Deeper Clinical Document Understanding Using Relation Extraction

Hasham Ul Haq[1], Veysel Kocaman[1] and David Talby[1]

[1]John Snow Labs inc. 16192 Coastal Highway, Lewes, DE 19958, USA

Abstract

The surging amount of biomedical literature & digital clinical records presents a growing need for text mining techniques that can not only identify but also semantically relate entities in unstructured data. In this paper we propose a text mining framework comprising of Named Entity Recognition (NER) and Relation Extraction (RE) models, which expands on previous work in three main ways. First, we introduce two new RE model architectures – an accuracy-optimized one based on BioBERT and a speed-optimized one utilizing crafted features over a Fully Connected Neural Network (FCNN). Second, we evaluate both models on public benchmark datasets and obtain new state-of-the-art F1 scores on the 2012 i2b2 Clinical Temporal Relations challenge (F1 of 73.6, +1.2% over the previous SOTA), the 2010 i2b2 Clinical Relations challenge (F1 of 69.1, +1.2%), the 2019 Phenotype-Gene Relations dataset (F1 of 87.9, +8.5%), the 2012 Adverse Drug Events Drug-Reaction dataset (F1 of 90.0, +6.3%), and the 2018 n2c2 Posology Relations dataset (F1 of 96.7, +0.6%). Third, we show two practical applications of this framework – for building a biomedical knowledge graph and for improving the accuracy of mapping entities to clinical codes. The system is built using the Spark NLP library which provides a production-grade, natively scalable, hardware-optimized, trainable & tunable NLP framework.

Keywords

Relation Extraction, Natural Language Understanding, Natural Language Processing, BERT, Spark, Deep Learning, Adverse Drug Events

1. Introduction

Biomedical literature has witnessed exponential rise in the past decade. MEDLINE currently holds more than 26 million records from 5639 publications, and has indexed more than 5 million records in the past seven years alone [1]. Furthermore, public databases like *https://clinicaltrials.gov* have seen an explosion of trials data as the aftermath of the novel Covid-19 outbreak.

In addition, wide-spread adoption of Electronic Health Records (EHRs), has made copious amount of free-text data available in digital format. This unstructured data is usually documented by healthcare professionals during the course of patient care, such as clinical notes, discharge summaries, lab reports, and pathology reports [2]. While publications and literature are growing rapidly, there still lacks structured knowledge that can be easily processed by computer programs. Relation Extraction becomes even more pertinent in biomedical research as it can provide the critical links required to generate knowledge graphs for better analysis and research, and even text summarization. Relating entities also help us improve medical coding by enriching vanilla entity chunks with surrounding information to get more accurate codes.

Relation extraction is generally regarded as a classification problem where entity pairs - usually identified by NER models - are classified to determine their relationship type in a given context. These models are trained to identify semantic relations between recognized entities as illustrated in Figure 1.

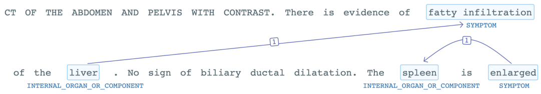

CT OF THE ABDOMEN AND PELVIS WITH CONTRAST. There is evidence of fatty infiltration
SYMPTOM
1
1
of the liver . No sign of biliary ductal dilatation. The spleen is enlarged
INTERNAL_ORGAN_OR_COMPONENT
INTERNAL_ORGAN_OR_COMPONENT
SYMPTOM

Figure 1: A Relation Extraction model semantically relating symptoms and body organs in a sample text.

Since the classification implicitly relies on context, transformers based models like BERT [3] have been shown to outperform traditional methods of dependency parsing. Recently, there is also an increasing trend of jointly training large BERT models on NER and RE tasks with shared layers and features [4]. However, even in joint learning, the RE classification is still contingent upon entity spans identified by the NER model.

