Chemical Entity Recognition for MEDLINE Indexing

Max E. Savery, Willie J. Rogers, Malvika Pillai, James G. Mork, MS, Dina Demner-Fushman, MD, PhD
Lister Hill National Center for Biomedical Communications, U.S. National Library of Medicine, National Institutes of Health, Bethesda, MD

Abstract

Chemical entity recognition is essential for indexing scientific literature in the MEDLINE database at the National Library of Medicine. However, the tool currently used to suggest terms for indexing, the Medical Text Indexer, was not originally conceived as a chemical recognition tool. It has instead been adapted to the task via its use of MetaMap and the addition of in-house patterns and rules. In order to develop a tool more suitable for chemical recognition, we have created a collection of 200 MEDLINE titles and abstracts annotated with genes, proteins, inorganic and organic chemicals, as well as other biological molecules. We use this collection to evaluate eleven chemical entity recognition systems, where we seek to identify a tool that effectively recognizes chemical entities for indexing and also performs well on chemical recognition beyond the indexing task. We observe the highest performance with a SciBERT ensemble.

Introduction

Over 900,000 articles in the U.S. National Library of Medicine's (NLM) MEDLINE® database are indexed with Medical Subject Headings (MeSH®). MeSH terms are recommended by the Medical Text Indexer (MTI)[1] tool and manually assigned to MEDLINE articles by human indexers. MeSH covers many biomedical categories such as diseases, psychiatry, anatomy, and of particular interest to this paper, chemicals. Chemical MeSH recommendations present a unique challenge to MTI. To recommend chemical MeSH terms for indexing, MTI must first perform chemical entity recognition (CER). To do so, it relies on pattern and rule-based approaches, as well as MetaMap.[2] This approach has been able to adequately meet the needs of the indexers, but it has serious drawbacks. For example, MetaMap is able to identify chemicals to the extent that they can be mapped to the Unified Medical Language System (UMLS®) Metathesaurus®. This covers a wide range of chemical entities but cannot account for all variations. Additionally, flagging new chemicals is important to indexing as well, but the current approach is capable of recognizing only in-vocabulary entities.

To address the issue of CER in the past, NLM has attempted to supplement MTI with other tools, but in practice, these approaches were not yet mature enough to improve performance as desired. However, no formal study has been conducted. Because the field of CER has developed rapidly in the past few years, we are interested in making a systematic attempt to identify a tool that effectively contributes to MTI. Our criteria are straightforward: A CER tool must provide MTI with an exhaustive list of critical chemical terms occurring in the text; and we would prefer a tool perform CER without the intensive use of lexical resources, as these are difficult to maintain and update.

Here, we evaluate existing automatic chemical annotation systems, as well as experimental deep learning approaches. To conduct our evaluation, we have created an annotated test collection of 200 MEDLINE titles and abstracts. To thoroughly evaluate CER in the indexing environment, we have annotated mentions of organic and inorganic chemicals, genes, proteins, and other biological molecules, including nested and fragmented entities in our annotations. To our knowledge, it is the only existing corpus in which all of these entities have been annotated in titles and abstract, and it has been made publicly available along with the code for this paper.* Using this corpus, referred to as the Chemical Entity Mentions for Assessing MTI (ChEMFAM) corpus, we evaluate eleven systems that encompass a wide range of entity recognition techniques, including pattern, rule, and lexical resource-based methods, traditional machine learning, and LSTM and transformer-based deep learning. We observe the best results with a SciBERT[3] ensemble, achieving a F1-score of 75.09%.

