Semantic Role Labeling

Emil Engelstad Moghaddam
moghaddam.2036494@studenti.uniroma1.it

1 Introduction

Semantic Role Labeling (SRL) was described by Màrquez et al. (2008) as the process of extracting the information of "Who did What to Whom, Where, When and How?" from natural language. SRL is a shallow semantic parsing technique that produces predicate-argument structures from natural language. Predicates refer to words, or multiword expressions, that indicate either an action or an event. On the other hand, arguments are those phrases in the sentence that relate to the predicate. SRL can be split into a four-step process. Predicate identification and disambiguation(picking the correct sense), argument extraction(for each predicate) and the classification of arguments into semantic roles. SRL has shown to be beneficial for tasks such as machine translation (Shi et al., 2016).

Embedding and encoding of the language was done using the Bidirectional Encoder Representations from Transformers(BERT) proposed by Devlin et al. (2018). BERT utilises the same attention mechanism proposed in the original transformer paper, "Attention is all you need" (Vaswani et al., 2017), but differs in the fact that there is no decoder. BERT was trained using masking and Next Sentence Prediction on unlabeled data. The version used is called BERT multilingual base model (cased) and was trained on text from 104 languages, making it suitable for multi-language transfer learning.

The use of BERT models for SRL was first done by Shi and Lin (2019). They feed the contextual embeddings generated by the BERT model through a multi-layer perceptron(mlp) followed by a bidirectional Long Short Term Memory (BiLSTM). This method is mentioned because it shows the possibility of using a BERT model in combination with another sequence handling architecture. In our case, this is important because the dataset contains additional sequential data, incompatible with the BERT tokenizer, but compatible with other sequential models.

2 Method

2.1 Training-parameters

The training loop was initialised to use a batch size of 16, Adam optimiser (Kingma and Ba, 2014), a gradient clipping value of 0.5 (Zhang et al., 2019), and weighted cross-entropy loss. In the loss function, the weight of the null class was set to be equal to the inverse probability of the class, which is 0.05. Even though Figure 6 shows that the remaining classes have a skewed distribution, they were weighted equally.

2.2 BERT model - English dataset

A model consisting of a classification layer stacked on top of the BERT embedding layers was trained for 100 epochs on the English data set. The model was checkpointed at the validation loss minima, where several metrics were evaluated.

2.3 Cross-lingual learning

Furthermore, the same model was used as a warm start and trained for an additional 15 epochs on the French and Spanish data set. To measure the effect of this method, the BERT model was also trained one time solely on the French and Spanish datasets.

2.4 Utilization of pos-tags and dependency relations

The effect of including pos-tags and dependency relations into the model was tested using two different architectures. Here embedding was done using randomly initiated vectors 30-dimensional vectors. The first architecture used a combination of Gated Recurrent Units(GRUs) and mlp's Figure 7. The second architecture used 3 additional encoder heads Figure 8.

3 Results

3.1 BERT model on English dataset

For the base model we see that validation loss reaches a minima after 20 epochs Figure 1. At this point to model obtains a classification f1 score (validation) of **0.82** (Table 1), and an identification f1 (validation) score of **0.90**. (Table 2).

3.2 Cross-lingual performance

Tuning the BERT model to the English dataset, before training on the Spanish/French dataset led to increased performance in both argument identification and argument classification. The comparatively better results occurred for both Spanish and French.

Classification f1 score was increased from **0.43** to **0.66** on the French dataset, and from **0.47** to **0.54** on the Spanish dataset Table 4.

3.3 Utilization of pos-tags and dependency relations

The two architectures that utilised pos-tags and dependency relations showed no significant improvement over the base model. In fact metrics obtained are very similar Table 3. The only real difference between the architectures is that the model with the recurrent head approaches the validation minima more slowly Figure 2

3.4 Classification results per class

Table 5 shows major differences among which classes the modelled successfully was able to categorise. Visually inspecting the table, one can see a strong correlation between the number of instances in which the class appears in the dataset, and the model's ability to categorise that class. Figure 5 shows that out of the 5000 arguments, 500 are wrongly marked by the model as belonging to the null class. When going from the identification to classification, another 400 of the initially correctly identified arguments are wrongly categorised (Figure 4). The number of false positives and false negatives are well balanced in both argument identification and categorisation

