Simple Ideas at CLinkaRT: LeaNER and MeaNER Relation Extraction

Marius Micluța-Câmpeanu[1], Liviu P. Dinu[1,2]

[1]Faculty of Mathematics and Computer Science, University of Bucharest, Romania
[2]Human Language Technologies Research Center, University of Bucharest, Romania

Abstract

In this paper, we present our approach for performing relation extraction on clinical texts in the context of the CLinkaRT task at EVALITA 2023. Our system ranked first in this task with an F1-score of 62.99, outperforming most other submissions by a significant margin, with an increase of 6.5% over the second best score of 59.16, while also improving over the mBERT baseline of 62.83. We pursue a simple yet unexplored method to determine sentence level relations in text by relying on Named Entity Recognition models to identify the components of a relation. We apply this method to link laboratory results to their appropriate events in medical reports.

Keywords

EVALITA, CLinkaRT, named entity recognition, relation extraction, transformers

1. Introduction

The availability of vast quantities of textual data in the biomedical domain from digital repositories like PubMed Central has led to the development of highly specialized resources and language models [1]. Nonetheless, most of these efforts have been focused on English, while other less-resourced languages were largely neglected due to lack of available datasets.

The typical approach for downstream tasks in these languages is to resort to multilingual models, such as mBERT [2]. The rising need for pretrained models in languages other than English for biomedical applications materialized in the past few years with the advent of BioBIT/MedBIT for Italian [3] and similar models for other lower-resource languages: Spanish [4], Turkish [5] and French [6].

In the context of creating better systems for Italian, the CLinkaRT shared task [7] at EVALITA 2023 [8] challenges participants to detect laboratory measurements and tests from clinical records in order to associate them with their corresponding results. The relevance of developing and improving relation extraction tasks is highlighted in the literature, since it provides the underlying core elements for creating advanced biomedical text mining systems. Some examples include discovering interactions between drugs, adverse effects, genes, chemicals and diseases; predicting inappropriate emergency room visits; generating educational documents; building interaction networks and unveiling effective treatment methods for complex symptoms [9, 10].

The dataset for CLinkaRT [7] is adapted from the Italian part of the E3C Corpus, a collection of clinical narratives in several European languages [11]. The relationship annotations for this task are given in a similar format to PubTator, with each relation pair on a separate line. A pair is given by the entity mentions involved in the form of start and end offsets for sources (RML entities or results) and targets (EVENT entities or test events). A source may be a multi-token entity, while a target is always a single token. These tokenizations are provided together with the dataset to minimize evaluation mismatches.

In the following sections, we describe our team's solution to perform relation extraction in the CLinkaRT task.

2. System description

2.1. Overview

Our proposed method for this challenge is comprised of two consecutive Named Entity Recognition (NER) models that identify sources and targets for each sentence, followed by linking the entities through post-processing to output relations in the required format. The target entities predicted by the first NER model are prefixed with a special token when fed to the second NER model, which in turn identifies source entities. Since we used a similar system for our submission in the TESTLINK twin task [12], we provide a shortened description of the implementation and focus more on experiments and findings carried specifically in the context of CLinkaRT.

EVALITA 2023: 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Sep 7 – 8, Parma, IT
marius.micluta-campeanu@unibuc.ro (M. Micluța-Câmpeanu); liviu.p.dinu@gmail.com (L. P. Dinu)
0000-0002-7559-6756 (L. P. Dinu)

2.2. Implementation details

The first NER model is trained to predict all target entities in a sentence. For instance, in the phrase "La creatinina oscillava tra 1,5–2 mg/dL con proteinuria sempre < 1 g die" there are two relations: "creatinina" target with "1,5–2 mg/dL" as source and "proteinuria" with "< 1 g die" as source. We begin by locating targets first because the annotations mark only the syntactic head of a target, e.g. *esami* "tests" is an appropriate target for both *esami colturali* "culture tests" and *esami ematici* "blood tests".

After determining all targets in a sentence, we transform the training examples to incorporate target locations directly in text by adding a special marker token `[T]` before each target token, which should help the second NER model find relevant source entities. This is a viable strategy to denote one-to-one, one-to-many and many-to-one relations between sources and targets, thus effectively eliminating the need of a relation classifier model. The target indicates just the syntactic head, so we do not add an end marker because it might hinder the second NER model's ability to properly learn representing adequate targets.

