

# Deep Learning Brasil - NLP at SemEval-2020 Task 9: Sentiment Analysis of Code-Mixed Tweets Using Ensemble of Language Models

**Manoel Veríssimo dos Santos Neto**      **Ayrton Denner da Silva Amaral**  
INF - Federal University of Goiás      INF - Federal University of Goiás  
verissimo.manoel@gmail.com      ayrtondenner2013@gmail.com

**Nádia F. F. da Silva**      **Anderson da Silva Soares**  
INF - Federal University of Goiás      INF - Federal University of Goiás  
nadia@inf.ufg.br      anderson@inf.ufg.br

## Abstract

In this paper, we describe a methodology to predict sentiment in code-mixed tweets (hindi-english). Our team called **verissimo.manoel** in CodaLab<sup>1</sup> developed an approach based on an ensemble of four models (MultiFiT, BERT, ALBERT, and XLNET). The final classification algorithm was an ensemble of some predictions of all softmax values from these four models. This architecture was used and evaluated in the context of the SemEval 2020 challenge (task 9), and our system got **72.7%** on the F1 score.

## 1 Introduction

It is a common tendency among multilingual people who are non-native English speakers to code-mix in their speech using English-based phonetic typing. This linguistic phenomenon, particularly in social media like Twitter<sup>2</sup>, poses a great challenge to the conventional Natural Language Processing (NLP) study area.

Within the context of the Sentiment Analysis, the study of the phenomenon of code-mixed language is important to the research community because this behavior is more common today. The interest in this area has grown due to the volume of data that social networks generate, and also by the value that this information has to understand people opinions when they are expressed in written texts.

In this paper, we explain our methodology to predict sentiment in tweets, describing how our method is based on a combination of the latest language models, and also how such models contributed to a great advance in this task. This configuration was employed and evaluated in the SemEval 2020 challenge (task 9), in which the goal is to predict the sentiment in code-mixed texts written in English and Hindi languages of a tweet (Patwa et al., 2020). The models used in this combination are: MultiFiT (Eisenschlos et al., 2019) that an evolution of ULMFiT (Howard and Ruder, 2018), BERT (Devlin et al., 2018), ALBERT (Lan et al., 2019) and XLNet (Yang et al., 2019b).

This work is organized as follows: Section 2 explains some related works, Section 4 describes the dataset used, Section 3 addresses the methodology applied in the task, Section 5 presents the results, and finally Section 6 expose our final considerations as well as possible future works.

## 2 Related Works

Sentiment Analysis in Twitter has been considered as a very important task from various academic and commercial perspectives. Many companies use the data on Twitter to decide about marketing and business decisions.

A challenge is to apply Sentiment Analysis in texts written in two different languages like English and Hindi. This happens because some people around the world speak two or more languages and sometimes when these people write texts they write it in more than one language.

<sup>1</sup><https://competitions.codalab.org/competitions/20654>

<sup>2</sup><https://twitter.com/>

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>.

The author (Sarkar, 2018) present their work for Sentiment Analysis for Indian Languages (SAIL) (code mixed). They implemented an algorithm using Multinomial Naïve Bayes trained using n-gram and SentiWordnet features.

For (Bhargava et al., 2016), users tend to express their thoughts by mixing words from multiple languages, because in most of the time, they are comfortable in their native language, and mixed languages are common when users write in social media. They divide their technique into two stages, viz Language Identification, and Sentiment Mining Approach. They evaluated their results and compared to a baseline obtained from machine-translated sentences in English, and found it to be around 8% better in terms of precision.

For (Ghosh et al., 2017) is very important a preprocessing step to remove noise from raw text. The authors developed a Multilayer Perceptron to determine sentiment polarity in code-mixed social media text from Facebook. Such example using texts from Facebook is important to show that this phenomenon can be applied in different social media than the one used on SemEval 2020.

