CS 522
Advanced Data Mining

Project Presentations

April 30, 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

DC.

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.

CS-522 Project - Group 4

BERT for Mapping Public Awareness

Eadhunath Venghatesan
Hiranmayi Raju
Naunidh Singh
Shivam Kulkarni

30th April 2020

Introduction to the Problem

- Create a model that is able find correlations and contexts within a large text corpus.
- Produce output to map it on the GeoD visualization software.

- A model exists which uses LDA to achieve this task of Topic Modelling.
- We are to harness the power of BERT to replace LDA from this task.
- Represent Wikified Words in contextual space and find underlying correlations.

Data Summary

- Total number of words in input data (entire Reddit Corpus): ~ **55 Million**
- Total number of wikified words in the corpus: ~ **1.3 Million**

Data Preprocessing

- Queried top 100 most frequently occuring Wiki words.
- Reduced Dataset to only those files that contain these frequent words.
- Handling Wiki words :
  - If the wiki-word is frequent - wiki_facebook_abc => wiki*facebook*abc
  - Non-frequent wiki-words - wiki__game_of_thrones__xyz => game of thrones
- Reconstructed the Corpus with the above rules for make the Pre-training data.

Approaches Tried

- Training BERT from scratch
  - Tried and discarded as lot of the information was lost and model too specific.
- Training BERT using Google's vocab and initial weights
  - Adopted and used to complete this project
  - BERT config used - **cased_L-12_H-768_A-12**

Quick Summary of Wikified Words

# 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 Frequency	43117

Experiments and Results

Approaches Tried

- Pre-train on the default Google Vocabulary
  - Tried and discarded as it could not learn wiki-words.
- Pre-train after adding the frequent words to the Google Vocabulary
  - Adopted and implemented

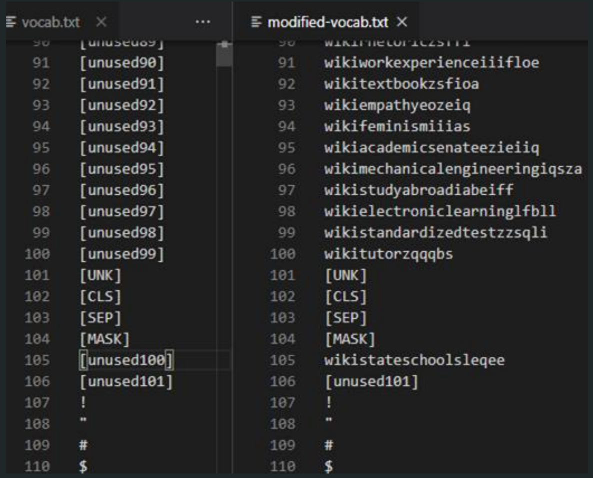
Pre-training

- Use the sampled corpus and modified vocab file to make the pre-training data as a *.tfrecord* file
- We trained the model on multiple different values of batch_size, learning_rate and training_steps.
- Best model while using *batch_size* of 128, *learning_rate* of 1e-4, *training_steps* of 50k and *seq_length* of 128.
- Made multiple checkpoints during the training processes (every 2.5k steps) to check for overfitting and comparison.
- Coding Environment
  - Python and Tensorflow
  - Google Colab (TPU)
  - Google Cloud Storage

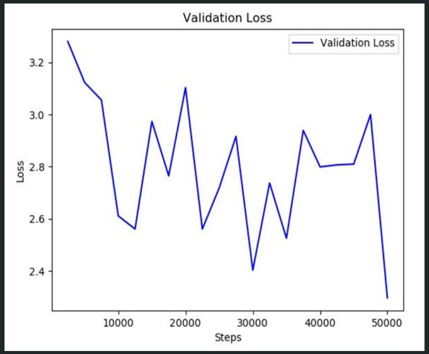
Results

- Final Loss Value : 1.25 (does not say much)
- We access with the results from the next steps.

Modifying vocab file



```
vocab.txt                  modified-vocab.txt
[unused89]
[unused90]                 wikiworkexperienceiiifloe
[unused91]                 wikitextbookzsfioa
[unused92]                 wikiempathyeozeiq
[unused93]                 wikifeminismiiias
[unused94]                 wikiacademicsenateezieiiq
[unused95]                 wikimechanicalengineeringiqsza
[unused96]                 wikistudyabroadiabeiff
[unused97]                 wikielectroniclearninglfbll
[unused98]                 wikistandardizedtestzzsqli
[unused99]                 wikitutorzqqqbs
[UNK]                      [UNK]
[CLS]                      [CLS]
[SEP]                      [SEP]
[MASK]                     [MASK]
[unused100]                wikistateschoolsleqee
[unused101]                [unused101]
!                          !
"                          "
#                          #
$                          $
```



Experiments and Results [contd.]

Extracting Word Embeddings

- We can make use of the checkpoint files (.ckpt) to get the weights and hence the word embeddings.
- We chose certain files which we want to analyze
- We added special tokens like [CLS] & [SEP]
- BERT tokenizer converts words into different tokens
- BERT tokenizer tries to map the tokens found in sentences to tokens in vocab.txt
- We then put the model in evaluation mode and fed the data as input to the model
- Each of the 12 layers in the model store word embeddings
- We extracted ours by summing the vectors of the last 4 layers.
  - Output : word representations as a 768 length vector.