All relation types (one-to-one, one-to-many, many-to-one) are handled in a uniform manner. For every target in a sentence, we generate one sample with a single target marker `[T]`. In this regard, there is no difference between a source with multiple targets and several one-to-one relations. For one target with many sources, only a single example is created. This way, we augment our training data for the second NER model.

As an example with two one-to-one relations, the sentence "La creatinina oscillava tra 1,5–2 mg/dL con proteinuria sempre < 1 g die" has two targets, "creatinina" and "proteinuria". The samples for our second NER model will be:

(1) "La `[T]` creatinina oscillava tra 1,5–2 mg/dL con proteinuria sempre < 1 g die" with only "1,5–2 mg/dL" labeled as source

(2) "La creatinina oscillava tra 1,5–2 mg/dL con `[T]` proteinuria sempre < 1 g die" with only "< 1 g die" labeled as source

In the following example, there is one source linked to three targets: "Gli esami colturali (germi comuni, BK) risultavano negativi.". The source is "negativi", while the targets are "esami", "germi" and "BK". Three sentences will be added, all with a single source to be predicted ("negativi"):

(3) "Gli `[T]` esami colturali (germi comuni, BK) risultavano negativi."

(4) "Gli esami colturali (`[T]` germi comuni, BK) risultavano negativi."

(5) "Gli esami colturali (germi comuni, `[T]` BK) risultavano negativi."

We encode annotations for both NER models with standard IOB2 tags (inside, outside, beginning) for either sources (RML entities) or targets (EVENT entities). A regular relation extraction pipeline would first employ a NER model to determine sources and targets at the same time and then apply a relation classifier on all possible source-target pairs.

With our approach, the first NER model is tasked to predict just target entities, while the second NER model is trained solely with labels for source entities. The consequence is that our models have a lower number of possible labels, determined by fewer IOB2 tags, therefore improving prediction performance. Each model has 3 tags: beggining, inside, outside. While the NER model used to predict targets only denotes the target head, we still need "inside" tags due to sub-word splitting required by transformer models. Contrast this with a traditional pipeline that would have a NER model with 5 tags (2 beginning tags, 2 inside tags, 1 outside tag) followed by a relation classifier model.

2.3. Data augmentation

The CLinkaRT training dataset consists of 83 Italian documents with 658 annotated relations. Due to the limited number of examples, we decide to augment the initial dataset with contextual word embeddings using the `nlpaug` library [13]. For this process, we replace random words with other similar words in the embedding space, except for labeled tokens, since there is a risk of injecting noisy labels.

We preserve the annotated entities and the target marker `[T]` in the augmented examples, ignoring sentences with 9 words or less. Samples that are not identical in terms of word count are discarded because the original labels would be misplaced.

Given that one of our main concern is finding laboratory tests and measurements, many of these entities are numeric values. To further our data augmentation, we introduce tiny changes of ± 2 for decimal values (age, year or quantities with a higher tolerance) and ± 0.1 for real values (tests or percentages). In most cases, this process should not significantly alter the labeling.

The training set has a much greater number of negative samples (examples without relations) than positive samples. We augment each example one or more times, with positive instances denoted by `in_multiplier` and negative instances by `out_multiplier`. We use the multipliers shown in Table 1, where the NER-tgt model predicts targets and the NER-src model predicts sources. The second model requires fewer auxiliary examples because the preprocessing step created additional samples for

Multiplier type	NER-tgt	NER-src
`in_multiplier`	4	2
`out_multiplier`	1	1

Table 1
Data augmentation multipliers: for each NER model and example type (labeled/unlabeled), we repeat the augmentation one or more times

System	Precision	Recall	F1-score
vocab transfer	29.95	31.83	30.88
mBERT	61.37	**64.37**	62.83
Second best team	**71.10**	50.65	59.16
Our team	65.55	60.62	**62.99**

Table 2
Baselines and best systems for linking measurements to results in clinical documents

sentences with more target entities.

