Cross-lingual Natural Language Inference - XNLI

Ahmed ElSheikh - 1873337

1 Introduction

Natural Language Inference (NLI) is a sequence pair classification task, which tells the logical relationship between 2 sentences whether it is entailment or contradiction or neutral[1]

Languages can be split into High resources languages and Low resources languages. High resources languages are languages which have many existing data resources[2] which allows the building of Deep Learning (DL) systems for them, English is by far the most well resourced language. On the other hand, low resources languages have very few -if any- resources available (e.g. Urdu/Swahili). Languages resources are costly to produce, hence, it would be better if there is a way to produces a cross-lingual Language Understanding (XLU) and NLI can be considered one of the fundamental blocks of this task.

XLU is a task where system is trained on a language (e.g. English) and evaluated on other languages (e.g. Italian/Arabic/Chinese). Which falls under the umbrella of few-shots learning, specifically, Zero-Shot Learning, model is trained on a language and test on another(s).

2 Dataset Preparation

Datasets used in this task are Multi-Genre NLI corpus (MNLI) -for training and validation- and Cross lingual NLI corpus (XNLI) -for testing, which is an extension of NLI corpora to 15 languages spanning over different languages' families, including both low and high resources languages (Facebook AI, 2018), refer to fig 1 for languages distribution. Further insights can be found in Appendix A

Datasets did not suffer from classes imbalance, labels were shared equally across all dataset instances, in the datasets as shown in the following figures 3-5. Consequently, no methods of oversampling or under-sampling were carried out. Datasets were tokenized either using the pre-trained language models' tokenizers, or using nltk regex tokenizer. All words were lowered to reduce vocabulary size. Sequences were trimmed/padded to max length of 128 tokens due to the following figures 7 - 12.

[1]For examples, check examples table 1.

[2]Data resources can be found in the form of high amounts of multi genre raw text, and lexical, syntactic and semantic resources, and task specific resources.

3 Network Architectures

3.1 Baseline Model

Bidirectional Long Short Term Memory (BiLSTM) networks (Graves and Schmidhuber, 2005), extension of LSTM networks, BiLSTMs can be trained to get context on the left and right of a tokenposition. BiLSTMs have shown tremendous success in sequence learning problems in a variety of NLP tasks. This model was developed as the baseline of the experiments done within this project's scope. For model's architecture refer to figure 14

3.2 K Model

K Model is an improvement over baseline model; as it is made up of a BiLSTM instead of 2 BiLSTMs network. This model was molded by the data processing approach in sequence pair classification tasks (e.g. XNLI task). The process was modified from passing unpaired sentences to paired premises-hypotheses sentences seperated by a special token, inspired from BERT (Devlin et al., 2018) model preprocessing technique where unpaired sequences are passed as follows '[CLS] tokens [SEP]', and sequence pair passed as '[CLS] sentence A [SEP] sentence B [SEP]'. Model's network is shown in figure 15

3.3 Pretrained Language Models

The adoption of transfer learning in the field of NLP lead to multiple breakthroughs in different NLP tasks. It can be thought of as the ability to

train a model on a dataset, then use this model to perform different NLP tasks on different data sets.

Pretrained models can be utilized in one of 2 ways. Firstly, features extraction: The network is used as a fixed feature extractor for the new dataset (e.g. ElMo), secondly, fine tuning: introduces minimal task specific parameters and is trained on downstream tasks by simply fine tuning of all pretrained paramerters, or, some parameters can be frozen, and fine tune the rest of parameters. (Devlin et al., 2018)

Most of the pretrained language models are based on transformer architecture which utilizes self attention, hence, modeling relations between different words in the sequence regardless of the tokens' positions.

