CBERTdp

Clustering BERT Embeddings for Classification in Sentiment Analysis via Dot Product

Thomas Vecchiato 880038 - Riccardo Zuliani 875532 - Alice Schirrmeister 1000371 - Isabel Marie Ritter 1000095

University Ca'Foscari of Venice
Department of Environmental Sciences, Informatics and Statistics

February 26, 2024

Ca' Foscari University of Venice

Objective

Enhaning the efficiency of Sentiment Analysis

- By redistributing the tasks laid upon neural networks
- While maintaining a good accuracy

How?

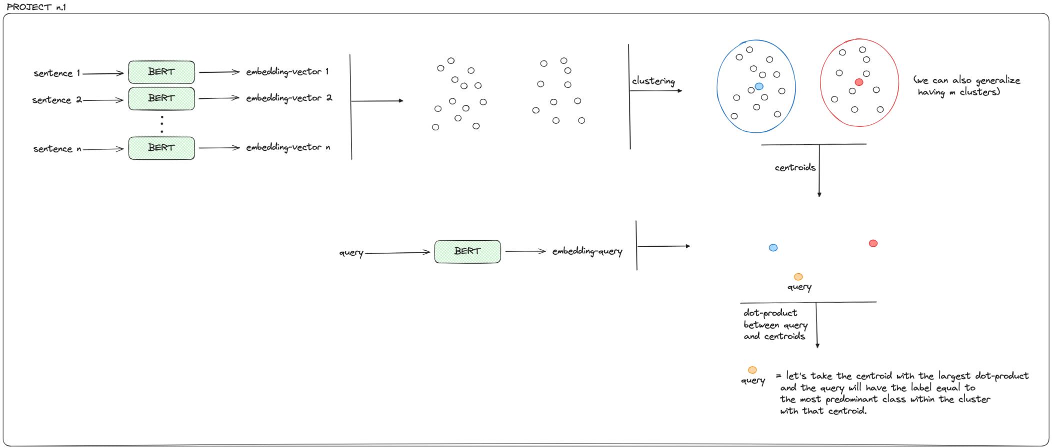
Employing **K-means clustering** and the **dot product**

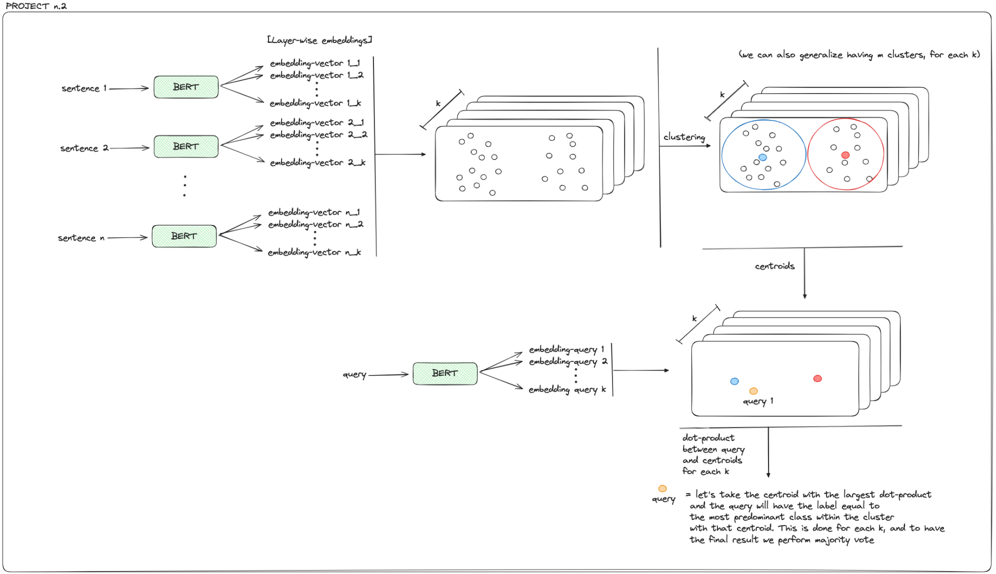
- Clustering BERT [2] embeddings
- Classification via dot product between centroids and new embedding
- *Three approaches*
  - Each of different complexity to increase the accuracy step-wise
- Can be extended to other classification problems

Which are the related studies?

