Homework 2: Semantic Role Labeling (SRL)

Alessio Orlando
Sapienza University of Rome
orlando.1792394@studenti.uniroma1.it

1 Introduction

Semantic Role Labeling consists of tagging words with labels that indicate their semantic role. The SRL pipeline usually is divided in 4 majour steps: Predicate identification, Predicate Classification, Argument identification and Argument Classification. Our main task is to address C and D, but I decided to address also task A and B to see how the evaluation drops.

2 Dataset

The dataset consists in different sentences tokenized with labels. In this section I'll briefly talk about some problem that exists in the dataset and that has not been addressed by me to allow a fair evaluation between the reports. The dataset presents some errors in the labeling of roles and predicates, this cannot be addressed manually since the labels should not be modified nor removed. In my case I decided to work with all the labels from VerbAtlas (Di Fabio et al., 2019) plus all the the remaining classes in the dataset 2.

2.1 Preprocessing

The dataset given does not require a lot of preprocessing. Is already divided in tokens, both lemmatized and not. The pos are given and also the dependency head and relations. Each sentence can have more than 1 predicate. Even if it is possible to predict and connect SRL to the desired predicate within the same sentence, to facilitate the model for roles identification and classification I decided to divide the sentences such that each sentence have one and only one predicate. This only for task C and D, instead for task A and B (predicate identification and classification) this was not necessary because the model can easily tag them. Such approach create an even bigger unbalance 1 in the classes. I decided to not address this issue in my experiments but it is surely a problem to consider for further analysis.

3 Words

The core of the model is based on a Transformer layer that acts as an embedding extractor for the words. Right after the transformer I added a Bi-LSTM layer to better capture the essence of the word embedding. The pre-trained embedding tried are 'xlm-roberta-base', 'bert-base-cased' 'bert-base-multilingual-cased'. For task A and B I ended up using 'bert-base-case' given the better F1 score, 'bert-base-multilingual-cased' was surely a possible alternative, even an XLM language model. Overall the difference was not very high, in fact in the possibility of a cross language approach I would have chosen the latter one. For training time and the approaching deadline I didn't managed to perform any cross-language test but it is surely a possible next step. 'xlm-roberta-base' instead was the best result overall for task C and D with a little margin over the other two, as shown in the graph 9.

4 Additional inputs

All of the models presented in this report used only words as input for the dataset, this because both models (For task AB and CD) needed to work with the full evaluation function on which where only given words and lemmas. It was still possible to extract those information from the words tokenization but after some early stages test I noticed that adding posses didn't changed the behaviour of the model at all. As for the dependencies relation I didn't find any useful way to use them. A possible experiment that came in my mind was to transform the dependency relation in a format similar to a natural language one and pass them to a Transformer embedding layer and try to create an embedding able to carry that information throughout the layers.

In the end I didn't managed to try this approach for time restriction but I think that, considering the fact that usually transformers performs really well with natural languages, it can still be an interesting approach to try .

5 Model

The core of both the models is a pre-trained embedder with a Bi-LSTM layer as feature extraction layer. After that there are different layout for the classification of the different tasks.

5.1 Tasks C & D

For the task C and D I used 2 Linear layers stacked on top of each other 4. This quantity is the results of different tests on which only few really started learning (others remained very low on f1 scores).

5.2 Tasks A & B

For task A and B instead I decided to try a Multi-Task approach with, of course multiple losses. The first part of the pipeline is the same as the one in the model for task C and D. After the Bi-LSTM layer I've placed a shared linear layer, a dropout and an activation layer. Then I started the separation for the classification of both A and B. In particular I added one single Linear layer for each task 3. Given the high number of classes I decided to maintain an high value for the hidden layers of both the LSTM and the linear layers (512 for both the LSTM and the Linear layer). This resulted also to be the best result overall 8.

6 Training

The training has been performed mostly by evaluating the F1-score for the validation dataset. The F1-score for the SRL is not really straightforward. As we can see in the graph 1, there is a big unbalance for the class ”_”, this need to be taken in account while performing the evaluation. For the training of the models I decided to use the same function used for the final evaluation by the committee. To be sure that the model was not in an overfitting path also the validation loss has been taken into account to perform early-stops.

