Topic Modelling with Transformer Models and Variational Autoencoders

Aleksander Mako 5507254

October 5, 2021

1 Introduction

Topic modelling is an unsupervised technique which aims to extract meaningful topics from a corpus of texts.
This is a well researched area of NLP and the traditional tools such as LDA and NMF often are the preferred techniques for such tasks. However the main drawback in these methods is the number of hyper parameters and the need to extensively tune them to get an optimal performance.
In contrast to these techniques Topic2Vec is one of the latest techniques that addresses the problem of hyper parameter tuning and produces topics which are easy to interpret. Inspired by this project I have provided an alternative implementation which relies on sentence embeddings using transformer models and a deep learning model which is called Variational Autoencoder.

2 Project Description

2.1 Data

I have used two data sources while developing the project. The first one is the 20newsgroups dataset which is a a collection of 20,000 newsgroup documents spread out almost evenly among 20 newsgroups.Each newsgroup corresponds to different topics and was also used by the authors of the Topic2Vec paper.
However I have removed the dataset from the demoable code because the task on this dataset is very easy and I only used it as an initial benchmark for my implementation.
The second data used in this project is a collection of Medium articles. There are in total 337 articles available in CSV format. The dataset contains 6 columns containing the author, the number of claps, reading time, link to the article, it's title and then the raw text.
The analysis has been performed on the raw text directly without using the title. The algorithm expects the data to be provided as a list of strings where each string should be a sentence. Therefore in during data preparation I have used NLTK's sentence tokenizer to extract sentences from each medium article.

2.2 Algorithm

The original Topic2Vec [Ang20] paper describes a five step Algorithm.

1. Create the joint embeddings,
2. Find low dimensional embeddings.
3. Cluster the low dimensional embeddings.
4. Compute the center of the cluster.
5. Find the k nearest word vectors to the cluster centroid.

In a similar manner I have also implemented these five steps with slight variations.

1. Create sentece embeddings using huggingface pre-trained transformer model.
2. Create low dimensional representation of the word embeddings using VAE instead of UMAP.
3. Cluster the low dimensional k-means.
4. Compute cluster based tf-idf scores.
5. Consider k words with the top tf-idf scores.

Starting with the transformer model we get a 384 dimensional dense vector representation per sentence or alternatively it can be used for mapping paragraphs as well. Transformer models have been able to achieve state of the art results in various NLP tasks due to their two way training which allows for forward and backward context learning. Due to their excellent results I decided to build my implementation on top of them.
The second point of divergence is in finding a low dimensional representation of these embeddings. In the original paper the authors have used UMAP as their algorithm of choice. UMAP [MHM20] is able to perform a reduction in dimensions while retaining neighborhood area information. This is possible because UMAP relies on the initial data topology and projects the data to a lower dimension along with topology information. This is useful for the next step which is clustering. If two embeddings were considered close or similar prior to the reduction in order to produce good and accurate clusters after the reduction of dimensions the vectors should still remain close or similar after it. As opposed to UMAP I used a different method here which relies on deep learning known as a Variational Autoencoder with and adaptive learning rate.
A VAE is a deep learning architecture which is designed to have a bottleneck layer where the dimensions are smaller than those of the input layer. The goal of this technique can be broken down into two parts.

1. Given some input x learn the conditional distribution z given x where z is the lower dimensional representation of x.
2. Given the lower dimensional representation z learn the higher dimensional mapping x' .

This property is important because in the lower dimensional space we can learn a range of values instead of fixed one. Which in turn makes it possible to achieve similar endings for two similar sentence embeddings.
In the third step of the algorithm the author uses DBSCAN to find dense clusters while I have implemented K-MEANS. I found K-MEANS to be a more suitable method for my implementation due to a drawback of the Variational Autoencoder which is it's smooth lower dimensional representation. Initially a vanilla VAE is better suited for tasks where cross class interpolation is a desired property however DBSCAN in this instance requires extensive configuration in order to produce a decent amount of clusters.
On the other hand K-MEANS being a centroid based method is able to detect clusters among points which should clearly be in different clusters.
The last point of difference is the tf-idf scores computed between clusters. The intution is that when we use tf-idf on k documents we are comparing how important is each word in each document. Therefore if we just assume one cluster is a single big document then apply tf-idf we should get how important is a word in each cluster thus forming the topic. [Gro20]

2.3 Results

To benchmark my results I have also completed the same task using a simple LDA model. In order to keep the comparison fair I have done the same amount of data preprocessing for my model as for the LDA one.
The metric used to evaluate the performance is topic coherence calculated with the u_mass formula.
Out of the box for given a small not optimised number of topics,not optimised beta and alpha the plain LDA model preforms worse.

	n	alpha	beta	u_mass_score
LDA	5	symmetric	0.6	-1.69465075371047
Project	5	-	-	-0.427210028288895

Table 1: Out of the box performance no optimisation of LDA.

	n	alpha	beta	u_mass_score
LDA	14	symmetric	auto	-0.31741277162613524
Project	5	-	-	-0.427210028288895

Table 2: Optimised LDA model.

The closer the u_mass score is to 0 the better the model performance and in this case the project out of the box with minimal optimisation performs better. However the LDA outperforms the project when the parameters have been tuned. In conclusion the best performance of the project comes from choosing only 5 topics. For the current implementation of the project a smaller number of topics will produce good results due to the smooth VAE latent space which would put all the embeddings closer to each other after the reduction in dimensions.
While LDA performs best on a total of 14 topics with the parameters shown in the second table. Even after optimisation the u_mass score between the two models is still comparable.

2.4 Further Research and Improvements

The main drawback of the implementation I have provided is the cluster quality. As discussed in previous section after dimensionality reduction using a Variational Autoencoder the space in which these smaller dimensional vectors reside is very smooth. It is in fact intended to be like that so that in generative tasks we can interpolate values between classes. However in this specific task I found that this property of VAEs makes it harder to produce well separated clusters.
However this problem is tackled well by Junyuan et.al [XGF15] in the paper Unsupervised Deep Embedding for Clustering Analysis where an approach is shown on how to achieve state of the art clustering performance on the latent space of Variational Autoencoders using K-MEANS. In this paper the authors propose to optimise the centroids of clusters with a deep learning technique while learning a lower dimensional representation of the input data.
Further more Sudipto et.al [MALK18] describe a completely new way of performing dimensionality reduction and clustering the latent space of Generative Adversarial models. Just like VAEs, GANs learn a probability distribution and have a similar architecture with a bottleneck which is part of the architecture and allows these networks to learn lower dimensional mappings of the input data. Further more the authors of the paper propose a new novel back propagation algorithm which would would be able to learn a latent space which is much more separable than what I have achieved in this project.
On a positive note this implementation requires only 1 parameter and that is the number of topics. In addition optimising the model performance over the number of topics is much faster than LDA. Lastly training the model takes time if training takes place on regular CPUs instead of GPUs however it can achieve decent results with minimal configuration.

References

[Ang20] Dimo Angelov. Top2vec: Distributed representations of topics. *CoRR*, abs/2008.09470, 2020.

[Gro20] Maarten Grootendorst. Topic modeling with bert, Oct 2020.

[MALK18] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. Clustergan : Latent space clustering in generative adversarial networks. *CoRR*, abs/1809.03627, 2018.

[MHM20] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[XGF15] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. *CoRR*, abs/1511.06335, 2015.