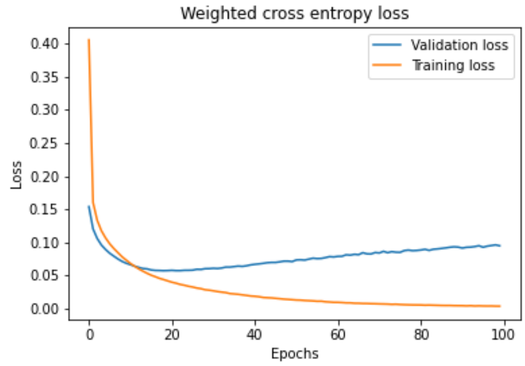

Figure 1: Weighted cross entropy loss for the base model trained - 100 epochs

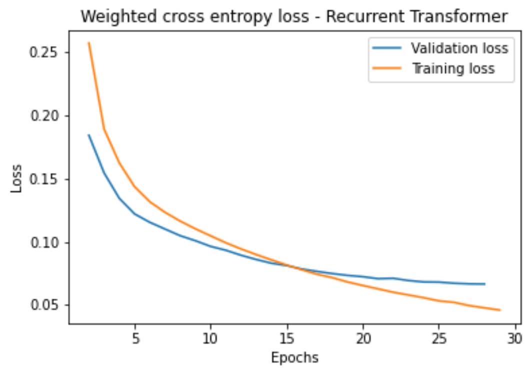

Figure 2: Weighted cross entropy loss for model with a recurrent head Figure 7

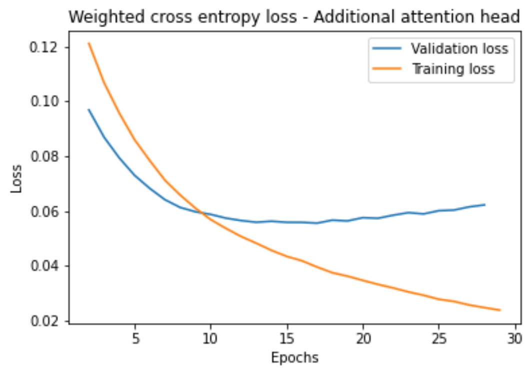

Figure 3: Weighted cross entropy loss for the model with additional encoder head Figure 8

Metric	Result
F1	0.821
False negatives	903.0
False positives	887.0
Precision	0.822
Recall	0.820
True positives	4110.0

Table 1: **Argument Classification** - on English validation dataset

Metric	Result
F1	0.901
False negatives	503.0
False positives	487.0
Precision	0.903
Recall	0.899
True positives	4510.0

Table 2: **Argument Identification** - on English validation dataset

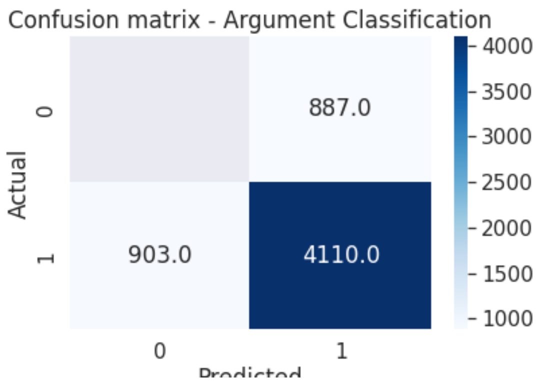

Figure 4: Shows the confusion matrix for argument classification visualized as a heat map

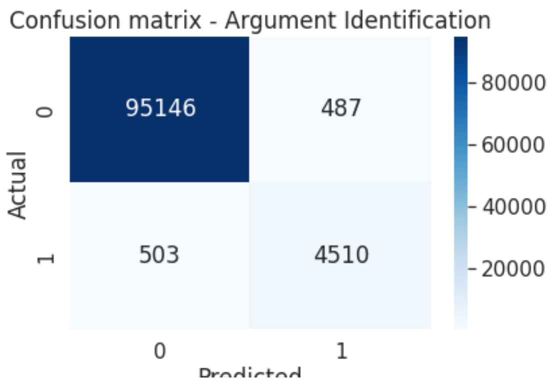

Figure 5: Shows the confusion matrix for argument identification visualized as a heat map