The bottleneck of this augmentation process is the library call that executes the transformation. Considering that the operation runs on GPU, it should be natural to attempt to speed up this step by augmenting several examples in parallel. While the `nlpaug` library has an API that allows augmenting multiple sentences at once and at first it appeared to work on a few samples in the train set, a significant number of augmented examples constructed by `nlpaug` turned out to be empty sentences due to limitations or issues of this library. The batch implementation required a bit of effort due to the need to apply different multipliers. Since this attempted optimization did not succeed, we resumed augmenting examples one by one.

2.4. Model training and inference

We implement our NER models as standard token classifiers with the help of HuggingFace Transformers library [14]. We perform fine-tuning on a model pretrained on Italian medical textbooks, web-crawled data and translated English PubMed abstracts, available as `IVN-RIN/medBIT-r3-plus` on HuggingFace Hub [3].

All of our training experiments are carried out by mostly preserving default parameters: AdamW optimizer with $5e^{-5}$ learning rate with linear decay and no warmup steps, $1e^{-2}$ weight decay, 8 samples per batch trained for 4 epochs, with 10% examples held out for validation. Both models are trained independently using gold labels, with the second NER model (NER-src) receiving the target marker tokens `[T]` from these gold annotations.

In inference mode, the models are asked to output source and target offsets with respect to the original raw text. For each sentence converted into an example, we store the offset of the first token. Since HuggingFace Datasets library employs a separate tokenization, we align the transformed concatenated sentences with the initial full texts by using spaCy [15].

3. Results

We conduct our experiments by creating a test set from the training set with a 90/10 split in order to simulate the final test set, switching to 10-fold cross-validation for selecting appropriate values for some of the parameters regarding training and augmentation. Although the models are trained at the sentence level, this split is by document id so we do not overestimate the model performance on unseen examples.

The main results for relation extraction in the CLinkaRT task are displayed in Table 2. Our team obtains the first place across all teams, with an F1-score of 62.99, an improvement of 6.5% in F1-score over the second best competing team of 59.16 and a slight increase over the mBERT baseline with a score of 62.83. We also achieve the highest recall of 60.62 among other participants, with the second best score of 50.65, while the mBERT baseline has a recall of 64.37. We improve the baseline precision of 61.37 with a 6.8% increase, reaching a score of 65.55.

We report the performance of our system on the validation set averaged across 10 folds together with the official results. We used cross-validation to carry out parameter and model selection. Besides these 10% reserved examples for testing, the models also set aside 10% of the remaining examples for validation and hyperparameter tuning. In spite of these efforts, we notice a possible tendency of overfit. One explanation for this phenomenon is the small size of the model validation set, with too few samples to properly adjust the parameters when training. Another related explanation is given by the high variation between some of the folds, with half of the folds obtaining F1-scores over 82%, while the other folds consistently scored lower, between 73% and 77% F1-score. Even so, it might simply be the case that the test set is intentionally constructed with novel situations to determine the performance on unseen data more accurately, which would justify the gap between test and validation.

Outside the evaluation window, we repeated the inference process a second time on the test set keeping the same parameters and achieved 64.09 F1-score, showing that our approach can outrun the other systems by a greater margin than in the official results. Still, this variation is caused by the nondeterministic nature of transformer networks. We plan to analyze the extent of this variation and to limit the randomness of our system.

System	Pr	Re	F1-score
MedBIT-R3-plus	81.71	79.06	80.20
MedBIT-R3-plus (no aug)	64.55	68.99	66.41
Italian BERT	71.22	69.09	69.99
DistilBERT	79.56	71.16	75.04
BioBIT	81.03	75.08	77.81

Table 3
Results on the development set to determine the transformer model to be used in experiments for parameter tuning

4. Discussion

In this section, we present some observations regarding the design choices of our implementation and conduct an error analysis.

The data augmentation process has three main parameters: the minimum number of words that should be replaced in a sentence, `min_aug`, and the two multipliers for positive and negative examples defined earlier as `in_multiplier` and `out_multiplier`. We pick values between 3 and 6 for `min_aug`, based on the number of failed replacements, fixing the value at 4 words. The reasoning behind this decision is that the augmentation library is sometimes unable to adequately generate valid examples due to misplaced or missing words, so the gold labels cannot be applied, in turn leading to fewer examples in the train set.