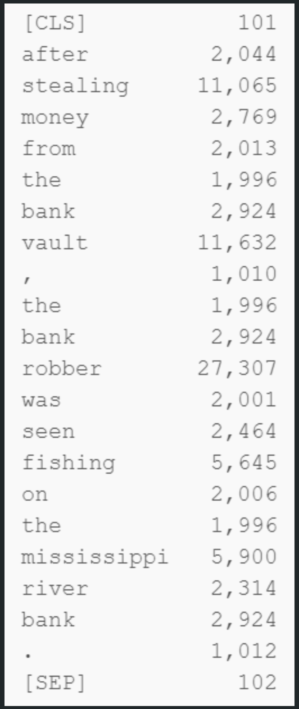
Examples

```
Input Text = "After stealing money from the bank vault, the bank robber was
seen fishing on the Mississippi river bank."

      Vector similarity for  *similar* bank meanings:  0.91
      Vector similarity for *different* bank meanings:  0.71

Vector Similarity between wiki words:
      Vector similarity between Facebook and Twitter:  0.82
      Vector similarity between Facebook and Biology:  0.62
```

Tokens mapping



[CLS]	101
after	2,044
stealing	11,065
money	2,769
from	2,013
the	1,996
bank	2,924
vault	11,632
,	1,010
the	1,996
bank	2,924
robber	27,307
was	2,001
seen	2,464
fishing	5,645
on	2,006
the	1,996
mississippi	5,900
river	2,314
bank	2,924
.	1,012
[SEP]	102

Experiments and Results [contd.]

K-Means Clustering

- To show the clustering of all similar words together, we use K-Means clustering
- We use cosine similarity as the metric to perform K-Means clustering instead of the usual Euclidean Distance
- In the clusters, we are all the wiki-words.

Example Clustering:

Cluster 0:
['wikiwikipediasofelef', 'wikimarxismiqofose', 'wikifascismiiosf', 'wikicapitalismsfib', 'Wiki__Socialism__zbafl', 'Wiki__Democracy__lqsq', 'Wiki__Far_right__zzizzo', 'Wiki__Nationalism__zilfa', 'Wiki__Propaganda__zezoe', 'Wiki__Representative_democracy__szeia', 'Wiki__Direct_democracy__qsaib']

Cluster 1:
['wikiideologylsfbq', 'Wiki__Algebra__ieqa', 'Wiki__Physical_fitness__fezqab', 'Wiki__Educational_toy__iboiblb', 'wikiinterdisciplinarityiszoi', 'wikisocialscienceszblai', 'Wiki__School_choice__zllbi', 'Wiki__Charter_school__eeibqe', 'Wiki__Faculty_development__esaqsqq', 'Wiki__Single_parent__zzbioq', 'Wiki__Rights_issue__ibieeif', 'Wiki__Wage_slavery__selob', 'Wiki__Consumer_organization__qeofzo', 'Wiki__Sales__issbqa', 'Wiki__Index_fund__ioboiz', 'Wiki__Brown_people__lsiaaaa', 'Wiki__Medical_journal__sfiqqz', 'wikitenureeifqsi', 'Wiki__Peer_review__zfiib', ''Wiki__State_Farm_Insurance__aioasq’]

Cluster 2:
['Wiki__Photo_shoot__ioeqbbal', 'wikibarackobamasefebb', 'Wiki__Sarcasm__zqsaql', 'wikirussiazseqi', 'wikilolzaaelq', 'wikiblogeebfs', 'Wiki__Troll__Internet___ifsqf', 'Wiki__Algebra__ieqa', 'wikiunemploymenteilfi', 'Wiki__Real_point__aoaaqeq', 'wikiredditeazqoos', 'wikiyoutubeeszflbb', 'wikituitionzfqezo', 'wikischolarshipiaoeel', 'wikistudentloanfbfqqo', 'wikirhetoriczsffl', 'Wiki__Surgery__fssqq', 'wikiacademictermsfssll', 'wikiundergraduateeducationzilzaq', 'wikiliberalartscollegeiabiz', 'wikiworkingtimessziba', 'wikisocialmediasaqllfz']

Conclusion

- The approach chosen, on the whole, seems pretty effective as seen from the similarities and clusterings.
- Major learning was the implementation of BERT including Pre-training on new datasets and also handling new domain-specific vocabulary.
- We were able to make use of wiki-words and consolidate their contextual meanings to the model.
- The dataset used was not the easiest to work with and required a good amount of analysis and processing.
- Pre-training BERT with this dataset, with the new vocabulary, was a challenging task.
    - Especially since the evaluation, in this case, required manual supervision.
    - BERT was made for language modelling, however we are using it for Topic Modelling.
- Based on the analysis, we feel that we have converged close to the actual solution but there is always a scope for improvement.

BERT: NATIVE LANGUAGE DETECTION

Abhishek Krishna Vandadi(A20426749)
Venkata Akshith Reddy Kasireddy(A20455209)
Barani Kumar (A20434813)
Deepa Bada (A20450368)

AIM: NATIVE LANGUAGE DETECTION

There are many authors who write in English and have different native language. There is always an influence of native language while speaking or writing English. We try to find out the native language of non-english writers to understand how much does a native language influence the English writing of a person.

SUMMARY OF THE DATA

- There are two datasets: Training set(6000 data samples) and Test set(2000 data samples). Each of these two datasets contains 10 non-english language languages. Exactly 10% of the data items belong to each language. There are 2 columns in common for both the sets: Native Language and English text written by author. The training set has one additional column: Language ID.
- Preprocessing: We remove the native language column from both datasets and converted all the text to lowercase for better classification. We use the language ID column and text column to train our model. We will then use the test dataset to test it.

EXPERIMENTS AND RESULTS - I

- We fine tuned the **BERT-Base Uncased model**(Pretrained) using pytorch.
- We worked on changing 2 parameters namely,
    - **epochs**
    - **learning rate,**
- We have tried about **35 combinations** of number of epochs and learning rate **to obtain optimal results.**
- Most Optimal Parameter values
    - **Learning Rate : 2e-5**
    - **Epochs : 4**
- **Anything over these parameter values** leads to a very significant training loss and overfitting of the model .
- **Anything lower than this parameter values** does not provide an efficient output. The possible reason could be related to the size of dataset that we have.

