Advanced Machine Learning

Final Project Report

tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection

by
Tomer Segall, ID301833833
Millis Sahar, ID300420379

Here is a link to see this report in it's optimal settings.

Links

- Paper: Article Github
- Final project proposal
- Anchor Paper Report
- Code for Reconstructing Paper's Results
- Code for Fake News Detection using BERT
- Code for Inovative Part
- Final Report (this document)

Datasets

- Microsoft Research Paraphrase Corpus (MRPC/MSRP)
- Augmented MRPC
- Fake News

1. Introduction

1. **Project statement**
    1. **Clearly state the problem/issue/challenge that your final project address**

        Semantic similarity detection is a fundamental task in natural language understanding(NLP).
        Adding topic information has been useful for previous feature-engineered semantic similarity models as well as neural models for other tasks.
        There is currently no standard way of combining topics with pretrained contextual representations such as BERT.
        `The paper` investigates if topic models can further improve BERT's performance for semantic similarity detection.

        The objective is to show that combining topic information with BERT improves performance over various datasets.

        `The inovation` part will also include:

        - Modifying this technique for binary classification on different dataset
        - Using Transfer Learning of BERT models
        - Data Augmentation

2. **Data sources - what are the data set(s) that you will use**
    - Microsoft Research Paraphrase Corpus (MRPC/MSRP)
    - Augmented MRPC
    - Fake News

3. **Evaluation - how are you going to evaluate performance**

    We will follow the article metric and use `Accuracy`.
    Also, while verifiying decreasing losses.

2. Related work

1. **Mention the relevant studies that you found that address a similar objective to your project statement**
    1. **Shortly write about each of those papers (not more than one paragraph for each)**
    2. **State at least two relevant studies**
2. **Please note that failing to mention a highly relevant study is an issue**

- Evolution of Semantic Similarity - A Survey

  This survey traces the evolution of Semantic Similarity Techniques and widely used datasets for semantic similarity. Also, has a detailed description of semantic similarity methods as:

    - Knowledge-based methods
    - Corpus-based methods
    - Deep neural network-based methods
    - Hybrid methods.

- A survey on the techniques, applications, and performance of short text semantic similarity (Additional Link)

  In this survey they have conducted a comprehensive and systematic analysis of semantic similarity. The main focus of the study is:

    - Introduces the theory and processing of text similarity.
    - Reviewed and summarized the techniques of non-DL measures and DL measures.
    - Presents Applications of different semantic similarity measures.
    - Typical methods to performed sentence pair similarity experiments.
    - The performance of models on different datasets.

- Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks
  Introduces, reviewed and evaluated Sentence-BERT(SBERT), which is a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity.

- How transfer learning impacts linguistic knowledge in deep NLP models?
  This study is analyze representations of 3 popular pre-trained models:
  - BERT
  - RoBERTa
  - XLnet

  with respect to semantic knowledge, linguistic information, and how it's redistributed across different layers and individual neurons.

- FakeBERT: Fake news detection in social media

  This study propose a BERT-based deep learning approach (FakeBERT) by combining different parallel blocks of the single-layer CNNs with the Bidirectional Encoder Representations from Transformers (BERT). This method improves the performance of fake news detection with the powerful ability to capture semantic and long-distance dependencies in sentences.

3. Anchor paper

1. **Anchor paper implementation:**
    1. **State the anchor paper.**
       As mentioned before, Our research will be based on the article "tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection"
    2. **Reference external code repositories that you have used so for (if any)**.
       This is its code.
    3. **Summarize your coding effort - what was done, whether you faced technical challenges and why, gaps you still have.**