6.1 Multi-Task train

During the training of the model for task A and B I decided, as already mentioned, to perform a multi task training. Since the output of this model needed to be used as input for the second model I really wanted to have a good identification model and so I gave a little bit more importance to that part of the task by adding an additional loss just for the predicate identification. I then merged the two losses and gave to the task B (Predicate Disambiguation) 9 time the importance compared to the task A. This number is not really important since there was not a big difference in the end result, my scope was to reduce the overfitting on the task A (easier) by applying a weighted average of the two losses. I didn't obtained the expected outcome but I didn't worsen it either so I decided to leave it as it was 6.

7 Best model

As already mentioned I putted the most effort in evaluating the different languages models and the behaviour with the main corpus (the English one). Even tho there is small differences between the results, the model that gave the best result with my architecture (described in chapter 5.1 and 5.2) is the 'XLM-roberta-base' for task C and D and 'bert-base-cased' for task A and B. In the evaluation of only C and D I achieved an F1 of 88.46% in task C and 82.24% in task D 2. In the complete evaluation of A, B, C and D, instead, I reached an F1 score of 93.96% in task A and 82.71% in B (Predicate Disambiguation)1. This scores of course reduced the overall performance of my CD model. In fact the F1 score for task C became 84.49% and in task D 79.06%. In all the models the loss behaved as expected and never reached a point of clear overfitting. I am still in doubt about a possible underfitting of the model but any increase in complexity resulted in a lower F1 score and higher loss.

8 Conclusion

In conclusion most of the outcome was pretty much as expected, the XLM model, more complex and harder to train achieved a better score in task C and D, and didn't managed to overcome the higher score of the BERT model in task A and B, simpler to tune. I am sure that with more tune and tweaks the XLM model can increase his lead in respect of the other two models and can surely beat the simpler yet effective ”Bert-base-cased”. As for 'BERT-multilingual-cased' he managed to not lose too much ground but given the aptitude for a multilingual approach he didn't managed to stay on top of any of the Tasks tried in the English corpora.

9 Next Steps

In this section I'll briefly explore the motivation behind the effort putted on training with an XLM model and some different approaches that could have been tried to increase the performance of the model. The XLM model is substantially a BERT model modified in such way that enhance the cross language cohesion in the corpus. Instead of using the word representation he uses Bytes representation of the data, this increase the overlap between different languages and opens up the possibility of training concurrently on multiple corpus in different languages. (Conneau et al., 2019) The cross language corpus given seems to be perfect for this kind of elaboration and train, with a little bit more time and resources it can be surely be exploited to perform good in the cross-language task. Also, further studies need to be done to understand what kind of structural changed can be applied to the architecture to increase the performance without increase the corpora given. In fact a second possible interpretation of such an high difference in loss between train and validation is that the corpora given does not fully capture the essence of each of the classes, this can also be seen in the confusion matrix where we can see how some classes are really hard to predict, some of them are even completely miss classified by the model.

