

Secondary Use of Clinical Problem List Entries for Neural Network-Based Disease Code Assignment

Markus KREUZTHALER^{a,1}, Bastian PFEIFER^a, Diether KRAMER^b and Stefan SCHULZ^a

^a*Institute for Medical Informatics, Statistics and Documentation,
Medical University of Graz, Austria*

^b*Department of Information and Process Management,
Steiermärkische Krankenanstaltengesellschaft m.b.H. (KAGes), Graz, Austria*

Abstract. Clinical information systems have become large repositories for semi-structured and partly annotated electronic health record data, which have reached a critical mass that makes them interesting for supervised data-driven neural network approaches. We explored automated coding of 50 character long clinical problem list entries using the International Classification of Diseases (ICD-10) and evaluated three different types of network architectures on the top 100 ICD-10 three-digit codes. A fastText baseline reached a macro-averaged F_1 -score of 0.83, followed by a character-level LSTM with a macro-averaged F_1 -score of 0.84. The top performing approach used a downstreamed RoBERTa model with a custom language model, yielding a macro-averaged F_1 -score of 0.88. A neural network activation analysis together with an investigation of the false positives and false negatives unveiled inconsistent manual coding as a main limiting factor.

Keywords. Natural Language Processing, Electronic Health Records, Machine Learning, Secondary Use

1. Introduction and Motivation

The clinical information system (CIS) of a large, multicentre public hospital provider in Austria stores short clinical problem descriptions in German (maximum 50 characters) together with which manually assigned codes from the International Classification of Diseases (ICD-10). This huge table fulfils three purposes: (i) collection of content that automatically fills the “Diagnoses” section when narrative discharge summaries are created, (ii) display of a problem-list like scrollable textbox in the CIS frontend, and (iii) provision of ICD codes for administrative purposes.

Due to the technical limitation of 50 characters, the often-lengthy ICD-10 descriptions are usually overwritten by the users, who use overly compact expressions, characterized by ellipsis, context-dependent abbreviations and acronyms, non-standardised numeric values, spelling variants and errors. These text snippets exemplify typical idiosyncrasies of clinical language [1].

¹ Corresponding Author: Markus Kreuzthaler, E-mail: markus.kreuzthaler@medunigraz.at

Our work is centred on this resource. We wanted to investigate to what extent clinical real-world data is suited as training material for different types of multi-class classification approaches for automatic assignment of codes from the classification system ICD-10, and where it reaches its limits. We applied three different types of neural network (NN) architectures: a shallow NN, a recurrent NN and a transformer-based architecture for our experimental secondary use-case scenario of clinical routine documentation.

2. Methods and Data

We used ~1.9 million unique de-identified problem list entries and ignored all entries without ICD codes. A 90/10 split was carried out for training and test set preparation. We trained one model using the top 100 occurring three-digit ICD-10 codes, which cover about 50% of all code instances in the data set. Therefore, we scaled up all training samples of this highly imbalanced data set to the most frequent code, resulting in ~6 million training samples (100 times ~60k observations). The test set based distribution of the 100 codes under investigation remained unchanged (~93k observations).

2.1. Neural Network Architectures

Shallow. As a baseline, we used *fastText* [2], exploiting pre-trained skip-gram embeddings at subword and word type level from the training set. We used a simple rule-based tokenizer² and normalized all resulting tokens to lower case.

Recurrent. Character inputs are modelled as 122 dimensional one-hot encoded vectors from the training set, together with an out-of-character-dictionary feature dimension in accordance to [3] for the input to the chosen LSTM [4] network. *Deeplearning4j* was used as implementation library.

Transformer. For transformer-based architectures, we decided to apply RoBERTa [5]. We build our own language model, in order to support the downstream task with a first understanding of the language under scrutiny. To this end, we used *ktrain*, a lightweight wrapper library, for *Hugging Face*.

2.2. Neural Network Interpretation

In health care scenarios, regulatory frameworks require that decision support systems are able to explain the path that led to a certain suggestion. This is optimally met by decision trees, but poses complex challenges for NNs. Established approaches are LIME (Local Interpretable Model-Agnostic Explanations), SHAP (SHapley Additive exPlanations) or the notion of *saliency* as being the norm of the gradient of the loss function to a given input, an approach successfully applied in clinical NLP domains [6] for explainable machine learning systems. In this work, we were particularly interested in character-wise feedback for the overall classification result via the inspection of certain class probabilities at certain positions of the LSTM sequence model. This allows the identification of activation levels at this granular input representation scheme, because single characters have significant impact on the correct interpretation of narrative content.

² $[\text{p}\{\text{IsAlphabetic}\}\text{p}\{\text{IsDigit}\}]$

2.3. Evaluation Metrics

We evaluated the performance of the trained model on the test data set using precision, recall, and F₁-score per ICD-10 code, as well as macro-evaluation statistics. The following definitions were used: true positives (*TPs*) – the number of correctly assigned codes; true negatives (*TNs*) – the number of correctly unassigned codes; false positives (*FPs*) – the number of incorrectly assigned codes; false negatives (*FNs*) – the number of incorrectly unassigned codes. Exploiting this definition, precision $P = TP / (TP + FP)$, recall $R = TP / (TP + FN)$ and $F_1\text{-score} = 2 \cdot P \cdot R / (P + R)$.

3. Results and Discussion

Table 1. Selection of test set based evaluation results using RoBERTa.

ICD-10	Precision	Recall	F ₁ -score	ICD-10	Precision	Recall	F ₁ -score
P07	0,99	1,00	0,99	D48	0,79	0,71	0,74
F17	0,99	0,97	0,98	D37	0,74	0,73	0,73
A46	0,96	0,97	0,97	I47	0,67	0,77	0,72
E78	0,96	0,97	0,97	T14	0,83	0,64	0,72
G47	0,97	0,98	0,97	I64	0,56	0,72	0,63
I35	0,97	0,97	0,97	E10	0,53	0,73	0,61
I71	0,95	0,98	0,97	N19	0,43	0,73	0,54
D64	0,95	0,97	0,96	E14	0,45	0,51	0,47
G40	0,96	0,96	0,96	C80	0,40	0,55	0,46
J44	0,96	0,96	0,96	Z03	0,38	0,44	0,41

Evaluation of the test set with 92,832 problem list entries yielded an overall macro-averaged F₁-score of 0.83 for the fastText baseline, 0.84 for the LSTM, and 0.88 for the RoBERTa approach. Table 1 shows the evaluation on an ICD-10 three-digit level. The left side of the table shows the top 10 best performing codes, the right side shows codes with a macro-averaged F₁-score less than 0.75. We got the highest F₁-score value for the ICD-10 code P07 (“disorders related to short gestation and low birth weight, not elsewhere classified”) and the lowest one for Z03 (“medical observation and evaluation for suspected diseases and conditions”) for all three different NN architectures. Interestingly, LSTM, which solely relies on character-level inputs, showed about the same performance as the fastText baseline. This provides strong evidence that the LSTM network is highly sensitive to even minor variations of character sequences that are relevant for the correct ICD code assignment. We decided to investigate this further by performing a network activation analysis of the LSTM.

3.1. Explainable AI

What can be seen at the top of Figure 1 is the network stimulation for the correct class I25 (chronic ischaemic heart disease) for the German input “Kononare Herzkrankh. 1 Gefäß – 1 x DES in LAD” (coronary heart disease. 1 vessel – 1 x DES in LAD). The intensity level corresponds to the class level probability for a given code at a given position and varies between zero and one. A low-level intensity therefore corresponds to

a low probability for the class shown on the right in the figure. This snippet exhibits the typing mistake “Kononare” instead of “Koronare” (coronary), an ad-hoc abbreviation “Herzkrankh.” for “Herzkrankheit” (heart disease) and two acronyms. “DES” stands for “drug-eluting stent” and “LAD” for “left anterior descending (artery)”. It is principally the last part of this input string that strongly supports the code I25. In contrast, the same string has a very low feedback activation for the ICD-10 class E11 (type 2 diabetes mellitus), as expected.

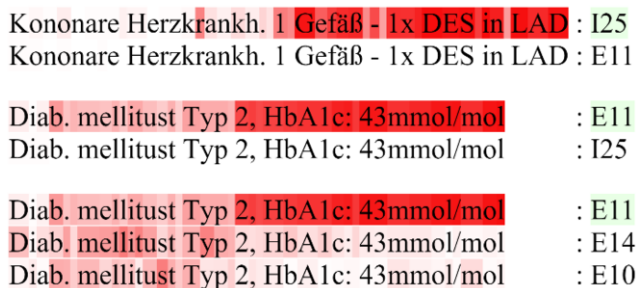


Figure 1. Activation heat map with respect to a given class at a certain character position.

The centre of Figure 1 displays the result of the experiment in reverse. We see the network activation for the input “Diab. mellitust Typ 2, HbA1c: 43 mmol/mol” (diabetes mellitus type 2, HbA1c: 43 mmol/mol) and its corresponding correct code E11. Again, there are ad-hoc abbreviations “Diab.” for “Diabetes” and the typing mistake “mellitust” (correct “mellitus”). With the occurrence of the digit “2” in the input string, the network responds with a very high activation at this position for type 2 diabetes mellitus. “HbA1c” (glycated haemoglobin) is also an important diabetes biomarker. Conversely, there is very little feedback activity for the class I25, resulting in low probability values for the whole input sequence.

At the bottom of Figure 1 the input activation for the same string is contrasted with E10 (type 1 diabetes mellitus) and E14 (unspecified diabetes mellitus). As seen before with the appearance of the character “2”, there is strong evidence for E11. Nevertheless, for E10 and E14, there are clearly observable activation levels. This inspection supports our assumption that, for certain ICD-10 code sections, the network will intermix classes if there is no clear consistent exclusive manual coding. Nevertheless, the network tries to interpret the input as well as possible with respect to the training data.

4. Conclusion and Outlook

We presented an approach for the assignment of short clinical problem list entries to ICD-10 three-digit codes using three different NN architectures: a shallow NN (fastText), a recurrent NN (LSTM) and a transformer-based architecture (RoBERTa) which performed best with an overall macro-averaged F_1 -score of 0.88. The fastText baseline evaluated with a macro-averaged F_1 -score of 0.83, the LSTM approach modelling the problem purely on character-level reached a macro-averaged F_1 -score of 0.84. In a character-level based heat map, we visualized LSTM network activations to a given input. Thus, we could trace classification decisions back. TP, FP and FN analyses revealed that the trained network suffered from coding inconsistencies (e.g., E14 versus E11).

The preliminary experiments presented in this paper emphasize the potential of secondary use scenarios of annotated semi-structured clinical real-world data for building NN-based applications such as the assignment of disease codes. Taking advantage of such resources is needed for data-driven applications such as the generation of clinical BERT models [7] for languages other than English, like German, where to the best of the authors knowledge a model is not available by now. There is yet an unexploited potential of NNs to bridge the gap between standardized code entries and the language used in the routine of clinical documentation and communication. Identifying and leveraging annotated language resources as presented in this paper could be a step in this direction.

Acknowledgments

This study was approved by the ethics committee of the Medical University of Graz (30-496 ex 17/18).

References

- [1] Névél A, Dalianis H, Velupillai S, Savova G, Zweigenbaum P. Clinical Natural Language Processing in languages other than English: opportunities and challenges. *J Biomed Semant* 2018;9:12. <https://doi.org/10.1186/s13326-018-0179-8>.
- [2] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching Word Vectors with Subword Information. *Trans Assoc Comput Linguist* 2017;5:135–46. https://doi.org/10.1162/tacl_a_00051.
- [3] Zhang X, LeCun Y. Text Understanding from Scratch 2016. <https://doi.org/10.48550/arXiv.1502.01710>.
- [4] Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput* 1997;9:1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [5] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach 2019. <https://doi.org/10.48550/arXiv.1907.11692>.
- [6] Gehrmann S, Dernoncourt F, Li Y, Carlson ET, Wu JT, Welt J, et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLOS ONE* 2018;13:e0192360. <https://doi.org/10.1371/journal.pone.0192360>.
- [7] Alsentzer E, Murphy J, Boag W, Weng W-H, Jindi D, Naumann T, et al. Publicly Available Clinical BERT Embeddings. *Proc. 2nd Clin. Nat. Lang. Process. Workshop, Minneapolis, Minnesota, USA: Association for Computational Linguistics*; 2019, p. 72–8. <https://doi.org/10.18653/v1/W19-1909>.