*The corpus, annotation guidelines, and code can be found at `https://github.com/saverymax/CER-for-MTI`

Related Work

A Annotated Corpora

Annotated data are essential for the evaluation of CER systems. Many corpora containing annotated chemical entities have been previously constructed: gene and gene product mentions in sentences from MEDLINE articles (BC2GM)[4]; indexing and mentions of chemicals in PubMed abstracts (BC4CHEMD)[5]; chemical and disease mentions and their interactions (BC5CDR)[6], chemical and gene/protein entity mentions in patents (BC5CHEMD-Patents)[7]; and DNA/RNA, proteins, and cell lines/cell types (JNLPBA)[8]. However, excepting the case of the BC5CHEMD-Patents task, these collections do not annotate proteins, genes, and chemicals in a single corpus in scientific abstracts, nor do any include nested or fragmented entities. To evaluate CER performance on multiple entity types, we could use multiple entity-specific datasets, such as in Crichton et al., 2017.[9] However, we need an evaluation that more closely mirrors the indexing environment, where a CER tool will encounter many classes of entities that may be nested and or noncontiguous (fragmented), occurring in scientific abstracts. The ChEMFAM corpus satisfies these requirements.

B Entity Recognition Systems

Named Entity Recognition (NER) has traditionally utilized pattern, rule, dictionary and gazetteer-based methods, as well as traditional machine learning methods such as Conditional Random Fields that use engineered features. Tools implementing these approaches for CER include ChemTagger,[10] ChemSpot,[11] ChemDataExtractor,[12] and LeadMine.[13] While much success has been achieved with these, they are limited by the time and expertise required to build the resources and feature sets, as well their ability to interpret contextual information and account for variation of chemical terminology.[14]

Recently, deep learning methods have achieved state-of-the-art performance in many CER tasks. These models learn to map n-dimensional distributed word or character embeddings to a desired output through non-linear activation functions. Often, embeddings pre-trained on biomedical text can be used to improve the performance of these models.[15] The typical deep learning model architecture for NER combines the pre-trained embeddings with LSTM and/or CNN layers, using these as input to a final CRF layer. For example, Habibi et al., 2017[16] presents a LSTM-CRF trained with character embeddings concatenated to word embeddings pre-trained on Wikipedia, PubMed, and PMC. ChemListem[17] employs an ensemble, where one network is trained on a feature set supplemented with GloVe embeddings and another network is trained using solely character embeddings.

However, the development of the multi-layer bidirectional transformer,[18] as implemented in BERT,[19] has been shown to outperform the LSTM-CRF models in NER. Pre-training and fine-tuning of the BERT architecture on biomedical and scientific text, as implemented in SciBERT and BioBERT,[20] has achieved state-of-the-art results in numerous biomedical tasks, including BC5CDR, BC2GM, and BC4CHEMD. The pre-trained weights of these models allow them to be adapted to new tasks relatively easily, in situations where large training datasets may not exist. Due to this flexibility, we apply these models to the recognition of chemical entities in our corpus, comparing them to previously developed, publicly available tools.

Methods

A Collection

To collect articles to be annotated, we used MeSH to limit our PubMed query to articles representative of each class of entity important for our purposes. We manually selected fifty citations representative of each class, though many citations have mentions of multiple classes.

The four class labels corresponded to MeSH tree terms: Inorganic Chemicals (D01), Organic Chemicals (D02), Amino Acids, Peptides, and Proteins (D12), and Nucleic Acids, Nucleotides, and Nucleosides (D13). While we did not map entities to MeSH, processing in MTI will do so; for this reason it was practical to consider the labels of our annotations in that context. Additionally, though the distinction between certain types of entities is biologically and chemically significant, such as lipids and carbohydrates, we do not require a tool to make this distinction. These types of entities were labeled with the general class they fit into and are further described in the annotation guidelines included with

the collection.

Using the BRAT tool,[21] the collection was manually annotated by two subject matter experts. The second annotator used the annotations of the first annotator as reference. Once initial annotations were complete, inter-annotator agreement was calculated with F1-score[22] using one set of annotations as the set of true labels. After discussing all decisions, the final collection was manually harmonized by the annotators and any further corrections to the corpus were made. A summary of the annotated entities can be seen in Table 1. The fully annotated collection can be found in the link provided in an earlier footnote.