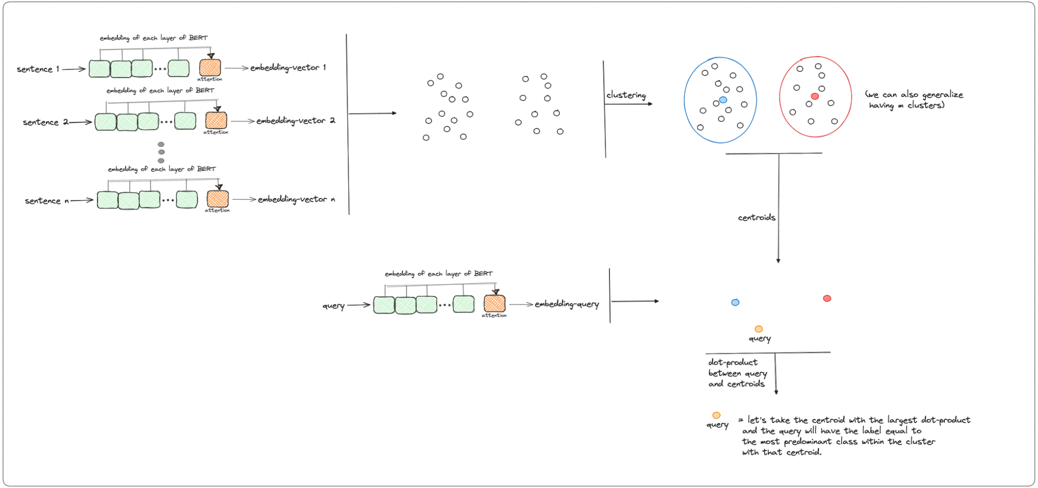
Clustering BERT embedding is not a new idea, indeed much research has been conducted in this field:

- Power of contextualized word embedding [8]
- Benchmarks for combination of different embeddings and clustering algorithm [10]
- BERT $\rightarrow$ UMAP $\rightarrow$ HDBSCAN [4]
- Embedding into LDA [3]
- Topic modelling and prototype selection [7] [5] [1]
- Prototype-Selection [11]


PROJECT n.1
sentence 1
sentence 2
sentence n
BERT
BERT
BERT
embedding-vector 1
embedding-vector 2
embedding-vector n
clustering
(we can also generalize
having m clusters)
centroids
query
BERT
embedding-query
query
dot-product
between query
and centroids
query
= let's take the centroid with the largest dot-product
and the query will have the label equal to
the most predominant class within the cluster
with that centroid.


PROJECT n.2
[Layer-wise embeddings]
sentence 1
BERT
embedding-vector 1_1
embedding-vector 1_2
embedding-vector 1_k
sentence 2
BERT
embedding-vector 2_1
embedding-vector 2_2
embedding-vector 2_k
sentence n
BERT
embedding-vector n_1
embedding-vector n_2
embedding-vector n_k
k
(we can also generalize having m clusters, for each k)
clustering
k
centroids
k
query
BERT
embedding-query 1
embedding-query 2
embedding query k
query 1
dot-product
between query
and centroids
for each k
query
= let's take the centroid with the largest dot-product
and the query will have the label equal to
the most predominant class within the cluster
with that centroid. This is done for each k, and to have
the final result we perform majority vote

PROJECT n.4



embedding of each layer of BERT
sentence 1
attention
embedding-vector 1
embedding of each layer of BERT
sentence 2
attention
embedding-vector 2
embedding of each layer of BERT
sentence n
attention
embedding-vector n
clustering
(we can also generalize having m clusters)
centroids
embedding of each layer of BERT
query
attention
embedding-query
query
dot-product between query and centroids
query
= let's take the centroid with the largest dot-product and the query will have the label equal to the most predominant class within the cluster with that centroid.

Experiments I

- **Google Colaboratory Pro +** using Tesla T4
- **Pytorch** and **HuggingFace** as deep learning libraries
- **FAISS** [6] library for the implementation of K-Means on the GPU
- **Datasets** are all available on HuggingFace and have positive and negative labels
  - *IMDb*[1]*:* movie reviews
  - *Stanford Sentiment Treebank*[2]*:* movie reviews
  - *Yelp Polarity Review Dataset*[3]*:* yelp reviews

[1] https://huggingface.co/datasets/imdb
[2] https://huggingface.co/datasets/sst2
[3] https://huggingface.co/datasets/yelp_polarity

Experiments II

- **Confidence measure** to assess the cluster goodness/purity [11]
- **Performance evaluation:**
  - Accuracy Score
  - F1-score
- **Comparison with baseline:**
  - *Naive:* predict the most common class
  - *Random choice*
  - *Machine learning:* SVM, Naive Bayes, Logistic Regression and KNN
- **Competitors:** multiple combinations of BERT plus one the following additional module: Linear layer, LSTM and GRU, the last two both uni and bidirectional
- **Training parameters:** 100 epochs, CrossEntropyLoss and AdamW, learning rate to $2e-5$, early stopping strategy on loss (after 20 epochs)

