NLNDE at CANTEMIST: Neural Sequence Labeling and Parsing Approaches for Clinical Concept Extraction

Lukas Lange[a,b], Xiang Dai[a,c], Heike Adel[a] and Jannik Strötgen[a]

[a]Bosch Center for Artificial Intelligence, Robert-Bosch-Campus 1, Renningen, 71272, Germany
[b]Saarland University, Saarland Informatics Campus, Saarbrücken, 66123, Germany
[c]University of Sydney, Sydney, 2006, Australia

Abstract

The recognition and normalization of clinical information, such as tumor morphology mentions, is an important, but complex process consisting of multiple subtasks. In this paper, we describe our system for the CANTEMIST shared task, which is able to extract, normalize and rank ICD codes from Spanish electronic health records using neural sequence labeling and parsing approaches with context-aware embeddings. Our best system achieves 85.3 F_1, 76.7 F_1, and 77.0 MAP for the three tasks, respectively.

Keywords

Named Entity Recognition, Context-Aware Embeddings, Recurrent Neural Networks, Biaffine Classifier

1. Introduction

Collecting and understanding key clinical information, such as disorders, symptoms, drugs, etc., from electronic health records (EHRs) has wide-ranging applications within clinical practice and transnational research [1, 2]. A better understanding of this information can facilitate novel clinical studies on the one hand, and help practitioners to optimize clinical workflows on the other hand. For example, to improve clinical decision support and personalized care of cancer patients, Jensen et al. [3] developed a methodology to estimate disease trajectories from EHRs, which can predict 80% of patient events in advance. However, free text is ubiquitous in EHRs. This leads to great difficulties in harvesting knowledge from EHRs. Therefore, natural language processing (NLP) systems, especially information extraction components, play a critical role in extracting and encoding information of interest from clinical narratives, as this information can then be fed into downstream applications.

The CANcer TExt Mining Shared Task (CANTEMIST) [4] focuses on identifying a critical type of concept related to cancer, namely tumor morphology. There are three independent subtasks as shown in Figure 1: (1) The extraction of tumor mentions, (2) the subsequent normalization to ICD codes and (3) the ranking by importance of the codes for each document.

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: lukas.lange@de.bosch.com (L. Lange); dai.xiang.au@gmail.com (X. Dai); heike.adel@de.bosch.com (H. Adel); jannik.stroetgen@de.bosch.com (J. Strötgen)
ORCID:

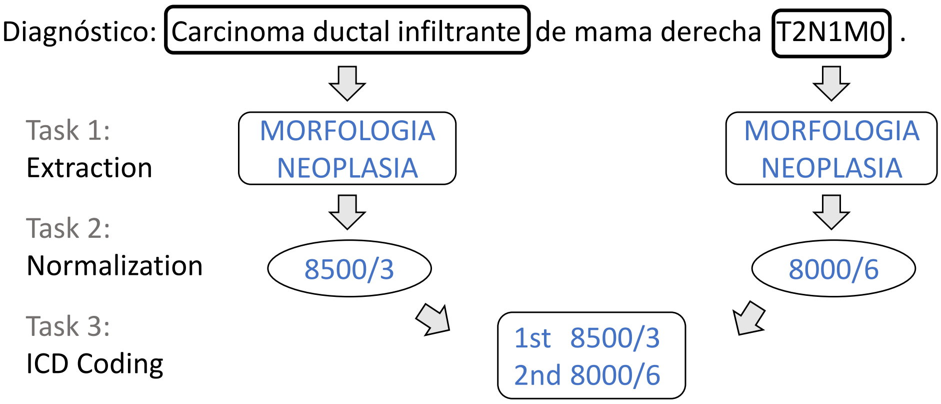


Figure 1: Sample sentence with normalized and ranked extractions.