(Patra et al., 2018) cites some data about the Census in India, showing the existence of 22 scheduled languages and 462 million users on the internet. To express their feelings, such users probably use more than one language to write text on social media.

In order to help NLP researchers, a corpus was created by (Swami et al., 2018) using English-Hindi code-mixed of tweets marked for the presence of sarcasm and irony where each token is also annotated with a language tag. In this present work, this corpus was used to training MultiFiT model.

### 3 Methodology

The methodology applied in this task consists of training and using prediction values of four models: MultiFiT, BERT, ALBERT, and XLNet. After retrieving prediction values, our ensemble calculates an average of all softmax values from these four models, as shown in Figure 1. Models BERT, ALBERT, and XLNet were trained on a DGX-1, while MultiFiT was trained on a GTX 1070 Ti 8GB. The hyperparameters of the four models are described in Table 1.

Model	Batch Size	Learning Rate	Max Length	Optimizer	Training	Base Model
MultiFiT	32	1e-3			20 epochs	
BERT	8	2e-5	128	AdamW	30 epochs	BERT-Base, Multilingual Cased
ALBERT	16	2e-5	256	AdamW	30 epochs	Xxlarge
XLNet	16	2e-5	256		8000 steps	XLNetLarge, Cased

Table 1: Hyperparameters



Figure 1: Solution Architecture.

#### 3.1 Preprocessing

This step consists in eliminating noises and terms that have no semantic significance in the sentiment prediction. For this, we perform the removal of links, removal of numbers, removal of special characters, and transform text in lowercase.

### 3.2 MultiFiT

Nowadays, there are many advances in NLP, but the majority of researches is based on the English language, and those advances can be slow to transfer beyond English.

The **MultiFiT** (Eisenschlos et al., 2019) method is based on **Universal Language Model Fine-tuning (ULMFiT)** (Howard and Ruder, 2018) and the goal of this model is to make it more efficient for modeling languages others than English.

There are two changes compared to the old model: it utilizes tokenization based on sub-words rather than words, and it also uses a QRNN (Bradbury et al., 2016) rather than an LSTM. The model architecture can be seen in Figure 2.

The architecture of the model consists of a subword embedding layer, four QRNN layers, an aggregation layer, and two linear layers. In special this architecture, subword tokenization has two very important properties:

- Subwords more easily represent inflections and this includes common prefixes and suffixes. For morphologically rich languages this is well-suited.
- It is a common problem out-of-vocabulary tokens and Subword tokenization is a good solution to prevent this problem.

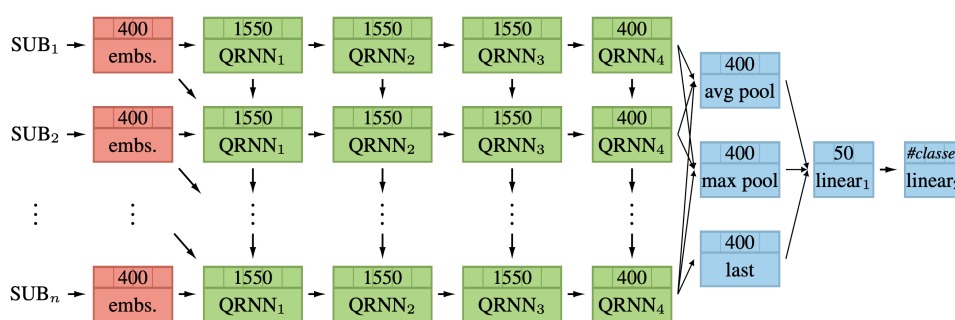


Figure 2: MultiFiT Architecture.<sup>3</sup>

### 3.3 BERT

**Bidirectional Encoder Representations from Transformers** (also abbreviated as **BERT**) (Devlin et al., 2018) is a model designed to pre-train deep bidirectional representations from unlabeled data. The pre-trained BERT model can be fine-tuned with just one single additional output layer, which can be used in sentiment analysis and others NLP tasks.

