Identification of Cancer Entities in Clinical Text Combining Transformers with Dictionary Features

John D. Osborne[a], PhD, Tobias O'Leary[a], BSc, James Del Monte[a], Kuleen Sasse[a] and Wayne H. Liang[a], MD MS

[a]*University of Alabama at Birmingham, 720 2nd Ave South, Birmingham, 35294, Alabama, USA*

Abstract

Clinical NLP tools that automatically extract cancer concepts from unstructured Electronic Health Record (EHR) text can benefit cancer treatment matching, clinical trials cohort identification, and reportable cancer abstraction. We used a combination of two BERT-based [1] language models, BETO [2] and MBERT [1]; with regular expressions constructed from training data; and ICD-O dictionary based features to participate in the tumor named-entity recognition subtask of the 2020 **CANTEMIST** (CANcer TExt Mining Shared Task) [3]. Our goal is to explore the incorporation of dictionary-based features into these models to provide better integration between machine learning models and external knowledge resources. Results on the test data set were highest with a regular expression based system (F-Score 0.73) and development set results showed a 5 point drop in F-Score (0.76 to 0.71) when integrating dictionary features into our BETO based system. We suggest that dictionary-based features will need careful integration to improve the performance of masked language models.

Keywords

clinical concept recognition, NLP, named entity recognition, information extraction, cancer

1. Introduction

The widespread adoption of Electronic Health Records (EHR) has resulted in an explosion in the volume of clinical data captured electronically. Computerized methods such as data analytics and clinical decision support (CDS) can be applied on clinical data to accelerate new scientific discoveries and improve clinical care delivery. This is particularly important in the field of oncology: cancer treatments are highly specific to cancer subtypes based upon clinical and tumor attributes (e.g., Philadelphia chromosome-positive B-cell Acute Lymphoblastic Leukemia); cancer clinical trials require identification of patients who meet high specific eligibility criteria (e.g., cancer subtype, clinical features, biomarker status); and cancer reporting for public health surveillance and quality assurance requires abstracting detailed cancer-related attributes from the clinical record. Each of the above examples would highly benefit from automated extraction of cancer concepts from the EHR[4]. However, much of the rich phenotype data required for the above examples are found not in machine-readable structured text, but are found

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: ozborn@uab.edu (J.D. Osborne); tobiasoleary@uab.edu (T. O'Leary); jvdelmon@uab.edu (J.D. Monte); ksasse@uab.edu (K. Sasse); wliang@uabmc.edu (W.H. Liang)

URL: https://github.com/ozborn/ (J.D. Osborne)

ORCID: 0000-0002-0851-1150 (J.D. Osborne); 0000-0001-7116-9338 (T. O'Leary); 0000-0003-2354-9787 (W.H. Liang)

IberLEF 2020, September 2020, Málaga, Spain.

CEUR Workshop Proceedings (CEUR-WS.org)

solely in unstructured texts (e.g., clinic notes, pathology reports, radiology reports)[5]. Natural Language Processing (NLP) tools that can automatically and accurately extract cancer concepts from unstructured clinical texts can increase the spectrum of data available for computational methods, thereby benefiting cancer research and care delivery. In this report, we compare multiple approaches to extracting cancer entities from CANTEMIST[3], a Spanish language clinical text data set.

1.1. Background

The first task of structured data extraction is the identification of the specific span of text (mention) containing the name of interest. This is referred to as Named-Entity Recognition (NER), or clinical entity recognition in the context of clinical text. NER software that has been developed specifically for clinical text includes cTAKES[6], CliNER[7] and other machine learning approaches utilizing support vector machines[8] and conditional random fields[9]. More recent methods have applied neural networks to clinical NER [10, 11, 12], including Deep Learning (DL) methods[13]. In particular, the development of the transformer architecture[14] and masked language models like BERT[1] and its siblings has yielded impressive results on non-clinical benchmarks like SuperGLUE[15]. Subsequently, a variety of English clinical language embeddings[16, 17, 18] have been developed, as well as non-clinical multilingual models such as MBERT[1] and language-specific models such as BETO[2].

