MNLP Homework-3
Relations Extraction

Francesco Palandra
Department of Computer Science
Sapienza University of Rome
palandra.1849712@studenti.uniroma1.it

Abstract

In the domain of Natural Language Processing (NLP), Relation Extraction emerges as a pivotal challenge notable for its complexity and its importance in retrieving information efficiently. Different approaches have been used, from table filling end-to-end to more recent generative approaches that tackle the problem in a new way using seq2seq models. In this report, an adaption of UniRel is presented. This approach exploits the use of an interaction Map built on both entity and relation interaction to extract subject-relation-object triplets from raw text. This method uses the self-attention mechanism of BERT to learn the interaction Map and achieve good results in the Relations Extraction task. The code is available at: https://github.com/Frklin/nlp2023-hw3

1 Introduction

Relation Extraction (RE) has recently gained prominence within the NLP community as a task of increasing interest and importance. The goal of Relation Extraction involves taking a sentence and extrapolate all the possible triplets of the form subject-predicate-object (<s-p-o>). Its difficulties make this task very hard to tackle, however, the contributions represent a significant leap in advancing Natural Language Understanding, with applications ranging from semantic search to knowledge graph construction.

2 Related Work

Various techniques have been proposed in the past years, transitioning from traditional methods to innovative approaches leveraging deep learning advancements. Initially, RE tasks were approached through table-filling methods, where entities and relationships were mapped in a structured form, facilitating the extraction process (Wang and Lu, 2020). Subsequently, the field saw a shift towards generative methods, exemplified by REBEL (Huguet Cabot and Navigli, 2021), which reimagines RE as a sequence-to-sequence (seq2seq) task. This innovation leverages the generative capabilities of models like BART, pushing the boundaries of how relationships are extracted from text, marking a departure from structured table filling to more fluid, context-aware methodologies. Building on these foundations, UniRel (Tang et al., 2022) introduces an approach that further refines the interaction between entities and relations. By leveraging the attention mechanism inherent in models like BERT, UniRel constructs an interaction map that captures the dynamic connection between entities and the relations. In the UniRel implementation a max len constant is defined based on the sentence with more tokens and all the sentences are padded based on this value. In the following approach it has been decided to pad each batch individually. This increased the difficulty on managing and keeping track on the indices but decrease drastically the time of each epoch up to 50% in time and 30% in memory.

3 Methodology

This section delves into an analysis of the methodological approach.

3.1 Problem Formulation

Let's give a formal definition of the problem. Given a sentence $X = \{x_1, x_2, ..., x_N\}$ where N is the number of tokens in the sentence, we would like to have all the triplets of the form $R = \{(s, p, o)\}_1^r$ where s, o, p are respectively the subject and the object the relation that connects the two entities and r is the number of relation triplets. More in detail, the subject and the object are composed from one or more tokens of the sentence $s = (x_i, x_j)$ and $o = (x_k, x_l)$ while the predicate is extracted from a set of possible predicates. Finally, to simplify the notation, as shown in figure 1 instead of taking the tokens we will use the entity start and end index.

Figure 1: Example of triplet relations in an sentence

3.2 Interaction Map

The main idea is to exploit the attention mechanism of the transformer to find the correlation between the tokens. In this way, an interaction map as shown in figure 3 can be generated. To precisely find the right span of the subject and the object, 3 different matrices are created: head H^{NxN}, tail T^{NxN} and span $SPAN^{NxN}$. These matrices capture three distinct features, respectively focusing on the start indices, end indices, and the span of the entities. The matrices are filled with zeros and ones are positioned where an interaction between two entities appears. There are 3 different types of interactions:

- **Entity-Entity**: the connection and interaction that occurs between subject and object entities.
- **Entity-Relation**: the link between the subject entity and its predicate.
- **Relation-Entity**: the association between the predicate and the object entity.

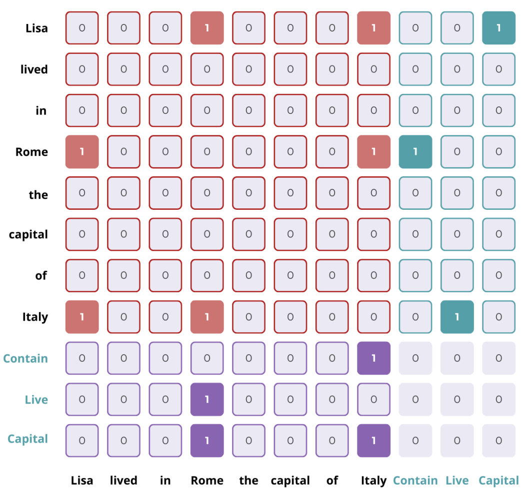

Figure 3: Interation Map from a sentence, in red all the entity-entity interaction, in teal the entity-relation and in purple the relation-object.