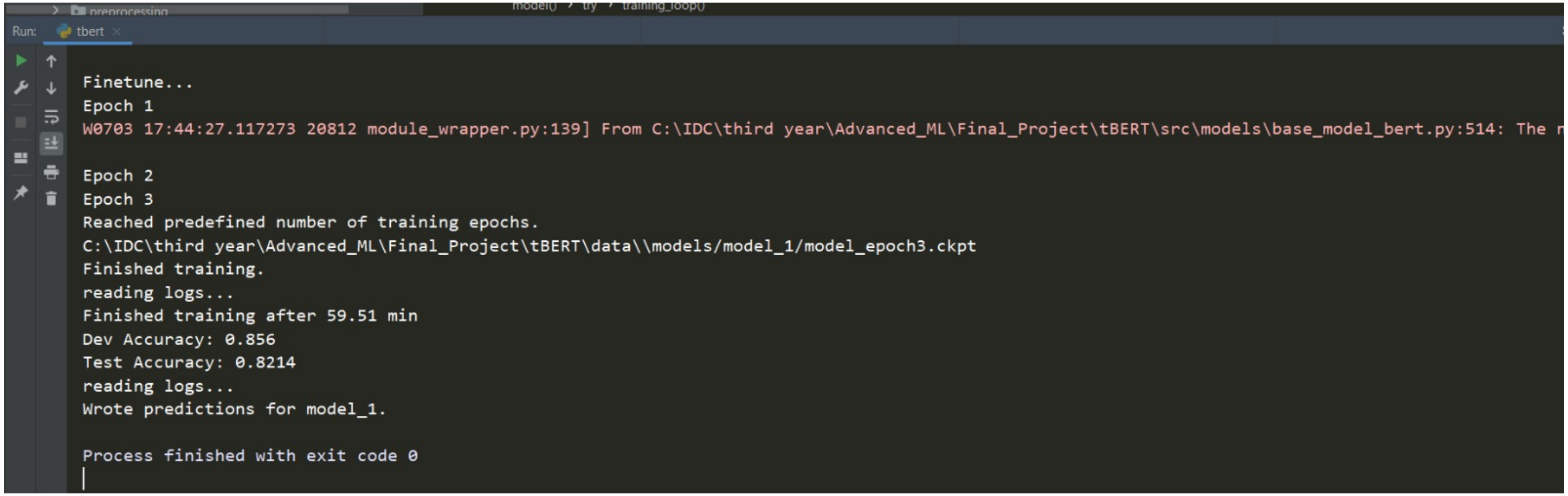
       We were not able to use the aticle's code, due to a very high complexity and "spaghetti code" - So we implemented our own. our code version used Torch framework to create the same method suggested by the paper. While our code version and training process were mush **easier, faster, and more explainable.**

       Due to the article's lack of information regarding all parameters we had to use hyper parameter tuning to mimic the article's results. We needed to run a big number of trial and error to find the optimal hyper-params.

       After that, using this hyper-params, we tried to find the optimal number of topics (t=...) for the LDA in tBERT.

       We were able to achieve the same results as the article. But we also were able to surpasse it.
    4. **Provide a link to your code repository (e.g., a Git repo) where you implemented the anchor paper algorithm.**
       Here is the Code for Reconstructing Paper’s Results.
2. **Anchor paper results**
    1. **Report the results you obtained out of your implementation and show a comparison to the reported results in the anchor paper**
    2. **(optional) Report any additional results you obtained based on another data set or another configuration you have tried**

> `Article's Accuracy Result:` 82.14%



```
Finetune...
Epoch 1
W0703 17:44:27.117273 20812 module_wrapper.py:139] From C:\IDC\third year\Advanced_ML\Final_Project\tBERT\src\models\base_model_bert.py:514: The

Epoch 2
Epoch 3
Reached predefined number of training epochs.
C:\IDC\third year\Advanced_ML\Final_Project\tBERT\data\\models/model_1/model_epoch3.ckpt
Finished training.
reading logs...
Finished training after 59.51 min
Dev Accuracy: 0.856
Test Accuracy: 0.8214
reading logs...
Wrote predictions for model_1.

Process finished with exit code 0
```