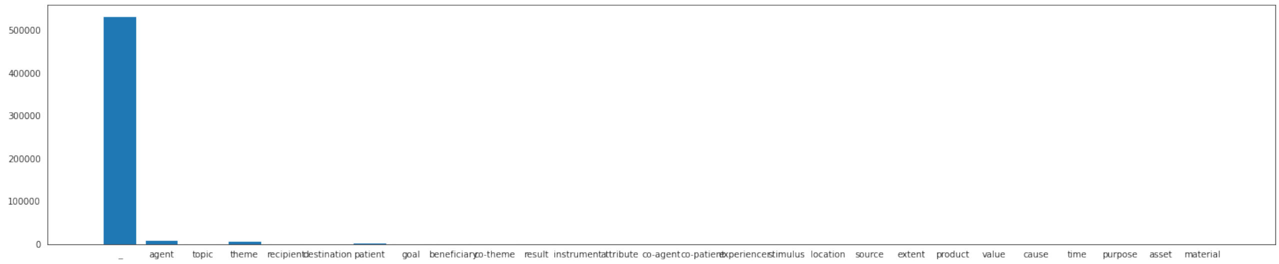

Figure 1: Class distribution: EN dataset

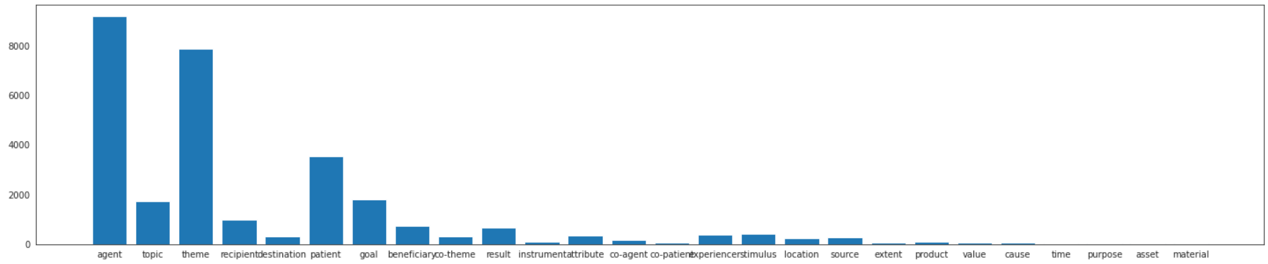

Figure 2: Class distribution without "_": EN dataset

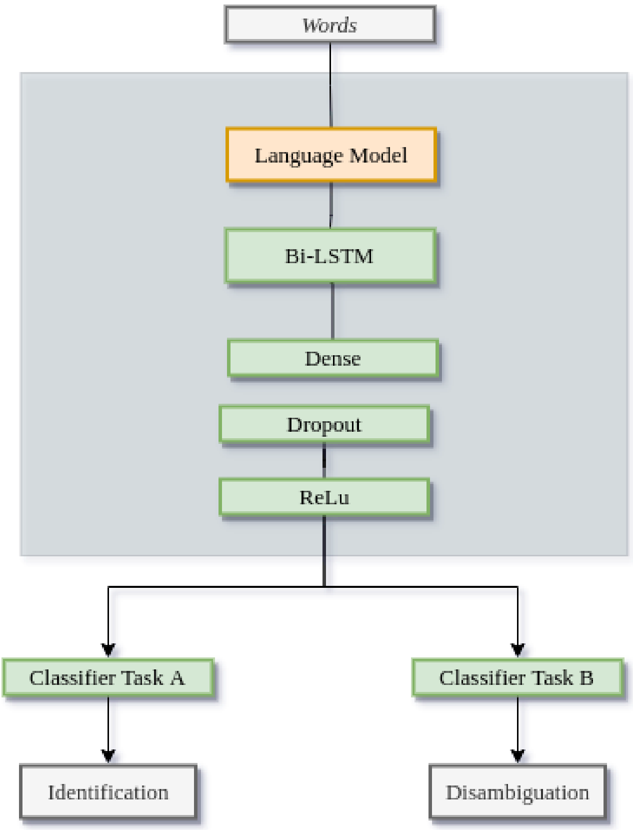

Figure 3: Model diagram for task A and B

Lang Model	**Dev F1 A**	**Dev F1 B**
BERT-base-cased	93.96%	**82.71**%
BERT-multilingual-cased	94.43%	81.19%
XLM-roberta-base	**94.56%**	82.55%