While the trend of training large transformer models continues, applying them on large datasets remains a challenge as they require significant computational resources. Furthermore, long documents containing high number of entity spans can exponentially increase probable entity pairs for RE classification - requiring signifi-

SDU 2022 workshop at AAAI

hasham@johnsnowlabs.com (H. U. Haq); veysel@johnsnowlabs.com (V. Kocaman); david@johnsnowlabs.com (D. Talby)
0000-0002-8417-3288 (H. U. Haq); 0000-0002-0065-6478 (V. Kocaman); 0000-0003-2782-5478 (D. Talby)

cantly more resources and processing time.

In this study we focus on three major aspects of RE; the model architectures and their scalability, evaluating the models on benchmark datasets, and training and using RE for general use-cases. We also study the application of RE for understanding different aspects of clinical documents like extracting and relating dates to generate timeline of a patient's data on a timeline, or parsing and understanding trial results on large cohorts for analysis.

Following are the novel contributions of this paper:

- Introducing two new RE architectures.
- Evaluating and comparing performance of the proposed models on benchmark datasets.
- Training the models on custom datasets and demonstrating how RE can be used to get a structured output for specific use-cases.
- Studying the use-case of putting the history and medical history of patients on a timeline.
- Analyzing the benefits of using RE to get more precise entity chunks for achieving better performance while mapping them to medical codes.

2. Approach

We treat RE as a classification problem where each example is a pair of biomedical entities appearing in a given context - the entities being NER chunks, and context being the sentence / entire document - and develop two novel solutions; the first one comprising of a simpler FCNN architecture for speed, and the second one based on the BioBERT [5] architecture for accuracy. We experiment both approaches and compare their results.

For our first RE solution we rely on entity spans and types identified by the NER model to develop distinct features to feed to an FCNN for classification. At first we generate distinct pairs of entities (e.g. symptom-treatment), and then generate custom features for each pair. These features include semantic similarity of the entities, syntactic distance of the two entities, dependency structure of the entire document, embedding vectors of the entity spans, as well as embedding vectors for 100 tokens within the vicinity of each entity. Figure 2 explains our model architecture in detail. We then concatenate these features and feed them to fully connected layers with leaky relu activation. We also use batch normalisation after each affine transformation before feeding to the final softmax layer with cross-entropy loss function. We use softmax cross-entropy instead of binary cross-entropy loss to keep the architecture flexible for scaling on datasets having multiple relation types.

Our second solution focuses on a higher accuracy, as well as exploration of relations across long documents, and is based on [6]. In our implementation, we implement the model in Apache Spark for scalability, take checkpoints from the BioBERT model, and train an end-to-end BERT model for RE. Similar to the first solution, this architecture also depends on the entity spans identified by the base NER model, and uses the entire document as context string while training the model. The original paper used sequence length of 128 tokens for the context string, which we keep constant, and instead experiment with the content of the context string, training data augmentation, and fine-tuning techniques.

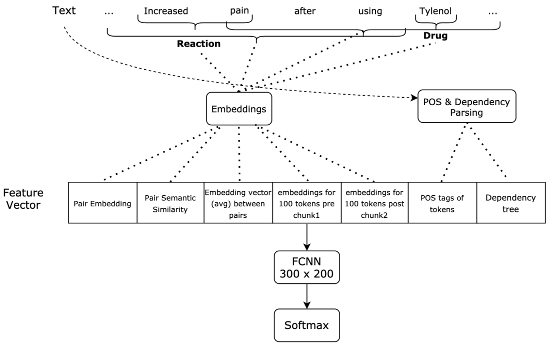


Figure 2: Overview of the first RE model. All the features are vertically stacked in a single feature vector. The feature vector is kept dynamic with additional padding for compatibility across different embedding sizes, and complex dependency structures.