> Our implementation: Accuracy of 86.6%

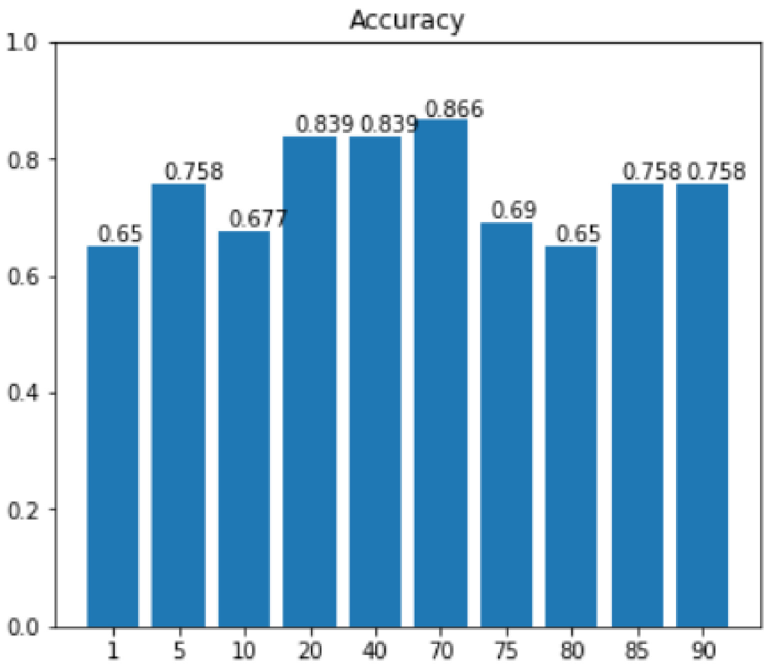
> Due to the article's lack of information regarding all parameters we used hyper parameter tuning to mimic the article's results - and got better ones.

> Parameters that were mentioned:
>
> - Batch size = 32
> - Epochs number = 3
> - Loss function = binary cross entropy
> - Preffered topics number in range of 70-90

> Parameters that were not mentioned:
>
> - Optimizer
> - Random Seeds
> - Learning rate
> - Classification Layers Specs

> We will also show how did we use this method on different tasks, different dataset, and various parameters.

> For example:
> Here are some additinal results, when we tried to use different number of topics to get the optimal results:



4. Innovative part

1. **Restate the problem/issue/challenge**
   We will combine this article approach with `Data Augmentation` and `Transfer Learning` for get better results. And finaly we will use this new approach on a new tasks which will need a huge architecture change to `classify` fake news.
2. **Explain why it is innovative**
   1. **Relate to the anchor paper and the related work that you found and explain how they are different than your work**

      While the article's approach had great results for sementic similairty, we believed we could achieve higher performence when utelizing Augmentation and transfer learning, which the article did not cover.

      Both approaches has shown great boost in results in various NLP research.

      For example, this article `How transfer learning impacts linguistic knowledge in deep NLP models?` displays how powerfull transfer learning is.
      While this article `A Survey of Data Augmentation Approaches for NLP` sets the tone for augmentations, explain the motivation for doing it and shows different approahes on how to augment.
      As you can see these are very recent articles.

      This leads us to believe this is an aspect worth exploring.
3. **Explain your solution**
   Our solution used the same parameters as the original article instructed, but we added two main components:

   > First, we `augmented` the dataset for better generalization.

   > Second, we used `transfer learning`, meaning we focus on BERT versions that would be the most effective for our tasks.

   Both this actions resulted in a better textual embeddings, a better understanding of the context, and a better representation for the meaning of the text.
4. **Implementation**
   1. **Reference external code repositories that you have used so for (if any).**
      - `HuggingFace transformers package` - we used it for transfer learning, BERT versions, etc.
      - `nlpaug package` - we use it to augment our textual data.