Three main steps of our approaches

1. **Saving Pre-trained BERT Embedding:**
   - Get the Layer-Wise embedding and store them in a separate *.npy* file
   - Done for each dataset saving the *training*, *validation* and *test* embedding
2. **Select Embeddings:**
   - **Main Approach** CLS from the last BERT layer
   - **Layer-Wise**
     1. Concatenation of the CLS token of all layer
     2. Mean of the CLS token of all layer
   - **Layer-Aggregation**
     - Same embeddings as the first variation of Layer-Wise
     - Embedding fed into a Multi-Head Self Attention layer
3. **Clustering with K-Means and Save Results**

- **Layer-Aggregation** outperforms its counterparts across all evaluation metrics
- Underperformance of the **Layer-Wise** approach
- Choice of dataset does not significantly influence the results

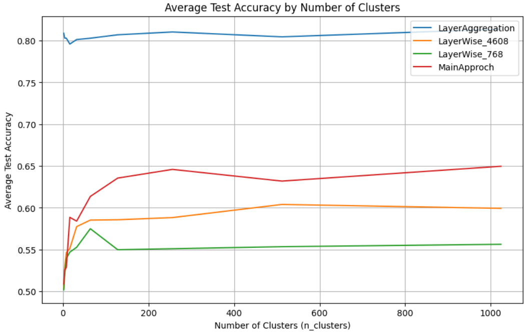


Figure 1: Mean accuracy for each of the separate methods we propose.

- **Layer-Aggregation** matches the baselines
- **Logistic Regression** outperforms it
- Remaining approaches fall short of the Machine Learning baselines performance
- **Competitors** outpace our approaches (higher accuracy)

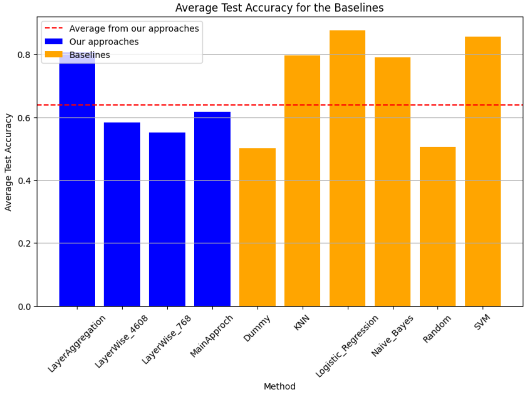


Figure 2: Mean accuracy in comparison with the baselines

- **KNN** and **SVM** have higher computational demands than our approaches (by elapsed time)
- **Logistic Regression** outperforms our own approaches
- **Competitor** models outperform our own approaches (different usage of GPU)
- For our own approaches approx. more computational costs with increasing accuracy

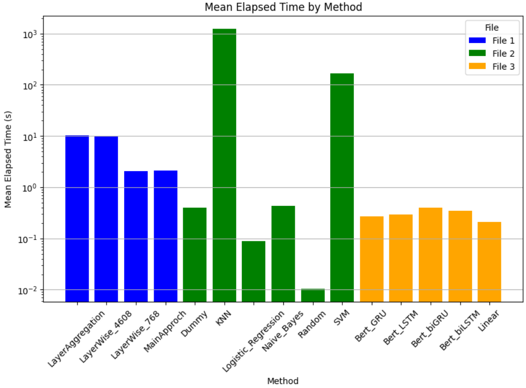


Figure 3: Average elapsed time scaled logarithmically

- Results of IMDb dataset
- $\#$ clusters $k = 128$, as optimal k on average
- No evidence of a clear pattern. However, $m = 8$ gives good results
- We can generally observe only 6% of the closest clusters in terms of dot-products
- To find the **best result** for a given *dataset* and a given *number of clusters*, the **choice of** m **is important**

Figure 4: Variation of the accuracy in test, varying the top-m clusters

Ablations - DistilBERT [9]

method	dataset	n_cls	top_k	t_acc	f1	conf	s_el
MA	imdb	1024	64	0.741	0.738	0.548	6.622
MA	sst2	1024	1	0.687	0.686	0.473	0.222
MA	y_p	256	32	0.735	0.722	0.558	2.784
LW_9216	imdb	1024	32	0.687	0.688	0.471	92.658
LW_9216	sst2	1024	1	0.627	0.624	0.354	3.297
LW_9216	y_p	1024	32	0.757	0.756	0.506	145.846
LW_768	imdb	1024	512	0.614	0.608	0.451	6.622
LW_768	sst2	1024	64	0.634	0.629	0.348	0.225
LW_768	y_p	256	1	0.634	0.631	0.401	2.698
LA	imdb	512	16	0.826	0.826	0.743	41.882
LA	sst2	4	2	0.509	0.338	0.695	0.025
LA	y_p	512	256	0.913	0.913	0.802	53.749