Relatively little attention has been given to integrating dictionary features for large clinical vocabularies into these types of architectures for clinical NER. One recent exception[19] incorporated dictionaries into a Bi-LSTM-CRF DL model by integrating feature vectors with character embeddings, obtaining good results for Chinese clinical NER. Incorporation of dictionaries into DL models could allow for both the higher performance of DL models, while yielding the user control and understanding provided by dictionaries. For example, dictionary integration could allow for easier incorporation of vocabulary updates, such as changes to the International Classification of Diseases for Oncology (ICD-O) codes, or changes to cancer reporting requirements for tools like the Cancer Registry Control Panel (CRCP)[20].

For this paper, we explore the integration of transformer-based language models (such as BETO and MBERT) with external knowledge resources, as well as their applicability to clinical entity normalization for cancer concepts.

2. Methodology

We developed a total of 8 different systems for this task, including *BETO-FLAIR*, *REGEX*, *BETO-FLAIR-REGEX*, *MBERT-REGEX*, *MBERT-PYTORCH*, *MBERT-DICT*, *BETO-PYTORCH* and *BETO-DICT*. Only 3 systems, *BETO-FLAIR*, *MBERT-PYTORCH* and *REGEX*, were finished in time to be official entries for the CANTEMIST shared task, but results are shown for all systems. All systems with masked language models were constructed utilizing Huggingface's implementations of transformer-based language models[21], specifically BETO[2] and MBERT[1], the multi-lingual extension of BERT. Specific details of the language models are in their own sections below.

We used three distinct methods for predicting annotations: **fine-tuned masked language models** (*BETO-FLAIR*, *BETO-PYTORCH* and *MBERT-PYTORCH*), **regular expressions** alone

Table 1
Cantemist Data Set

Data Set	Reports	Sentences	Tokens
train-set	501	18540	456447
dev-set1	250	9092	226371
dev-set2	250	8332	183356
test-set	300	10727	248770

Table 2
Number of Reports, Sentences, and Tokens in each annotated data set.

or in conjunction with masked language models (*REGEX*, *BETO-FLAIR-REGEX* and *MBERT-REGEX*), and **dictionary features** in conjunction with masked language models (*MBERT-DICT* and *BETO-DICT*). BETO is a BERT language model pretrained on a large Spanish corpus and outperforms MBERT on several Spanish tasks, including natural language inference, paraphrasing, NER, and document classification[2]. Multilingual BERT or MBERT is a cross-lingual extension of BERT, trained to perform on 104 languages (https://github.com/google-research/bert/blob/master/multilingual.md#list-of-languages). Similar to the *BERT-base* and *BETO* models, *MBERT* has 110 million parameters. *MBERT* was pretrained using text data from Wikipedia in each of the top 104 largest languages by number of articles, and features a wordpiece vocabulary of 119,000 words, shared for all languages. Regular expressions were included to create a baseline for comparison.

2.1. Data

The Cantemist NER subtask is an information extraction task to identify tumor morphology mentions in a Spanish language corpus of synthetic oncological clinical case reports. The corpus was annotated with a single class, "MORFOLOGIA_NEOPLASIA," in the BRAT[22] standoff format. The corpus was split into 4 sets: train-set, dev-set1, dev-set2 and test-set, and included an unannotated background set that was not utilized. Table 1 contains the overall size of these sets. Also available was a list of ICD-O-3 codes (valid_codes.txt) containing the morphology codes and an associated term and comment. This was used for dictionary-based systems; no other third party data were used to develop our systems.