3.3.1 BERT

BERT model is a transformer based pretrained language model that utilizes both directons (left and right) to learn general language representations, on the other hand, ElMo is a unidirectional language model, consequently, it can be considered a sub-optimal for sentence level tasks where it is crucial to incorporate text in both directions to fetch a better contextual represenetation of the sentence. BERT as a bidirectional language model uses Masked Language Modeling (MLM) training objective, which randomly masks 15% of the tokens per sentence and try to predict them. this objective was paired with Next Sentence Prediction (NSP) which jointly pretrains sequence pair representations. Sequence/sentence pair classification task it relies on the understanding of the relationship between 2 sentences which might not be captured by language modeling which is addressed by NSP.

BERT utilizes $30K$ tokens as its vocabulary, its tokenizer is based on a WordPiece model, bridges between Byte Pair Encoding (BPE) and unigrams, BPE segments words into subwords level, hence vocabulary contains [Words, front occurring SubWords, middle occurring subwords , Individual Characters]. [3]

BERT was trained only on English sentences, which will not be of use in XNLI task, hence, a variant of it is used which is "multilingual-BERT", mBERT gained a lot of popularity as a contextual representation for various multilingual tasks as it is trained to provide sentence representations in 104 languages. The vocabulary was shared across languages -which helps in aligning of contextual embeddings of subwords in the same shared embeddings space- with $\approx 119K$ tokens, tokenization was frequency based to extract potential merging of tokens but fina1 decision is based on the likelihood of the merged token.

3.3.2 XLM

Cross Lingual Model (XLM) (G Lample and A. Conneau, 2019) Transformer based language model based on BERT's architecture, however, it was trained with an extra training objective which is Translated Language Modeling (TLM), TLM is an extension of MLM, using parallel data and masking random tokens in both source & target sentences, for example to predict a word in a source sentence, model is forced to leverage the target sentence's context if the source's context doesn't suffice. Refer to figure 17 for an example.

XLM uses shared subword vocabulary across all languages using BPE, which greatly improve the alignment of embeddings space across languages that share same alphabet or same anchor tokens. BPE helps alleviate the bias towards high level language resources and prevents words from low resources languages to be split into characters.[4]

3.3.3 XLM-R

XLM-R stands for XLM-RoBERTa, XLM-R abstains from training the XLM model with TLM objective, but trains a RoBERTa model on unlabeled data across 100 languages extracted from CommonCrawl datasets (2.5TB of data). (Facebook AI Research, 2020) RoBERTa is short for Robustly optimized BERT Pretraining Approach (Facebook AI and University of Washington, 2019), it is based on BERT but with a modified set of parameters, dropping the NSP training objective, as well as training on much larger batches, and learning rates on more data for longer times, and instead of using static masking (MLM) it uses Dynamic masking technique in its MLM training objective

3.4 Training

All the models were trained using GPUs with different batch sizes, baseline & K_Model were using Adam optimizer with default learning rate (lr=0.001) for 5 epochs, sequences were padded to

[3] Subwords are n-grams of the word. e.g. for the word "Embeddings" Front occurring can be "em" or "emb" back occurring can be "##ing" or "##bed"

[4] which is an improvement over BERT vocabulary building technique

max length(128) before passing them to the models. While pretrained models (mBERT, XLM, XLM-R) were finetuned using AdamW optimizer[5](Ilya Loshchilov & Frank Hutter, 2019) for 3 epochs refer to table 7 for optimizers used and their learning rates. Refer to table 8 for more insights about training resources, training time, GPUs , batch sizes used.

Gradients Clipping was deployed with clipping value of 1.0 to alleviate the problem of exploding gradients LSTMs usually face. As well as, deploying Dropout layer after the embeddings layer before LSTM layer or after pretrained model preventing model's dependency on certain tokens while learning; as Dropout Training (Y. Gal, Z. Ghahramani) drop words randomly from input sequences while training.