Table 1: Models results

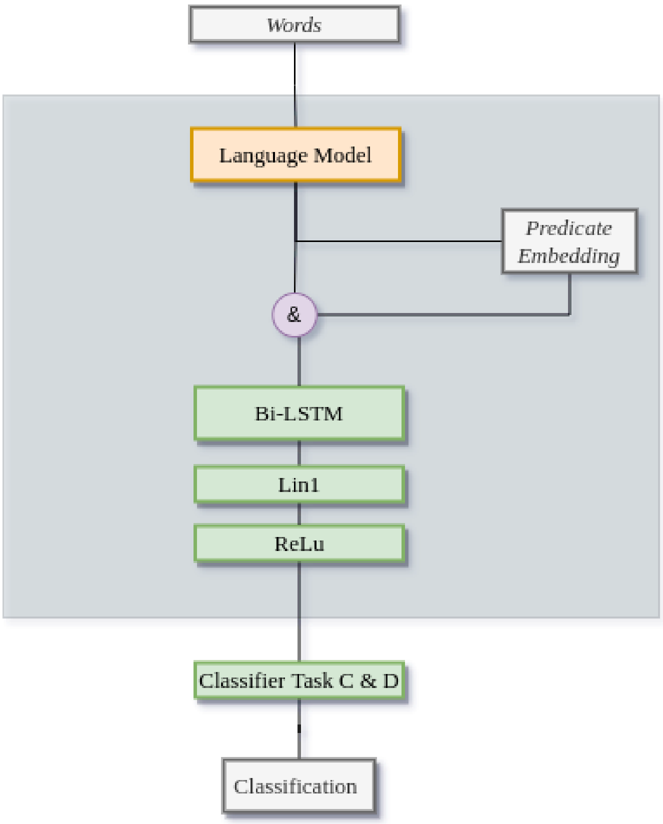

Figure 4: Model diagram for task C and D

Lang Model	**Dev F1 C**	**Dev F1 D**
BERT-base-cased	88.31%	80.14%
BERT-multilingual-cased	88.12%	81.35%
XLM-roberta-base	**88.46%**	**82.24%**