2.2. Pre-Processing

Cantemist input files in BRAT standoff format (.ann files) were converted to CoNLL format for processing by all systems, except REGEX. Input text was tokenized using the NLTK Spanish tokenizer and converted to either a 2 or 3 column format. The first column specified the token, and the second column specified the Cantemist tag in IOB (Inside-Outside-Begin) format. An optional third column was used in *BETO-FLAIR-REGEX* and *MBERT-REGEX* to specify the logits resulting from the pytorch-based *MBERT-REGEX* system, which was used to adjust the cutoff

Table 3
Example of Regular Expression Construction

<table>
<tr><th>Annotated Strings</th><th>Regular Expression</th></tr>
<tr><td></td><td>(?:\W)</td></tr>
<tr><td>adenocarcinoma mucinoso</td><td>([Aa]denocarcinoma\s+mucinoso\s+</td></tr>
<tr><td>moderadamente diferenciado</td><td>moderadamente\s+diferenciado|</td></tr>
<tr><td>tumoración sólida, estirpe mesenquimal</td><td>[Tt]umoración\s+sólida,\s+estirpe\s+mesenquimal|</td></tr>
<tr><td>diseminación intracraneal</td><td>[Dd]iseminación\s+intracraneal|</td></tr>
<tr><td>(LOEs) hepáticas</td><td>[^a-zA-ZÁÉÍÓÚÑÜáéíóúñü]?LOEs
[^a-zA-ZÁÉÍÓÚÑÜáéíóúñü]?\s+hepáticas|</td></tr>
<tr><td>pT2bN0M1</td><td>[Pp]T2bN0M1|</td></tr>
<tr><td>HCC</td><td>HCC|</td></tr>
<tr><td>CCR</td><td>CCR)</td></tr>
<tr><td>Ca</td><td></td></tr>
<tr><td></td><td>(?=\W)</td></tr>
</table>

frequency for the regular expression component (*REGEX*). During the conversion, many of the annotations had overlaps which affected the performance of the converter. To handle overlapping annotations, we kept only the longest of the overlapping annotation, based on span length.

2.3. *BETO-FLAIR*

For our first system, we used the sequence tagger from Flair version 0.4.2. We loaded the pretrained BETO[1] cased model as the base model for our sequence tagger [2]. BETO is a BERT model that was pretrained with a Spanish corpus of approximately 3 billion Spanish words using a similar architecture to BERT-base. Both have the same 110 million parameters. However, BETO has 32,000 words in its vocabulary, compared with 30,000 in BERT-base. We trained the model for 20 epochs with a batch size of 32 using the train set as the training data. We validated on dev-set1 and tested on dev-set2. This language-specific masked language model system was used as a reference point for an "off the shelf" clinical NER implementation.

2.4. *REGEX*

We constructed a regular expression by joining together a unique list of each annotated mention of cancer in the training and development sets. When evaluating against dev-set2 we excluded annotation in that set. We removed a single two-letter string, 'Ca', because it generated more false positives than true positives. After replacing any regex escape character to match any non-Spanish letter, we allowed the first character of the string to match both its upper and lowercase forms, if the length of the string was greater than 5. This boundary was chosen by manually reviewing the output. We listed these expressions from longest to shortest and concatenated all of them together with the regex 'or' operator, enforcing a word-boundary before and after. See Table 3 for a clarifying example. Any regex match was predicted as "MORFOLOGIA_NEOPLASIA".

2.5. *MBERT-PYTORCH* and *BETO-PYTORCH*

Both implementations use Huggingface's Transformers[21] library to provide the *MBERT-PYTORCH* model (using the pretrained 'bert-base-multilingual-cased' model) or the large cased model of *BETO BETO-PYTORCH*. Data from CoNLL files are packed into samples as close as possible to *BERT*'s maximum of 512 tokens. The assigned labels consist of IOB tokens, indicating whether a given subword is the beginning ("B") of a match, the interior ("I") of a match, or outside ("O") the scope. The model functions as a standard pytorch model and trained for 4 epochs. The batch size was 8; however, the implementation makes use of gradient accumulation to only calculate gradients after a specified number of training steps, effectively giving the model a batch size of 32 to match the *BERT* paper. Since Huggingface provides tools only for subword (token) classification and sample classification, we used the label generated for the first subword in a word as the label for that word for both *MBERT-PYTORCH* and *BETO-PYTORCH*.