3.3 Relations Extraction

From these three matrices all the triplets can be extracted. The constraints are very rigid since a triplet is valid if all these three conditions are satisfied:

$$\forall s_{start}, p, o_{start} \in H^{NxN} : \\ (s_{start}, p) \wedge (p, o_{start}) \wedge (s_{start}, o_{start}) \in H_{\mathbb{1}} \tag{1}$$

$$\forall s_{end}, o_{end} \in T^{NxN} : \\ (s_{end}, p) \wedge (p, o_{end}) \wedge (s_{end}, o_{end}) \in T_{\mathbb{1}} \tag{2}$$

$$(s_{start}, s_{end}) \wedge (o_{start}, o_{end}) \in SPAN_{\mathbb{1}} \tag{3}$$

Meaning that all the $< s_{start},\ p,\ o_{start} >$ are extracted from the matrix H^{NxN} s.t. the tuple (s_{start}, p), (p, o_{start}) and $(s_{start}, o_{start}) \in H_{\mathbb{1}}$ (the set of $(x,\ y)$ coordinates in H^{NxN} where the cell [x, y] = 1). Then, keeping the same predicate p all the $< s_{end},\ p,\ o_{end} >$ from T^{NxN} are extracted s.t. (s_{end}, p), (p, o_{end}) and $(s_{end}, o_{end}) \in T_{\mathbb{1}}$. Finally, a last check is done on the $SPAN^{NxN}$ matrix to verify that the span of the subject and the object is the right one.
It seems very challenging to align all these conditions, however, it has been observed that without so many controls many more triplets are included and while the precision might increase, the recall decreases drastically and so does the F1-score since more triplets were allowed. However, to favor the extraction of triplets the matrices are built symmetrically, meaning that both $[s, o]$ and $[o, s]$ are set to one in the matrices. This results in a slight increment in the performance.

4 Evaluation

In order to evaluate the model we need to distinguish two different aspects of the metrics. The first refers to the metrics concerning the matrices and the latter refers to the relations extracted. The two metrics are connected and are directly proportional one from each other, however achieving high scores in matrix metrics does not automatically lead to improved results in relation metrics. They indicate a favorable scenario for tuple extraction, provided all conditions are meticulously adhered to. The primary metric employed is the F1-micro, which evaluates the model by considering both the accuracy of the extracted triplets and their quantity, while the selected is Binary Cross Entropy since the matrix contains 0 and 1s.

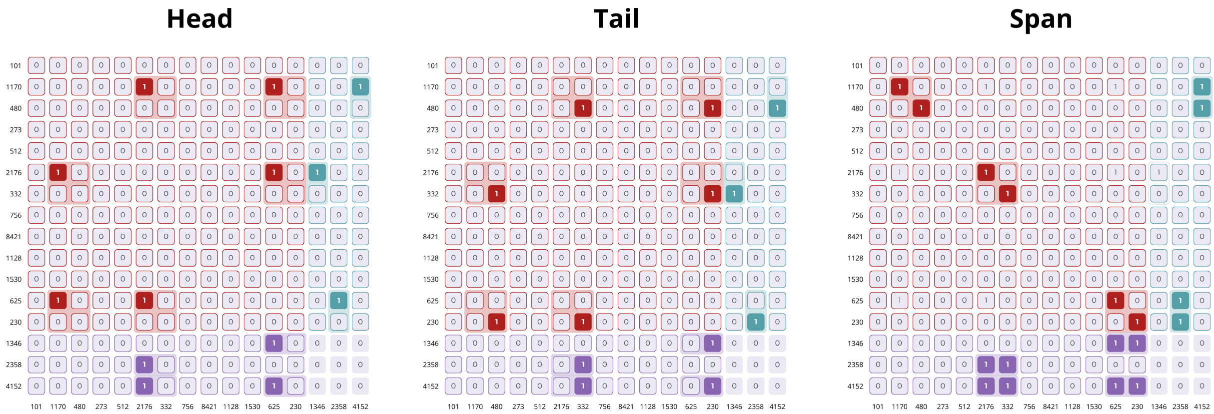

Figure 2: The head, tail and span matrices generated from 3, as it showed in the figure, the number of rows and columns is higher since the encoding from bert can tokenize one word in more pieces.

4.1 Unidirectional vs Bidirectional

As mentioned in chapter 3.3 to raise the probability of correctly locating the position of the entities it is possible to construct the part of the entities of the matrix symmetrically. An exploration into the impact of bidirectional matrix usage reveals that symmetrically configuring entity sections significantly increases the precision of entity position detection.

