CSCI 6509 P1

Shakhboz Abdulazizov
B00870379
Dalhousie University
Halifax NS, Canada
sh873370@dal.ca

Bhuvaneshwari Basquarane
B00853122
Dalhousie University
Halifax NS, Canada
bh563857@dal.ca

Rakshit Makan
B00883651
Dalhousie University
Halifax NS, Canada
r.makan@dal.ca

Noah Sealy
B00726289
Dalhousie University
Halifax NS, Canada
noah.sealy@dal.ca

1. Problem Statement

Document clustering is an important problem in Machine Learning which combined with NLP language models can be useful in exploratory analysis of document corpora. In this project, we try to understand how different clustering algorithms in combination with dimensionality reduction and a state-of-the-art Deep Language Model over a text corpus taken from the Abstract of research papers in a domain of machine learning, natural language processing and deep language models. We are interested in algorithms which can model the semantics and the context of the frequently occurring terms in the corpus and are robust in the presence of vocabulary mismatch.

2. Possible Approaches

2.1. Dataset

The data-set which we are trying to utilize is the research papers from the library of the MALNIS lab. The data format which we received is of CSV that consists of headers like Bibliography Type, Identifier, Author, Title, Journal, Month, Year, URL, Abstract. The collection holds one thousand papers in the area of machine learning, natural language processing and deep language models. The idea behind this approach is, we will use deep language models specifically to process our text corpus from the different research papers abstract section and project them onto a vector embedded space. Here we have performed pre-processing activities like duplicate removal, eliminating null records and usage of stop words and special characters on top of the given data-set.

2.2. Preprocessing

As described in previous section the text has been extracted from the PDFs of research paper of MALNIS lab, due to which there are multiple anomalies in the data. We have to clean and pre-process the data before we can pass it through the deep language models. For this task we are going to use the NLTK python library, as well as regular expressions to clean the corpus. Firstly, we are going to drop the duplicates, then remove stop-words, special characters, and numerical characters as they might lead to skewed results in clustering . After that we will be using WordNet Lemmatization on the corpus to get the roots of the words in the text. This part is going to be expanded as per the requirements and results of the clustering.

2.3. Deep Language Model

In this part we will be embedding the text using the following deep language models and will use the vector form of the text to cluster the documents:

BERT - BERT (Bidirectional Encoder Representations from Transformers) is a recent paper published by researchers at Google AI Language [1]. It has caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others. BERT's key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training.

InferSent - This a model developed by Facebook in 2018 paper to embed large chunk of text [4]. The model is trained on the supervised data of the Stanford Natural Language Inference datasets. Much like how computer vision uses ImageNet to obtain features, which can then be transferred to other tasks, our work tends to indicate the suitability of natural language inference for transfer learning to other NLP tasks.

2.4. Dimensionality Reduction

Currently, one of the chosen approaches for reducing the dimensions of the word embedding representations is Principal Component Analysis, or PCA. PCA is a method of dimensionality reduction which will be employed in order to represent the documents at two or three dimensions. To start, this is convenient for the researcher as it will allow for visual analysis of the clusters which are easily readable to the human eye. Another factor of convenience associated

with dimensionality reduction techniques, such as PCA, is that it will reduce the overall complexity of a given word embedding by a large factor, making clustering a more efficient task. This is crucial as the word embedding's are usually represented in very high dimensions by the deep language models.

Though PCA is used as an example in this section, it is only one of the various options for dimensionality reduction. An alternative approach is t-Distributed Stochastic Neighbourhood Embedding, or t-SNE [5]. Similar to PCA, t-SNE will allow the word embedding to be projected onto a two or three-dimensional space for better clustering and visual analysis of the results.

Both dimensionality reduction options will be considered throughout testing of the model, in order to find which brings the most valuable and efficient representation of the documents. The chosen technique will be used in the final implementation of the project.

2.5. Clustering

For clustering part of the problem we are going to use one of the following clustering algorithms with the aforementioned Deep Language Model (BERT) and compare the results in order to find out which Deep Language Model and Dimensionality reduction techniques works best for clustering:

K-Means - being one of the most widely known clustering algorithm, it is fast and easy to understand. However it is a partitioning algorithm rather than clustering as it partitions the dataset into as many parts as requested through trying to minimize intra-partition distances [2].

DBSCAN - is a density based algorithm and considers dense regions as clusters. DBSCAN groups together the points that are closely packed together, assuming points that lie in low density regions as 'noise'. It needs a minimum cluster size and a distance threshold epsilon as user-defined input parameters.

HDBSCAN - a relatively new algorithm which allows varying density clusters. In addition to being better for data with varying density, it's also faster than regular DBSCAN. HDBSCAN needs the minimum cluster size as single input parameter [3].

3. Project Plan

Table 3.1 shows the anticipated schedule for the project's development from February 22 to March 22. This month of development will entail a full development sprint; from data collection to model evaluation. As discussed throughout this proposal, this sprint is focused on creating evaluating different word embedding models based on clusters.

In reference to the week of February 22 in Table 3.1, data collection and preprocessing refers to finding data which is appropriate to the problem. This data will most likely be a large collection of documents, each with an explicit topic. The preprocessing aspect entails preparing the given text data in order for it to be able to interface with the chosen word embedding models effectively and efficiently. Although preprocessing is not mentioned further in the development timeline, it is expected that the preprocessing itself may change as the project evolves.

Table 3.1: Project development timeline, in anticipation for the given deadlines. Note, this table includes work from past weeks for reference.

Week Of	Tasks
Feb 22	Group formation. Data collection and preprocessing.
Mar 1	Explore deep language models. Explore dimensionality reduction techniques.
Mar 8	Data clustering and model evaluation.
Mar 15	Data clustering and model evaluation.
Mar 22	Process results. Prepare project report and presentations.

March 1 in Table 3.1 refers to the background research which must take place for this project; here the group aims to explore different model solutions, as well as potential dimensionality reduction techniques which may suffice in furthering the progress of the project. The options so far found at the time of this proposal are discussed above.

March 8 and March 15 in Table 3.1 refer to where most of the actual development for the project will occur. At this time, the group will be implementing the various chosen models and techniques. Once implemented, each model will be evaluated and recorded, in order to be discussed in the final project report.

Lastly, March 22 in Table 3.1 refers to analysis of the model results from the previous weeks. This analysis will allow the group to write a project report on the given results. On top of preparing this report, the group will also prepare presentations relating to the project as a whole. As the group is made up of masters students, taking the course at the 6000-level, these presentations will be presented separately, but prepared synchronously.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, et al. "Attention is all you need," Advances in neural information processing systems, 2017.

[2] L. McInnes, J. Healy, S. Astels (2016) *Comparing Python Clustering Algorithms*. Retrieved from https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html.

[3] B. Bailey (2017) *Lightning Talk: Clustering with HDBScan*. Retrieved from https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html.

[4] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. EMNLP, 2017.

[5] E. Sherkat, E. Milios, and R. Minghim. A Visual Analytics Approach for Interactive Document Clustering. ACM 2019.