The implementation of BERT there has two steps: pre-training and fine-tuning. In the pre-training step, the model is trained on unlabeled data over different pre-training tasks using a corpus in a specific language or in multiples corpus with different languages. For the fine-tuning step, the BERT model is first initialized with the pre-trained parameters, and all of the parameters are fine-tuned using labeled data from the specific tasks.

Dataset repositories like NLP-progress<sup>4</sup> track different model results and progress in many Natural Language Processing (NLP) benchmarks, and also the current state for the most common NLP tasks. When doing a comparison between results available for reference in such repositories, BERT was able to achieve state-of-the-art in many NLP-related tasks, which gives an excellent reason to use BERT in our architecture, even while many reasons of BERT state-of-art performance are not fully understood (Kovaleva et al., 2019) (Clark et al., 2019).

<sup>3</sup><https://nlp.fast.ai/classification/2019/09/10/multifit.html>

<sup>4</sup>[http://nlpprogress.com/english/sentiment\\_analysis.html](http://nlpprogress.com/english/sentiment_analysis.html)

### 3.4 ALBERT

Recent language models had shown a tendency to increase in size and quantity of parameters for training. They often offer many improvements in many NLP tasks, but they suffer as a consequence of the need for many hours of training, which consequently increases its costs of operation. **ALBERT** (Lan et al., 2019): **A Lite BERT for Self-supervised Learning of Language Representations**, offers an alternative of parameters reduction to solve this problem.

There are two changes to reduce the size of the model based on BERT. The first is a factorized embedding parameterization, this decomposing the large vocabulary embedding matrix into two small matrices. This decomposition approach reduces the trainable parameters and reduces a significant time during the training phase. The second change is the share parameter cross-layer, which also prevents the parameter from growing with the depth of the network.

### 3.5 XLNet

**XLNet** (Yang et al., 2019a) is a model that uses a bidirectional learning mechanism, doing that as an alternative to word corruption via masks implemented by **BERT**. XLNet uses a permutation operation over tokens in the same input sequence, being able to use a single phrase through different training steps while providing different examples. Phrase permutation in training fixes token position, but iterating in every token in training phrases, rendering the model able to deal with information gathered from tokens and its positions in a given phrase. XLNet also draws inspiration from **Transformer-XL** (Dai et al., 2019), relying specially in pre-training ideas.

## 4 Dataset and Task

**Dataset.** The data for the task consists of 17,000 tweets for training and 3,000 for test/evaluation. The data format is in CONLL format. The amount of tweets for the dataset can be seen in Figure 3.

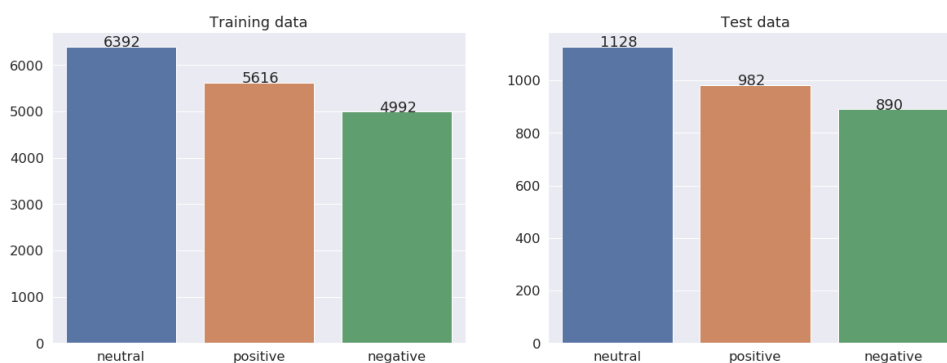


Figure 3: Number of labels per classes.

**Task details.** The task objective is to predict the sentiment of a given code-mixed tweet. The sentiment labels are positive, negative, or neutral, and the code-mixed languages will be English-Hindi. The challenge is to predict the sentiment in texts written in these two languages (Patwa et al., 2020).