5 Results

The performance outcomes of the developed model are presented in the table 2 and figures 4, showcasing the model's efficacy in the Relation Extraction task.The model demonstrates excellent performance, achieving an F1-score of 0.89 on the validation set and **0.90** on the test set.
A detailed examination of precision and recall metrics (figure 6) for triplet extraction further establishes the model's capability. The synergistic effect of utilizing three distinct matrices—head, tail, and span plays a pivotal role in retrieving the web of relations present within sentences. This strategic approach allows for a comprehensive capture of relationships, significantly contributing to the model's overall performance. Moreover, an analysis of the precision across single batches and throughout each epoch reveals a rapid assimilation of knowledge by the model, as evidenced by its swift convergence to a validation precision of approximately 0.90, not only demonstrating the model's efficiency in understanding the underlying patterns but also its effectiveness in generalizing from the training data to unseen examples. Particularly interesting is the impact of implementing a bidirectional matrix on the model's validation performance. As illustrated in the plot (Figure 5), the introduction of a bidirectional (symmetrical) matrix configuration results in a remarkable improvement of 0.1 in the validation F1-score. This significant enhancement highlights the critical importance of a symmetrical matrix

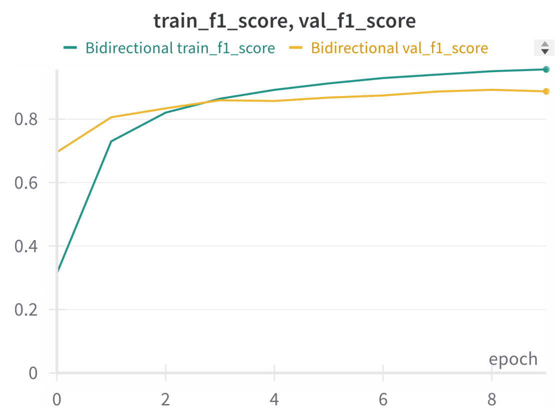

Figure 4: F1 score in training and validation of the Bidirectional Model

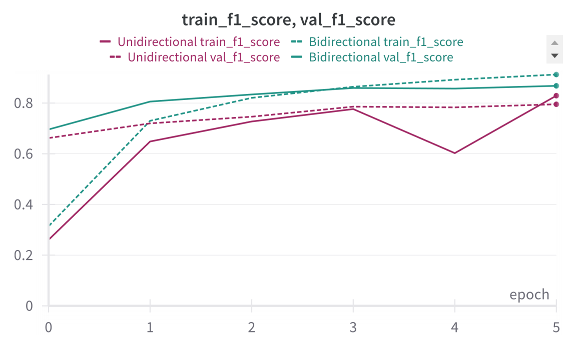

Figure 5: Unidrectional vs Bidirectional

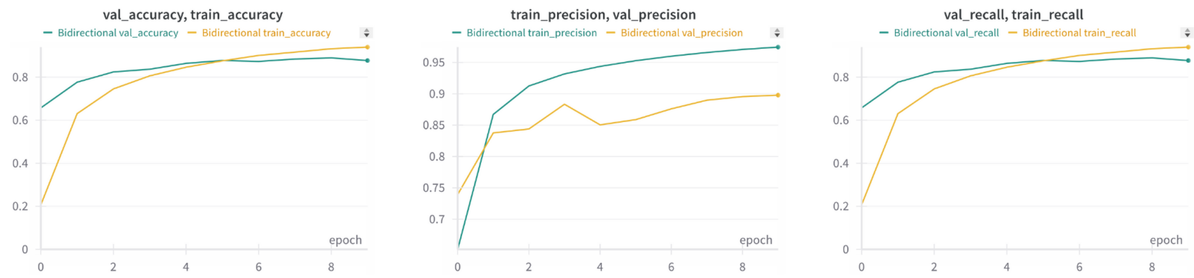

Figure 6: Accuracy, precision and recall of the triplets extraction from the bidirectional model

structure in augmenting the model's capacity to decipher and interpret the complex interactions between entities more accurately.

Parameter	Value	Parameter	Value
embeddings	BERT	optimizer	Adam
epochs	10	lr	1e-4
threshold	0.5	weight decay	0.01
bidirectional	true	loss	BCE
batch size	64	att heads	12

Table 1: List with the best hyperparameters found

6 Conclusion

This work builds on the UniRel framework, incorporating a dynamic padding strategy to boost efficiency. The apprach involves leveraging the attention mechanism of transformers to retrieve complex textual relationships. The model selected in this study constructs three separate matrices to dissect the interactions among subjects, objects, and predicates. This systematic approach enables the model to learn the dynamics of relations with remarkable results. The performance of the model are convining demonstrating its proficiency in accurately identifying relationships. The introduction of bidirectional matrices further boosts the model's performance, underscoring the value of symmetrically considering entity interactions. Relation Extraction is acknowledged as a challenging area within Natural Language Processing. Yet, the success of our model illustrates that, with innovative approaches and effective utilization of neural network architectures, overcoming these challenges is within reach. The implications of mastering Relation Extraction are vast, offering potential breakthroughs in how machines understand and interpret human language.