4 Conclusions

- The BERT-based model achieved high results on the dataset compared with the 0.25 baseline.
- Pos-tags and dependency relations did not improve the model's results when using the architectures proposed in this paper.
- Knowlegde learned by multi-lingual BERT in one language is highly transferable to other languages.
- Mistakes were equally balanced between identification and categorization, leaving room for improvement on both aspects.
- Discrepancies between how often instances of different classes are correctly categorised seem to be correlated with the frequency of the class in the dataset.

References

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Lluís Màrquez, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Special issue introduction: Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159.

Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. 2016. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2245–2254, Berlin, Germany. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. 2019. Why gradient clipping accelerates training: A theoretical justification for adaptivity.

Metric	BERT	BERT + GRU	BERT + Attention-head
Identification			
F1	0.901	0.899	0.901
Precision	0.903	0.892	0.901
Recall	0.899	0.906	0.901
Classification			
F1	0.821	0.810	0.827
Precision	0.804	0.798	0.827
Recall	0.816	0.804	0.827

Table 3: **Model comparison** - Comparing the effect of additional lexical information with the use of Gated recurrent units and encoder layer

Model type	Identification F1 score	Classification F1 score	Language
Base model	0.72	0.47	Spanish
Fine Tuned(on English)	0.78	0.54	Spanish
Base model	0.73	0.43	French
Fine Tuned(on English)	0.85	0.66	French