- BERT pre-trained model results

method	dataset	n_cls	top_k	t_acc	f1	conf	s_el
MA	imdb	1024	64	0.678	0.672	0.544	7.647
MA	sst2	1024	1	0.581	0.522	0.523	0.224
MA	y_p	256	32	0.789	0.787	0.581	2.755
LW_4608	imdb	1024	32	0.626	0.621	0.416	34.575
LW_4608	sst2	1024	1	0.569	0.558	0.384	1.18
LW_4608	y_p	1024	32	0.573	0.492	0.456	53.831
LW_768	imdb	1024	512	0.601	0.59	0.405	6.51
LW_768	sst2	1024	64	0.565	0.501	0.375	0.22
LW_768	y_p	256	1	0.526	0.405	0.391	2.795
LA	imdb	128	4	0.5	0.334	0.066	5.03
LA	sst2	4	2	0.688	0.666	0.703	0.016
LA	y_p	512	256	0.89	0.89	0.793	26.999

- DistilBERT pre-trained model results

Ablations - Base embedding with PCA

method	dataset	n_cls	top_k	t_acc	f1	conf	s_el
MA	imdb	1024	64	0.53	0.502	0.201	0.527
MA	sst2	1024	1	0.528	0.528	0.171	0.011
LW_9216	imdb	1024	32	0.535	0.528	0.235	0.487
LW_9216	sst2	1024	1	0.516	0.514	0.17	0.011
LW_768	imdb	1024	512	0.544	0.54	0.239	0.619
LW_768	sst2	1024	64	0.509	0.338	0.168	0.019
LA	imdb	128	4	0.811	0.811	0.139	0.243
LA	sst2	4	2	0.509	0.338	0.179	0.008

Table 1: Results using PCA on pre-trained BERT

- $PCA(\phi(\text{sentence})) = \vec{v} \in \mathbb{R}^2$
- Results are still good, losing approximately only 10% from the initial dimensionality
- Speed up in computational terms
- *yelp_polarity* dataset is excluded

Conclusion

- **CBERTdp**: new approach for sentiment analysis classification
- The method is composed essentially by two simple but powerful points:
  - Bidirectional transformer model
  - Dot-product
- **Results**: the results of our approach in terms of accuracy are not up to the level of the competitors and are similar to the baseline
- **Hyper-parameters**: k for the number of clusters and m for the top-m clusters after the dot-product
- **Property**: simplicity
- **New directions**:
  - Testing new centroid-based clustering algorithms
  - Finding representatives more suited to the problem for each cluster
  - Improve the embeddings
  - Extending the approach to n different sentiments

Thank You for your Attention!

The project code is available at this link, or you can scan the following QR Code

[1] Ercan Atagün, Bengisu Hartoka, and Ahmet Albayrak. "Topic Modeling Using LDA and BERT Techniques: Teknofest Example". In: *2021 6th International Conference on Computer Science and Engineering (UBMK)*. 2021, pp. 660–664. DOI: 10.1109/UBMK52708.2021.9558988.

[2] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[3] Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. *Topic Modeling in Embedding Spaces*. 2019. arXiv: 1907.04907 [cs.IR].

[4] Anton Eklund and Mona Forsman. "Topic Modeling by Clustering Language Model Embeddings: Human Validation on an Industry Dataset". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Ed. by Yunyao Li and Angeliki Lazaridou. Abu Dhabi, UAE: Association for Computational Linguistics, Dec. 2022. DOI: 10.18653/v1/2022.emnlp-industry.65. URL: https://aclanthology.org/2022.emnlp-industry.65.

[5] Maarten Grootendorst. *BERTopic: Neural topic modeling with a class-based TF-IDF procedure*. 2022. arXiv: 2203.05794 [cs.CL].

[6] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[7] Sarojadevi Palani, Prabhu Rajagopal, and Sidharth Pancholi. *T-BERT – Model for Sentiment Analysis of Micro-blogs Integrating Topic Model and BERT*. 2021. arXiv: 2106.01097 [cs.CL].

[8] Nils Reimers et al. *Classification and Clustering of Arguments with Contextualized Word Embeddings*. 2019. arXiv: 1906.09821 [cs.CL].

[9] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL].

[10] Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. *Tired of Topic Models? Clusters of Pretrained Word Embeddings Make for Fast and Good Topics too!* 2020. arXiv: 2004.14914 [cs.CL].

[11] Sebastiano Vascon et al. *Using Dominant Sets for k-NN Prototype Selection*. 2013. DOI: 10.1007/978-3-642-41184-7_14. URL: https://link.springer.com/chapter/10.1007/978-3-642-41184-7_14.