The multipliers are selected by cross-validation, stopping early in case of unsatisfactory results on the first folds. For `out_multiplier`, we vary this parameter between 0 and 2 for both NER models, while for `in_multiplier` we use values in the range 1–5 for NER-tgt and 1–4 for NER-src. Our experiments confirm that augmentation is also needed for negative samples. This step has a significant impact in our system, boosting the score on the validation set with over 20 percentage points in F1-score. As it would be expected, adding too many examples by using larger multipliers eventually induces overfit.

The main drawback of augmentation is the slow data generation. As we mentioned earlier, `nlpaug` runs sequentially, so we had to make educated guesses of what combinations of parameters to include in our experiments.

For most of our experiments, we rely on the model called MedBIT-R3-plus [3] accessible on HuggingFace Hub. In order to determine if this is the right choice, we briefly examine the effectiveness of other transformer models. We consider three alternative options: Italian BERT [16], the multilingual version of DistilBERT [17] and the BioBIT model trained only on medical textbooks [3]. Undoubtedly, Italian BERT is less suitable for this task, with a substantial drop in F1-score of 14 percentage points. The results of the multilingual model are averaged only over the first 5 folds (out of 10 folds) because the training process takes much longer. We believe the increased training time is not justified given that a much smaller model can achieve better results. For this reason, we do not conduct additional experiments with multilingual transformers. Concerning the BioBIT model, it offers slightly worse results than the MedBIT-R3-plus counterpart, as noted likewise in the original paper [3]. The results are summarized in Table 3, using data augmentation for all variants except where noted otherwise. Due to time constraints, we did not run additional experiments with these models.

In addition to typical false positives and false negatives, we observe types of errors that show the system is on the right track, but fails to output the exact offsets in the reference file.

There are a couple of incomplete entity spans:

(6) The true source is “pari 0 inferiori a 1.5 mg/dl” and the predicted source is “inferiori a 1.5 mg/dl”. Other similar examples: true source is “fino a 12.8 mg/dL”, predicted source is “12.8 mg/dL”; true source is “punte di circa 1200 pg/mL”, predicted source is “circa 1200 pg/mL”.

(7) The true target is “antitrombina”, while the predicted target is “anti”

The first situation appears due to modifying comparatives not being present or being scarcely existent in the training data, which one could solve with additional examples or through careful augmentation. The second issue seems to be a defect on our side that can be handled in post-processing by inspecting the initial tokenization.

Another common mistake is the prediction of one relation instead of two (or vice versa) in the case of intervals, which we explain by ambiguities in the training set. For example, our system outputs “1.9 – 2.5 mg/dl” linked with “creatininemia”, but there are two expected relations: “1.9” linked with “creatininemia” and “2.5 mg/dl” linked with “creatininemia”. Conversely, our system detects two relations, “sostanzialmente” linked with “obiettività” and “nei limiti di norma” linked with “obiettività”, while there is only one true relation, “sostanzialmente nei limiti di norma” linked with “obiettività”.

Lastly, a challenging facet of this task is the presence of reference values for some tests, which are picked up by our model, although they are not found as gold labels because they do not represent test results. Future work in this direction should find means to distinguish between reference values and actual measurements and test values.

5. Conclusion and future work

In this paper, we detailed our contribution in the CLinkaRT task [7] at EVALITA 2023 [8], demonstrating that intuitive solutions yield competitive results. Our proposed approach achieves the best F1-score among other systems in the task of correlating laboratory tests and measurements with their results, with a 6.5% improvement in F1-score over the second best contestant.

We present a straightforward strategy to extract sentence-level relations based on two plain NER models, illustrating the learning capabilities of transformer networks to solve challenging tasks with the help of special tokens. We intend to further explore this direction since NER models are well established and usually require fewer resources than alternative relation extraction (RE) models. The presented method is not limited to the clinical domain and it can be easily applied in other contexts, with the added benefit of shorter development cycles. In certain domains and applications, the overhead of a generic RE model may be unjustified if the relations in question are simple enough.

Data augmentation is a valuable, but underused technique in natural language processing contexts. We look forward to enhancing the augmentation procedure to account for in-domain information. Another area we believe to be worth pursuing is the handling of numeric values and ranges, either by finding a way to inject fuzzy intervals or by masking these values altogether, therefore simplifying the initial problem.