Experiments and results -II

1) **Overall accuracy of BERT fine tuned model=17%**

   Best classification= Arabic

   Worst classification=Vietnamese

1) **Overall accuracy of Logistic Regression Model developed using BERT vectors=46%**

   Best classification=Spanish

   Worst classification= Cantonese

Optimal Results from Fine-Tuned model				**Optimal Results from logistic Regression model**			
Native Language	**Precision**	**Recall**	**F1 Score**	**Native Language**	**Precision**	**Recall**	**F1 score**
Japanese	0.16	0.19	0.18	Japanese	0.44	0.50	0.47
Korean	0.22	0.13	0.17	Korean	0.45	0.39	0.42
Vietnamese	0.11	0.03	0.05	Vietnamese	0.47	0.40	0.43
Mandarin	0.13	0.21	0.16	Mandarin	0.34	0.32	0.33
Russian	0.09	0.07	0.08	Russian	0.50	0.56	0.53
Thai	0.18	0.26	0.21	Thai	0.50	0.51	0.50
Spanish	0.05	0.09	0.06	Spanish	0.62	0.57	0.60
Cantonese	0.13	0.12	0.16	Cantonese	0.31	0.32	0.32
Polish	0.10	0.14	0.12	Polish	0.49	0.49	0.49
Arabic	0.40	0.41	0.41	Arabic	0.47	0.51	0.49

Experiments and results III: Confusion Matrix

We have obtained a 10x10 confusion matrix.

0-9 represent the languages in the same order as the results.

As we can notice, the misclassification is very high.

	Jap	Kor	Viet	Man	Rus	Thai	Spa	Can	Pol	Ara
Jap	108	10	11	15	203	139	15	17	38	44
Kor	12	101	65	56	23	65	39	123	78	38
Viet	25	54	31	61	45	167	69	89	52	7
Man	27	68	166	97	89	98	14	13	20	8
Rus	98	19	32	41	47	129	9	98	89	79
Thai	14	51	40	60	21	126	78	44	61	105
Spa	88	106	13	35	67	32	35	189	58	12
Can	34	77	91	50	11	9	134	97	54	43
Pol	22	63	122	143	8	17	76	10	73	66
Ara	10	54	29	31	103	20	68	31	7	247

Conclusion

- Working with BERT was indeed very fruitful and we obtained some interesting results. The model is comparatively doing better with Asian languages(Arabic, Korean,Mandarin, Cantonese, Japanese and Thai) and able to understand that the author is from non-english background. The possible reasons could be that the writing pattern, the words used, etc. by the authors which might have helped BERT predict their native language.
- On the other hand, the model is very poor with the non-asian languages such as Russian, Vietnamese and Spanish. I believe that most of these authors have written the text in such a way that there might have been any evidence of their native language and it became difficult for the model to predict. Coming to another non-asian language polish, the F1 score is not close to asian languages but also not nearby Russian, Spanish and Vietnamese.
- BERT is very interesting and easy to use. The logistic regression model developed from the BERT vectors seems to be more reliable and promising than BERT fine tuned model in our project. However, even the logistic regression model does not result in great accuracy.
- Also, the size of the dataset should have been larger which we realised after reaching towards the end of the project. Maybe 1200 samples of each language would lead to better results than 600.
- Our future goal is to work on newer approaches using BERT which could lead to better classification for our project.

TEAM 2

BERT for Geo Mapping of Entities

Team Members :

Akash Tanwani(A20448831)
Disha Sharma (A20443440)
Ninad Parikh (A20427382)
Sathyaveer Karmarkar (A20445690)

ILLINOIS INSTITUTE OF TECHNOLOGY

ILLINOIS INSTITUTE OF TECHNOLOGY

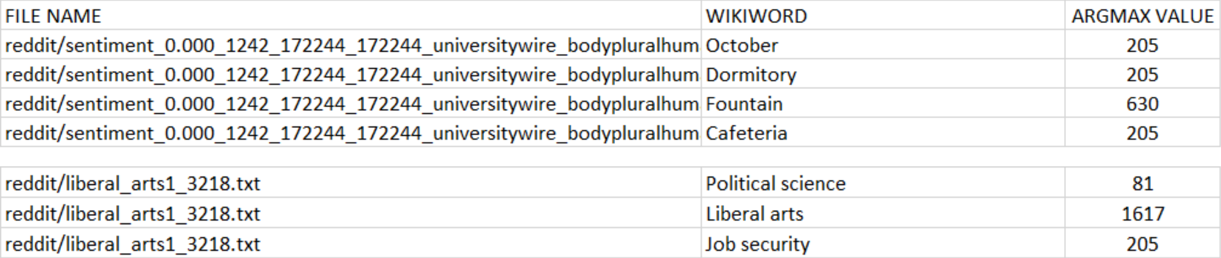
Goal of the project

- Discover topics or clusters of wikiwords

- For each document, wikiword pair shows the topics / cluster assignment

Data

- **Input data :** The dataset contains wikified data scraped from reddit on 'humanity' keyword.
- **Output data**



FILE NAME	WIKIWORD	ARGMAX VALUE
reddit/sentiment_0.000_1242_172244_172244_universitywire_bodypluralhum	October	205
reddit/sentiment_0.000_1242_172244_172244_universitywire_bodypluralhum	Dormitory	205
reddit/sentiment_0.000_1242_172244_172244_universitywire_bodypluralhum	Fountain	630
reddit/sentiment_0.000_1242_172244_172244_universitywire_bodypluralhum	Cafeteria	205
reddit/liberal_arts1_3218.txt	Political science	81
reddit/liberal_arts1_3218.txt	Liberal arts	1617
reddit/liberal_arts1_3218.txt	Job security	205

- **Approaches:**

Preprocessing	Extracting wikiwords with and without context
Pretraining	Pretraining the data from the checkpoint of Bert-base-cased
Embedding extraction	Using Bert-as-a-service to encode the input string
Clustering	Using K-Means algorithm to cluster the wiki word embeddings.

Results per train_steps

global_steps	Base	20	125	325	1000	3000	6000
loss	-	2.885396	2.885396	1.593328	1.2925639	0.52907777	0.14534494
masked_LM_accuracy	-	0.56349343	0.56349343	0.71474963	0.7514955	0.88078964	0.9623141
masked_LM_loss	-	2.5958247	2.5958247	1.510927	1.2594049	0.57911205	0.17664616
Next_sentence_accuracy	-	0.8875	0.8875	0.97125	0.9525	0.99375	1.0
Next_sentence_loss	-	0.27448446	0.27448446	0.0983968	0.1312545	0.020964736	0.0005384741

Analysis for the cosine similarity of the below embeddings.