4 Results

4.1 Experiment 1 & 2

In this experiment, both the Baseline model and the K-model were trained for 5 epochs each with the same set of hyperparameters as in table 5. Both models failed to pass the random selection results (33.3%). Due to the fact that the models were not supplied with pretrained aligned multilingual word embeddings and the model has to train its own embeddings which was not successful as to no parallel data was provided, hence, our models only learned embeddings for English words, so when models were evaluated on the testing dataset, most of the sentences' tokens were considered as OOVs. Yet another reason to mention is that, when models were supplied with multilingual embeddings provided by fasttext embeddings, both models also failed to achieve better results due to the fact that the embeddings were not aligned in the same shared space, hence, our models were not capable of understanding that "Home" in English is same as "Casa" in Italian nor 'Hogar' in Spanish.

K-model's advantage over the baseline model was the reduced number of parameters going from $\approx 57M$ parameters to $\approx 2.5M$ parameters, consequently, training time was reduced significantly.

4.2 Experiment 3

Using mBERT for XNLI task yielded accuracy $\approx 68\%$ which is more than double of the results achieved by both baseline model and K model, due to the fact that it a pretrained transformer based language model on larger amounts of data with different training objectives, also, mBERT vocab is built and shared across 104 languages with $\approx 119K$ wordpiece generated tokens compared to $\approx 89K$ tokens extracted using Regex Tokenizer, and due to the fact the vocabulary is shared across languages, consequently, embeddings were better aligned in the same embeddings space compared to monolingual unaligned word embeddings trained from the previous 2 experiments[6].

4.3 Experiment 4

XLM model achieved an accuracy of $\approx 70\%$ when using MLM objective only, and achieved $\approx 72\%$ when using MLM & TLM as training objectives. This change in performance between XLM 2 variants is due to the model trained with TLM objective is forced to learn similar representations for the same sentence in different languages. As well as, that XLM uses a better initialization technique for its word embeddings, which is taken from MLM to be fed into the translation model, allowing the model a better learning capability. Hence the improved performance compared to model only using MLM objective

4.4 Experiment 5

XLM model achieved $\approx 72\%$ when using MLM & TLM as training objectives, Which outperforms mBERT with 68% this is due to that XLM is trained on $\approx 223M$ sentences of parallel data which is relatively larger than the data mBERT used for pretraining, added to the XLM model requires languages of tokens as an extra metadata, helping it to learn the relations between tokens in different languages. Another reason for this increase in performance is that XLM model is trained with 2 objectives (MLM & TLM) while mBERT is only trained on MLM

4.5 Experiment 6

XLM-R model achieved $\approx$ 79% outperforming XLM model as it is self-supervised trained RoBERTa model on 2.5TB of data, while XLM requires parallel data which might not be available on a sufficient scale. As well as, upsampling low resources language, XLM-R vocabulary size which is $5\times$ XLM vocabulary size.

[5]Same as Adam optimizer but with weight decay is decoupled from optimization step, allowing optimization for Learning Rate & Weight decay seperately, this improves the overall model generalization capability

[6]Fasttext Monolinugal and Multilingual pretrained embeddings were used, but no alignment of word vectors was done

References

XNLI: Evaluating Cross-lingual Sentence Representations, Facebook AI and New York University

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks, 18(5-6):602–610.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Cross-lingual Language Model Pretraining, XLM, from Facebook AI Research

Unsupervised Cross-lingual Representation Learning at Scale, XLM-R, Facebook AI Research Team

RoBERTa: A Robustly Optimized BERT Pretraining Approach, Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike, Lewis Luke Zettlemoyer, Veselin Stoyano

DECOUPLED WEIGHT DECAY REGULARIZATION, Ilya Loshchilov & Frank Hutter, 2019

A Theoretically Grounded Application of Dropout in Recurrent Neural Networks, Yarin Gal, Zoubin Ghahramani

A Data Analysis

Premises	Hypotheses	Relation
I am holding a gun	I am eating a fruit	Neutral
Titanic sank in the Pacific Ocean	Titanic sank in the Atlantic Ocean	Contradiction
Soccer game with multiple males playing	Men plays soccer	Entailment