We use Spark NLP's [7] NER models [8] as foundation for the RE models as these NER models provide entity spans required for performing RE. In a single inference pipeline, the RE models are placed sequentially after the the NER model, and are fed the results of the NER model, the context, embeddings, and dependency tree for feature generation. Apart from feature generation, the dependency tree also helps regularize candidate entity pairs for RE classification as we can eliminate pairs having a larger syntactic distance. This modular approach of arranging components reduces coupling and achieves a higher degree of memory and computational efficiency as components like sentences, tokens, and embeddings are shared between NER and RE models and don't need to be executed again. Since the NER model is essentially a token classifier and produces prediction per token, we convert the tokens to chunks using BIO tags.

3. Experiments

We test the models on public datasets, report evaluation metrics, and analyse the results on examples. In addition to public datasets, we explain the process of annotating and training models on new datasets. We then study the utility of applying RE for some use-cases like knowledge graph generation and improved entity resolution (the process of mapping entity chunks to medical codes).

3.1. Performance on Public Datasets

We test both model architectures on seven public datasets by using the official training-test split for training and testing the models, and report macro-average f1 scores for each one of them in Table 1. These datasets include the 2012 i2b2 challenge for evaluating temporal relations in clinical text [9], the 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text [10], the Drug-Drug-Interaction (DDI) dataset for linking drugs with dispositions and reactions [11], the Chemical–protein interaction (CPI) dataset for linking genes/proteins with drug chemicals [12], the Phenotype-Gene Relations (PGR) dataset for relating human phenotypes and genes [13], the adverse drug events dataset for relating drugs with their reactions [14], and the posology relations task based on the 2018 n2c2 task [15]. For the sake of brevity we don't delve into the details for each dataset, and specific details for each dataset can be found in the cited resources. As explained in Table 1, the BERT model achieves new SOTA metrics on 5 public datasets, and out performs the lighter FCNN model due to better contextual awareness. However, it is more than 3 time slower and has much higher memory requirements. Table 2 compares the speed difference of the two architectures. Hyperparameter setting and sample Python code for training an RE model from scratch is in Appendix A & C.

Dataset	FCNN	BioBERT	Curr-SOTA
i2b2-Temporal	68.7	**73.6**	72.41
i2b2-Clinical	60.4	**69.1**	67.97
DDI	69.2	72.1	**84.1**
CPI	65.8	74.3	**88.9**
PGR	81.2	**87.9**	79.4
ADE Corpus	89.2	**90.0**	83.7
Posology	87.8	**96.7**	96.1

Table 1
Macro-averaged F1 scores of both RE models on public datasets. FCNN refers to the Speed-Optimized FCNN architecture, while BioBERT refers to the Accuracy-Optimized BioBERT architecture. The SOTA metrics are obtained from [16], [17], [18], [19], [20], [21], and [22] respectively.

3.2. Performance on Private Datasets

In addition to the public datasets, we sampled approximately 5000 clinical notes and manually annotated them with the help of domain experts on the following guidelines: We selected general entities (e.g, body part, date, test result) that can compliment core entities (e.g, symptom, procedure, test) as the first entity and disjoint entity types - meaning the entities should not have relation among themselves - from the the core entities for the second entity as explained in Table 3. Since the first entity can relate to multiple entities in the second column, we can define the relation between the two entity types as one-to-many, and can keep the relation types to a minimum i.e. are the two entities related or not. This approach helps reduce annotation complexity resulting in faster annotation times, and a higher inter-annotator agreement. For annotation purposes we utilized the publicly available Annotation Lab tool.

Dataset	FCNN RE Model	BERT RE Model
1k Notes	104	584
10k Notes	925	5197

Table 2
Time in seconds taken by both models to run a sample batch of documents on a single linux machine with with 64GB of RAM, and an 8-Core CPU (without GPU hardware acceleration).

model	Entity 1	Entity 2
re_bodypart_procedure_test	Body Part	Procedure Test
re_test_result_date	Test	Test Result Date
re_bodypart_problem	Body Part	Symptom
re_test_problem_finding	Test	Symptom
re_bodypart_directions	Body Part	Direction

Table 3
Entity types in column *Entity 2* do not have relations among themselves

4. Practical Applications of Relation Extraction

The ability to semantically relate entities paves way for a lot of opportunities and use-cases. For example, the RE model for Adverse Drug Events can be used to identify drugs that caused reactions in large trial datasets. Figure 3 shows the output of running the ADE RE model on sample text. Similarly, lab results, discharge notes, and prescriptions can be parsed to get a structured output as illustrated in Figure 4.