A: “harvard”,“yale”

B: “he graduated from stanford university”,“stanford university is located in california”

C: “I studied at Illinois Institute of technology”, “I graduated from university of Illinois institute of technology”

Embeddings used	Bert-base-uncased	20	125	325	1000	3000	6000
A	0.94169194	0.93526334	0.882898	0.86279756	0.84584016	0.781684	0.7448325
B	0.6295593	0.8175078	0.7761315	0.7618442	0.6670189	0.54867387	0.5043123
C	0.8857086	0.91715294	0.827649	0.8401249	0.71199524	0.7135249	0.7182884

- Used bert as a service for finding word embeddings
- Bert Parameters:
  1. Pooling_layer= -4 -3 -2 -1
  2. Pooling_strategy=NONE

	bert-base-uncased	**bert-base-cased**
Total number of clusters	**617**	**519**
Minimum cluster size:	**10**	**10**
Maximum cluster size:	**30913**	**45918**
Most Clusters Range	10 to 90 few in hundreds and thousands	10 to 90 few in hundreds

For bert-base-uncased model:

Coherent Clusters:

Cluster id - 477; Size - 36 wikiwords

['Basic income', 'Livestock', 'Financial adviser', 'Patent attorney', 'Registered nurse', 'Estate agent', 'Independent contractor', 'Criminal record', 'Automobile salesperson', 'Public figure', 'Government contractor', 'Oil reserves', 'Tax collector', 'Arts district', 'Freedom of assembly', 'Military dictatorship', 'War Veteran', 'Media consultant', 'Veterinary technician', 'Crime writer', 'Business license', 'Venomous snake', 'Ordinary income', 'Economic entity', 'Comic book creator', 'Barber surgeon', 'Landscape architect', 'Art dealer', 'Arts District', 'Amateur radio operator', 'Tax advisor', 'Crown land', 'Plant operator', 'Agricultural machinery']

Noisy Clusters:

Cluster id - 949; Size - 28wikiwords

['Chlorophyll', 'Executive producer', 'Holland Purchase', 'Korean sword', 'Vietnamese literature', 'Chinese International School', 'Partner dance', 'Anheuser Busch', 'Carolingian Empire', 'East Anglia', … ,'Post Soviet states', 'Vietnamese pronouns', 'Nonylphenol', 'Epimerase and racemase', 'Pop rock', 'East China Normal University', 'Epigastrium', 'Parenthesis', 'Vietnam National University']

For bert-base-cased model:

Coherent Clusters: Cluster id - 114 ; Size - 22 wikiwords

['Tutor', 'Linguistics', 'Rabbi', 'Pastor', 'Psychiatrist', 'Demon', 'Sniper', 'Double agent', 'Minister', 'Mortgage broker', 'Butler', 'Ambassador', 'Reporter', 'Talent agent', 'Small claims court', 'Course evaluation', 'Franklin Humanities Institute', 'Man to man marking', 'Credit default swap', 'TUTOR', 'Prophet', 'Minister']

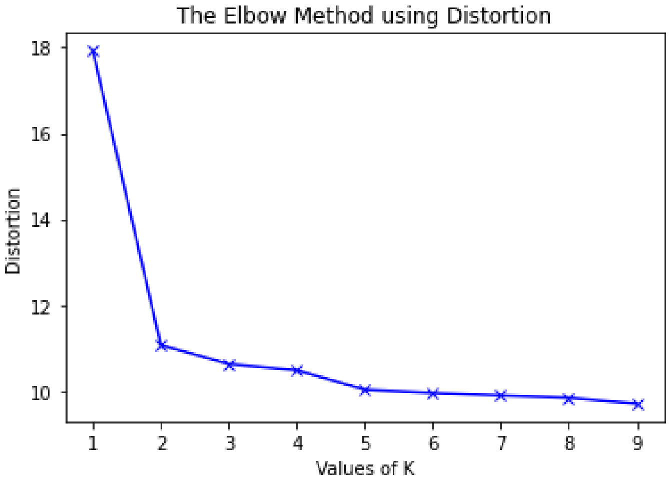
Noisy Clusters:

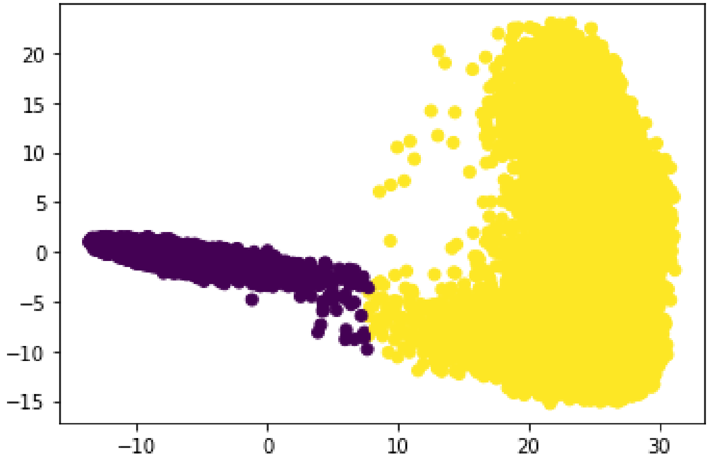
Cluster id - 98 ; Size - 45918 wikiwords

'Perfume', 'Multiracial', 'Tonne', 'Income tax', 'Muscle', 'Knife', 'Skull', 'Russian language', 'Rick Perry', ------ 'Front end and back end', 'Paper and pencil game', 'Higher education in the United States', 'Abraham Lincoln', 'Deportation', 'Ford Foundation'

Experiments and results:

Clustering based on Embeddings of Wiki Words

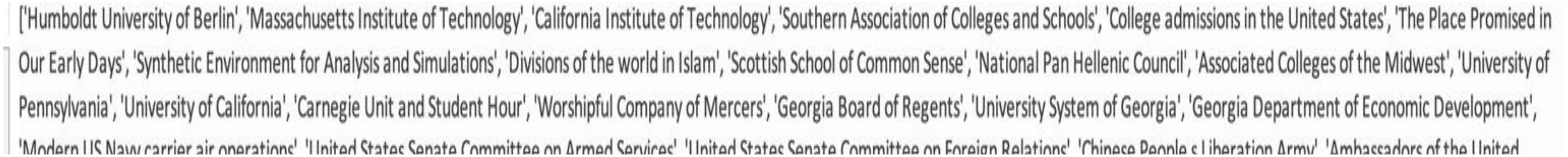




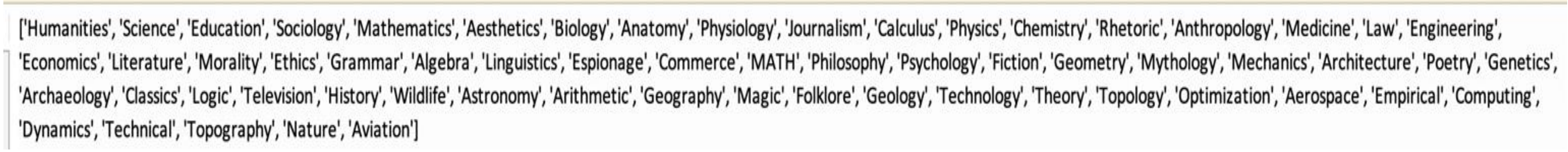
Analysis on K Means Clustering:

- Based on the elbow method, the ideal clusters should be 2.
- The purple cluster were all single wiki words.
- The yellow cluster were all multi word wiki words and just 26 single wiki words.

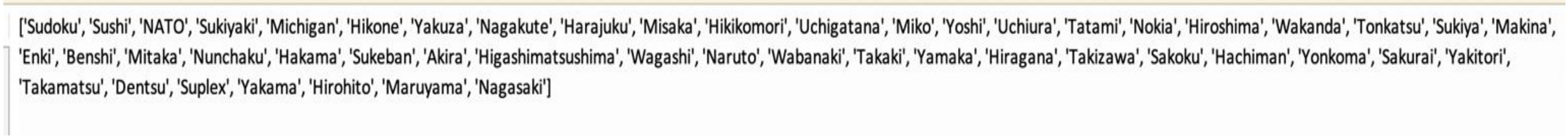
Coherent Clusters formed:



```
['Humboldt University of Berlin', 'Massachusetts Institute of Technology', 'California Institute of Technology', 'Southern Association of Colleges and Schools', 'College admissions in the United States', 'The Place Promised in
Our Early Days', 'Synthetic Environment for Analysis and Simulations', 'Divisions of the world in Islam', 'Scottish School of Common Sense', 'National Pan Hellenic Council', 'Associated Colleges of the Midwest', 'University of
Pennsylvania', 'University of California', 'Carnegie Unit and Student Hour', 'Worshipful Company of Mercers', 'Georgia Board of Regents', 'University System of Georgia', 'Georgia Department of Economic Development',
'Modern US Navy carrier air operations', 'United States Senate Committee on Armed Services', 'United States Senate Committee on Foreign Relations', 'Chinese People s Liberation Army', 'Ambassadors of the United
```

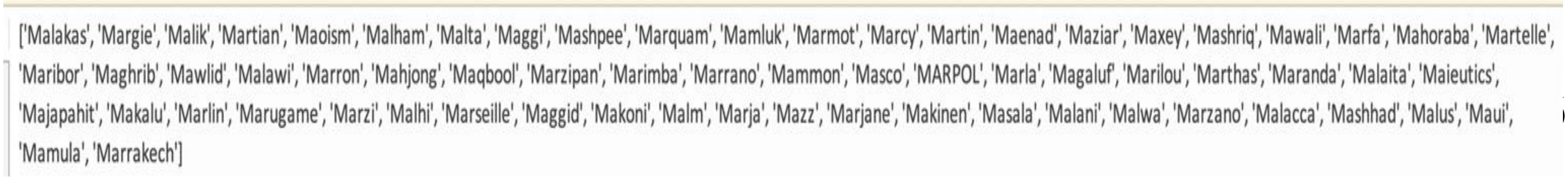


```
['Humanities', 'Science', 'Education', 'Sociology', 'Mathematics', 'Aesthetics', 'Biology', 'Anatomy', 'Physiology', 'Journalism', 'Calculus', 'Physics', 'Chemistry', 'Rhetoric', 'Anthropology', 'Medicine', 'Law', 'Engineering',
'Economics', 'Literature', 'Morality', 'Ethics', 'Grammar', 'Algebra', 'Linguistics', 'Espionage', 'Commerce', 'MATH', 'Philosophy', 'Psychology', 'Fiction', 'Geometry', 'Mythology', 'Mechanics', 'Architecture', 'Poetry', 'Genetics',
'Archaeology', 'Classics', 'Logic', 'Television', 'History', 'Wildlife', 'Astronomy', 'Arithmetic', 'Geography', 'Magic', 'Folklore', 'Geology', 'Technology', 'Theory', 'Topology', 'Optimization', 'Aerospace', 'Empirical', 'Computing',
'Dynamics', 'Technical', 'Topography', 'Nature', 'Aviation']
```



```
['Sudoku', 'Sushi', 'NATO', 'Sukiyaki', 'Michigan', 'Hikone', 'Yakuza', 'Nagakute', 'Harajuku', 'Misaka', 'Hikikomori', 'Uchigatana', 'Miko', 'Yoshi', 'Uchiura', 'Tatami', 'Nokia', 'Hiroshima', 'Wakanda', 'Tonkatsu', 'Sukiya', 'Makina',
'Enki', 'Benshi', 'Mitaka', 'Nunchaku', 'Hakama', 'Sukeban', 'Akira', 'Higashimatsushima', 'Wagashi', 'Naruto', 'Wabanaki', 'Takaki', 'Yamaka', 'Hiragana', 'Takizawa', 'Sakoku', 'Hachiman', 'Yonkoma', 'Sakurai', 'Yakitori',
'Takamatsu', 'Dentsu', 'Suplex', 'Yakama', 'Hirohito', 'Maruyama', 'Nagasaki']
```