Our system implementation is available at https://gitlab.com/marius.micluta-campeanu/testlink-clinkart-2023 to encourage an open environment for future work.

References

[1] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371. doi:10.18653/v1/D19-1371.

[2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.

[3] T. M. Buonocore, C. Crema, A. Redolfi, R. Bellazzi, E. Parimbelli, Localising In-Domain Adaptation of Transformer-Based Biomedical Language Models, 2022. arXiv:2212.10422.

[4] C. P. Carrino, J. Armengol-Estapé, A. Gutiérrez-Fandiño, J. Llop-Palao, M. Pàmies, A. Gonzalez-Agirre, M. Villegas, Biomedical and Clinical Language Models for Spanish: On the Benefits of Domain-Specific Pretraining in a Mid-Resource Scenario, 2021. arXiv:2109.03570.

[5] H. Türkmen, O. Dikenelli, C. Eraslan, M. C. Çalli, S. S. Ozbek, Developing Pretrained Language Models for Turkish Biomedical Domain, in: 2022 IEEE 10th International Conference on Healthcare Informatics (ICHI), IEEE, 2022, pp. 597–598.

[6] Y. Labrak, A. Bazoge, R. Dufour, M. Rouvier, E. Morin, B. Daille, P.-A. Gourraud, DrBERT: A Robust Pre-trained Model in French for Biomedical and Clinical domains, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 16207–16221. URL: https://aclanthology.org/2023.acl-long.896.

[7] B. Altuna, G. Karunakaran, A. Lavelli, B. Magnini, M. Speranza, R. Zanoli, CLinkaRT at EVALITA 2023: Overview of the Task on Linking a Lab Result to its Test Event in the Clinical Domain, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[8] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, EVALITA 2023: Overview of the 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR.org, Parma, Italy, 2023.

[9] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, H. Liu, Clinical information extraction applications: A literature review, Journal of Biomedical Informatics 77 (2018) 34–49. URL: https://www.sciencedirect.com/science/article/pii/S1532046417302563. doi:https://doi.org/10.1016/j.jbi.2017.11.011.

[10] N. Perera, M. Dehmer, F. Emmert-Streib, Named Entity Recognition and Relation Detection for Biomedical Information Extraction, Frontiers in Cell and Developmental Biology 8 (2020). URL: https://www.frontiersin.org/articles/10.3389/fcell.2020.00673.

doi:10.3389/fcell.2020.00673.

[11] B. Magnini, B. Altuna, A. Lavelli, M. Speranza, R. Zanoli, The E3C Project: Collection and Annotation of a Multilingual Corpus of Clinical Cases, in: J. Monti, F. Dell'Orletta, F. Tamburini (Eds.), Proceedings of the Seventh Italian Conference on Computational Linguistics, volume 2769 of *CLiC-It*, CEUR-WS, Milan Italy, 2020, pp. 422–431.

[12] B. Altuna, R. Agerri, L. Salas-Espejo, J. J. Saiz, A. Lavelli, B. Magnini, M. Speranza, R. Zanoli, G. Karunakaran, Overview of TESTLINK at IberLEF 2023: Linking Results to Clinical Laboratory Tests and Measurements, Procesamiento del Lenguaje Natural 71 (2023).

[13] E. Ma, NLP Augmentation, https://github.com/makcedward/nlpaug, 2019.

[14] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[15] I. Montani, M. Honnibal, M. Honnibal, S. V. Landeghem, A. Boyd, H. Peters, P. O. McCann, jim geovedi, J. O'Regan, M. Samsonov, G. Orosz, D. de Kok, M. Blättermann, D. Altinok, S. L. Kristiansen, M. Kannan, R. Mitsch, R. Bournhonesque, Edward, L. Miranda, P. Baumgartner, R. Hudson, E. Bot, Roman, L. Fiedler, R. Daniels, W. Phatthiyaphaibun, G. Howard, Y. Tamura, spaCy: Industrial-strength Natural Language Processing in Python, 2023. URL: https://doi.org/10.5281/zenodo.7715077. doi:10.5281/zenodo.7715077.

[16] S. Schweter, Italian BERT and ELECTRA models, 2020. URL: https://doi.org/10.5281/zenodo.4263142. doi:10.5281/zenodo.4263142.

[17] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).