In addition of using the public models, following are some of the use-cases we explored with our general-purpose models:

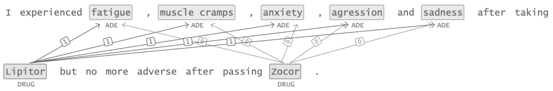

I experienced fatigue , muscle cramps , anxiety , agression and sadness after taking
ADE
ADE
ADE
ADE
ADE
1
1
1
1
0
1
0
0
0
0
Lipitor but no more adverse after passing Zocor .
DRUG
DRUG

Figure 3: Output of the ADE RE model on sample data. Arrows with *0* represent the two entities are not related, while *1* represents that the reaction is caused by the drug.

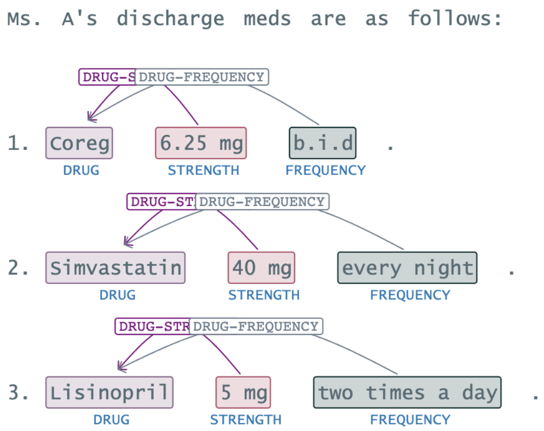
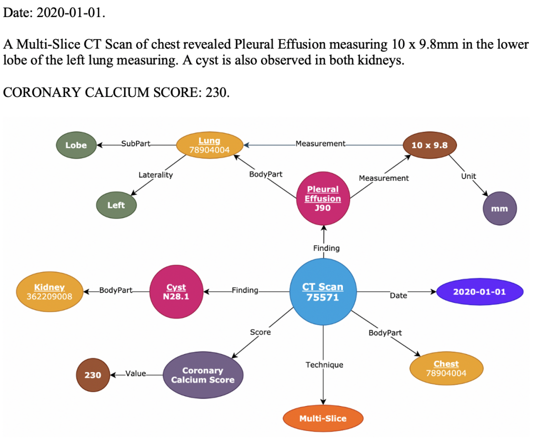


Figure 4: Output of the Posology RE model on sample data.

4.1. Generating Knowledge Graph with Relations

Most notable benefit of RE is the ability to generate knowledge graphs from unstructured text. For this experiment, we used pretrained Spark NLP NER models and the general-purpose RE models explained in the previous section to process medical reports with the primary goal of generating a concise structured output of a report. For instance, we relate procedures with dates and findings to recognize dates of a procedure and its findings along with any existing condition. We use the relations between body parts and procedures to get more specific details of the location of the procedure. Similarly, relating body parts with findings like test results and measurements can add more details to the final output in specific use-cases. More granularity can be achieved by having further subdivisions of body parts. For instance, in our experiment, we divide the body part in three parts; the primary body part (e.g, lung), a sub-part (e.g, lobe), and direction/laterlity (e.g, left) of the body part. In practice, these specifc entities trickle from the NER model down to the RE models. A graph generated from a sample report can be seen in Figure 5.


Date: 2020-01-01.

A Multi-Slice CT Scan of chest revealed Pleural Effusion measuring 10 x 9.8mm in the lower lobe of the left lung measuring. A cyst is also observed in both kidneys.

CORONARY CALCIUM SCORE: 230.