Table 1: Sequence Pair Examples

Showing examples of premises-hypotheses and their corresponding labels

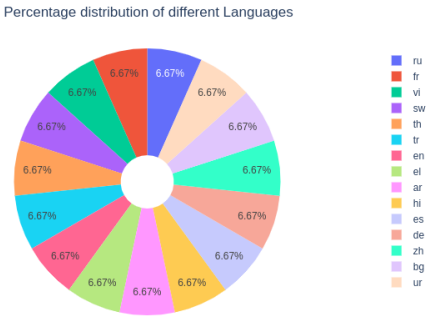
Language	Code
Arabic	ar
Bulgarian	bg
Deutsch	de
Greek	el
English	en
Spanish	es
French	fr
Hindi	hi
Russian	ru
Swahili	sw
Thai	th
Turkish	tr
Vietnamese	vi
Chinese	zh

Table 2: XNLI corpus languages

Dataset	Number of Sentences	Language
MNLI - Train	392,702	English
MNLI - Dev	9,815	English
XNLI - Test	75,150	15 Languages

Table 3: Number of sentences per dataset

MNLI dataset is a multi-genre English based dataset made up of $392K$ premises-hypotheses pairs for training data, $9K$ pairs for validation dataset, and $75K$ pairs in different languages from cross lingual dataset XNLI



Sentences equally distributed across all languages. (5010 sentences per language, totalling 75,150 sentences)

Figure 1: Percentage distribution of languages in Test datasets

Figure 2: Distribution of languages across the datasets

It is obvious that English is the dominating language in the training dataset, meanwhile, sentences are equally distributed in the testing dataset

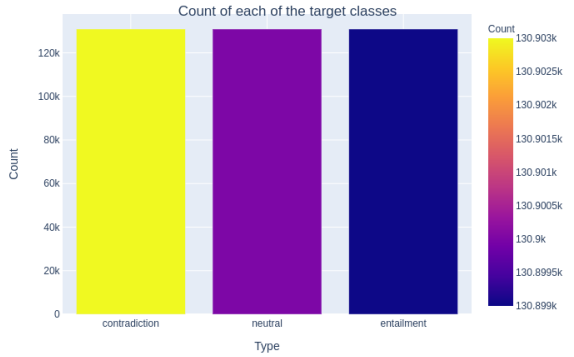


Figure 3: Distribution of labels across the train dataset

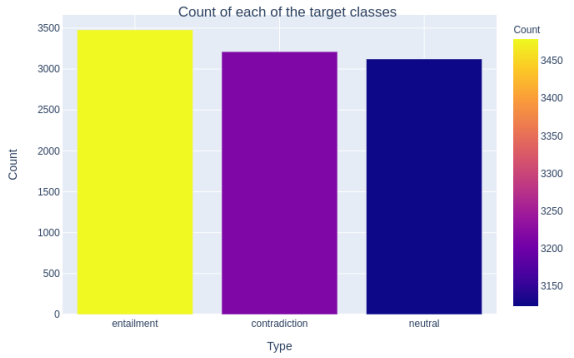


Figure 4: Distribution of labels across the dev dataset

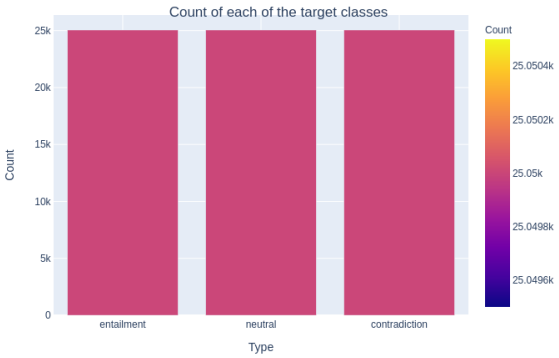


Figure 5: Distribution of labels across the test dataset

Label	Neutral	Entailment	Contradict
Dataset	Number of labels		
MNLI (Train)	130,900	130,899	130,903
MNLI (Dev)	3,123	3,479	3,213
XNLI (Test)	25,050	25,050	25,050