Table 4: Multi language transfer learning

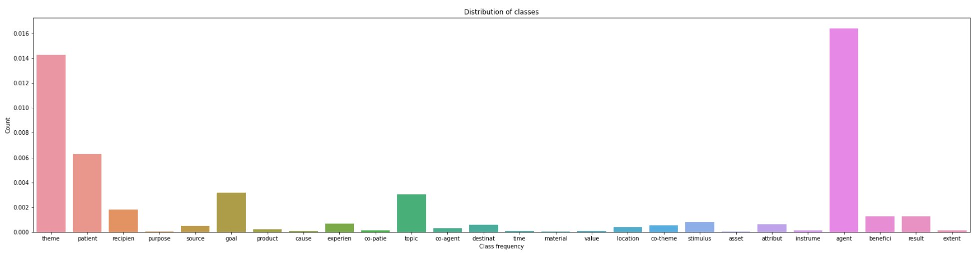

Figure 6: Distribution of classes – no argument class (0.964%) is removed

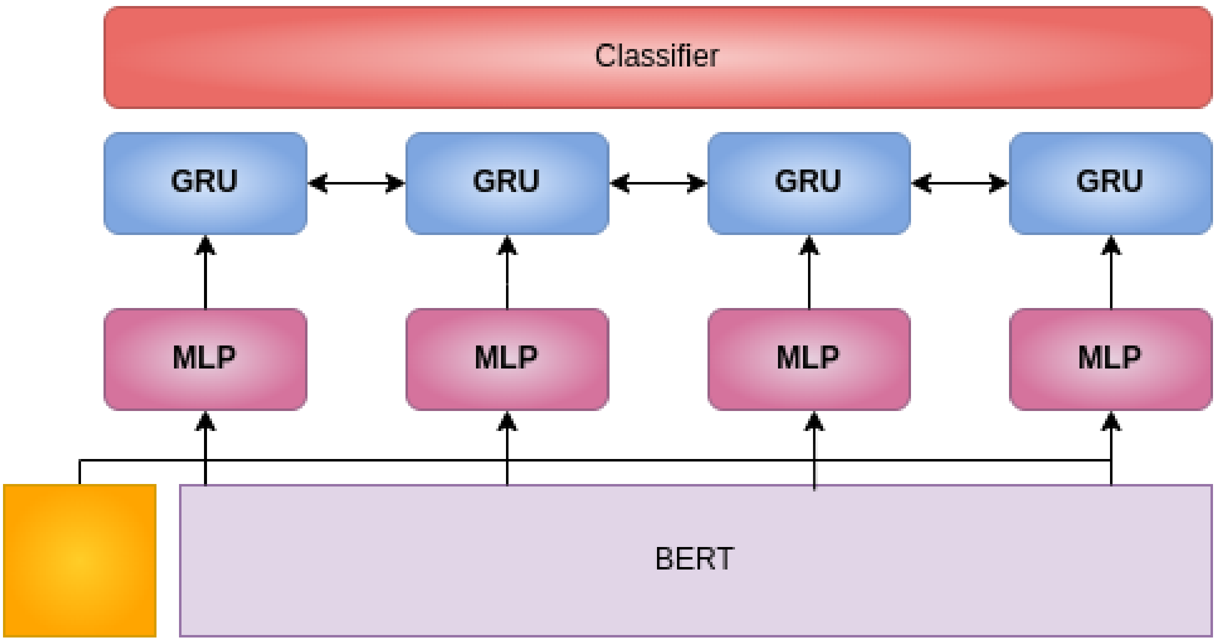

Figure 7: Figure shows BERT architecture with Gated Recurrent Unit head - Yellow box shows how additional embedding can be feed into architecture

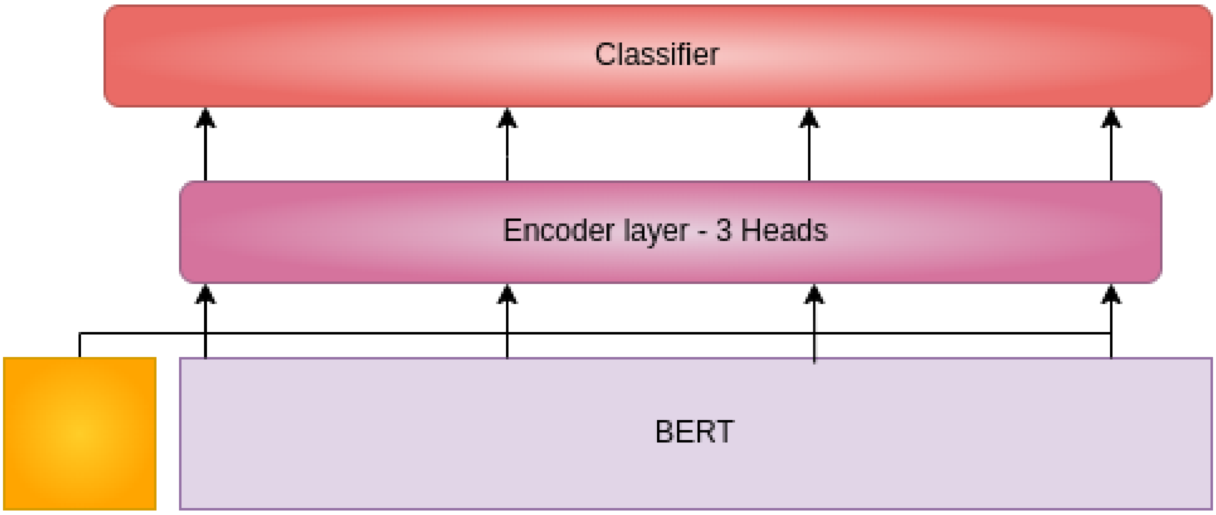

Figure 8: Figure shows BERT architecture with additional transformer head - Yellow box shows how additional embedding can be feed into architecture

Metric	Count	Correctly classified	Percentage
theme	1276	1068	84 %
patient	643	571	89 %
recipient	161	140	87 %
purpose	4	0	0 %
source	38	25	66 %
goal	322	232	72 %
product	14	0	0 %
cause	7	0	0 %
experiencer	62	37	60 %
co-patient	12	7	58 %
topic	340	306	90 %
co-agent	32	24	75 %
destination	44	25	57 %
time	6	2	33 %
material	1	0	0 %
value	10	0	0 %
location	29	12	41 %
co-theme	45	29	64 %
Null class	95633	95146	99 %
stimulus	61	33	54 %
asset	1	0	0 %
attribute	54	31	57 %
instrument	16	1	6 %
agent	1603	1397	87 %
beneficiary	133	100	75 %
result	96	69	72 %
extent	3	1	33 %

Table 5: **Argument Classification by class** - on English validation dataset