Figure 5: A graphical representation (with CPT, ICD & SNOMED codes) of the structured data extracted from a sample text.

Furthermore, the structured data can help create a patient timeline which can show progress of a certain condition over a certain duration. A sample timeline monitoring coronary calcium score and cyst can be seen in Figure 6. Such information can be used to analyse multiple trends like effectiveness of a drug for treating a certain condition on large datasets.

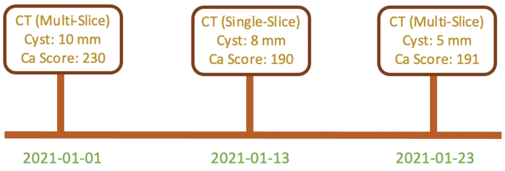


Figure 6: A sample timeline of a patient showing calcium score trend, and evolution of cyst over multiple scans in a month.

4.2. Enriching Chunks for more Accurate Coding

Entity Resolver models map entity chunks to medical codes like CPT [23], ICD [24], SNOMED [25], MeSH [26], RxNorm [27] etc based on semantic similarity. This task becomes challenging due to two major reasons. First, the inherent noise of the text like abbreviations, acronyms, and synonyms can result in false positive results. Second, medical codes are sensitive to variables like severity, location in human body, administration type, diagnosis method, etc; For a given condition or treatment, there could be different codes (within the same ontology) depending on the aforementioned factors. This challenge

Ontology	Base Chunk	Base Code	Enriched Chunk	Enriched Code
CPT	CT Scan	3324F	CT Scan Chest (multi-slice) contrast	71260
SNOMED CT	CT Scan	169072007	CT Scan Chest (multi-slice) contrast	169069000
ICD-10	Lesion	L98. 9	Lesion liver	K76. 9
SNOMED	Lesion	300577008	Lesion liver	300331000

Table 4
Comparison of entity resolution results - more enriched and specifc entity chunks result in a more accurate code.

is more prominent in ontologies with wider vocabularies like SNOMED.

RE provides solution to both problems; First, it intrinsically cleans the input for the resolver models of stop words and noise without additional effort. Second, it adds additional information to the core entity chunks from surrounding context; With the help of relations, simple entities can be enriched with precise information to get accurate codes. For example, a chunk *CT Scan* - identified as a procedure - can be enriched with the imaging technique to achieve a more accurate CPT/SNOMED code. Enriching it further with the location of the procedure (e.g, chest) would result in an even accurate chunk that can be resolved to a more specific CPT/SNOMED code. Table 4 compares base chunks with enriched chunks that include body parts, demonstrating the benefits of enriched entity chunks for improved coding.

5. Conclusion

In this paper we presented two new model architectures for RE while enabling scalability. We then tested the models on public datasets and reported evaluation metrics. The model metrics show that the BioBERT based model outperforms the lighter FCNN model, and obtains new state-of-the-art accuracy on on three benchmarks. However, for datasets with a small number of relation types, the simpler FCNN model may be a compelling option not only due to faster run times, but also much lower memory requirements compared to the BioBERT model, allowing to process larger datasets on commodity hardware. We also explain how to train RE models from scratch and describe the design behind the pre-trained models available as part of Spark NLP library.

We then study practical use cases where RE plays the salient role of linking entities together to generate knowledge graphs, patient timelines, and structured summaries of medical notes. Relating dates to primary procedures and problems can help create a timeline for each patient. Finally, using granular NER models together with discrete RE models to clean and enrich entity chunks enables better entity resolution to clinical codes.

Given the complex nature of RE, and the pivotal role of contextual information, a common approach is to limit relations within a certain syntactic span as even BERT models have token sequence limit. A future research direction could be to focus on improving contextual representation of large documents to allow relations over lengthy contextual spans. A second future research direction is to test whether auxiliary data - either from medically annotated data or through transfer learning from healthcare-specific language models - can deliver higher accuracy Relation Extraction on the same neural network architectures.

References

[1] S. Yadav, S. Ramesh, S. Saha, A. Ekbal, Relation extraction from biomedical and clinical text: Unified multitask learning framework, CoRR abs/2009.09509 (2020). URL: https://arxiv.org/abs/2009.09509. arXiv:2009.09509.

[2] Q. Wei, Z. Ji, Y. Si, J. Du, J. Wang, F. Tiryaki, S. Wu, C. Tao, K. Roberts, H. Xu, Relation extraction from clinical narratives using pre-trained language models, in: AMIA Annual Symposium Proceedings, volume 2019, American Medical Informatics Association, 2019, p. 1236.

[3] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[4] J. Wang, W. Lu, Two are better than one: Joint entity and relation extraction with table-sequence encoders, CoRR abs/2010.03851 (2020). URL: https://arxiv.org/abs/2010.03851. arXiv:2010.03851.

[5] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, CoRR abs/1901.08746 (2019). URL: http://arxiv.org/abs/1901.08746. arXiv:1901.08746.

[6] L. B. Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the blanks: Distributional similarity for relation learning, CoRR abs/1906.03158 (2019). URL: http://arxiv.org/abs/1906.03158. arXiv:1906.03158.

[7] V. Kocaman, D. Talby, Spark nlp: Natural language

understanding at scale, Software Impacts 8 (2021) 100058. URL: https://www.sciencedirect.com/science/article/pii/S2665963821000063. doi:https://doi.org/10.1016/j.simpa.2021.100058.

[8] V. Kocaman, D. Talby, Improving clinical document understanding on COVID-19 research with spark NLP, CoRR abs/2012.04005 (2020). URL: https://arxiv.org/abs/2012.04005. arXiv:2012.04005.

[9] W. Sun, A. Rumshisky, O. Uzuner, Evaluating temporal relations in clinical text: 2012 i2b2 challenge, Journal of the American Medical Informatics Association : JAMIA 20 (2013). doi:10.1136/amiajnl-2013-001628.

[10] Ö. Uzuner, B. R. South, S. Shen, S. L. DuVall, 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text, Journal of the American Medical Informatics Association 18 (2011) 552–556.

[11] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, T. Declerck, The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions, Journal of Biomedical Informatics 46 (2013) 914–920. URL: https://www.sciencedirect.com/science/article/pii/S1532046413001123. doi:https://doi.org/10.1016/j.jbi.2013.07.011.

[12] M. Krallinger, O. Rabal, S. A. Akhondi, M. P. Pérez, J. Santamaría, G. P. Rodríguez, G. Tsatsaronis, A. Intxaurrondo, J. A. B. López, U. K. Nandal, E. M. van Buel, A. Chandrasekhar, M. Rodenburg, A. Lægreid, M. A. Doornenbal, J. Oyarzábal, A. Lourenço, A. Valencia, Overview of the biocreative vi chemical-protein interaction track, 2017.

[13] D. Sousa, A. Lamurias, F. M. Couto, A silver standard corpus of human phenotype-gene relations, 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. 1487–1492. URL: https://aclanthology.org/N19-1152. doi:10.18653/v1/N19-1152.

[14] H. Gurulingappa, A. M. Rajput, A. Roberts, J. Fluck, M. Hofmann-Apitius, L. Toldo, Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports, Journal of Biomedical Informatics 45 (2012) 885–892. URL: https://www.sciencedirect.com/science/article/pii/S1532046412000615. doi:https://doi.org/10.1016/j.jbi.2012.04.008, text Mining and Natural Language Processing in Pharmacogenomics.

[15] S. Henry, K. Buchan, M. Filannino, A. Stubbs, O. Uzuner, 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records, Journal of the American Medical Informatics Association 27 (2020) 3–12.

[16] H. Guan, J. Li, H. Xu, M. V. Devarakonda, Robustly pre-trained neural model for direct temporal relation extraction, CoRR abs/2004.06216 (2020). URL: https://arxiv.org/abs/2004.06216. arXiv:2004.06216.

