Chemical detection and indexing in PubMed full text articles using deep learning and rule-based methods

Tiago Almeida[1], Rui Antunes[1], João Figueira Silva[1], João Rafael Almeida[1,2], and Sérgio Matos[1§]
[1]DETI/IEETA, University of Aveiro, Aveiro, Portugal
[2]Department of Computation, University of A Coruña, Spain
§ Corresponding author. E-mail: aleixomatos@ua.pt.

***Abstract*—Identifying chemicals in biomedical scientific literature is a crucial task for drug development research. The BioCreative NLM-Chem challenge promoted the development of automatic systems that can identify chemicals in full-text articles and decide which chemical concepts are relevant to be indexed. This work describes the participation of the BIT.UA team from the University of Aveiro, where we propose a three-stage automatic pipeline that individually tackles (i) chemical mention detection, (ii) entity normalization and (iii) indexing. We adopted a deep learning solution based on a biomedical BERT variant for chemical identification. For normalization we used a rule-based approach and a hybrid version that explores a dense retrieval mechanism. Similarly, for indexing we also followed two distinct approaches: a rule-based, and a TF-IDF based method. Our best official results are consistently above the official median and benchmark in the three subtasks, with respectively 0.8454, 0.8136, and 0.4664 F1-scores.**

Keywords—chemical identification; named entity recognition; normalization; chemical indexing; deep learning; transformer based model.

I. Introduction

Automatic information extraction from biomedical scientific literature is an essential step for helping in curation tasks, although it is a challenging task far from being solved (1). Particularly, the identification of chemical names advances drug development research. This task, known as named entity recognition (NER), is usually followed by a normalization step where entity mentions are linked to unique codes from a standard vocabulary. Predominantly, only PubMed abstracts have been used for assessing biomedical information extraction systems, as despite the added value of using the extra information in PubMed full-text articles, these pose new challenges stemming from the more detailed explanations and statements, and more complex writing style when compared to abstracts. PubMed provides biomedical researchers, biologists, pharmacologists, epidemiologists, physicians (and others) a way to search for the most relevant research articles. Offering accurate search results expedites their work, and to improve the quality of PubMed search results it is imperative that related information is added to every article. MeSH (Medical Subject Headings) identifiers are used to index articles in PubMed, however, the addition of the appropriate MeSH identifiers for each article is performed manually in a process that costs time and requires expertise. The BioCreative VII Track 2 (NLM-Chem) challenge (2) aims to bring the text mining community to tackle this issue. Participating teams are encouraged to develop computerized solutions and share their systems, since automatic annotations may help expert curators with their manual work.

In this paper we describe the methods from our participation in BioCreative VII Track 2 (NLM-Chem). This track comprises two tasks: (i) chemical identification and (ii) chemical indexing. In the first task, the goal is to recognize chemical mentions (named entity recognition) and link predicted entities to their respective MeSH identifiers (normalization). The second task aims to predict the chemical MeSH identifiers that should be used to index each document (that is, find the more relevant MeSH terms for each document).

II. Data

Task organizers provided two main datasets (3): *training* and *evaluation*, both consisting of PubMed full-text articles. The *training* dataset corresponds to the NLM-Chem corpus (4) containing 150 documents, whereas the *evaluation* dataset is comprised of 1387 documents that were scheduled for human indexing in 2021.

During the challenge we only had access to the ground truth annotations of the *training* dataset to develop our system. Regarding the *evaluation* dataset, only a subset of those articles was manually annotated for the chemical identification task evaluation, whilst for the evaluation of chemical indexing task all articles were used (all documents were manually indexed by human curators).

To foster the implementation of enhanced systems, the organizers also shared two other compatible datasets that could help improving the participants' systems: CHEMDNER (5) and CDR (6). These datasets contain 10000 and 1500 documents respectively, but these documents correspond to PubMed abstracts and not full-text articles as in the NLM-Chem dataset. Both datasets contain the chemical mention annotations and the chemical MeSH indexing identifiers, but only the CDR dataset contains the MeSH identifiers for each chemical mention (normalization).