Incoherent and Interesting Clusters:



```
['Malakas', 'Margie', 'Malik', 'Martian', 'Maoism', 'Malham', 'Malta', 'Maggi', 'Mashpee', 'Marquam', 'Mamluk', 'Marmot', 'Marcy', 'Martin', 'Maenad', 'Maziar', 'Maxey', 'Mashriq', 'Mawali', 'Marfa', 'Mahoraba', 'Martelle',
'Maribor', 'Maghrib', 'Mawlid', 'Malawi', 'Marron', 'Mahjong', 'Maqbool', 'Marzipan', 'Marimba', 'Marrano', 'Mammon', 'Masco', 'MARPOL', 'Marla', 'Magaluf', 'Marilou', 'Marthas', 'Maranda', 'Malaita', 'Maieutics',
'Majapahit', 'Makalu', 'Marlin', 'Marugame', 'Marzi', 'Malhi', 'Marseille', 'Maggid', 'Makoni', 'Malm', 'Marja', 'Mazz', 'Marjane', 'Makinen', 'Masala', 'Malani', 'Malwa', 'Marzano', 'Malacca', 'Mashhad', 'Malus', 'Maui',
'Mamula', 'Marrakech']
```

Conclusion

- Using argmax to cluster the wiki words resulted in one major cluster while others being really small clusters.
- Using the embeddings of the wiki words resulted in coherent clusters of consistent sizes but also some of the clusters formed were not coherent to a topic.
- The results after pretraining with various train steps were very interesting and gave a brief idea about overfit and underfit of data.

COVID-19 Bert

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

Project Goal

- We evaluate different configurations of Bert on creating a language representation for COVID-19 research papers in order to aid in answering research questions

Data

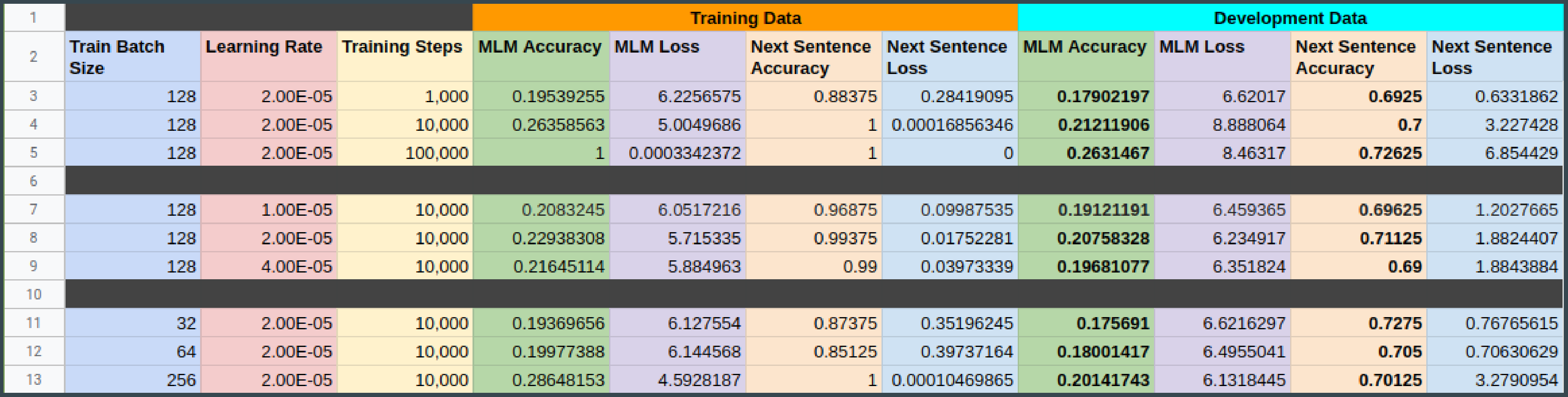
- Corpus of ~51k (6GB) Scholarly Articles on COVID-19
- We installed and configured SentencePiece to generate a new vocab
    - 32K word pieces
    - Everything is lowercase
- Example Words:
    - coronavirus
    - transmission

Experiments and Results - I

- Vary learning rate, training batch size, and training steps
  - Best Learning Rate: 2E-5
  - Best Training Batch Size: 128
  - Best Training Steps: 1,000
- Evaluate BERT configurations using MLM
  - .16 MLM accuracy
  - .69 Next Sentence accuracy
- Evaluate SciBert using MLM
  - .66 MLM accuracy
  - .86 Next Sentence accuracy
- Predict masked words using SciBert and BERT

Experiments and Results - II

- Rows 2-5 show varying train steps
- Rows 7-9 show varying learning rate
- Rows 11-13 show varying training batch size



1				Training Data				Development Data			
2	Train Batch Size	Learning Rate	Training Steps	MLM Accuracy	MLM Loss	Next Sentence Accuracy	Next Sentence Loss	MLM Accuracy	MLM Loss	Next Sentence Accuracy	Next Sentence Loss
3	128	2.00E-05	1,000	0.19539255	6.2256575	0.88375	0.28419095	**0.17902197**	6.62017	**0.6925**	0.6331862
4	128	2.00E-05	10,000	0.26358563	5.0049686	1	0.00016856346	**0.21211906**	8.888064	**0.7**	3.227428
5	128	2.00E-05	100,000	1	0.0003342372	1	0	**0.2631467**	8.46317	**0.72625**	6.854429
6											
7	128	1.00E-05	10,000	0.2083245	6.0517216	0.96875	0.09987535	**0.19121191**	6.459365	**0.69625**	1.2027665
8	128	2.00E-05	10,000	0.22938308	5.715335	0.99375	0.01752281	**0.20758328**	6.234917	**0.71125**	1.8824407
9	128	4.00E-05	10,000	0.21645114	5.884963	0.99	0.03973339	**0.19681077**	6.351824	**0.69**	1.8843884
10											
11	32	2.00E-05	10,000	0.19369656	6.127554	0.87375	0.35196245	**0.175691**	6.6216297	**0.7275**	0.76765615
12	64	2.00E-05	10,000	0.19977388	6.144568	0.85125	0.39737164	**0.18001417**	6.4955041	**0.705**	0.70630629
13	256	2.00E-05	10,000	0.28648153	4.5928187	1	0.00010469865	**0.20141743**	6.1318445	**0.70125**	3.2790954