Table 4: Labels Distribution table

Summary of labels distribution across 3 datasets in numbers

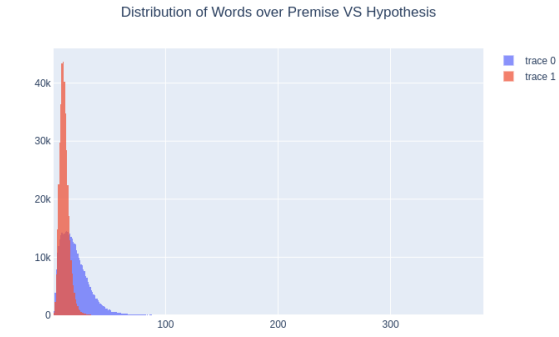


Figure 6: Distribution of Words over Premise VS Hypothesis

Distribution of premises words fall within the range of hypothesis.

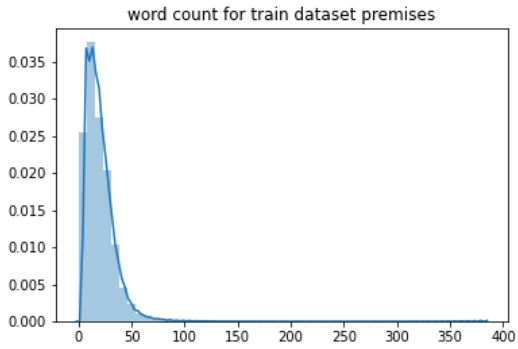


Figure 7: Train dataset premises Words Counts

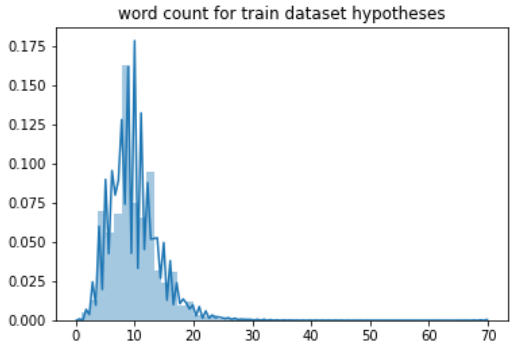


Figure 8: Train dataset hypotheses Words Counts

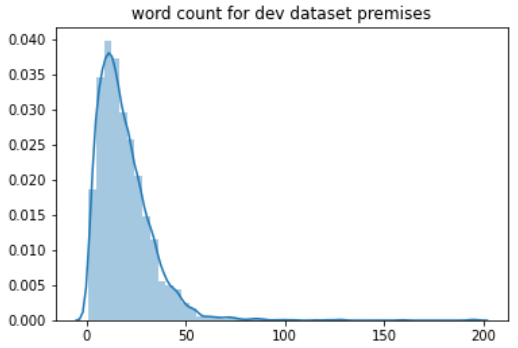


Figure 9: Dev dataset premises Words Counts

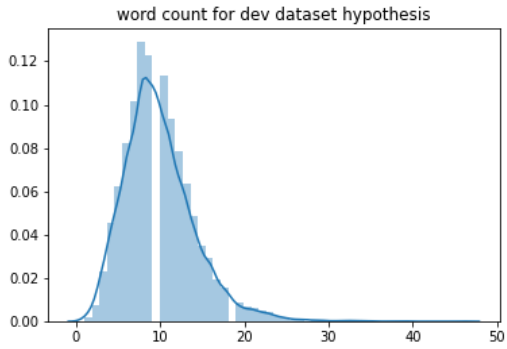


Figure 10: Dev dataset hypotheses Words Counts

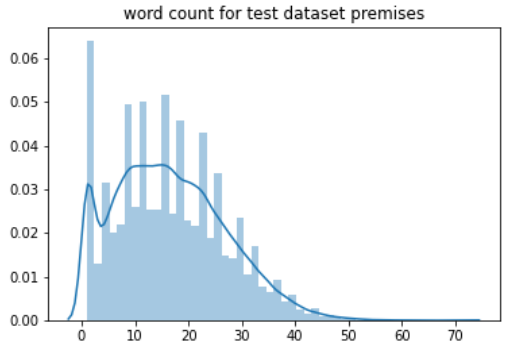


Figure 11: Test dataset premises Words Counts

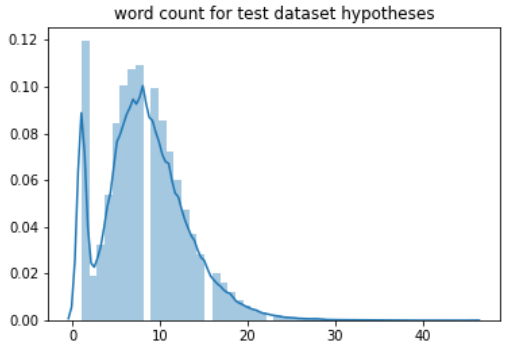


Figure 12: Test dataset hypotheses Words Counts

From figures (07 - 12), the distribution of tokens' length as per sentence is of a high density in the region of 0 - 50 tokens as per sequence. And very few sequences that