In this paper, we describe our submission as Neither Language Nor Domain Experts (NLNDE) to the shared task. We treat the first subtask as a named entity recognition (NER) task and use neural sequence labeling and parsing approaches as frequently done to address NER in low resource settings. For the other two subtasks, we use rather simple non-deep learning methods, due to the very limited amount of training data: For the second subtask, the extracted entities are normalized using string matching and Levenshtein distance and the ranking of the third subtask is based on frequency.

2. Related Work

To identify medical concepts within the clinical narratives in EHRs, several machine learning-based named entity recognition (NER) and normalization systems were implemented [1, 5, 6]. Current state-of-the-art models for the extraction of clinical concepts are typically implemented as recurrent neural networks based on multiple different embeddings [7, 8]. DNorm, introduced in [5], applied a pairwise learning to rank approach to automatically learn a mapping from disease mentions to disease concepts from the training data. Evaluation results show that the machine learning method can effectively model term variations and achieves much better results than traditional techniques based on lexical normalization and matching, such as MetaMap [9]. Leaman et al. [1] introduced an extension of DNorm, called DNorm-C, which approaches both discontinuous NER and normalization using a pipeline approach. A joint model for NER and normalization was introduced in [6], aiming to overcome the cascading errors caused by the pipeline approach and enable the NER component to exploit the lexical information provided by the normalization component.

Other efforts on addressing both medical NER and normalization in other text types also exist. Metke-Jimenez and Karimi [10] compared different techniques for identifying medical concepts and drugs from medical forums. Zhao et al. [11] proposed a deep neural multi-task learning method to jointly model NER and normalization from biomedical publications, where stacked recurrent layers are shared among different tasks, enabling mutual support between

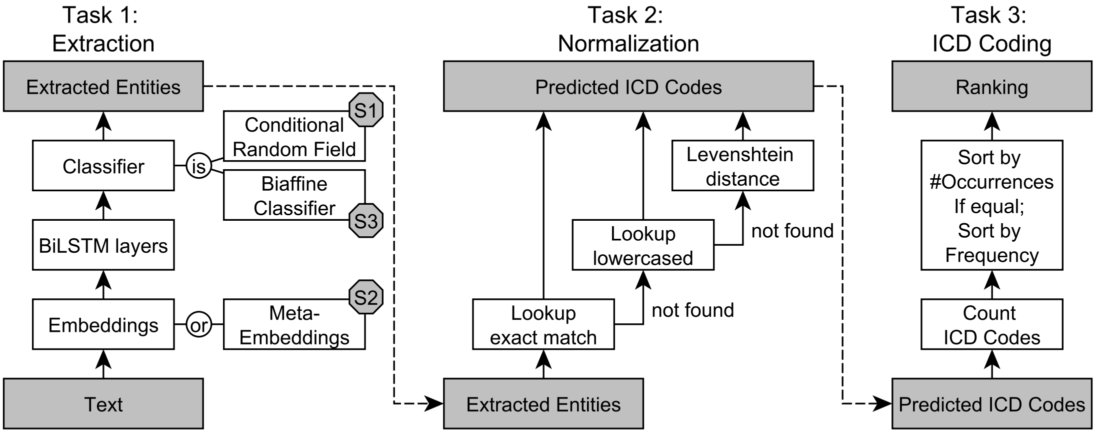


Figure 2: Overview of the NLNDE system architecture. *S1*, *S2* and *S3* are system variants in our different submissions.

tasks. Similarly, Lou et al. [12] proposed a transition-based model to jointly perform disease NER and normalization, combined with beam search and online structured learning. Experiments show that their joint model performs well on PubMed abstracts.

In contrast to concept normalization, which identifies a one-to-one mapping between text snippet and medical concept, ICD coding assigns most relevant ICD codes to a document as a whole [13, 14]. Most previous methods simplified this task as a text classification problem, and built classifiers using CNNs [15] or tree-of-sequences LSTMs [16]. Since ICD codes are organized under a hierarchical structure, Mullenbach et al. [17] and Cao et al. [18] proposed models to exploit code co-occurrence using label attention mechanism and graph convolutional networks, respectively.