Here are some examples of sentences taken from the dataset.

#### Positive Sentence

*@AmitShah @narendramodi All India me nrc lagu kare w Kashmir se dhara 370ko khatam kare ham Indian ko apse yahi umid hai*

#### Negative Sentence

*@RahulGandhi television media congress ke liye nhi h . Ye toh aapko pata chal hi gya hoga . Achha hoga ki Congress ke ... <https://t.co/HmH8M7PTaK>*

## Neutral Sentence

*@sardanarohit jaaz saab ko salo saal ke pending case ko soultion me maza nahi aata \* inko to public paise monthly case miljaye \**

## 5 Results

In this section, we report the obtained results by our model according to the metric evaluation used by the challenge: macro f1, precision and recall, accuracy, and f1 for all classes. Results are reported for each model and an ensemble using a combination of results of the four models XLNet, BERT, ALBERT, and MultiFiT. In Table 2 we show model’s performances and in Table 3 we present the F1 score per class.

Model	F1	P	R	Acc
<b>Ensemble</b>	<b>0.727</b>	<b>0.729</b>	<b>0.726</b>	<b>0.723</b>
XLNet	0.679	0.696	0.692	0.690
ALBERT	0.679	0.684	0.676	0.675
BERT	0.675	0.680	0.672	0.670
MultiFiT	0.665	0.665	0.669	0.662

Table 2: Result Semeval-2020.




	Class	Acc
	Negative	0.671
	Positive	<b>0.760</b>
	Neutral	0.606

Table 3: Ensemble f1 by class.

The obtained results on the testing data indicate that our ensemble produces the best F1; on the other hand, the XLNet model represents the best result among the other models. It is important to quote that the final ensemble is a large combination of results of the four models used in this architecture.

For the official<sup>5</sup> results in competition, the organizers used only the first three submissions and in our case, our models were only MultiFiT and BERT. Using only these architectures our results are only **66.5%**.

## 6 Conclusion

In this paper, we propose a combination of four models for the Semeval 2020 (task 9), and our team got **72.7%** on F1 score in the competition. All of these models are based on using language models and transfer learning. They alone performed well, but together in an ensemble combination, they performed even better.

In some applications, it is difficult to use an ensemble consisted of four models, especially because of the overhead coming from time spent on inference, culminating in an approach that sometimes will not perform well. On the other hand, the individual results of these four models are very close, meaning that for this task, any model can be used.

It is important to note that MultiFit has the worst result, but the difference is very small, and this specific model takes a lot less time to train, being the lightest model of the ensemble.

As future works, we intend to explore these models for Sentiment Analysis in other multilingual and monolingual scenarios.

<sup>5</sup>[https://competitions.codalab.org/competitions/20654#learn\\_the\\_details-results](https://competitions.codalab.org/competitions/20654#learn_the_details-results)

## References

- R. Bhargava, Y. Sharma, and S. Sharma. 2016. Sentiment analysis for mixed script indic sentences. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 524–529, Sep.
- James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2016. Quasi-recurrent neural networks.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT’s attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August. Association for Computational Linguistics.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.
- Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kardas, Sylvain Gugger, and Jeremy Howard. 2019. Multifit: Efficient multi-lingual language model fine-tuning.
- Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China, November. Association for Computational Linguistics.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.
- Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail\_code-mixed shared task @icon-2017.
- Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, December. Association for Computational Linguistics.
- Kamal Sarkar. 2018. Ju\_ks@sail\_codemixed-2017: Sentiment analysis for indian code mixed social media texts.
- Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection.
- Z Yang, Z Dai, Y Yang, JG Carbonell, R Salakhutdinov, and QV Le. 2019a. Xlnet: generalized autoregressive pretraining for language understanding. corr abs/1906.08237 (2019). URL: <http://arxiv.org/abs/1906.08237>.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding.