Sentiment classification of social media comments using semi-supervised learning

Group 15
Carlos Lago - clago@kth.se
Xuecong Liu - xuecongl@kth.se
Eliott Remmer - eliottr@kth.se
Zhenlin Zhou - zhenlinz@kth.se

March 29, 2022

1 Introduction

The main objective of the project is to develop a model capable of discerning positive and negative comments in SEB social media channels. For this task, we take advantage of transformers, a widely used pre-training technology for natural language processing (NLP). Transformers use a mechanism called Self-Attention to capture the relationship among words. They have shown excellent performance in general language understanding, natural language QA, and generative-adversarial tasks. However, the fine-tuning of the pre-trained model usually requires a large labelled data set.

In real-life scenarios, like ours, there is hardly any labelled data, and the manual annotating of new data is time-consuming and expensive; in contrast, unlabelled examples characterising the target task can be easily collected. This makes it natural to use semi-supervised learning to maximise the utilisation of unlabelled data. Therefore, we adopt the GAN-BERT framework, which can fine-tune BERT-like architectures with unlabelled data.

To understand the reception of SEB advertising campaigns, we obtain SEB-related comment texts from social media, 10% of which are manually labelled. We use a semi-supervised learning framework to train classifiers for sentiment analysis to leverage the abundant unlabelled data. In summary, the aim of the project is to implement and evaluate BERT and GAN-BERT, and generate reproducible model implementations.

2 Related work

Broadly speaking, the field of NLP covers the collective knowledge of linguistics, artificial intelligence and computer science, and use cases include different ways in which computers strive to understand human-written (natural) text. The historical progress of the field can be summarised as an evolution from a symbolic, rule-based approach into a statistical revolution with powerful machine learning methods to push progress forward and finally – into the era of neural NLP with deep neural networks. Among many, applications of NLP include summarization of sentences and paragraphs, named entity recognition, question answering, grammatical error correction, machine translation and sentiment classification. To successfully and efficiently extract the sentiment of text has shown to be a popular use case of NLP models, especially within social media where text and opinions exist in abundance. For many years, Long short-term memory (LSTM) models have been popular but since 2017, these recurrent neural networks (RNN) based models have been replaced by transformer-based networks. Transformer networks use attention layers to emphasize certain parts of the input, thus identifying the context of the words in a sentence. This is done as to not reinvent the wheel for every use case and rather give the model a head start in the race to understand language.

In 2018, Devlin et. al. published a paper [1] describing a new language representation model based on the concept of bidirectional encoder representations from transformers which are known as BERT. This model is a transformer-based model designed to pre-train representations using unlabeled text. A powerful aspect of BERT is that it conditions both left and right context. Using the fact that the model can be pre-trained on large corpora, BERT can easily be used for several NLP tasks by fine-tuning it for specific settings. BERT is constructed as a transformer language model with multiple attention and encoder layers.

One of the techniques that are widely used in NLP is Generative Adversarial Nets, or GANs. The research by Ian J. et al. [2], introduced the concept of GANs, which trains two models at the same time – a generative model G to generate fake samples, and a discriminative model D that predicts whether a sample comes from the real training data or G. This method is very useful to improve the performance of classifiers via semi-supervised learning. In essence, both networks try to optimise an opposing and different objective function, or loss function, in a zero-sum game, which is similar to an actor-critic model.

In 2020, Danilo et al. published a paper [3] based on the BERT model, fused with the idea of GAN, and uses SS-GAN to expand BERT for fine-tuning, reducing the need for labelled samples. Even if there are less than 200 labelled samples, it is possible to obtain the equivalent of fully supervised learning. Their results show that fine-tuning a BERT model with GAN-BERT architecture improved its performance in sentiment analysis.

As an alternative for this task, Kingma, et al. proposed a scalable and accurate deep generative method in a semi-supervised learning task in 2014.[4]. They developed a stochastic variational inference algorithm, enabling scaling to large datasets by joint optimisation of both model and variational parameters. This model shows great results in benchmark classification tasks.

3 Problem description

The aforementioned model that exploits the ideas surrounding transformers is the BERT model, which is pre-trained on a large corpus. Because of the extensive pre-training, the model simply needs to be fine-tuned for the context of the problem – which is beneficial compared to previously used methods in terms of cuts in computational (and environmental) resources. A majority of the comments in the dataset gathered from SEB's facebook channels are in Swedish. Therefore, to classify these comments, the pre-trained BERT model for Swedish ('KB-BERT') can be used as a starting point. This language model was pre-trained on around 15-20GB of Swedish text of different origin [5]. The main rationale of this study is to use the presented semi-supervised learning approaches to avoid the tedious and time-consuming process of manually labelling data.

4 Methodology

In this project, we fine-tuned a Swedish BERT model using GAN-BERT semi-supervised learning, along with another identical model using supervised learning for comparison. This section covers the explanation of how the models are adapted to solving our problem and what methods are used.

4.1 BERT fine-tuning

The motivation behind the use of BERT is that as a bidirectional encoder transformer, it can represent better the context of words, and allows fast fine-tuning, making it perfectly suited for classification tasks.

BERT is fine-tuned by adding a head layer, as seen in Fig. 1, which takes the extracted BERT representation as input and learns to classify the comments. This enables us to fine-tune BERT for our task with a low number of comments and a short training time.

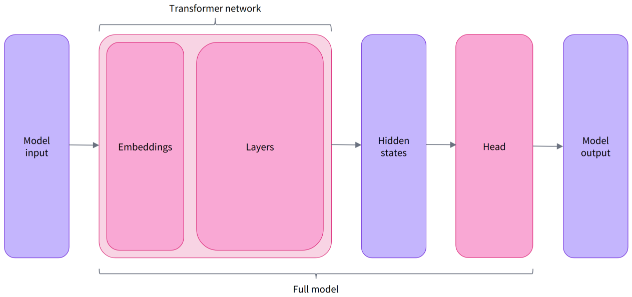

Figure 1: BERT fine-tuning

Apart from fine-tuning BERT with our data, we also test MARMA[1], a BERT model already fine-tuned for Swedish sentiment analysis, as a baseline. It has been trained on 20k App Store reviews. This is useful to identify the performance difference when fine-tuning to our specific task.

4.2 GAN-BERT semi-supervised learning

GAN-BERT algorithm is a semi-supervised learning algorithm, meaning that it can learn from unlabelled data as well as labelled. It is a promising method in this problem, as we have a limited number of labelled data. As shown in Fig. 2, the GAN-BERT architecture fine-tunes BERT for the sentiment analysis task by introducing a discriminator for classification and a fake sample generator to act adversarially.

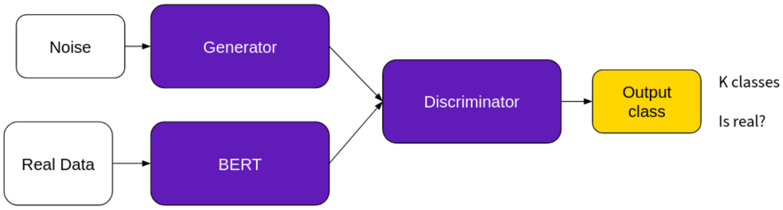

Figure 2: GAN-BERT architecture

The Multi-Layer Perceptron (MLP) generator takes 100-dimensional Gaussian noise with mean 0 and standard deviation 1 as input. Its loss comprises two parts: a) the difference between its generated sample and the sentence embedding from BERT, and b) the logarithm of the probability that the discriminator classifies its sample as fake. This loss encourages the generator to generate samples similar to BERT and believable to the discriminator.

The input for the MLP discriminator is either BERT's sentence embedding of a real sample or a fake one from the generator. Here, the sentence embedding

[1] https://huggingface.co/marma/bert-base-swedish-cased-sentiment

is the vector representation of the classification token [CLS] in the last hidden layer of BERT. The discriminator then predicts whether the sample is fake or belongs to a real class. It gains loss when it classifies a fake sample as real, an unlabelled sample as fake, or misclassifies a labelled sample. The loss propagates back to both the discriminator and BERT, fine-tuning its inner representation.

After training, the generator is discarded while retaining the rest of the architecture. When analysing sentiments, the discriminator picks the real class with the highest probability as the prediction.

5 Implementation

This section on the implementation covers the data processing and the training of the models.

5.1 Data-preprocessing

Before training our model, we used the following procedure and techniques for data labelling and pre-processing.

5.1.1 Data labelling

Given the nature of the semi-supervised learning domain, we needed some labelled data. Therefore, we manually labelled comments as positive, negative or neutral based on our own judgement. Because the comments were in Swedish, the labelling was done by the Swedish speakers in both groups working on the project. We exported a series of comments from the GCP workspace and labelled them in a shared Google sheets document. At first, we did a SQL query selecting a portion of the most recent comments. This was not ideal because it left us with labelled data that was biased towards recent dates. Therefore, in the next query, we selected comment contributions given at random times. Following some general common-sense guidelines for data labelling [2], we set up a series of rules to follow when labelling comments. These rules were to remove comments that were completely irrelevant (bots, product placement etc) and comments that were very sarcastic. Also, questions and comments that consisted of constructive criticism were labelled as neutral (0). We quickly realised that sentiment labelling was a non-trivial task and that a certain dose of subjective judgement was inevitable. Some trends in commenting behaviour were noted, e.g. most positive comments were significantly shorter (consisting of single emojis or <10 characters) than negative and neutral ones.

[2] https://towardsdatascience.com/how-to-label-text-for-sentiment-analysis-good-practises-2dce9e470708

5.1.2 Data Cleaning

For data cleaning, we were inspired by approaches used in previous work on non-English BERT-based sentiment analysis where input data also was collected from social media [6]. The data cleaning process consisted of excluding non-Swedish comments and replacing email addresses and URLs with tags. For language detection, the library langdetect [7] was used which uses Naive Bayes to detect language based on probabilities from features of spelling. The model is trained on data from Wikipedia articles in 49 different languages and has a reported accuracy of 99%. Because of the varied nature of social media comments consisting of emojis, URLs, email addresses, telephone numbers, abbreviations and slang, the threshold for approved comments was raised slightly. This meant not only including Swedish comments but also comments classified as being written in Norwegian, Danish and German. This was done to avoid disregarding too much data. See Figure 3 for the distribution of languages according to langdetect. On the resulting comments, URLs and email addresses were replaced with the tags <url> and <email> using regular expressions substitutions.

Some previous work on using BERT for sentiment analysis on social media comments [6] suggests transforming emojis to text as part of pre-processing. When examining how our tokenizer embeds emojis, we found that several emojis are coded as unknown [UNK] as expected. However, it also keeps the Unicode encoding for several common emojis when tokenizing. Feeding the tokenizer with 562 common emojis[3], the tokenizer was able to encode 87, i.e. 15% of emojis. With this in mind, we decided to not pursue translating emojis into text.

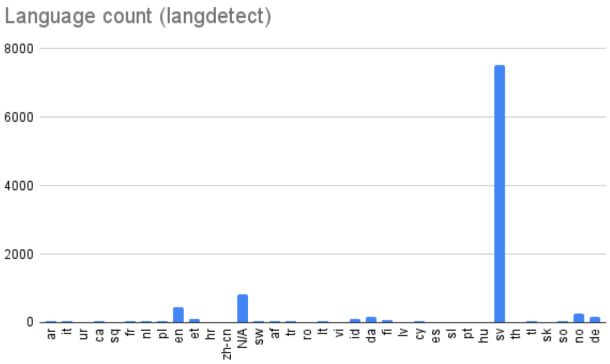

Figure 3: Language of comments in the fb_comments_all dataset

Apart from these techniques, we have also tested the following:

- Removal of stop words, which consists of filtering out words like "the",

[3] https://unicode.org/Public/emoji/14.0

"a", "an", "so", "what". This removes the low-level information to give more focus to the important information.

- lemmatization. It considers the morphological analysis of words to link the words back to their lemma.
- Stemming. It cuts the end or the beginning of the word by detecting common suffixes and suffixes.

5.1.3 Tokenization

To use a pre-trained BERT model, the data needs to be converted into embeddings, which should be able to handle multiple sequences of different input lengths. In order to do this, there are several steps that need to be performed:

- Adding a token [CLS] at the start of the message to represent the message and a [SEP] token at the end of the message.
- Adding padding tokens [PAD]. BERT receives a fixed-length message as input, so it is necessary to add padding tokens up to the required lengths.
- Converting tokens into corresponding IDs. Tokens are converted to unique IDs according to a vocabulary, making use of WordPiece algorithm to break words into common sub-words. If there is an unseen token they are converted into [UNK].

5.2 Training procedure

In order to set up the data for fine-tuning BERT and training the GAN-BERT model, we perform a train/test split with 80/20%. The data distribution is the same for both the train and test sets, with 74% negative and 26% positive samples. Both for the BERT and GAN-BERT, the models are trained with 5 to 10 epochs, using a decaying linear learning rate starting at 5e-5 and a batch size of 8. The other parameters in the GAN-BERT such as the number of hidden layers, dropout rate and activation functions are drawn from the original implementation.

Considering the imbalance in the data, the evaluation metrics used for evaluating the models are accuracy, F1, recall and precision scores. The F1 score, recall and precision can give insights on how well the model is doing with different sentiment classes, the motivation behind these metrics is the following:

- Accuracy: to measure the percentage of the correctly identified cases. It is used for an overview of the model performance.
- Recall: to measure the percentage of the correctly identified positive cases among all the positive cases. It shows how good the model is at identifying the true relevant samples.

- Precision: to measure the percentage of the correctly identified positive cases in all the predictions. It reveals the quality of the prediction.

- F1: combination of precision and recall.

6 Experimental evaluations

In the project, we first tested the performance of a baseline model, MARMA, and fine-tuned a BERT model with supervised learning using different preprocessing techniques. With 5% of the labelled data, the results are shown in Fig. 1. We can observe that with a small number of labelled comments the supervised fine-tuning increases the performance.

Case	Accuracy	F1		Recall		Precision	
		pos	neg	pos	neg	pos	neg
MARMA	0.85	0.73	0.89	0.78	0.87	0.69	0.91
BERT fine-tuned	0.88	0.77	0.92	0.74	0.93	0.81	0.90

Table 1: Evaluation results 5% labelled

After the initial test, we tried different data pre-processing techniques to determine which are better suited for our data. The results of supervised fine-tuning with several methods are shown in Fig. 2. The best results are obtained by removing stop words and performing lemmatization.

Case	Accuracy	F1		Recall		Precision	
		pos	neg	pos	neg	pos	neg
remove stop-words	0.90	0.81	0.94	0.74	0.97	0.89	0.91
stop-words+lemmatization	0.92	0.83	0.94	0.74	0.98	0.94	0.91
stop-words+stemming	0.87	0.73	0.91	0.65	0.95	0.83	0.88

Table 2: Evaluation results 5% labelled with different pre-processing techniques

Following the same procedures after labelling more comments, up to 10% of the dataset, brings the results shown in Fig. 3. The results show that, when using a higher portion of the dataset, the preprocessing doesn't improve the performance of the model.

Case	Accuracy	F1		Recall		Precision	
		pos	neg	pos	neg	pos	neg
MARMA	0.90	0.80	0.93	0.82	0.92	0.79	0.94
BERT fine-tuned (no pre-processing)	0.95	0.90	0.97	0.86	0.98	0.93	0.95
BERT fine-tuned (pre-processing)	0.94	0.89	0.96	0.86	0.97	0.91	0.95

Table 3: Evaluation results 10% labelled

With the supervised fine-tuning results as a baseline, we implement and test GAN-BERT on the same train and test data. Our first trials of GAN-BERT show no improvement compared to the supervised fine-tuned BERT performance. While the loss of both the discriminator and generator networks decreases, the performance seems to be stuck and stops improving. A sample training can be seen in Fig. 4. The removal of stop words and lemmatization is used for pre-processing, as they were beneficial with 5% of labelled data as shown in Table 2.

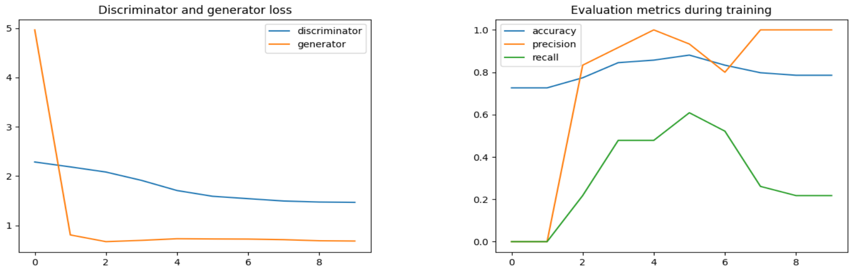

Figure 4: Evaluation metrics during training

In order to explore how the unlabelled data affects the training performance, we have tried using different amounts of data, results shown in Table 4. The results show that when we use a bigger number of the unlabelled data, the performance decreases.