3. Approach

This section provides an overview of the different methods tested for the three tasks, starting with the extraction, followed by the normalization and finally the ranking of the entities. Our architecture for the complete sequence of all three tasks is shown in Figure 2.

3.1. Task 1: Named Entity Recognition

We mainly experiment with two different methods for the extraction of tumor mentions. The first model treats the extraction as a sequence labeling problem without nested mentions, while the second model treats the problem as a parsing problem that allows the detection of nested mentions.

Sequence Labeling Model. For the sequence labeling model, the data is converted into the BIO format [19] using SpaCy[1] as the tokenizer. Overlapping annotations are resolved to

[1] https://spacy.io/api/tokenizer

a single annotation by selecting the longest sequence. We use a recurrent neural network, in particular, a bidirectional long short-term memory network (BiLSTM) with a conditional random field (CRF) output layer similar to [20]. For our choice of embeddings, we follow [21] who used a similar system for de-identification of Spanish clinical documents. In particular, we use pre-trained fastText embeddings [22] that were trained on articles from Wikipedia and the Common Crawl, as well as domain-specific fastText embeddings [23] that were pre-trained on articles of the Spanish online archive SciELO[2] for clinical documents. In addition, we include byte-pair-encoding embeddings [24] with 300 dimensions and a vocabulary size of 200,000 syllables. Finally, we add pre-trained FLAIR embeddings [25], which are calculated by contextualized character language models. All the n different embeddings are concatenated into a single embedding vector e

$$e_{CONCAT}(i) = [e_1(i); \cdots ; e_n(i)] \tag{1}$$

The embeddings are then fed into a stacked BiLSTM network that generates the feature presentation f given the embeddings e for each word in the sentence. f is then mapped to the size of the label space and fed into a conditional random field (CRF) classifier [26] that computes the most probable sequence of labels. We found that 3 stacked LSTM layers with a hidden size of 128 units each worked best in our experiments. The stacking of up to three layers increased the extraction performance by more than 1 F_1 point compared to a single LSTM layer.

Tokenization. We further analyze the effects of tokenization errors on the extraction. The BiLSTM-CRF using the SpaCy tokenizer achieves an F_1 of 82.4 on the development set (Precision (P): 84.9, Recall (R): 80.1). We then derive the following custom splitting rules according to annotation boundary problems from the training data.

- Suffix Rule: Cut off the suffix if the word is ending with a "." or "-"
- Prefix Rule: Cut off the prefix if the word starts with a "-"
- Infix Rule: split each word at hyphens, punctuation and quotation marks into three parts.

The rules increase performance for all three metrics by 0.4–0.5 points (P: 85.4, R: 80.5, F_1: 82.9).

Meta-Embeddings. Related work has shown significant improvements when the simple concatenation of embeddings is replaced with a different meta-embedding method. We experiment with an attention mechanism as described by Kiela et al. [27] to create meta-embeddings of several different embedding types. Such meta-embeddings were shown to be useful in multiple extraction tasks [27, 28, 29, 30]. As all embeddings have a different size of up to 2048 dimensions, all embeddings are mapped to the same space with dimension E first. We set E to the size of the largest embeddings. For this, we use a non-linear mapping Q_i with bias b_i for embedding e_i:

$$x_i = \tanh(Q_i \cdot e_i + b_i) \tag{2}$$

[2] https://scielo.org/

We take the attention method proposed by Lange et al. [30] who used feature-based attention. With this, the attention function has access to additional word information, in our case the word's shape, frequency and length. This helps to infer linguistic information about the word that can be useful for the attention weight computation but is not encoded in the word vectors. The features are added as a vector f_w to the attention function:

$$\alpha_i = \frac{\exp(V \cdot \tanh(Wx_i + f_w))}{\sum_{l=1}^{n} \exp(V \cdot \tanh(Wx_l + f_w))} \tag{3}$$

$$e_{META} = \sum_i \alpha_i \cdot x_i \tag{4}$$

with x_i being the mapped embeddings e_i and V and W being parameters of the model that are learnt during training. The final meta-embedding e_{META} is then used as input to the stacked BiLSTM network. The meta-embedding model has a hidden size of 25 dimensions for the attention computation.

