Overview of the CLEF eHealth 2019 Multilingual Information Extraction

Mariana Neves[1 [0000−0002−6488−2394]], Daniel Butzke[1 [0000−0002−4800−4655]], Antje Dörendahl[1], Nora Leich[1], Benedikt Hummel[1 [0000−0003−2016−7441]], Gilbert Schönfelder[1,2], and Barbara Grune[1]

[1] German Centre for the Protection of Laboratory Animals (Bf3R), German Federal Institute for Risk Assessment (BfR), Diedersdorfer Weg 1, 12277, Berlin, Germany
mariana.lara-neves@bfr.bund.de
[2] Charité - Universitätsmedizin Berlin, Institute of Clinical Pharmacology and Toxicology, Charitéplatz 1, 10117 Berlin, Germany

Abstract. Non-technical summaries (NTSs) of animal experimentation can be valuable resources to foster more transparency of research made with animals and to better inform the community about this topic. The NTSs of planned animal experiments in Germany are publicly available and have been manually assigned to ICD-10 codes. We used this data in the scope of organizing the Multilingual Information Extraction Task (Task 1) in the CLEF eHealth challenge. For the development phase, we released a training dataset containing more than 8,000 NTSs and their corresponding codes (if any assigned). For the test phase, we released 407 unseen NTSs for which the participants should submit the predictions made by their systems. The best performing system obtained a P, R, and FM of 0.83, 0.77, and 0.80, respectively.

Keywords: Document indexing, ICD-10 codes, summaries of animal experiments.

1 Introduction

Non-technical summaries (NTSs) are short descriptions of the planned animal experiments to be carried out in a country and are stipulated when requesting permission for the experiment. The European Union (EU) requires the member states to collect these summaries and to make them available to the community in order to foster more transparency in animal research [12]. The German Federal Institute for Risk Assessment (BfR, in its acronym in German) publishes the German NTSs online in the AnimalTestInfo database[3].

[3] https://www.animaltestinfo.de/

These NTSs are regularly manually annotated with ICD-10 codes for the identification of the diseases that are the focus of the planned experiments. Indexing the NTSs using terms from standard teminologies provides additional information on the research goals of the animal experiments and supports a detailed analysis of the data. [3].

We utilized our annotated NTSs in the scope of a shared task in the CLEF eHealh challenge. Our shared task aimed to evaluate systems for the automatic detection of the ICD-10 codes in German NTSs[4]. For this purpose, we utilized the manually annotated data for building training, development, and test datasets, which were to be used by the participants in the shared task. Previous editions of CLEF eHealth addressed similar tasks, such as the extraction of ICD-10 codes in death certificates for English and French [8], and the following year for French, Italian, and Hungarian [9].

The remainder of the paper is structured as follows: we describe details of the shared task in Section 2 and the participating teams and systems in Section 3. We presented the baselines that we developed in Section 4 and the results obtained by participants and baselines in Section 5.

2 Details of the Shared Task

In this section we describe details of the challenge, including the schedule of the event, the data that we released, and the evaluation that we carried out.

Schedule. We released the training data, which is split into a training and development datasets, to the participants on February 1st, 2019. During three months, the participants could utilize this data for training, tuning, and evaluating their system. We released the official test set on May 6th, 2019. The participants had one week to process the test data and prepare the submissions files, that had to be uploaded in to the submission system until May 13th, 2019. Each team was allowed to submit up to three runs for their systems, i.e., different configurations or approaches that they experimented during the development period. Manual (human-annotated) approaches were not allowed in the shared task.

Data. Our training data consisted of a set of 8,386 manually annotated NTSs which was split into two datasets: 7,544 NTSs for the training dataset and 842 NTSs for the development dataset. For the test set, we released a collection of 407 unseen NTSs, i.e., which were not included in the training data. Each NTS is divided in six sections, namely, title, objectives, benefits, harms, replacement, reduction and refinement.

Evaluation. We evaluated the predictions returned by the participating systems based on an automatic and a manual approach. We automatically evaluated the

[4] Task 1: `https://clefehealth.imag.fr/?page_id=26`

submissions based on the standard metrics of precision (P), recall (R) and f-measure (FM). We utilized the Python script that we released to the participants during the shared task.[5] For the manual validation, one of our annotators manually checked a total of 100 NTSs originated from false positives (FPs) and false negatives (FNs) returned by the best runs. We randomly selected 25 FPs and FNs from the best run of the two best-scoring teams, thus a total of 100 NTSs. During the manual validation, our expert checked whether the wrong predictions (FP or FN) were indeed false.

3 Teams and systems

We received 14 submissions from six teams originated from a total of six countries, as summarized in Table 1. We present a summary of each team and their systems below.

Table 1. List of participating teams.

Team	Institution	Country
DEMIR	Dokuz Eylul University	Turkey
IMS_UNIPD	University of Padua	Italy
MLT-DFKI	German Research Center for Artificial Intelligence (DFKI)	Germany
SSN_NLP	SSN College of Engineering	India
TALP_UPC	Universitat Politècnica de Catalunya, University of Sheffield	Spain, UK
WBI	Humboldt-Universität zu Berlin	Germany

DEMIR [1]. The DEMIR team developed an approach based on two phases. In the first phase, they utilized the Elasticsearch tool to perform k-Nearest Neighbor (kNN) and threshold-Nearest Neighbor (tNN). In the second phase, the codes were selected from the top ones using two majority voting approaches based on either the pre-defined top M codes or on the similarity scores of the corresponding NTSs. The team submitted three runs, namely, k-NN based on k=5 and M=2 (run1), tNN based on T=30 and M=3 (run2), and tNN based on T=80 and adaptive M.

IMS-UNIPD [5]. The IMS-UNIPD team experimented with three probabilistic Naïve Bayes (NB) classifiers, following the same approach that they used in previous editions of the Multilingual Information Extraction Task in CLEF eHealth. All models were based on a two-dimensional representation of probabilities. They submitted three runs based on the three NB classifiers, namely, Bernoulli (run1), Multinomial (run2) and Poisson (run3).

[5] `https://github.com/mariananeves/clef19ehealth-task1`

MLT-DFKI [2]. The MLT-DFKI team tried a variety of approaches, such as Conditional Neural Networks (CNN) and Attention models, which that usually used for Neural Machine Translation (NMT), among others. They obtained the best results when relying on Bidirectional Encoder Representations from Transformers (BERT) and, more specifically, on BioBERT which was trained on biomedical documents [7]. For using this approach, which is available for the English language, the team had to first automatically translate the NTSs using the Google Translate API. The team only submitted one run.

SSN-NLP [6]. The SSN-NLP team developed a multi-layer Recurrent Neural Network (RNN) with a Long Short Term Memory (LSTM) as recurrent unit. They experimented with two attention mechanisms, namely Normed Bahdanau (NB) and Scaled Luong (SL), and with the requirement of a minimum number of occurrences of a code as generated by the model. They submitted three runs, namely, NB attention and minimum two occurrences (run1), SL attention and minimum of two occurrences (run2), and SL attention, minimum 2 occurrences and all codes if no code is repeated more than once (run3).

TALP_UPC. The TALP-UPC team developed a simple semi-supervised system based on Machine Translation and Named Entity Recognition (NER). In a first step, the "Benefits" section was translated into English using the Amazon Translate API[6]. For NER, they used MetaMap[7] (online batch submission system) and considered only the ICD-10 vocabulary source. After the identification of the entities (codes), their parents in the ICD-10 hierarchy were also selected to the prediction list.

WBI [11]. The WBI team utilized a multilingual BERT text encoding model [4] and additional training data of German clinical trials[8] also annotated with ICD-10 codes. They also experimented with training various instances of the models and ensembling the predictions based on their average or on a logistic regression classifier. The team submitted three runs, namely, BERT multi-label (run1), ensemble based on the average (run2), and an ensemble based on logistic regression (run3).

4 Baseline Approaches

We developed some baselines systems to compare the results from the participants to a simple text classification approach. The automatic classification of NTSs according to the ICD-10 codes consists of a multi-class and multi-label problem. It is multi-class because the ICD-10-GM-2016 ontology contains a total of 270 (until level4) that could potentially be assigned to an NTS, while it is multi-label because more than one code can be assigned to a each NTS.

[6] `https://aws.amazon.com/translate/`
[7] `https://metamap.nlm.nih.gov/`
[8] from `https://www.drks.de/drks_web/`

We considered only supervised learning approach based on our training data, i.e. codes that do not appear in the training data cannot be identified by our baseline approaches. Given it is a multi-label problem, during the training phase and for the training dataset, one classifier is trained for each of the 270 codes (if training data is available). During the test phase, for the development and test datasets, and for each NTS, each of the above classifier is used for deciding regarding the assignment of the corresponding code to the summary. All documents were pre-processed using the standard tokenization and TF-IDF functionality available in the Python scikitlearn library[9]. We considered two types of experiments, one using all sections of the summaries, and one using only only the title and the benefits sections.

We followed the approaches based on Support Vector Machine (SVM) that was previously utilized for the MIMIC II dataset [10]. The authors proposed flat and hierarchical SVMs in which the hierarchical structure of the ICD-10 terminology is considered in the latter. Both SVM algorithms were based on the SVM implementation available in the Python scikitlearn library and the differences between the two approaches are described below.

Flat This approach does not make use of the hierarchical structure of the terminology neither when building the classifiers nor when classifying the NTSs from the test set. For the flat SVM approach, we built one classifier for each code based on the totality of the summaries in the training dataset, i.e. for each code, the positive training examples were the NTSs that contained the particular code, while the negative examples were all NTSs that did not contain the code. Therefore, the classifiers were trained on a very unbalanced data for those codes that occur very seldom in our training data.

Hierarchical In this approach, we consider the four levels of the hierarchy of the ICD-10 ontology, as considered in our manual annotations of the NTSs. The classifiers related to codes on level 1 were trained on the whole training data, in which the positive examples were the ones that contained the particular code and the negative examples were the one that did not contain the code. Therefore, the classifier for level 1 are not different from the ones built in the flat approach for these same codes. As for next levels, the classifier for a particular code was only trained on the NTSs which belonged to the corresponding parent code. For instance, the classifier for code C00-C97 (level 2) was trained on all NTSs that were assigned to chapter II. The positive examples were the one assigned to code C00-C97, while the negative examples were the all the others assigned to chapter II but not to C00-C97, for instance, those that belong to the other codes in this chapter, such as D00-D09, D10-D36 or D37-D48. Therefore, each classifier has a different number of training examples, but a more balanced one with regard to the proportion of positive and negative examples, in comparison to the flat approach.

[9] `https://scikit-learn.org/stable/`

Table 2. List of the results for baselines and submitted runs. All results are presented in descending order of the scores for f-measure, precision and recall. We highlight in bold the highest values for f-measure, precision and recall.

Team and Runs	TPs	FPs	FNs	Precision	Recall	F-Measure
WBI-run1	602	124	181	0.83	0.77	**0.80**
WBI-run2	581	108	202	0.84	0.74	0.79
WBI-run3	615	154	168	0.80	0.78	0.79
MLT-DFKI	670	382	113	0.64	**0.86**	0.73
DEMIR-run1	394	454	389	0.46	0.50	0.48
DEMIR-run2	341	348	442	0.49	0.44	0.46
DEMIR-run3	386	455	397	0.46	0.49	0.48
baseline-hierar-All	178	20	605	0.93	0.27	0.42
baseline-flat-All	154	4	629	**0.98**	0.23	0.38
baseline-hierar-TB	189	7	594	**0.98**	0.22	0.36
TALP_UPC	275	462	508	0.37	0.35	0.36
baseline-flat-TB	167	1	616	0.92	0.22	0.35
SSN_NLP-run2	210	871	573	0.19	0.27	0.23
SSN_NLP-run1	213	889	570	0.19	0.27	0.22
SSN_NLP-run3	265	1788	518	0.13	0.34	0.19
IMS_UNIPD-run3	40	361	743	0.10	0.05	0.07
IMS_UNIPD-run2	394	44278	389	0.009	0.50	0.017
IMS_UNIPD-run1	0	0	783	0	0	0

5 Results

In this section we present the results obtained by the runs submitted by the participating teams and by our baseline systems.

5.1 Automatic Evaluation

Table 2 summarizes the results for all runs and baselines. Details for all runs are described in Section 1. Regarding the baselines systems, we evaluated both approaches (flat and hierarchical) and using the whole text of the NTS (All) or just the title and benefits (TB) sections.

The results for all metrics varied considerably, ranging from null to up to more than 0.80. The best scores were the following, 0.8 of f.measure from the WBI (run1) team, 0.86 of recall for the MLT-DFKI team, and 0.98 of precision of two of our baselines. Excepted for our baseline systems, results for precision, recall and f-measure were quite balanced for all runs. In contrast to these, our baselines obtained a much higher precision (above 0.9) over the recall (around 0.2-0.3).

As expected, the current state-of-the-art approach for many natural language processing (NLP) tasks, i.e. BERT, obtained the best performance in the runs submitted by teams WBI and MLT-DFKI. However, other machine learning approaches, e.g. kNN and tNN from team DEMIR, could outperform the deep learning approaches proposed by team SSN_NLP.

Table 3. List of the identified FNs which were validated as incorrect, i.e., they have not been missed by the systems.

NTS identifier	WBI run1	MLT-DFKI
19568	H55-H59	H55-H59
19663	J40-J47, J80-J84	J40-J47
19776	R10-R19, XVIII	R10-R19
21184	C76-C80, C00-C97, II	C76-C80, C00-C97, II
21802	P05-P08, XVI	P05-P08, XVI
21953	X, J09-J18	X, J09-J18
21969	T80-T88, XIX	T80-T88, XIX

5.2 Manual Evaluation

As described in Section 2, one expert manually validated a random sample of 100 FPs and FNs from the best runs from the two best-scoring teams, namely, run1 from WBI and the only run submitted by team MLT-DFKI. The FNs and FPs were automatically detected by our evaluation script (cf. Section 2) with regards to our gold standard test set. We provide a discussion below about the errors in our gold standards that we found.

FNs. From the 25 NTSs from run1 of the WBI team, our expert found seven NTSs in which a total of 14 FNs codes were wrong (cf. Table 3). These were not codes missed by the run, but rather codes that were mistakenly assigned to the NTSs in our gold standard. The same seven NTSs also contained 12 wrong FNs codes detected for the run from team MLT-DFKI. Curiously, even though we randomly selected the FNs codes, both runs had practically the same FNs codes from the same seven NTSs.

FPs. From the 25 NTSs from run1 of the WBI team, our expert found 12 NTSs in which a total of 22 FPs codes were wrong (cf. Table 4). These were codes that the expert judged as correct but that were not originally included in our gold standard. For the run from team MLT-DFKI, our expert judged as correct predictions just nine codes from four NTSs, from the total of 25 NTSs that were manually evaluated.

6 Conclusions

We presented the first corpus of non-technical summaries (NTS) of animal experiments for the German language. We annotated the NTS with the ICD-10 codes and utilized the data in the scope of a shared task in the CLEF eHealth challenge. Runs from two of the participants obtained results above 0.80 of f-measure and outperformed our baseline systems. The results obtained by the participants show that automatizing this task is indeed feasible, for instance, for the development of a semi-automatic system to support the experts in the manual annotation of the NTSs.

Table 4. List of the identified FPs which were validated as incorrect, i.e., they are indeed correct predictions from the systems.

NTS identifier	WBI run1	MLT-DFKI
18805	F10-F19, V	
19776	N80-N98, XIV	N80-N98, XIV
21969	C00-C75, C00-C97, C50-C50, II	C00-C75, C00-C97, C50-C50, II
20906	XXI, Z80-Z99	
16241	C76-C80	
21953	N17-N19, XIV	N17-N19, XIV
19599	XIX	
20619	C00-C75, C15-C26	
17716	C76-C80	
17108	D80-D90, III	
18865	I10-I15	
22344	C00-C75, C60-C63	
18318		C76-C80

Acknowledgment

We would like to thank all participants for their interest in our task, and Felipe Soares for providing a description of his team's system. We would like to acknowledge the Australian National University for supporting the submission Web site in EasyChair.

References

1. Ahmed, N., Arıbaş, A., Alpkocak, A.: DEMIR at CLEF eHealth 2019: Information Retrieval based Classification of Animal Experiment Summaries. In: CLEF (Working Notes) (2019)
2. Amin, S., Neumann, G., Dunfield, K., Vechkaeva, A., Chapman, K., Wixted, M.: MLT-DFKI at CLEF eHealth 2019: Multi-label Classification of ICD-10 Codes with BERT. In: CLEF (Working Notes) (2019)
3. Bert, B., Dörendahl, A., Leich, N., Vietze, J., Steinfath, M., Chmielewska, J., Hensel, A., Grune, B., Schönfelder, G.: Rethinking 3r strategies: Digging deeper into AnimalTestInfo promotes transparency in in vivo biomedical research. PLOS Biology **15**(12), 1–20 (12 2017). https://doi.org/10.1371/journal.pbio.2003217, `https://doi.org/10.1371/journal.pbio.2003217`
4. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018), `http://arxiv.org/abs/1810.04805`
5. Di Nunzio, G.M.: Classification of Animal Experiments: A Reproducible Study. IMS Unipd at CLEF eHealth Task 1. In: CLEF (Working Notes) (2019)
6. Kayalvizhi, S., Thenmozhi, D., Aravindan, C.: Deep Learning Approach for Semantic Indexing of Animal Experiments Summaries in German Language. In: CLEF (Working Notes) (2019)
7. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. CoRR **abs/1901.08746** (2019), `http://arxiv.org/abs/1901.08746`

8. Névéol, A., Anderson, R.N., Cohen, K.B., Grouin, C., Lavergne, T., Rey, G., Robert, A., Zweigenbaum, P.: CLEF eHealth 2017 multilingual information extraction task overview: ICD10 coding of death certificates in English and French. In: Proc of CLEF eHealth Evaluation lab. Dublin, Ireland (September 2017)
9. Névéol, A., Robert, A., Grippo, F., Morgand, C., Orsi, C., Pelikan, L., Ramadier, L., Rey, G., Zweigenbaum, P.: CLEF eHealth 2018 Multilingual Information Extraction Task Overview: ICD10 Coding of Death Certificates in French, Hungarian and Italian. In: Proc of CLEF eHealth Evaluation lab. Avignon, France (September 2018)
10. Perotte, A., Pivovarov, R., Natarajan, K., Weiskopf, N., Wood, F., Elhadad, N.: Diagnosis code assignment: models and evaluation metrics. Journal of the American Medical Informatics Association **21**(2), 231–237 (2014). https://doi.org/10.1136/amiajnl-2013-002159, http://dx.doi.org/10.1136/amiajnl-2013-002159
11. Sänger, M., Weber, L., Kittner, M., Leser, U.: Classifying German Animal Experiment Summaries with Multi-lingual BERT at CLEF eHealth 2019 Task 1. In: CLEF (Working Notes) (2019)
12. Taylor, K., Rego, L., Weber, T.: Recommendations to improve the EU non-technical summaries of animal experiments. ALTEX - Alternatives to animal experimentation **35**(2), 193–210 (Apr 2018). https://doi.org/10.14573/altex.1708111, https://www.altex.org/index.php/altex/article/view/90