had number of tokens surpassing 50 tokens. It can be deduced that most of the premises sentences are double the length of hypotheses' sentences. Given that the premises sentences were mostly double length of hypothesis sentences, a maximum length of 128 tokens was chosen

Hyperparameter	Value
Pretrained Embeddings	Fasttext
Embeddings Dim	300D
Hidden Dim	256D
Dropout Rate	0.4
Stacked Layers	2
BiLSTM	True

Table 5: Experiments 1 & 2 Hyperparameters

B Models

B.1 Comparison between Models

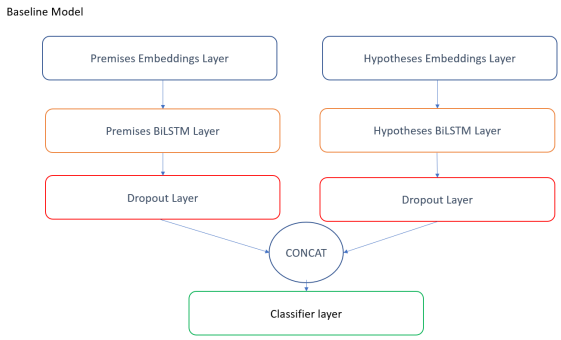


Figure 13: Baseline model architecture

Figure 14: Premises LSTM model output is concatenated with Hypotheses LSTM model before being passed to the classifier

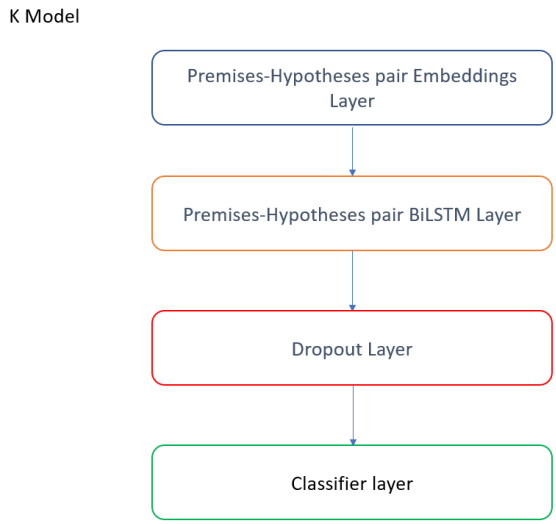


Figure 15: K model architecture

Model Name	Number of model params	Vocabulary Size (tokens number)
Multilingual BERT	$\approx 178M$	$\approx 119K$
XLM (MLM & TLM)	$\approx 249M$	$50K$[7]
XLM-R	$\approx 560M$	$250K$[8]
RoBERTa	$\approx 350M$	$50K$