We also used other datasets for helping the NER part. We used the DrugProt *training* and *development* subsets provided in BioCreative VII Track 1 (DrugProt), since these contain manually annotated chemical mentions. Documents from DrugProt that also appear in CDR and CHEMDNER (repeated

T.A., R.A., J.F.S. and J.R.A. are funded by the FCT - Foundation for Science and Technology (Portuguese national funds) - under the grants 2020.05784.BD, SFRH/BD/137000/2018, PD/BD/142878/2018 and SFRH/BD/147837/2019, respectively.

TABLE I. DATASETS STATISTICS.

	Number of documents	Number of chemical mentions
NLM-Chem	150	38339
CHEMDNER	10000	84331
CDR	1500	15943
DrugProt (filtered)	2180	33866
CRAFT	22030	6802
BioNLP11ID	5178	973
BioNLP13CG	5942	2270
BioNLP13PC	5051	2487

PMIDs) were discarded. We also used some of the datasets prepared by Crichton et al. (7). The following datasets with chemical-related mentions were used: CRAFT, BioNLP11ID, BioNLP13CG, and BioNLP13PC. For more details on these corpora, we refer the reader to the original paper (7). In our work, we experimented using all the datasets as additional training data for our deep learning NER model. Table I presents brief statistics about these corpora.

III. METHODS

NLM-Chem track organizers split the problem into two main tasks: (i) chemical identification and (ii) chemical indexing. For simplicity, we decided to divide the first task into two individual subtasks: entity recognition, and normalization. We organized the following sections considering these three sub-tasks: (A) chemical recognition, (B) chemical normalization, and (C) chemical indexing. We view these subtasks as discrete objectives that an automatic system should solve. Therefore, our approach follows a three-stage pipeline, where each stage directly corresponds to a subtask.

A. Chemical recognition

The objective of the first stage of our pipeline is to detect the boundaries of chemical mentions in the raw document text. Our main method relies on the current state-of-the-art BERT (Bidirectional Encoder Representations from Transformers) model for creating contextualized word representations, that are then used as features to train a classifier model. More precisely, we adopted the PubMedBERT (8) variant that reports state-of-the-art results in almost every biomedical task, including NER. Regarding the classifier, we followed the BIO notation schema to discriminate if a sub-word belongs to an entity (B-Begin, I-Inside) or does not (O-Outside). Additionally, we also tried several architectural variants. However, we quickly found that a multilayer perceptron (MLP) followed by a CRF layer (conditional random field) yielded the best results whilst also being a simple architecture.

During model conceptualization we also wanted to take full advantage of the contextualization power of the transformer architecture. Therefore, we decided to set our input size to 512 tokens (max size of BERT), of which we only forward the 256 tokens in the center for classification whereas the remaining 256 tokens (128 to the left and 128 to the right) are only used for context. The full-text documents from the NLM-Chem dataset are divided into passages (sections) such as abstract, introduction, methods, and others, and each passage may comprise several sentences or paragraphs. Thus, we split a passage into successive sequences shifted by 256 tokens (left and right context is kept), and each sequence is fed into BERT. An advantage of this method is that we can sequentially feed each passage of the document without performing any additional splitting such as sentence or paragraph segmentation.

For model training, we treated it as a simple classification problem and adopted the modern AdamW optimizer and the non-monotonic Mish activation function for the MLP. Additionally, we also experimented with training the last layer of the PubMedBERT model in an end-to-end fashion w.r.t. the classifier. In this case, since even a single layer of the BERT model is very large, we explored its training with additional datasets (Table I). This strategy gives the model the opportunity to recognize entities that it has never seen, opposed to only using the NLM-Chem dataset, at the cost of structural data biases.

B. Chemical normalization

After detecting chemical entities using the NER approach described in Section III.A, a named entity normalization process was developed to convert entities to their corresponding MeSH codes. This normalization workflow was divided in two major components: (i) a rule-based system and (ii) a deep-learning solution based on transformers. To supply both normalization components with curated concept-code mappings, two dictionary files were created by filtering and restructuring the 2021 MeSH and SCR (Supplementary Chemical Records) files. During this filtering procedure, the MeSH file only retained concepts belonging to the "Drugs and Chemical" category as these were within the scope of the present challenge.