Experiments and Results - III

- Masked Word Prediction
- Example:
  - “[CLS] severe acute respiratory syndrome coronavirus covid19 , cause of the potentially deadly atypical **[MASK]** , infects many organs , such as lung , liver. [SEP]”

Our Pretrained Model

26.	study
27.	identified
29.	dogs
30.	patients
32.	cells

**our model found stop words to be most probable*

SciBERT

1. **pneumonia**
2. syndrome
3. disease
4. infection
5. influenza
6. virus

Conclusion

- SciBert outperforms pretraining BERT-BASE on COVID-19 data
- SciBert was able to predict a masked word exactly from a COVID-19 article
- In the future:
    - Fine-tune SciBert on the COVID-19 corpus
    - Mask out specific words during the training phase so that we don’t get stop words

BERT for Geo Mapping Entities
Team 3

Sai Charan Akenna
Sai Vishal Kodimela
Siddhant Jain
Suriya Prakaash Sundaram Kasi Thirunavukkarasu

Project Objective

- Discovering topics/clusters of Wiki_Words.
- Form groups with wiki words within the same topic
- Map the topics according to the geographic location.
- For each document, Wiki_Words pair show the topic assignment

Data

- The dataset contains comments made by people on topic ‘humanities’ from reddit and the data was already wikified using Illinois Wikifier.

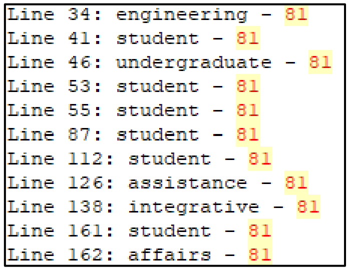
	Count
Total Wiki_Words	1.321,967
Wiki_Words(After removing duplicates)	78,598
Matching Single_Words with BERT Vocab	3,718
Matching Multiple_Words with BERT Vocab	14,039

Different types of wiki words	**Type**
Wiki__Journalism__isqza	Single word
Wiki__Mass_communication__ibosles	Multiple words
Wiki__W__E__B__Du_Bois__aqqaa	Multiple words with Double underscore in between
Wiki____st_century__efslz	Words with more than double undercore
Wiki_______efqis	Just wikifier tokens

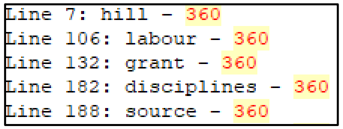
Approach – 1 (Grouped using the topic IDs)

- Used pretrained bert uncased model with 12 layers and 768 features.
- Used Bert-as-a-service to extract word embeddings, and assigned index of highest vector value as topic ID.
- Some groups have words only related to that topic.
- Disadvantage of this approach is that we found that groups might contain words irrelevant to that topic too.

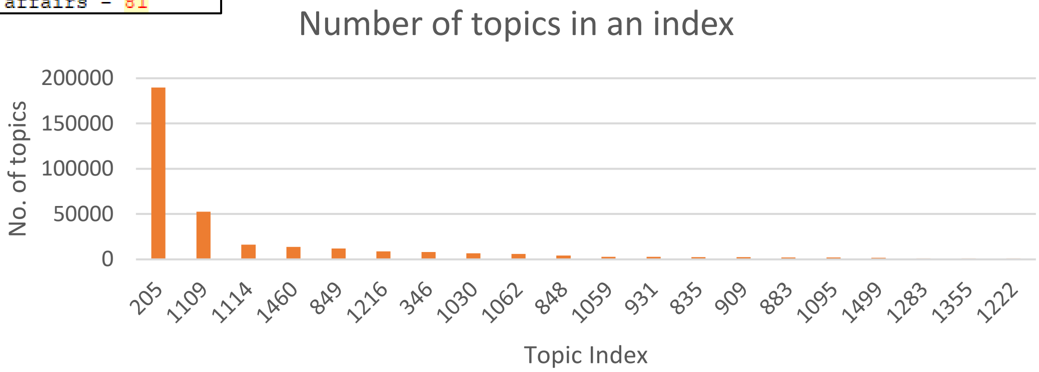
Wiki Word	Index	Words Count
Professor	205	189,666
Humanities	1109	52,502
Language	1114	16,247
Curriculum	1460	13,542
Journalism	849	11,760
medicine	1216	8,706
astronomy	346	8,122
tuition	1030	6,651
facebook	1062	5,725
fascism	848	4,008
university	1059	2,574
europe	931	2,529
youtube	835	2,411
fireworks	909	2,205
biology	883	1,973



```
Line 34: engineering - 81
Line 41: student - 81
Line 46: undergraduate - 81
Line 53: student - 81
Line 55: student - 81
Line 87: student - 81
Line 112: student - 81
Line 126: assistance - 81
Line 138: integrative - 81
Line 161: student - 81
Line 162: affairs - 81
```

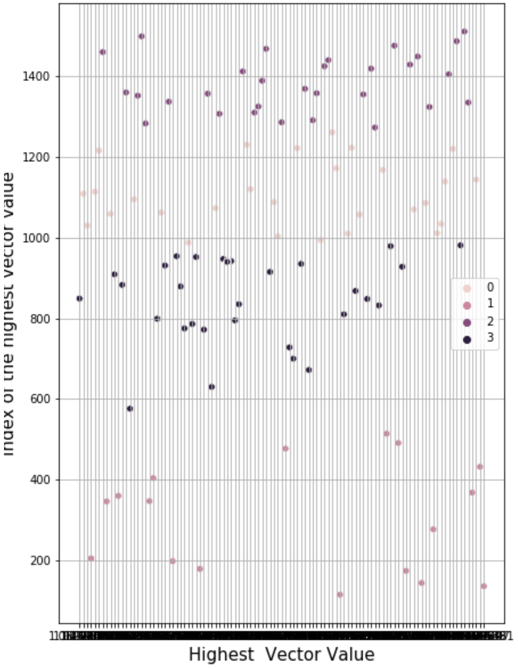


```
Line 7: hill - 360
Line 106: labour - 360
Line 132: grant - 360
Line 182: disciplines - 360
Line 188: source - 360
```