2. **Summarize your coding effort - what was done, whether you faced 3. technical challenges and why, gaps you still have.**

   `tBERT for Semantic Similarity`

   1. We had to run the article's code to get a baseline.
   2. We had to write our own implemention for their approach.
   3. We choose a dataset to focus on, did explanatory analysis and augment it.
   4. We devided the dataset to 3 sets - train, validation and test.
   5. We prepered our torch framework objects - Dataset, Dataloader, Trainer, etc.
   6. We trained various BERT versions on the train datasets.
   7. We searched and found the best hyper parameters.
   8. We evaluated the models.

   `tBERT for Classification`

   We did the same work to classify Fake News. But in order to do so we had to change the entire architecture.
   The semantic similarity works with 2 sentences as input, but the classification works only with one.
   As you can see in our code, we had to change a lot in order for this to work well.

3. **Provide a link to your code repository (e.g., a Git repo)**

   Google's colab notebook - Code for Inovative Part

   Warning: running this notebook will take more then 6 hours

5. **Results**

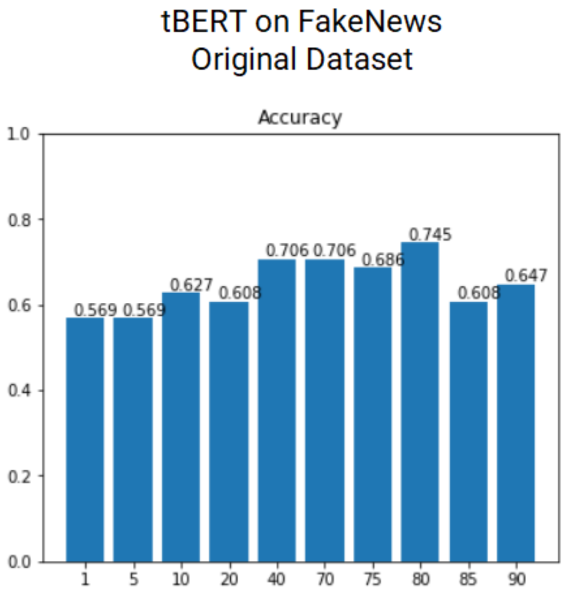
   1. **Report the results you obtained out of your implementation**
   2. **Show a comparison to other methods (e.g., the anchor paper method, methods from related work)**

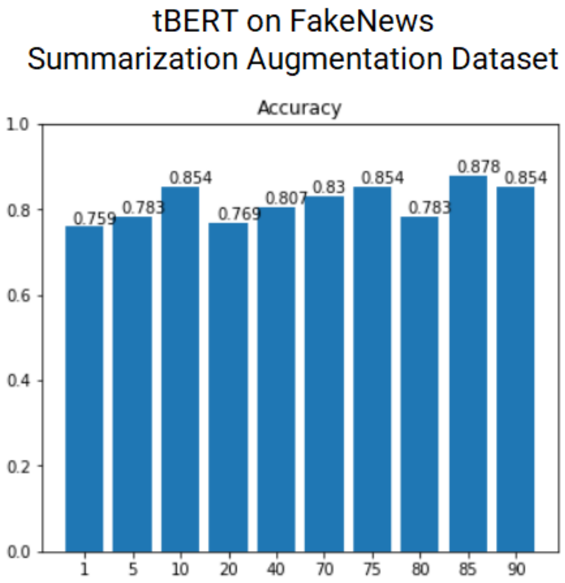
> tBERT on Fake News
> In these plots we can see how powerfull is Augmentations on the FakeNews dataset.
> And that we have to choose the right augmentation technique.