Table 2: Models results

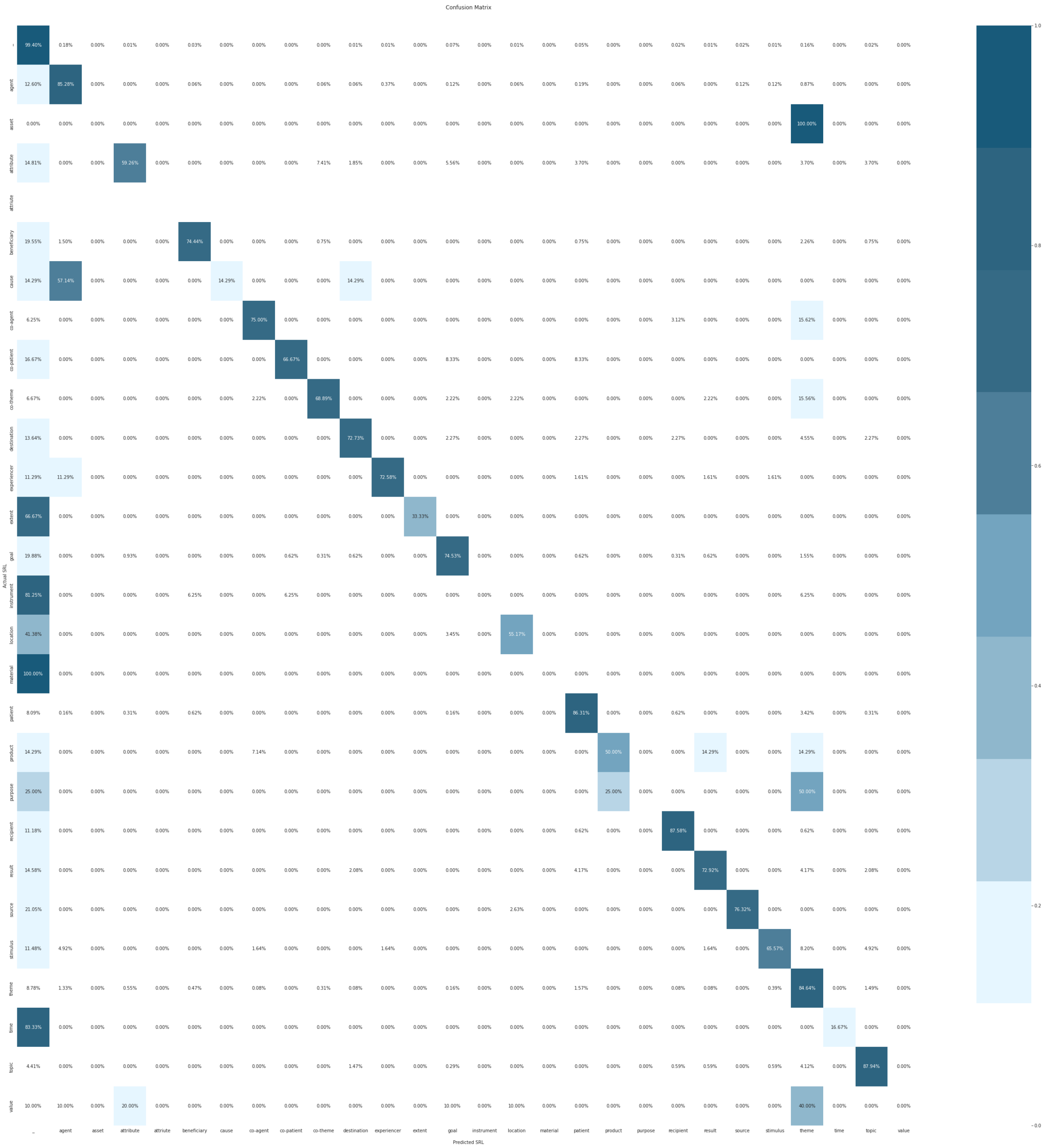

Figure 5: Confusion matrix for task D

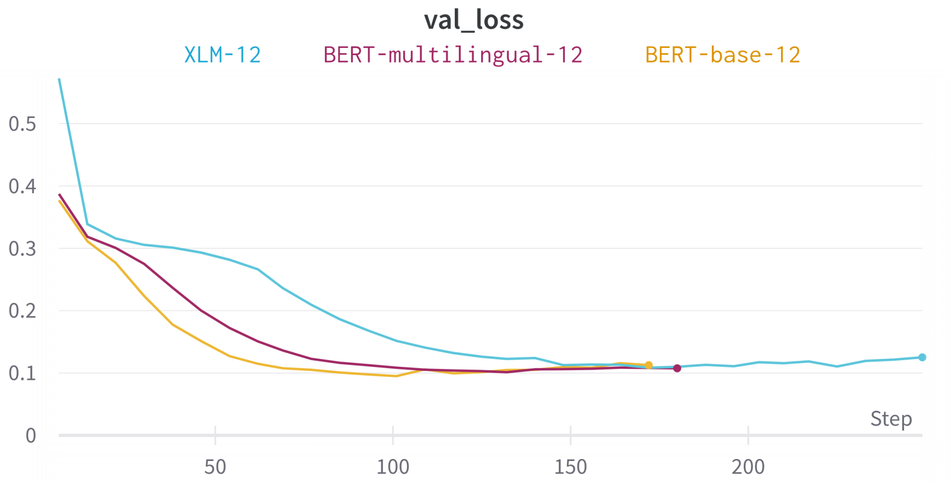

Figure 6: Validation Loss: Task A and B

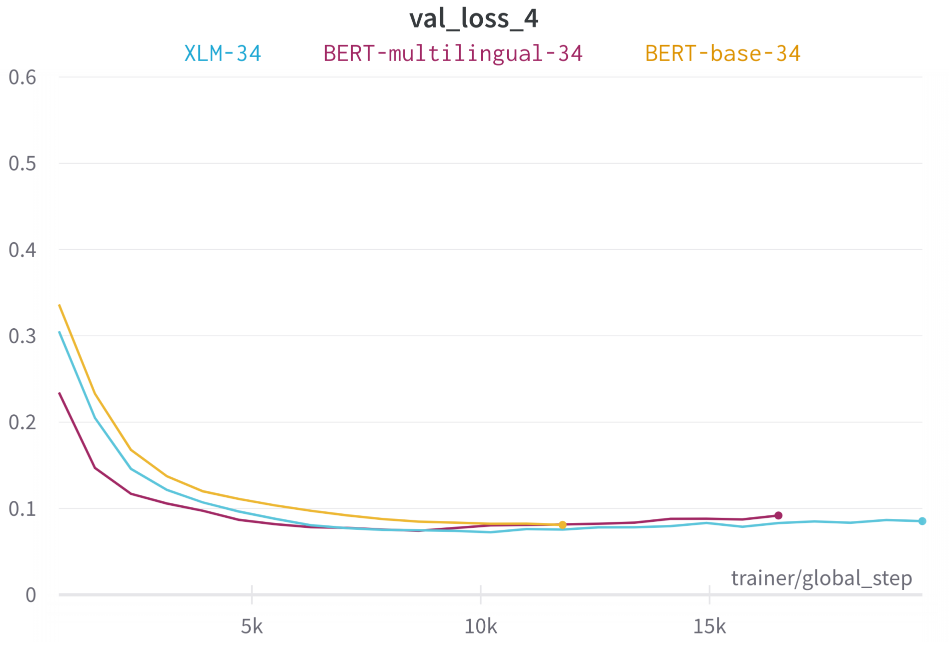

Figure 7: Validation Loss: Task C and D

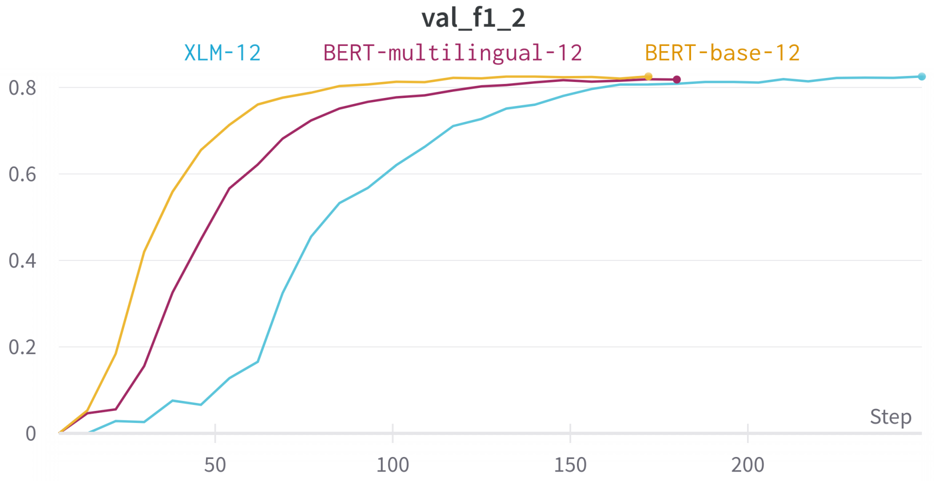