1) Rule-based component

The rule-based component attempts to map entities to their corresponding MeSH codes through exact matching mechanisms. The development of this component followed an incremental workflow as described next.

For the first iteration of the rule-based system, a simple dictionary was configured using only the base mappings from the MeSH filtered file, *i.e.* using only the DescriptorUI-DescriptorName mappings from the MeSH filtered file. Exact matching was then performed using raw text entities and lowercased entities, with the latter providing better results. Next, to assess the impact of the mapping dictionary in system performance, the dictionary was expanded to incorporate mappings from the entry terms related to each concept (DescriptorName), resulting in an improved performance.

Since it is common to find plenty of abbreviations within biomedical literature, an abbreviation expansion step was added to the rule-based system through the integration of the Ab3P tool (9). This step was added in two different configurations, the first storing a list of previously seen abbreviations per document and a second storing the same list per corpus. Obtained results showed an overall improved entity-code mapping process in all training data splits, with the corpus-level configuration obtaining better results than the document-level counterpart.

In the following iteration, the source dictionary was further expanded by adding mappings from the SCR file to the previously described dictionary (first system iteration). During this merging process, and following a similar approach as before, information from the entry terms and heading mappings related to each DescriptorName was also integrated in the dictionary. By exploring this novel source of information, the rule-based system attained improved better results throughout all data splits.

Since there was still a significant amount of entities that the system could not map, a partial matching mechanism was added to process and map the remaining non-mapped entities. To accomplish this, the MetaMap (10) based pyMeSHSim (11) Python package was integrated in the rule-based system. However, this partial matching mechanism was unsuccessful as (i) pyMeSHSim was very slow and thus unusable considering the large size of the test dataset, and (ii) pyMeSHSim yielded numerous false positives, consequently downgrading the rule-based system performance. As a result, this partial matching mechanism was removed from the solution.

Finally, complex mappings present in the gold-standard annotations (*e.g.* entities with multiple MeSH codes) were added to the source dictionary, improving its coverage, and a deep-learning component was used to process the remaining unmapped entities as described in Section III.B.2.

2) Deep learning component

Inspired by the undeniable success of the transformer architecture, we built a complementary component that uses a dense retrieval technique to map entities to their corresponding MeSH codes. This method consists of building a dense representation for each MeSH code and every recognized entity. Then, we measure the similarity between each entity representation and all MeSH code representations, returning the top MeSH code with a similarity above a specific threshold.

More precisely, we leverage the SapBERT (12) model to create the dense representations, also known as embeddings, for the entities and MeSH codes. SapBERT is a BERT-based model that was pretrained for biomedical entity representations by clustering similar biomedical terms. Despite not being directly trained on MeSH terms, we believe that the domains are closely related and, therefore, we used it as a zero-shot approach. We create the dense representations for each term (entity or MeSH DescriptorName) by feeding their associated textual representation to SapBERT. Next, we use the produced `[CLS]` embedding as the dense representation for each term, which was the same method used in SapBERT. As similarity measure, we adopted the traditional cosine similarity between the two embeddings.

Due to computational limitations, we were only able to use the deep learning component as a complementary step to the rule-based component. More precisely, we first applied the rule-based component to map every entity found during NER, and the deep learning component was only applied to the remaining entities that were not normalized by the previous method.

C. Chemical indexing

Finally, after assigning the corresponding MeSH codes to all entities, the next step involves selecting from the previous identified MeSH codes which ones should be indexed. Similarly to the last subtask, we devised two approaches, the first being rule-based and the latter being based on TF-IDF scores.

1) Rule-based approach

The rule-based approach is a two-stage pipeline focused on extracting the MeSH codes present in specific parts of the documents, namely the title, abstract and all the captions from tables and figures. We considered that these elements of the documents would be those where it could be more common to find mentions for MeSH codes of interest. This rule was used to perform an initial extraction.

In the second stage, we evaluated the percentage of occurrence of each recognized MeSH code in the documents. This was used to reduce the previous list of codes, which reflected positively in the precision metric on the training datasets. Rules applied in this stage had different weights for each part of the document, *i.e.*, if the MeSH code was recognized in the title, this code needed a percentage of occurrence equal or superior to 10%. In the case of the caption, a MeSH code to be indexed required a percentage of occurrence of at least 20%, and in the abstract 7%.