2.6. *BETO-FLAIR-REGEX* and *MBERT-REGEX*

The *MBERT-PYTORCH* and *BETO-FLAIR* systems were extended by integration with the *REGEX* system. Both systems were modified to output a confidence score, ranging from 0 to 1, associated with the prediction for each token. We obtain these scores by applying softmax to the logit values returned by the *BERT* model. For tokens that were initially classified as not "MORFOLOGIA_NEOPLASIA" (the "O" class) that were later contained in a span of text the *REGEX* system classified as "MORFOLOGIA_NEOPLASIA," we would adjust the confidence score by +0.15. If the adjusted confidence score of any of the tokens crossed the 0.5 boundary, we changed the classification to "MORFOLOGIA_NEOPLASIA" for all tokens matched in the regular expression. The confidence score adjustment of +0.15 was chosen empirically after manually reviewing the output of the two systems. The decision to apply the adjustment solely to the "O" class was made in hopes of improving recall without reducing precision. This resulted in only a few adjustments to the predicted results of the two systems: *BETO-FLAIR-REGEX* resulted in 1998 annotation changes compared with *BETO-FLAIR* alone across both the test and background sets, and *MBERT-REGEX* resulted in 59 changes compared with *MBERT-PYTORCH* alone.

2.7. *MBERT-DICT* and *BETO-DICT*

The *MBERT-DICT* and *BETO-DICT* systems extends MBERT-PYTORCH and *BETO-PYTORCH* respectively by the use of dictionary-based features described in the next section. We wrote a custom head for the BertForTokenClassification model which concatenates these dictionary-based features with the logits corresponding to each subword in a sample. These extended samples run through two stacks of dropout and fully-connected layers, first mapping to the *HIDDEN_SIZE*, then applying the standard mapping to *NUM_LABELS*. Results for this model converge in 4 epochs.

2.8. Dictionary Features

Table 4 summarizes dictionary-based features used in the *BETO-DICT* and *MBERT-DICT* systems. Features were selected in order to assess both the coverage and the cohesiveness of input mentions relative to a *term representation* of all term names and synonyms in the Cantemist-provided ICD-O dictionary (valid-codes.txt file) for an entry.

For subword-based dictionary features, the presence of input mention text dictionary-based features were calculated for each input subword present in the mention. We utilized the entire 512 BETO subword limit as our lookup window.

Character-based dictionary features were calculated using a Python string similarity library (https://github.com/luozhouyang/python-string-similarity). Character-based comparisons were made between a character-based *term representation* of the dictionary term and the lookup window. Parameters for character-based dictionaries, including overlap co-efficient and shingle size, were determined empirically, resulting in values of 3 and 5 respectively. All dictionary features are described in detail below.

Highest subword coverage The highest subword coverage calculates the number of overlapping subwords in the *subword window* for each ICD-O term's dictionary name and any of its respective synonyms (*term representation*) to compute a subword overlap count for each term. The highest subword coverage is the count of subword overlaps for the term with the maximum subword overlaps for the dictionary.

Distinct subword Each subword in the *subword window* is matched against all dictionary *term representation*s resulting in a set of all the entries with at least one subword. One or more subwords will return a set with the lowest cardinality. That minimum cardinality over all returned sets from all subwords in the *subword window* is the distinct subword.

Average matches The average number of *subword window term representation* over the entire dictionary.

Fraction of entries with highest subword coverage The fraction of entries whose *term representation* is the maximum number of subwords represented in the *subword window*.

Differential subword coverage The differential subword coverage is computed by taking the "Highest subword coverage" described above and subtracting the average subword coverage or overlap between the *subword window* and all subword *term representation* in the ICD-O dictionary.

Best entry log subword frequencies The subword frequency for each subword in the ICD-O dictionary is computed for all subwords overlapping between the *subword window* and the subword *term representation*. The log of these frequencies is summed for each comparison and the highest value is used as the "best entry log subword frequencies".