Case	Accuracy	F1		Recall		Precision	
		pos	neg	pos	neg	pos	neg
100	0.90	0.80	0.94	0.70	0.98	0.94	0.90
500	0.89	0.79	0.93	0.74	0.95	0.85	0.91
2000	0.87	0.70	0.92	0.57	0.98	0.93	0.86
All (7258)	0.79	0.36	0.87	0.22	1.00	1.00	0.77

Table 4: Evaluation results 5% labelled with different amount of unlabelled samples

To explain the drop in performance, we first tried training a model to differentiate positive, negative and neutral comments. This is done in order to test if the semi-supervised learning performs better with the neutral label present, as there could be many unlabelled samples that are neither positive nor negative. The performance shows a similar trend, with the performance decreasing more, as can be seen in Table 5.

Case	Accuracy	F1			Recall			Precision		
		pos	neg	neu	pos	neg	neu	pos	neg	neu
0	0.72	0.68	0.73	0.75	0.72	0.70	0.72	0.64	0.76	0.77
500	0.69	0.67	0.75	0.70	0.77	0.65	0.66	0.59	0.88	0.75
2000	0.63	0.47	0.56	0.74	0.40	0.43	0.89	0.56	0.77	0.64

Table 5: Evaluation results 5% for negative/neutral/positive classification

To analyse if the issue with learning from unlabelled samples was the data quality, we tested training GAN-BERT just with 5% of the labelled data, and then adding the other 5% of labelled data as unlabelled, as we know the quality of this data is good. The results show that in this case, the performance increases and the model is able to learn from unlabelled samples, as shown in Table 6. The manually filtered unlabelled data improves the performance while the randomly selected unlabelled data decreases it when using GAN-BERT. This shows that data quality is the key issue here.

Case	Accuracy	F1		Recall		Precision	
		pos	neg	pos	neg	pos	neg
5% labelled	0.90	0.80	0.94	0.70	0.98	0.94	0.90
+ 5% unlabelled (manually selected)	0.93	0.85	0.95	0.74	1.00	1.00	0.91
+ 5% unlabelled	0.89	0.79	0.93	0.74	0.95	0.85	0.91

Table 6: Evaluation results 5% labelled data

7 Summary and discussion

In this project, to analyse the sentiments of the comments on Facebook, we used GAN-BERT architecture to fine-tune a Swedish BERT model in a semi-supervised fashion. In parallel, we tested the performance of MARMA and fine-tuned BERT with supervised learning for comparison. In each of the three settings, we tested different labelled data percentages and data pre-processing techniques.

As mentioned, we decided not to translate emojis into text as a part of the pre-processing pipeline because the tokenizer was able to encode many common emojis. Extending this project, one idea is to investigate further the distribution of emojis in the dataset and make sure that frequent emojis are translated into text that is interpretable by the language model.

Fine-tuning BERT with supervised learning gives very good results even with a few hundred labelled samples, improving the baseline performance by 7%. Semi-supervised learning can be effective, but the unlabelled sample percentage influences the results to some extent. The performance of the model worsened when we use all of the unlabelled data, but improved when we use a small sub-sample. We suspect that this difference results from the quality difference between labelled and unlabelled data.

In regards to data balancing, it would have been interesting to investigate whether the overall data balance was propagated in all comment channels (i.e. Facebook ad post comment, Instagram standard post comment, Facebook visitor post comment etc)

In conclusion, if we were to choose a model to solve this problem, fine-tuning BERT with supervised learning will be the most efficient solution, as its training is fast and the performance is very good with a small number of labelled comments. GAN-BERT could be more beneficial in cases where we have more knowledge about the data quality of the unlabelled data.

8 Collaboration with opposing group

We have collaborated with the other group in the following tasks:

- Data labelling: the Swedish speakers from both groups manually labelled ~ 1500 comments as positive, negative or neutral.
- Data processing: structuring of the dataset into labelled and unlabelled comments for training, and discussion about pre-processing techniques.
- Experiments architectures/methods: we analysed together with the architectures and different techniques.

We have also had meetings together, to check on the status of the project and organise the WARA presentation we did together, introducing the problem and showcasing results and findings of each group.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[3] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, 2020.

[4] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[5] Martin Malmsten, Love Börjeson, and Chris Haffenden. Playing with words at the national library of sweden – making a swedish bert, 2020.

[6] Marco Pota, Mirko Ventura, Rosario Catelli, and Massimo Esposito. An effective bert-based pipeline for twitter sentiment analysis: A case study in italian. *Sensors*, 21(1), 2021.

[7] Nakatani Shuyo. Language detection library for java, 2010.