In a post-contest phase, we refined these percentages and integrated the MeSH codes identified in the conclusion section of the documents, when available. The system used percentages of occurrence of 6%, 16%, 17%, and 6% for the title, captions, abstract, and conclusions, respectively. This change improved the F1-score of our approach by approximately 5 percentage points in all splits of the *training* dataset but could not surpass our official results in the *evaluation* dataset. Therefore, we conclude that these adjustments severely suffered from overfitting and require further investigation.

2) TF-IDF approach

The TF-IDF approach takes inspiration on the inner workings of traditional information retrieval (IR) models, and the main intuition is that ultimately an IR system would be used to retrieve the documents by exploring the indexed MeSH codes. Therefore, we hypothesize that the indexing task can be viewed as an optimization problem w.r.t. the ranking score given by a retrieval search engine. Given that not every MeSH code contributes equally to the final ranking score, we can select the top-k that contribute the most and use those as our indexed list since, from an IR system point of view, these are the MeSH codes that largely contribute to the final ranking score.

Unfortunately, due to time constraints we only explored this idea in a naive way, where we adopted it to model the importance of each MeSH code by using the TF-IDF weighting scheme with different SMART (13) variations. The TF-IDF scheme models term importance as function of its non-linear frequency times its rarity. After computing the importance of each MeSH code per document, the next task was to select the most important ones. For that, we envisioned several selection methods, the main ones being threshold based and probabilistic based. After some experiments we decided to use only a simple threshold-based method focused on precision.

TABLE II. DIFFERENCES OF SUBMITTED RUNS REGARDING THE NAMED ENTITY RECOGNITION PART.

	Training data	Selected epoch
Run 1	NLM-Chem *train*, *dev*, and *test* subsets	30
Run 2[c]	NLM-Chem *train* and *dev* subsets	Best epoch in NLM-Chem *test* subset
Run 3[a]	All datasets	15
Run 4[a,b,c]	All datasets (except the NLM-Chem *test* subset)	Best epoch in NLM-Chem *test* subset
Run 5[a,c]	All datasets (except the NLM-Chem *test* subset)	Best epoch in NLM-Chem *test* subset

[a]In Runs 3, 4, and 5 we also trained the last layer of BERT.

[b]In Run 4 we further fine-tuned the model (we did a second training pass) using only the NLM-Chem *train* and *dev* subsets.

[c]Runs 2, 4, and 5 were trained for 30 epochs.

D. Submitted runs

In this subsection, we detail the submitted runs showing their differences. Since this challenge was split into three smaller steps (NER, normalization, and indexing) we present the choices that were made for each step. Table II highlights the differences between each submitted run regarding the named entity recognition part. Regarding the normalization step, runs 1, 4, and 5 used the rule-based method alone, while runs 2 and 3 used the rule-based method followed by the deep learning method. In the indexing subtask, runs 1 and 5 used the rule-based approach, while 2, 3 and 4 used the TF-IDF based approach.

IV. RESULTS AND DISCUSSION

Table III presents the results of our official submissions and includes additional official metrics shared by the organizers. From the presented results, it is noticeable a superior performance from run 4 in NER and normalization, meaning that it was beneficial to train in several datasets if then fine-tuned on the NLM-Chem dataset. Another interesting observation is that in the normalization, the rule-based method seemed to achieve high precision values whilst being competitive in terms of recall when compared to the median, giving us a comparable higher F1 measure. Furthermore, dense retrieval managed to increase recall at the cost of precision, resulting in a similar F1 score. Therefore, the gains of the hybrid approach remain inconclusive, and more experiments are required. In terms of the last task, the rule-based approach managed to achieve competitive results, outscoring the benchmark by more than 4 percentage points. On the other hand, TF-IDF did not manage to beat the benchmark, showing an overall poor performance, which may disprove the main hypothesis behind the idea, or that the naive approach was too simple to model this problem.