Table 6: Difference between pretrained models number of params & vocab size

Model Name	Optimizer	Learning Rates
Baseline	Adam	$1e-3$
K_Model	Adam	$1e-3$
Multilingual BERT	AdamW	$2e-5$
XLM (MLM & TLM)	AdamW	$5e-6$
XLM-R	AdamW	$5e-6$

Table 7: Models Optimizers & Learning Rates

AdamW optimizer was used with the default epsilon value of $1e-6$

Table 8: Difference between models

Model Name	BS	GPU Used	Training time (epoch)	Acc (%)
Baseline	8	Tesla K80	2hrs and 30m	33.3%
K Model	8	Tesla K80	50m	33.3%
mBERT	8	Tesla K80	5hrs and 15m	68.1%
XLM (MLM)	8	Tesla K80	5hrs and 35m	70.3%
XLM (MLM & TLM)	8	Tesla K80	6hrs	72.3%
XLM-R	16	Tesla P100	5hrs	79.6%

Difference between model batch size, training resources, time, performance. Regarding the baseline & K_Model both achieved the same accuracy in both cases training word embeddings from scratch or using fasttext unaligned pretrained word embeddings, due to the fact that the word embeddings were not aligned in the same space hence model can't tell if Home (en) is same as Casa (it)

Table 9: XLM-R Model Performance

Language Code	Accuracy (%)
Overall	0.7965
AR	0.7856
BG	0.8301
DE	0.8246
EL	0.8182
EN	0.8814
ES	0.8457
FR	0.8283
HI	0.7593
RU	0.7992
SW	0.7168
TH	0.7722
TR	0.7880
UR	0.7224
VI	0.7914
ZH	0.7842

Model's performance across the 15 languages XNLI, As seen results differ from a language to another as some languages belong to the same family (e.g. Italian, French, Spanish are all Latin Languages & Deutsch, English belong to Germanic Languages), languages belong to the same family will show results in same range. Low resources languages (Swahili, Urdu) has shown the lowest results

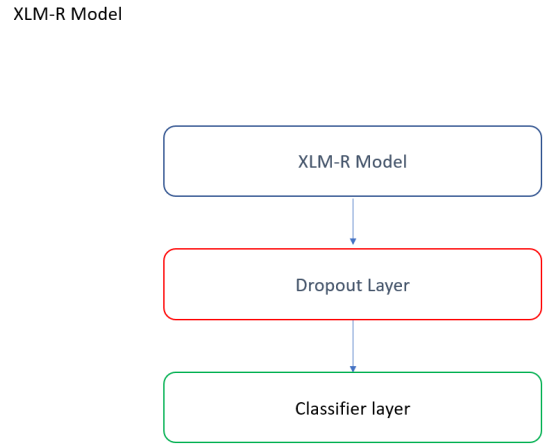


Figure 16: XLM-R Model architecture (best performing model)

XLM-R For sentence level classification, what is needed just to add a classifier on top of the model using its sentence representation output to classify the relationship between 2 sentences.

B.2 MLM & TLM Training Objectives

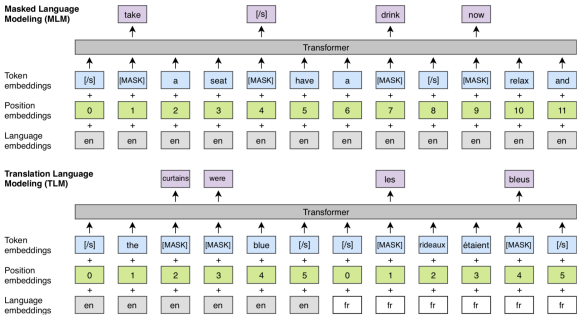


Figure 17: XLM model different Training objectives

XLM model different Training objectives, MLM just focuses on predicting the masked tokens given the current sentence context (single sentence classification), while TLM focuses on predicting masked tokens as well, but with a pair of sentences from 2 different languages allowing it to leverage the context of whichever language suffice, consequently forcing the model to better align sentence representations in the same embeddings space