Table 4
Dictionary Features for Named Entity Normalization

Feature Name	Feature Type
Highest subword coverage	Subword
Lowest subword specificity	Subword
Average matches	Subword
Fraction of entries with highest subword coverage	Subword
Differential subword coverage	
Best entry log subword frequencies	Subword
Overlap coefficient	Character
N-Gram	Character

Table 5
Official Test Results

System	Prec	Recall	F-Score
BETO-FLAIR	**0.736**	0.609	0.667
MBERT-PYTORCH	0.673	0.357	0.467
REGEX	0.688	**0.744**	**0.715**

Overlap coefficient An overlap coefficient of 3 is used to calculate a Szymkiewicz-Simpson similarity score between the mention subword and 10 adjacent characters on either side ("*character extended subword window*") to a *character-based term representation.*

Shingle N-Grams Character 5-gram profiles were pre-computed for each *character-based term representation* and compared to a profile from the input mention's *character extended subword window* using cosine similarity .

3. Results

Official test results are shown in Table 5, and unofficial test results including all systems developed are shown in Table 6. Only 3 systems were submitted in time for completion, but we show test results for all data using the included eval script. The discrepancies between the official and unofficial test results are caused by changes made to the utility that converted .conll and .ann files, changes in the regex system to more loosely match escape characters, and processing all files with *MBERT-PYTORCH.* Our original submission included only 80% of files. Our best performing system for the official test results was the baseline REGEX system which reflects underdevelopment of the masked language model systems, although *BETO-FLAIR* had the highest precision. This is replicated in the unofficial results. Additionally we show results on the development set in Table 7. The *BETO-PYTORCH* obtained the best results in all 3 evaluation metrics.

Table 6
Updated Unofficial Test Results

System	Prec	Recall	F-Score
BETO-FLAIR	**0.74**	0.61	0.67
BETO-PYTORCH	0.71	0.68	0.70
MBERT-PYTORCH	0.69	0.30	0.42
REGEX	0.70	**0.77**	**0.73**
BETO-FLAIR-REGEX	0.67	0.64	0.66
MBERT-REGEX	0.63	0.68	0.66
BETO-DICT	0.61	0.71	0.66
MBERT-DICT	0.64	0.67	0.65

Table 7
Development Data Set Results

System	Prec	Recall	F-Score
BETO-FLAIR	0.68	0.61	0.64
BETO-PYTORCH	**0.76**	**0.76**	**0.76**
MBERT-PYTORCH	0.69	0.73	0.71
REGEX	0.67	0.74	0.70
BETO-FLAIR-REGEX	0.68	0.61	0.64
MBERT-REGEX	0.69	0.73	0.71
BETO-DICT	0.67	0.58	0.62
MBERT-DICT	0.67	0.74	0.70

4. Discussion

We were disappointed by the poor performance of pure transformer-based systems on the test data relative to the simple regular expression-based system *REGEX*. We also tested using a pure regular expression dictionary matching approach, but performance was worse than simply looking for exact matches in the training data (data not shown). However, on the development data *BETO-PYTORCH* did produce the best results. Our efforts to integrate dictionary features with masked language models also yielded disappointing results. We suspect the poor performance of the dictionary-based approach is due to limited development time, the reliance on subwords (versus words), overly large lookup window (512 subwords instead of a sentence or smaller window) and the lack of dictionary feature validation and testing rather than the integration of these features into the BERT model.

Combining BERT based model with REGEX did not result in a significant improvement. Recall was slightly higher when evaluating on the test data set, but at a cost of lower precision. High recall with lower precision is naturally expected when using regular expression-based systems for NER. Picking a single confidence score adjustment and applying that across multiple trained models also likely caused lower performance, since each model's average confidence score for a given class was significantly different.

Future Directions In the short term, we plan to expand the number of dictionary-based features to better account for term variation and head nouns. Word level features also need to be introduced, and the utility of subwords to handle medical abbreviations and relevant Latin and Greek roots needs to be evaluated. Appropriate medical stemming, or use of a clinical subword vocabulary, also needs to be evaluated. We are also interested in cross-language evaluation (English-Spanish) of cancer extraction terms.