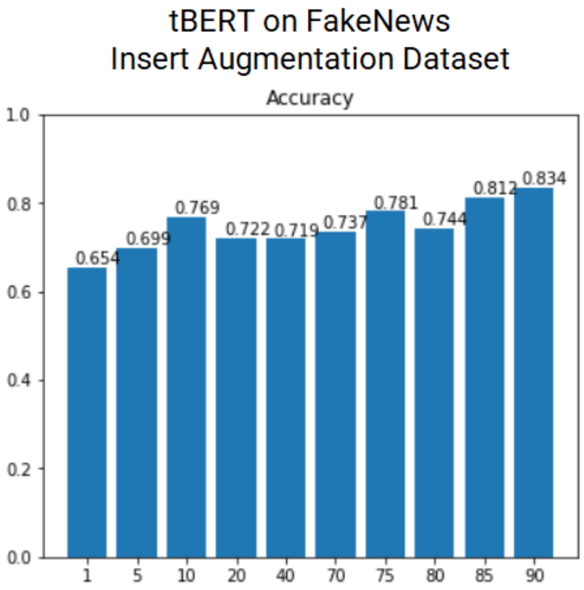
> We can see that the type of augmantation more important then the optimal number of t in tBERT.

> The state of the art results for this task were presented on January 2021, by FakeBERT, with an accuracy of 98.9%, on this paper `FakeBERT: Fake news detection in social media.`

> We did not thought we could perform as well, and our main goal was to implement the tBERT approach on a different dataset and see that it does much better then a original BERT. Which we did in this side colab notebook you can see that tBERT outperformes the original BERT.


tBERT on FakeNews
Original Dataset


tBERT on FakeNews
Summarization Augmentation Dataset

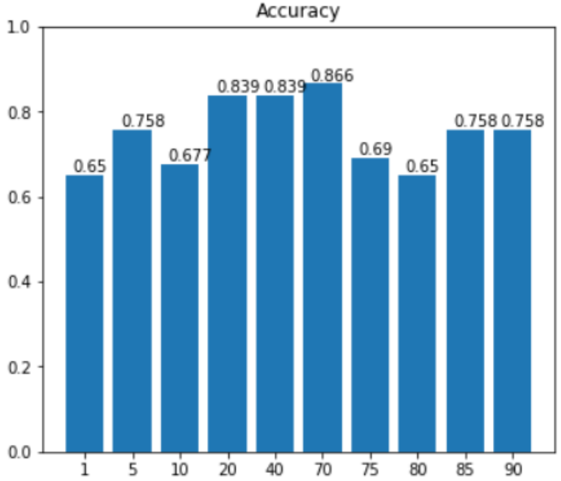

tBERT on FakeNews
Insert Augmentation Dataset

> tBERT on MSRP
> In these plots we can see a search for the optimal number of topic to represent our dataset (This is the optimal value of 't' in tBERT).
> The fact that t=70 were performing well was very noticeable accross all our research.
> (As always, there is a small error margins in these results)

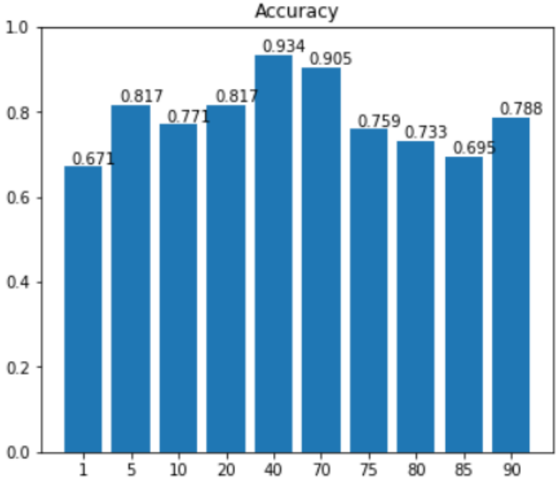
> Then we tried to find the optimal BERT version.
> (Note that those BERT version models were frozen while training)

> It comes as no surprise that the best version of BERT was the version `bert-base-cased-finetuned-mrpc` .
> This is a version of the original BERT that was finetune on the MSRP dataset, therefore it has the best embeddings representation for this textual dataset.