Model	Train	Val	Test
Unidirectional	0.77	0.79	0.79
Bidirectional	**0.96**	**0.89**	**0.90**

Table 2: F1-scores of the models in the train, validation and test set

References

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wei Tang, Benfeng Xu, Yuyue Zhao, Zhendong Mao, Yifeng Liu, Yong Liao, and Haiyong Xie. 2022. Unirel: Unified representation and interaction for joint relational triple extraction.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

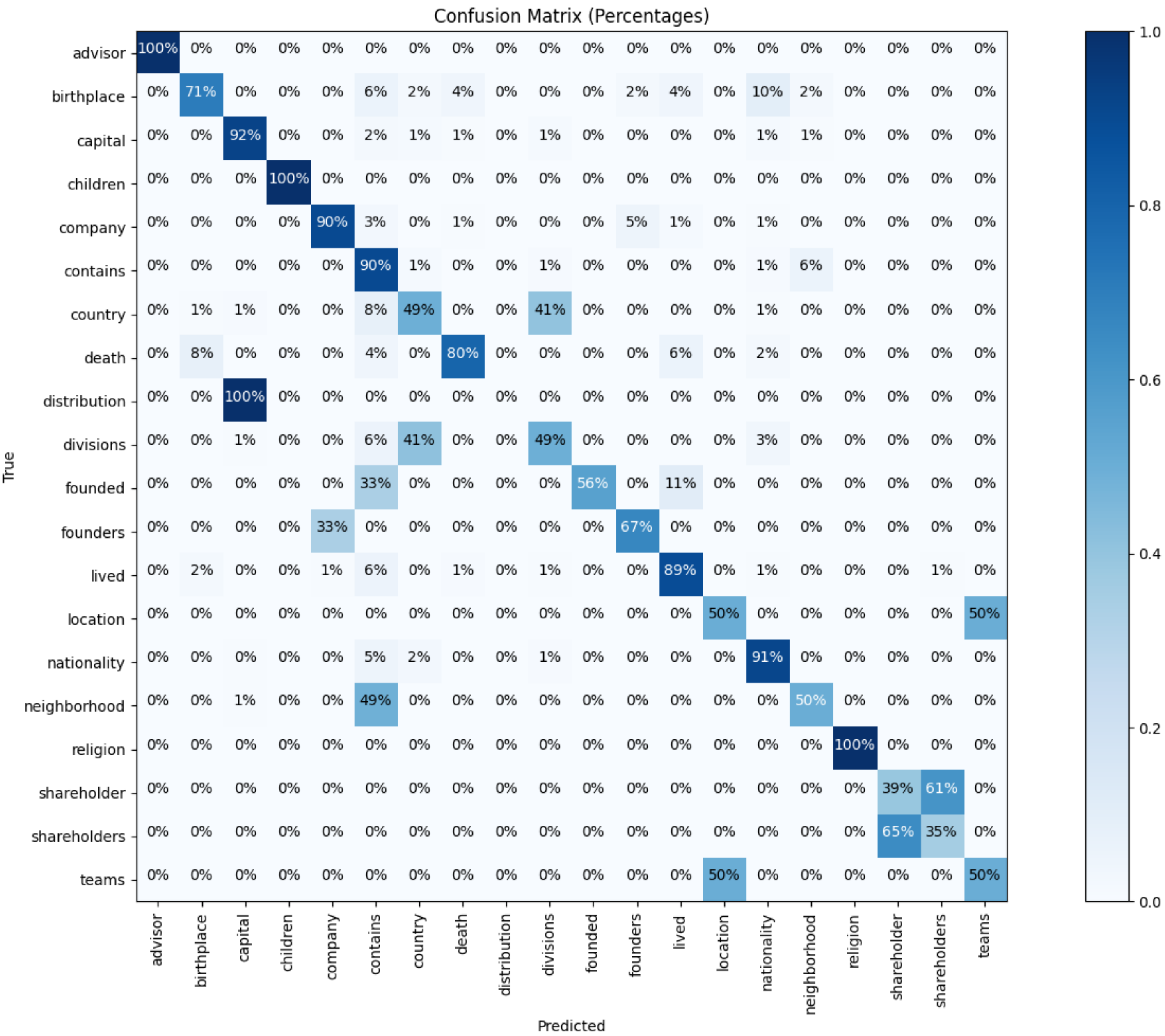

Figure 7: Confusion Matrix for Predicate Predictions. The matrix illustrates the model's performance in predicting predicates. Notably, the model achieves high accuracy in most cases, accurately predicting the majority of predicates. However, it struggles with rare predicates, such as 'distribution', indicating potential data scarcity issues. Additionally, the similarity between some incorrect predictions, such as 'death' and 'birthplace', suggests that the embeddings of these predicates may be close, contributing to the misclassification.