4.1. Limitations

Our work suffered from a number of limitations, the most important being the lack of a Spanish speaker in our group, forcing us to rely on Google Translate and the similarity of Latin-based medical terms. Due to time constraints, we did not perform a principled dictionary feature evaluation to assess the relative importance of features. Parameter settings were not fully evaluated for similar reasons.

Acknowledgments

This publication was supported by internal funding from the Informatics Institute at University of Alabama at Birmingham, and a Nvidia$^{\text{TM}}$ grant of a Titan XP GPU used for machine learning.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[2] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Spanish pre-trained bert model and evaluation data, in: to appear in PML4DC at ICLR 2020, 2020.
[3] 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.
[4] G. K. Savova, I. Danciu, F. Alamudun, T. Miller, C. Lin, D. S. Bitterman, G. Tourassi, J. L. Warner, Use of natural language processing to extract clinical cancer phenotypes from electronic medical records, Cancer Research 79 (2019) 5463–5470.
[5] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, K. B. Johnson, Data from clinical notes: a perspective on the tension between structure and flexible documentation, Journal of the American Medical Informatics Association 18 (2011) 181–186.
[6] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, C. G. Chute, Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications, Journal of the American Medical Informatics Association 17 (2010) 507–513.
[7] W. Boag, K. Wacome, T. Naumann, A. Rumshisky, Cliner: a lightweight tool for clinical named entity recognition, AMIA Joint Summits on Clinical Research Informatics (poster) (2015).

[8] B. Tang, H. Cao, Y. Wu, M. Jiang, H. Xu, Clinical entity recognition using structural support vector machines with rich features, in: Proceedings of the ACM Sixth International Workshop on Data and Text Mining in Biomedical Informatics, 2012, pp. 13–20.
[9] Y. Zhang, X. Wang, Z. Hou, J. Li, Clinical named entity recognition from chinese electronic health records via machine learning methods, JMIR Medical Informatics 6 (2018) e50.
[10] Z. Liu, M. Yang, X. Wang, Q. Chen, B. Tang, Z. Wang, H. Xu, Entity recognition from clinical texts via recurrent neural network, BMC Medical Informatics and Decision Making 17 (2017) 67.
[11] F. Dernoncourt, J. Y. Lee, P. Szolovits, Neuroner: an easy-to-use program for named-entity recognition based on neural networks, arXiv preprint arXiv:1705.05487 (2017).
[12] F. Dernoncourt, J. Y. Lee, O. Uzuner, P. Szolovits, De-identification of patient notes with recurrent neural networks, Journal of the American Medical Informatics Association 24 (2017) 596–606.
[13] Y. Wu, M. Jiang, J. Xu, D. Zhi, H. Xu, Clinical named entity recognition using deep learning models, in: AMIA Annual Symposium Proceedings, volume 2017, American Medical Informatics Association, 2017, p. 1812.
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[15] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, Superglue: A stickier benchmark for general-purpose language understanding systems, in: Advances in Neural Information Processing Systems, 2019, pp. 3266–3280.
[16] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, M. McDermott, Publicly available clinical bert embeddings, arXiv preprint arXiv:1904.03323 (2019).
[17] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2020) 1234–1240.
[18] K. Huang, J. Altosaar, R. Ranganath, Clinicalbert: Modeling clinical notes and predicting hospital readmission, arXiv preprint arXiv:1904.05342 (2019).
[19] Q. Wang, Y. Zhou, T. Ruan, D. Gao, Y. Xia, P. He, Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition, Journal of Biomedical Informatics 92 (2019) 103133.
[20] J. D. Osborne, M. Wyatt, A. O. Westfall, J. Willig, S. Bethard, G. Gordon, Efficient identification of nationally mandated reportable cancer cases using natural language processing and machine learning, Journal of the American Medical Informatics Association (2016) ocw006.
[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface's transformers: State-of-the-art natural language processing, arXiv preprint arXiv:1910.03771 (2019).
[22] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, J. Tsujii, BRAT: a web-based tool for NLP-assisted text annotation, in: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2012, pp. 102–107.