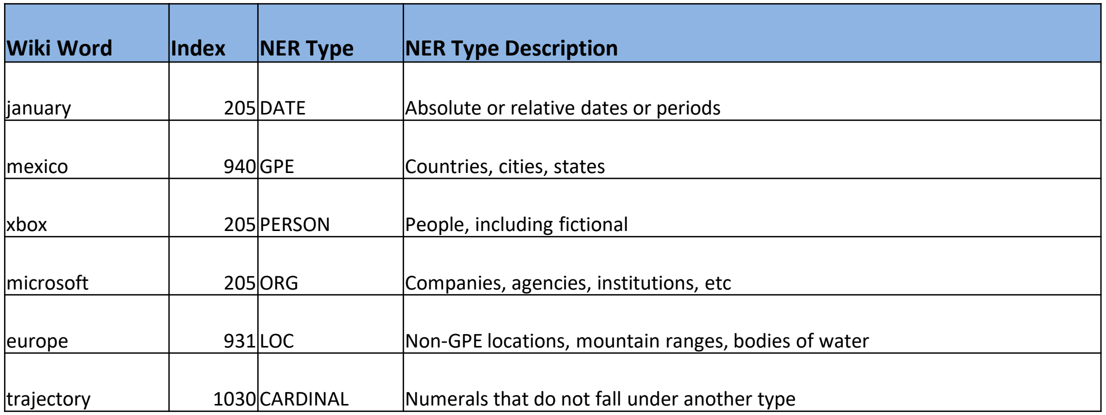
Approach – 2 (Grouped using K-means algorithm)

- We used the highest vector values and index of the same of every wiki word as input for kmeans model.
- Out of 104 unique index, the optimal number of clusters for kmeans is only 4.
- We could see that in this approach, clusters are formed with respect to the range of index value, because of which this is inefficient.



Approach – 3 (Grouped using NER)

- We used Named Entity Recognition from spacy library to identify the type of the word.
- Some of our results were interesting, but the disadvantage with this approach is, it identifies the word type without considering the context.
- Thus this way of grouping in this dataset is that much efficient.



Wiki Word	Index	NER Type	NER Type Description
january	205	DATE	Absolute or relative dates or periods
mexico	940	GPE	Countries, cities, states
xbox	205	PERSON	People, including fictional
microsoft	205	ORG	Companies, agencies, institutions, etc
europe	931	LOC	Non-GPE locations, mountain ranges, bodies of water
trajectory	1030	CARDINAL	Numerals that do not fall under another type

Conclusion

The data provided is small and it has repeated wiki words in different concepts.

From all the three approaches we found that, grouping of wiki words for this dataset is not efficient.

Some of the final groups in each approach have words that are irrelevant to that topic and can have different meaning with respect to context in which that occurs.

For example,

when we find the cosine similarity of two words,

cos_sim('humanities', 'professor') = 0.64957803

when the same is used in a contextual text, the similarity changes,

cos_sim('the professor teaches humanities', 'he studies humanities') = 0.429105

Since there is such a discrepancy, it was not feasible to use this approach.

BERT, kmeans, and spacy are not suitable classification techniques for this type of data.

CS-522
Environment project

•••

Agnes Gaspard, Kanyakumari Kashyap
Kavya Ravella, Shweta Metkar

Introduction

Goal of the project

- Get contextual word representations for our data using BERT, when we have new words (Wiki words) that available BERT models were not trained with
- Define topics for those Wiki words using embeddings and clustering

Data

Summary of data

- 118,059 comments from Reddit
- 78,597 distincts Wiki words
- Only 16% of the Wiki words occur more than 10 times
- “Wiki_Liberal_arts_iablf” has occurred 43117 times - considered as most occurrent

Summary of approaches used

- Pre-train BERT with frequent Wiki words
- BERT embeddings to cluster Wiki words

Experiment results - I

Pre-train BERT	BERT embeddings
Take wiki word “Wiki_University_eilqf” (2574 times repeated). Take 10 words occurring before and after the wiki word in a .tsv file. Remove the stop words in order to decrease the effect of accuracy. Build a TensorFlow module. Export the BERT Vocabulary into the module. Build a Text Preprocessing Pipeline Implement a Bert Keras layer by wrapping the TensorFlow module with hub.KerasLayer. Implement Sentence-Pair-Classification upon it.	Get sentences and remove wikification Use heuristic for sentences and get wiki word indexes Tokenize sentences and get token ids Get BERT embeddings, 3072 dimensions Remove stop words and punctuation Get max values and dimension per vector Cluster by dimension for topic Analyze manually to see if it makes sense

Experiments results - II

Pre training

- Taken the approach to use Bert-as-service for pre training previously.
- Noticed few limitations -> increasing the number of layers leads to overfitting the model.
- Pretrained the model using another approach -> Building Keras layer on the TensorFlow module.
- Pretrained the sentences containing the wiki words so that the wiki words which we have added to our Bert vocabulary achieve contextual word representations.

Experiment results - III

BERT embeddings

- 1,094,852 sentences
- Wikification irregularities: wiki____th_century__efbzs, wiki_______esbqs
- Max length sentences: 300 then 150 for tokenizing (BERT 512 tokens max): +151,481 sentences
- Argmax without absolute value
- Some clusters elements
  - 'bugs', 'question', 'creativity', 'technology', 'labs', 'computer', 'knowledge'
  - 'students', 'attend', 'percentage', 'older', 'younger', 'brilliant', 'great', 'ace', 'superstar', 'world'

Conclusion

- Data: size, RAM, sentences length, K Means failed
- We should have worked with the whole data since the beginning.
- Dimension clustering: clusters do not really make sense, K Means probably would have been better
- Pretrained the wiki words for 200 sentences