[17] D. Ningthoujam, S. Yadav, P. Bhattacharyya, A. Ekbal, Relation extraction between the clinical entities based on the shortest dependency path based LSTM, CoRR abs/1903.09941 (2019). URL: http://arxiv.org/abs/1903.09941. arXiv:1903.09941.

[18] M. Asada, M. Miwa, Y. Sasaki, Using drug descriptions and molecular structures for drug–drug interaction extraction from literature, Bioinformatics 37 (2020) 1739–1746. URL: https://doi.org/10.1093/bioinformatics/btaa907. doi:10.1093/bioinformatics/btaa907. arXiv:https://academic.oup.com/bioinformatics/article-pdf/37/12/1739/39119268/btaa907.pdf.

[19] L. N. Phan, J. T. Anibal, H. Tran, S. Chanana, E. Bahadroglu, A. Peltekian, G. Altan-Bonnet, Scifive: a text-to-text transformer model for biomedical literature, CoRR abs/2106.03598 (2021). URL: https://arxiv.org/abs/2106.03598. arXiv:2106.03598.

[20] D. Sousa, F. M. Couto, Biont: Deep learning using multiple biomedical ontologies for relation extraction, CoRR abs/2001.07139 (2020). URL: https://arxiv.org/abs/2001.07139. arXiv:2001.07139.

[21] P. Crone, Deeper task-specificity improves joint entity and relation extraction, CoRR abs/2002.06424 (2020). URL: https://arxiv.org/abs/2002.06424. arXiv:2002.06424.

[22] X. Yang, Z. Yu, Y. Guo, J. Bian, Y. Wu, Clinical relation extraction using transformer-based models, CoRR abs/2107.08957 (2021). URL: https://arxiv.org/abs/2107.08957. arXiv:2107.08957.

[23] AMA, Cpt, https://www.ama-assn.org/amaone/cpt-current-procedural-terminology, 2020. Accessed: 2021-12-22.

[24] WHO, Icd10, https://www.who.int/standards/classifications/classification-of-diseases, 2019. Accessed: 2021-12-22.

[25] NLM, Snomed ct, https://www.nlm.nih.gov/healthit/snomedct/index.html, 2019. Accessed: 2021-12-22.

[26] NLM, Mesh, https://www.nlm.nih.gov/mesh/meshhome.html, 2021. Accessed: 2021-12-22.

[27] NLM, Rxnorm, https://www.nlm.nih.gov/research/umls/rxnorm/index.html, 2021. Accessed: 2021-12-22.

[28] JSL, Training code for re, https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Healthcare/10.3.

Clinical_RE_SparkNLP_Paper_Reproduce.ipynb, 2021. Accessed: 2021-12-23.

A. A. Hyperparameter Settings

Since optimal hyperparameter values vary for each dataset, a range of values which performed best in all the datasets can be seen in Table 5.

Parameter	FCNN	BERT	Tested Range
Dropout rate	0.5	0.2	0.2-0.5
Batch size	64 - 128	64	8-128
Learning rate	0.0003	0.001	0.0001-0.005
Epoch	50 - 70	2-5	1-100
Optimizer	Adam	Adam	Adam
LR Decay	0.005	0.005	0.002-0.005

Table 5
Best performing hyperparameter value ranges on multiple datasets.

B. B. Preparing training data for RE model in Spark NLP

Since RE is a classification task, the primary inputs are the context string (sentence), and a pair of entities. If there are multiple pairs in a single context string, we treat them as disjoint inputs as each input encapsulates the required inputs like entity chunk pairs and context - which are then used to create input features. We can create a csv formatted file where each row is a training example for the model, and contains the aforementioned inputs. Exact schema of the training file can be found in the training notebook [28].

C. C. Training an RE Model in Spark NLP

Code for training an RE mode is provided as a google colab notebook [28]. As majority of the public datasets are protected and can not be shared, they need to be obtain from their official websites and converted to the required format before training.