V. CONCLUSIONS

In this work we performed chemical identification and normalization followed by chemical MeSH indexing. Our best results were 0.8454, 0.8136, and 0.4664 F1-scores in NER, normalization, and indexing respectively. We show PubMedBERT helps NER to perform competitively. Chemical MeSH indexing is the hardest task since there is error propagation from the first two steps (NER and normalization), and many MeSH terms do not exist after the normalization step (that is, are not mentioned directly in the text).

In the future we aim to (i) improve chemical recognition by exploring other model architectures and weighted losses, (ii) make further experiments with the SapBERT embeddings for normalization, and (iii) investigate the use of the MeSH tree structure to analyze if adding parent-related MeSH terms can help in the indexing task.

TABLE III. OFFICIAL OBTAINED RESULTS USING THE EVALUATION DATASET. ALL THE RESULTS PRESENTED USE THE STRICT EVALUATION METHOD. OUR TOP SCORE RESULTS ARE HIGHLIGHTED IN BOLD.

	Precision	Recall	F1-score
Chemical mention recognition			
Run 1	0.8354	0.8429	0.8392
Run 2	0.8421	0.8350	0.8386
Run 3	**0.8505**	0.7662	0.8062
Run 4	0.8394	**0.8515**	**0.8454**
Run 5	0.8372	0.7416	0.7865
Median	0.8476	0.8136	0.8373
Benchmark	0.8440	0.7877	0.8149
Chemical normalization to MeSH IDs			
Run 1	0.8582	0.7641	0.8084
Run 2	0.8221	**0.7898**	0.8056
Run 3	0.8124	0.7760	0.7938
Run 4	**0.8621**	0.7702	**0.8136**
Run 5	0.8310	0.7411	0.7835
Median	0.7120	0.7760	0.7749
Benchmark	0.8151	0.7644	0.7889
Chemical indexing			
Run 1	**0.5351**	**0.4133**	**0.4664**
Run 2	0.4882	0.3284	0.3927
Run 3	0.4910	0.3236	0.3901
Run 4	0.5173	0.3236	0.3981
Run 5	0.5308	0.3812	0.4437
Median	0.5173	0.3284	0.3981
Benchmark	0.3134	0.6101	0.4141

REFERENCES

1. Karp,P.D. (2016) Can we replace curation with information extraction software? *Database*, 2016(baw150).

2. Leaman,R., Islamaj,R., and Lu,Z. (2021) Overview of the NLM-Chem BioCreative VII track: full-text chemical identification and indexing in PubMed articles. *Proceedings of the seventh BioCreative challenge evaluation workshop*.

3. Islamaj,R., Leaman,R., Cissel,D., et al. (2021) The chemical corpus of the NLM-Chem BioCreative VII track: full-text chemical identification and

indexing in PubMed articles. *Proceedings of the seventh BioCreative challenge evaluation workshop*.

4. Islamaj,R., Leaman,R., et al. (2021) NLM-Chem, a new resource for chemical entity recognition in PubMed full text literature. *Scientific Data*, 8(91).

5. Krallinger,M., Rabal,O., Leitner,F., et al. (2015) The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(S2).

6. Li,J., Sun,Y., Johnson,R., et al. (2016) BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016(baw068).

7. Crichton,G., Pyysalo,S., Chiu,B., and Korhonen,A. (2017) A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, 18(368).

8. Gu,Y., Tinn,R., Cheng,H., et al. (2020) Domain-specific language model pretraining for biomedical natural language processing. *arXiv:2007.15779*.

9. Sohn,S., Comeau,D.C., Kim,W., and Wilbur,W.J. (2008) Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, 9(402).

10. Aronson,A.R., and Lang,F.M. (2010) An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3), pp. 229-236.

11. Luo,Z.H., Shi,M.W., Yang,Z., et al. (2020) pyMeSHSim: an integrative python package for biomedical named entity recognition, normalization, and comparison of MeSH terms. *BMC Bioinformatics*, 21(252).

12. Liu,F., Shareghi,E., Meng,Z., et al. (2021) Self-alignment pretraining for biomedical entity representations. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4228-4238.

13. Salton,G. (1972) A new comparison between conventional indexing (MEDLARS) and automatic text processing (SMART). *Journal of the American Society for Information Science*, 23(2), pp. 75-84.