Biaffine Classifier. Recently, a trend emerged of modeling different natural language processing tasks as parsing tasks and thus, solve them by using a dependency parser. Examples are named entity recognition [31] and negation resolution [32].

We experiment with such a system and model the extraction task as a parsing task. For this, we replace the CRF classifier with a biaffine classifier [33]. Following Yu et al. [31], we apply two separate feed-forward networks (FFNN) to the features f generated from the stacked BiLSTM to create start and end representations of all possible spans (h_s/h_e). Then, we use biaffine attention [33] over the sentence to compute the scores r_m for each span i in the sentence that could refer to a named entitiy.

$$h_s(i) = FFNNs(f_{s_i}) \tag{5}$$

$$h_e(i) = FFNNe(f_{e_i}) \tag{6}$$

$$r_m(i) = h_s^{\top}(i) U_m h_e(i) + W_m(h_s(i) \oplus h_e(i)) + b_m \tag{7}$$

Similar to Yu et al. [31], we use multilingual BERT, character and fastText embeddings. We experimented with the same set of embeddings that we used for the BiLSTM-CRF model as well, but the performance decreased for the biaffine model. Again, the embeddings are fed into the BiLSTM to obtain the word representations f. We found that 5 stacked LSTM layers with a size of 200 hidden units each worked best for the biaffine model. Using this combination of hyperparameters improved the model by roughly 1 F_1 point compared to the originally proposed model consisting of 3 layers of size 200.

For the BiLSTM-CRF and biaffine models we mostly follow the hyperparameter configurations and training routines of Akbik et al. [25] and Yu et al. [31], respectively, with exceptions regarding the number and sizes of the recurrent layers mentioned above.

3.2. Task 2: Normalization

The second task requires the normalization of the previously extracted entities to ICD-O-3 codes (Spanish version: eCIE-O-3.1). As a large number of possible ICD codes appears only

Table 1
Results of different normalization methods on the development set.

Method	Correct	False	Unmatched	F_1 on gold extractions
String matching	2341	21	979	70.07
+ lowercased	2374	22	945	71.06
++ Levenshtein distance	2910	431	0	87.10

once or never in the training data, we decided against deep-learning methods, as simply not enough training instances are available for this large label set. Instead, we use an approach based on string matching and Levenshtein distance [34].

For this, we collect all entities from the training set and their ICD code. As there is only little ambiguity among these entities, we use a context-independent method for the normalization. Using the entities from the training set, we are able to correctly assign 70% of the ICD codes to entities from the development set using exact string matching with a very low false-positive rate (< 1%). Using lower-cased matching, the number of correctly assigned codes slightly increases. Given that these methods assign codes almost perfectly to known entities, we first apply exact string matching and then lower-cased matching. For the remaining unmatched entities, we compute the Levenshtein distance between the given string and strings from the training data to find the closest neighbor among the known training instances and assign the corresponding code. This method achieves 87% F_1 on the gold extractions of the unseen development set.

3.3. Task 3: ICD Coding

The purpose of the last subtask is the creation of a ranked list of ICD codes for a given document. For this ICD coding, we create a ranking with a sorting function based on code frequency. We sort by the number of times each code occurs in the given document under the assumption that codes that appear more often inside a document are more important. Whenever two codes appeared an equal amount of times, they are ranked by their general frequency as found on the training set. This method achieves a MAP of 73.82 using the gold extractions of the unseen development set.

3.4. Submissions

The following five runs are the NLNDE submissions to the CANTEMIST shared task. The difference between the runs lies in the model architecture used in the extraction track. The normalization and ICD coding methods are equal across the submissions and solely based on the predicted extraction of the first subtask:

S1 : A BiLSTM-CRF model with a concatenation of FLAIR, fastText, BPEmb and domain-specific fastText embeddings.

S2 : A BiLSTM-CRF model with feature-based meta-embeddings as a replacement for the concatenation of embeddings used in *S1*.

S3 : A biaffine model with multilingual BERT and fastText embeddings for nested named entity recognition.

S4 : A similar biaffine model trained on the development set in addition to the training set.

S5 : An ensemble of *S1/S2/S3* based on majority voting. Predictions are accepted into the ensemble classifier whenever at least two models predicted identical entity offsets.

4. Results

The official results for the three tracks of the CANTEMIST shared task are shown in Tables 2, 3 and 4, respectively. The official evaluation metric of the test set is highlighted in gray and the best model is highlighted in bold.

4.1. Results for Task 1 and 2: Named Entity Recognition and Normalization

The **BiLSTM-CRF** (*S1*) is a competitive baseline model for our experiments with 82.7 F_1 for the extraction and 72.9 F_1 for the normalization. Even though the **meta-embeddings** (*S2*) improve performance on the development set, at least for the normalization, we observe contrary results on the test data, as the concatenation of embeddings works better for this.

The **biaffine model** (*S3*) achieves a much higher precision than the BiLSTM-CRF with +2 F_1 points for the extraction and +1 F_1 point for the normalization on the development set. This gap further increases on the unseen test data. The difference in recall is not that large, even though the biaffine model is able to extract nested entities. However, the number of nested mentions is rather low and the ability to extract them does not seem to make a big difference in practice for this shared task. Overall, the biaffine model dominates because of the better precision, which might be explained by the fact that many of the tumor mentions cover multiple tokens and the parsing model is better in capturing those long-distant dependencies. A more detailed analysis on this is provided in Section 4.3. In addition, the biaffine model can be further improved by training on a combination of training and development set, resulting in our best submission (*S4*).

The **ensemble model** (*S5*) effectively increases the precision compared to the single models, in particular for the normalization, but it does not have the same recall, as only entities predicted by at least two of the three models get accepted into the output. Thus, only high-confidence entities are output by the ensemble classifier. As a result, this model may be the better choice if precision is preferred over recall.

4.2. Results for Task 3: ICD Coding

The results for the third subtask, the ranked coding, are close to the results on the gold extractions. This indicates that the systems are able to extract the most important entities correctly. Overall, the differences between the systems are rather small as shown in Table 4. For example, the MAP score for the biaffine model (*S3*) is only 0.2 points higher than the BiLSTM-CRF (*S1*). Only the biaffine model trained on the combination of training and development data (*S4*) achieves a slightly higher performance of up to a MAP score of 77.0.

Table 2
Results of task 1: Extraction of tumor morphology mentions.

		Dev			Test		
		P	R	F_1	P	R	F_1
S1	BiLSTM-CRF	84.7	82.4	83.6	82.4	83.0	82.7
S2	MetaEmbeddings	84.7	82.4	83.6	81.5	82.3	81.9
S3	Biaffine	86.8	82.1	84.4	85.0	83.5	84.2
S4	Biaffine-Dev	-	-	-	***85.4***	***85.2***	***85.3***
S5	Ensemble	87.8	81.3	84.4	84.7	80.8	82.7

Table 3
Results of task 2: Normalization of extractions to corresponding ICD-O-3 codes.

		Dev			Test			Test w/o code 8000/6		
		P	R	F_1	P	R	F_1	P	R	F_1
S1	BiLSTM-CRF	76.4	74.3	75.3	74.3	74.9	74.6	75.0	70.9	72.9
S2	MetaEmbeddings	76.6	74.5	75.6	73.5	74.1	73.8	74.6	70.9	72.7
S3	Biaffine	79.0	74.7	76.8	***76.7***	75.3	76.0	76.4	71.4	73.8
S4	Biaffine-Dev	-	-	-	***76.7***	***76.6***	***76.7***	77.3	***72.6***	***74.9***
S5	Ensemble	80.0	74.0	76.9	***76.7***	73.2	74.9	***77.4***	70.2	73.6

Table 4
Results of task 3: Creating a ranked coding of the given document.

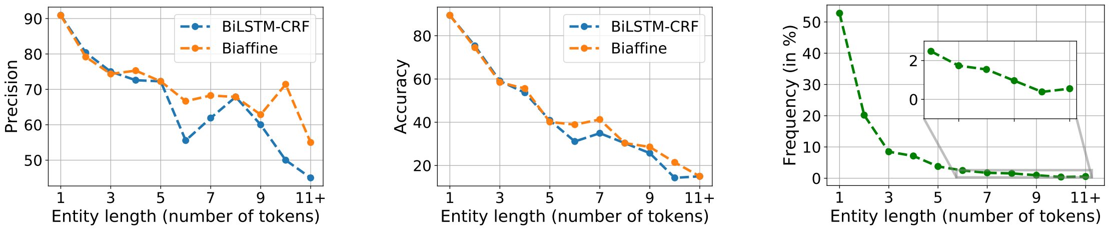
		Dev	Test				Test w/o code 8000/6			
		MAP	P	R	F_1	MAP	P	R	F_1	MAP
S1	BiLSTM-CRF	74.2	75.5	76.2	75.9	73.7	72.7	72.1	72.4	69.7
S2	MetaEmbeddings	74.4	74.8	75.8	75.3	73.5	71.9	71.6	71.8	69.4
S3	Biaffine	75.0	75.9	76.3	76.1	73.9	73.0	72.2	72.6	70.2
S4	Biaffine-Dev	-	77.0	***77.1***	***77.0***	***74.9***	74.3	***72.8***	***73.6***	***71.4***
S5	Ensemble	74.2	***77.2***	74.9	76.0	73.1	***74.6***	70.7	72.6	69.3

Following the official evaluation, we include the results without the most frequent code "8000/6" (Metastatic Cancer) for the normalization and coding tasks in Tables 3 and 4. With this, we observe a performance drop for all submissions between 2 and 3 F_1 or MAP points.

To conclude, our results show that the individual task-specific components deliver good results on the development as well as on the test set. Furthermore, the sequential execution as a pipeline model of extraction, normalization and ranking works well in practice.

4.3. Analysis: BiLSTM-CRF vs. Biaffine Classifier

In the following, the performance differences between the BiLSTM-CRF and biaffine models are analyzed with a focus on the lengths of the entities. As shown in Table 2, the main difference lies in the higher precision of the biaffine model. Figure 3a shows the precision for entities with



(a) Task 1: Extraction

(b) Task 2: Normalization

(c) Relative frequency

Figure 3: Results for entities of different lengths. (a) displays the impact of entity length on the extraction and (b) for the normalization. In (c) the relative frequency of these entities is shown. The last data point (11+) in all plots is the aggregation of all entities longer than 10 tokens.

respect to their length. In particular, for shorter entities, there are no differences in performance between the two model architectures. Starting with entities consisting of 6 and more tokens, the biaffine model begins to outperform the BiLSTM-CRF model for the extraction and also the subsequent normalization (Fig. 3b). The performance difference reaches up to 20 points in precision for the extraction of multi-token entities consisting of 10 tokens and 10 points for entities longer than at least 11 tokens.

For both model types, we observe that the performance drop correlates with the length of the entities. In general, there are fewer training instances for longer entities, as shorter entities are more frequent than longer ones with a tail of infrequent but long entities (Fig. 3c). This performance gap between short and long entities is even larger for the normalization which ranges from 85 F_1 for single-token entities to 15 F_1 for entities with more than 10 tokens. However, as more than half of the entities consist of a single token, the impact of longer entities on the overall F_1 score is limited and, thus, the difference of the BiLSTM-CRF and biaffine models regarding the overall precision is 2 points, even though the biaffine model is better suited for the extraction of longer multi-token entities.

5. Conclusion

In this paper, we described our system for the CANTEMIST shared task to extract, normalize and rank ICD codes from Spanish clinical documents. As neither language nor domain experts, we tested neural sequence labeling, as well as parsing approaches for the extraction, string matching and Levenshtein distance for the normalization and frequency for the ranking. We found that the best model is based on a biaffine classifier that achieves 85.3 F_1, 76.7 F_1 and 77.0 MAP for the three tracks, respectively. Future work includes the optimization of the extraction models for long multi-token entities.

References

[1] R. Leaman, R. Khare, Z. Lu, Challenges in clinical natural language processing for automated disorder normalization, Journal of biomedical informatics 57 (2015) 28–37.

[2] Y. Wang, L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, Clinical information extraction applications: a literature review, Journal of biomedical informatics 77 (2018) 34–49.

[3] K. Jensen, C. Soguero-Ruiz, K. O. Mikalsen, R.-O. Lindsetmo, I. Kouskoumvekaki, M. Girolami, S. O. Skrovseth, K. M. Augestad, Analysis of free text in electronic health records for identification of cancer patient trajectories, Scientific reports 7 (2017) 46226.

[4] A. Miranda-Escalada, E. Farré, M. Krallinger, Named entity recognition, concept normalization and clinical coding: Overview of the cantemist track for cancer text mining in spanish, corpus, guidelines, methods and results, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), CEUR Workshop Proceedings, 2020.

[5] R. Leaman, R. Islamaj Doğan, Z. Lu, Dnorm: disease name normalization with pairwise learning to rank, Bioinformatics 29 (2013) 2909–2917.

[6] R. Leaman, Z. Lu, Taggerone: joint named entity recognition and normalization with semi-markov models, Bioinformatics 32 (2016) 2839–2846.

[7] A. Gonzalez-Agirre, M. Marimon, A. Intxaurrondo, O. Rabal, M. Villegas, M. Krallinger, PharmaCoNER: Pharmacological substances, compounds and proteins named entity recognition track, in: Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1–10. URL: https://www.aclweb.org/anthology/D19-5701. doi:10.18653/v1/D19-5701.

[8] L. Lange, H. Adel, J. Strötgen, Closing the gap: Joint de-identification and concept extraction in the clinical domain, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6945–6952. URL: https://www.aclweb.org/anthology/2020.acl-main.621. doi:10.18653/v1/2020.acl-main.621.

[9] A. R. Aronson, Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program, in: Proceedings of the AMIA Symposium, Washington, DC, 2001, p. 17.

[10] A. Metke-Jimenez, S. Karimi, Concept identification and normalisation for adverse drug event discovery in medical forums, in: Proceedings of the First International Workshop on Biomedical Data Integration and Discovery, Kobe, Japan, 2016.

[11] S. Zhao, T. Liu, S. Zhao, F. Wang, A neural multi-task learning framework to jointly model medical named entity recognition and normalization, in: Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HawaiI, 2019, pp. 817–824.

[12] Y. Lou, Y. Zhang, T. Qian, F. Li, S. Xiong, D. Ji, A transition-based joint model for disease named entity recognition and normalization, Bioinformatics 33 (2017) 2363–2371.

[13] J. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, W. Duch, A shared task involving multi-label classification of clinical free text, in: Biological, translational, and clinical language processing, Prague, Czech Republic, 2007, pp. 97–104.

[14] A. Névéol, A. Robert, F. Grippo, C. Morgand, C. Orsi, L. Pelikan, L. Ramadier, G. Rey, P. Zweigenbaum, Clef ehealth 2018 multilingual information extraction task overview: Icd10 coding of death certificates in french, hungarian and italian, in: CLEF (Working Notes), 2018.

[15] S. Karimi, X. Dai, H. Hassanzadeh, A. Nguyen, Automatic diagnosis coding of radiology reports: a comparison of deep learning and conventional classification methods, in: BioNLP 2017, Vancouver, Canada, 2017, pp. 328–332.