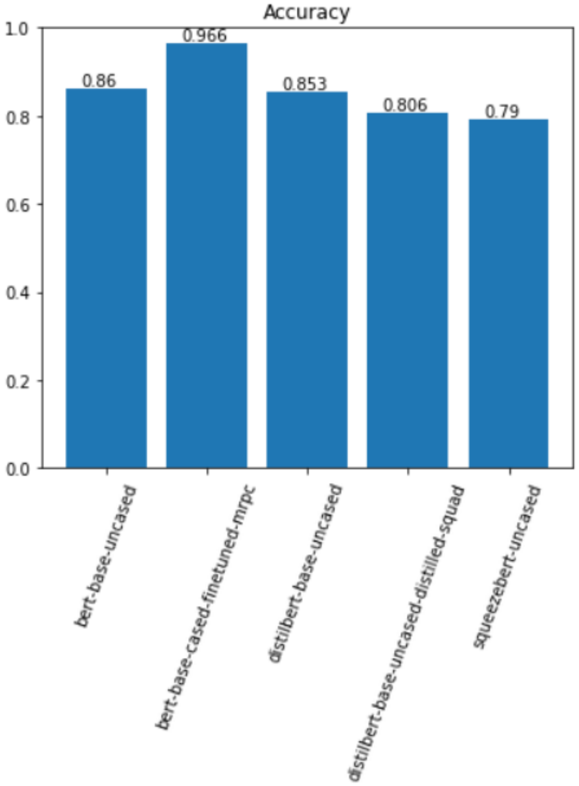
> In addition, we can see very nice results by `bert-base-uncased` and `distilbert-base-uncased` .
> This is amazing that a `distilbert` had the same performence, while being 40% smaller then the original BERT.



tBERT on MSRP
Original Dataset
Searching for optimal t



tBERT on MSRP
Augmentation Dataset
Searching for optimal t



tBERT on MSRP
Augmentation Dataset
Searching for optimal BERT version
t=70

5. Summary

1. **Summarize this effort**

    1. **What was achieved**

        `Anchor paper`

        - Running article tBERT's code.
        - Create our own tBERT's code version.
        - Train, test, and evaluate both tBERT versions on `MSRP` dataset.

        `Inovation - Different Tasks`

        - Change the architecture - input one sentence instead of two.
        - Fake News classification - train, test and evaluate.

        `Inovation - Augmentation`

        - Augment MSRP dataset with various methods.
        - Augment FakeNews dataset with various methods.
        - Train, test, and evaluate various models on the augmented datasets.

        `Inovation - Transfer Learning`

        - Utilize various pre-trained models - distilBERT, finetuneBERT, etc.
        - Train, test, and evaluate these models.

        `Inovation - Outperform Anchor Paper`

        - Use Augmentation & Transfer Learning to get optimal results.
        - Perform hyper parameters optimization search.
        - Achieve better results then anchor paper.

2. **How is it in comparison to the anchor paper and/or other related work**

   While using the same parameters as the anchor paper, we were able to achieve outperform the results in the nchor paper, when utilize known NLP techniques.

   > Anchor Paper Accuracy on MSRP: 82.14%

   > Our Approach Accuracy on MSRP: 96.6%

3. **(scientific) Insights gained**

   > The additinal information(e.g vector) from the TopicModel(e.g LDA) helps BERT successfully achieve higher performence.

   > Augmentation takes a long time, but it's improving model prediction accuracy, adding more training data into the models, preventing data scarcity for better models, and helps the models to generalize better.

   > Transfer Learning is a huge advantage in NLP. If done right, it resulted in simpler training, smaller memory requirements, and considerably shortened target model training.

2. **Open questions and future direction for further research**

   > Future work may focus on:
   >
   > - Investigate if introducing more sophisticated topic models will further improve results, such as `Named Entity`, `Sentiment`, etc.
   > - How to directly induce topic information into BERT without corrupting pretrained information
   > - Whether combining topics with other pretrained contextual models can lead to similar gains

Thanks for reading... :)