique input words
- Further, we analyzed the frequency of all wikified words.

Data Statistics

Count (distinct wikified words)	78597
Mean	16.82
Standard Deviation	276.09
Minimum	1
25% (1st quartile)	1
50% (2nd quartile)	2
75% (3rd quartile)	6
Maximum	43117

Approaches Tried

- Pre-training BERT from scratch
- Using Google’s architecture to train BERT on Reddit dataset
  - Without tampering Google’s default vocabulary
  - Tampering Google’s default vocabulary

The Setup

- BERT base model used : cased_L-12_H-768_A-12
- Number of wikified words added to vocab.txt : 10
- Number of steps : 30k
- Input size : 20k text files

Results

- Using Google’s checkpoint and Google’s vocabulary on our data

  Loss for final step: 0.7068774

- Using Google’s checkpoint but altering Google’s vocabulary on our data

  Loss for final step: 0.7979901

Conclusion So Far

- We need a better metric to compare the models
- Adding more words to vocabulary may yield better results
- Increasing number of steps may give us different results

- Team 5

Bert Team 5

•••

Luke Logan, Blake Ehrenbeck, Swathi Sudha Suresha, Nicholas Saveas

Preliminaries

- We downloaded the COVID-19 dataset of research papers
  - ~6GB and ~51K articles
- We installed Google Bert and SciBert
  - SciBert is trained on Scientific Research Papers
  - We are training specifically on COVID-19 Research Papers
- We installed and configured SentencePiece to generate vocab
  - It outputs a vocabulary in WordPiece format

Results

- We created a vocab of 32K word pieces
  - It is case-sensitive; we're going to change that
  - Overall, looked correct
  - Coronavirus, coronavirus
- We pretrained a Bert model with this Vocab
  - 10% of data used to train
  - 100 training steps
- We used SciBert to predict masked words on a sentence
  - Obtaining large-scale annotated [MASK] for NLP tasks in the scientific domain is challenging and expensive.
  - Model output: corpora
  - True value: data

- Team 6

Approach and Data Statistics

Problem Statement : Every author who writes in English might not be the native speaker of English. Native language will impact the English writing style of author, here we are trying to predict the native language of the author based on the data we have about the author's writing style.

Approach:

1.Cleaning the data.

2.Processing the data. Convert the .csv files to .txt files.

3. Obtaining the pre-trained model. And using BERT as a classifier for predicting the native language.

4.Result Evaluation

Datasets: Train dataset consists of 6000 data points, evaluation dataset consists of 2000 data points and test dataset consists of 2000 data points. With total of 10 languages in each set and 600 data points for each language in train data and 200 data points for each language in test and validation data sets.

Results

- Dataset which we obtained is in .csv format. After cleaning the data we pre-processed the data from .csv to .txt format eliminating the native language column to train the Bert Classifier.
- We have obtained the Bert pretrained model and we are in process of training the classifier using BERT neural network as we processed.