Figure 8: Validation F1: Task A and B

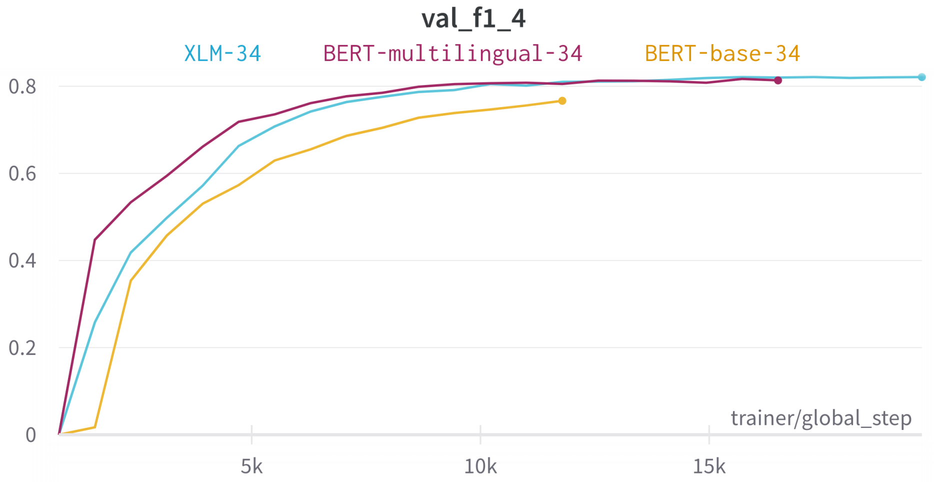

Figure 9: Validation F1: Task C and D

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. 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 627–637, Hong Kong, China. Association for Computational Linguistics.