C Training Loss & Accuracy

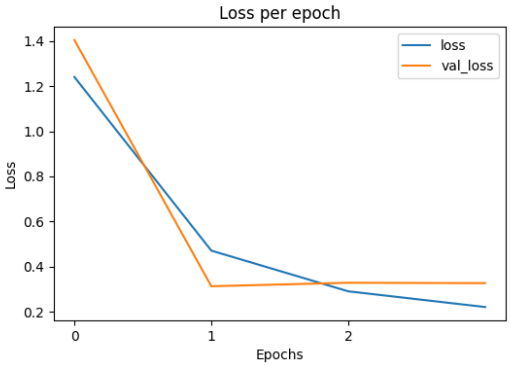


Figure 18: XLM-R Model training Losses

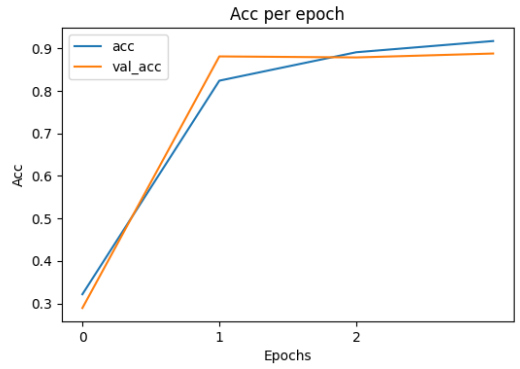


Figure 19: XLM-R Model training accuracy

XLM-R Model training epochs shows that the model is training, yet after the 1st epoch val_loss started to increase as well as the val_acc started to plateau, however, this pattern are not enough to deduce whether the model was overfitting or not, yet XLM-R model with such learning capability might not overfit that early in the training process

D Confusion Matrix

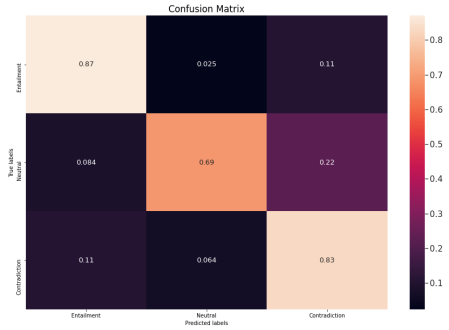


Figure 20: XLM-R Model Confusion Matrix

Confusion matrix describes the performance of a multi-class model in a visualized form, allowing identification between which classes were confused as others. $C_{(1,1)}$ indicates True positive instances of class 1, and the other columns (in row 1) are False Positives, same applies for $C_{(2,2)}$, $C_{(3,3)}$,. First Row is actual Class 1, 0.87 correctly predicted, $\approx$ 0.025 confused as class 2, $\approx$ 0.11 confused as class 3. Same applies to every other row/class.
It is obvious that the model tend to confuse neutral relationship between pairs of sequences with contradiction rather than entailment

E Word Clouds

Word clouds are a data visualization technique used to represent the text data where the words' size represent its importance in context and frequency of occurrence in a given text, excluding stopwords as they are very frequent in a given language, yet they don't add much of information to text.

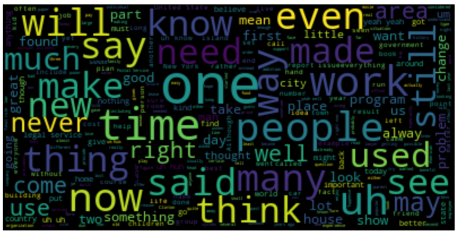


Figure 21: English Word Cloud

F Future Work

- Train pos2vec model using multilingual corpus in order to fetch crosslingual PoS embeddings in the same shared space, and use its embeddings to train a model which output will be concatenated to the embeddings of XLM-R model last hidden state before being fed to classifier layer, hypothetically, this might have a positive impact on the model's overall generalization capability.