[16] P. Xie, E. Xing, A neural architecture for automated icd coding, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 2018, pp. 1066–1076.

[17] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, J. Eisenstein, Explainable prediction of medical codes from clinical text, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, 2018, pp. 1101–1111.

[18] P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu, W. Chong, Hypercore: Hyperbolic and co-graph representation for automatic icd coding, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 3105–3114.

[19] L. A. Ramshaw, M. P. Marcus, Text chunking using transformation-based learning, in: Natural language processing using very large corpora, Springer, 1999, pp. 157–176.

[20] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270. URL: https://www.aclweb.org/anthology/N16-1030. doi:10.18653/v1/N16-1030.

[21] L. Lange, H. Adel, J. Strötgen, NLNDE: The neither-language-nor-domain-experts' way of spanish medical document de-identification, in: Proceedings of The Iberian Languages Evaluation Forum (IberLEF 2019), CEUR Workshop Proceedings, 2019. URL: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_5.pdf.

[22] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146. URL: https://www.aclweb.org/anthology/Q17-1010. doi:10.1162/tacl_a_00051.

[23] F. Soares, M. Villegas, A. Gonzalez-Agirre, M. Krallinger, J. Armengol-Estapé, Medical word embeddings for Spanish: Development and evaluation, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 124–133. URL: https://www.aclweb.org/anthology/W19-1916. doi:10.18653/v1/W19-1916.

[24] B. Heinzerling, M. Strube, BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://www.aclweb.org/anthology/L18-1473.

[25] A. Akbik, D. Blythe, R. Vollgraf, Contextual string embeddings for sequence labeling, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 1638–1649. URL: https://www.aclweb.org/anthology/C18-1139.

[26] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 282–289. URL: http://dl.acm.org/citation.cfm?id=645530.655813.

[27] D. Kiela, C. Wang, K. Cho, Dynamic meta-embeddings for improved sentence representations, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language

Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1466–1477. URL: https://www.aclweb.org/anthology/D18-1176. doi:10.18653/v1/D18-1176.

[28] L. Lange, H. Adel, J. Strötgen, On the choice of auxiliary languages for improved sequence tagging, in: Proceedings of the 5th Workshop on Representation Learning for NLP, Association for Computational Linguistics, Online, 2020, pp. 95–102. URL: https://www.aclweb.org/anthology/2020.repl4nlp-1.13. doi:10.18653/v1/2020.repl4nlp-1.13.

[29] G. I. Winata, Z. Lin, J. Shin, Z. Liu, P. Fung, Hierarchical meta-embeddings for code-switching named entity recognition, 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. 3541–3547. URL: https://www.aclweb.org/anthology/D19-1360. doi:10.18653/v1/D19-1360.

[30] L. Lange, H. Adel, J. Strötgen, NLNDE: Enhancing neural sequence taggers with attention and noisy channel for robust pharmacological entity detection, in: Proceedings of The 5th Workshop on BioNLP Open Shared Tasks, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 26–32. URL: https://www.aclweb.org/anthology/D19-5705. doi:10.18653/v1/D19-5705.

[31] J. Yu, B. Bohnet, M. Poesio, Named entity recognition as dependency parsing, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6470–6476. URL: https://www.aclweb.org/anthology/2020.acl-main.577. doi:10.18653/v1/2020.acl-main.577.

[32] R. Kurtz, S. Oepen, M. Kuhlmann, End-to-end negation resolution as graph parsing, in: Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies, Association for Computational Linguistics, Online, 2020, pp. 14–24. URL: https://www.aclweb.org/anthology/2020.iwpt-1.3. doi:10.18653/v1/2020.iwpt-1.3.

[33] T. Dozat, C. D. Manning, Deep biaffine attention for neural dependency parsing, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017. URL: https://openreview.net/forum?id=Hk95PK9le.

[34] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, in: Soviet physics doklady, volume 10, 1966, pp. 707–710.