Table 1: Summary of annotations in ChEMFAM Corpus

Entity Type	Mentions	Unique Mentions
Organic	880	414
Inorganic	704	306
Genes	724	299
Proteins	1128	537
Total	**3436**	**1556**

Briefly, our guidelines are described here: Entities are annotated only if they represent a specific instance of a chemical. We define an instance of a chemical to be an entity that can be mapped to IUPAC nomenclature, a named gene, gene product, or named protein expressed by a single gene or formed of multiple units expressed by multiple genes. Entities representing general concepts or sub-classes were not annotated, e.g., nucleotide or oligonucleotide, as we are only interested in evaluating identification of specific instances of a chemical. These general entities were added to a stop word list and removed from the tools' predictions in the evaluation. In the case of genes and proteins, nested entities were annotated. The fragment feature in BRAT was used to annotate all noncontiguous entities. Examples of nested and fragmented entities can be seen in Figure 1.

(a) **(b)**

Figure 1: BRAT screenshots of (a) nested entity and (b) fragmented entity

B Tools

We evaluated the automatic annotations of MTI as well as the following eleven entity recognition systems. Here we provide brief descriptions of each tool:

MetaMap Lite MetaMap Lite (MML)[23] is a speed-oriented, Java implementation of MetaMap that can be provided with custom vocabularies. We experimented with multiple versions, but we only report the version giving the highest F-score: Taking the set of all MeSH terms within the Chemical and Drugs (D) tree mentioned in articles in MEDLINE, we performed synonymy expansion with Custom Taxonomy Builder[24] to generate concept records for MML.

ChemDataExtractor ChemDataExtractor utilizes dictionaries, rule-based grammars, and supervised and unsupervised machine learning approaches for CER.

PubTator Central PubTator Central[25] is a web service combining multiple annotation tools into a single API. It is able to annotate diseases, chemicals, genes, proteins, variants, species, and cell lines, performing context disambiguation on the entity types with a deep learning module. Here we only consider PubTator's annotations of genes, proteins, and chemicals.

ChemListem ChemListem employs an ensemble described in the related work section.

LSTM-CRF The LSTM-CRF model[26] relies on character and GloVe[27] embeddings as input. We trained the model on chemical mentions in BC4CHEMD, using code provided in the repository listed below.[†]

BERT-Chem To generate the BERT-Chem model, we fine-tuned the BERT weights on chemical mentions in BC4CHEMD.

BERT-Gene For the BERT-Gene model, BERT was fine-tuned on gene and gene product mentions in BC2GM.

BERT-Ensemble The BERT-Ensemble was constructed by taking the set of predictions from the two BERT models above.

SciBERT-Ensemble Using the BERT architecture, the SciBERT author's pre-trained the model on biomedical and computer science papers from Semantic Scholar. After downloading the SciBERT model, we constructed the SciBERT-Ensemble with the same principle as the BERT-Ensemble: The set of predictions was taken from two SciBERT models respectively fine-tuned on BC2GM and BC4CHEMD.

BioBERT-Ensemble Similarly to SciBERT, BioBERT is pre-trained on biomedical text: PubMed and PMC. We constructed the BioBERT-Ensemble as described above.

XLNet-Ensemble XLNet[28] implements pre-training with autoregression, using permutation language modeling to learn the context of the tokens. Using the XLNet architecture, we constructed the XLNet-Ensemble as described above.

The publicly available tools were implemented with out-of-the-box settings. For all models using BERT architecture, we used the cased, base version of the pre-trained BERT. These models were fine-tuned for 3 epochs, with a learning rate of 3e-5, batch size of 4, and sequence length of 512. The cased, base version of XLNet was also used. As the authors of XLNet did not provide hyperparameters for NER, we used the hyperparameters they provided for the question answering task (SQuAD), as this is also a span-based task in which the system has to extract short spans of text from passages containing answers to the questions.[29] All deep learning models output annotations in the BIO annotation scheme, where tokens predicted to be at the beginning of an entity are labeled with B, and tokens predicted to be inside or at the end of an entity are labeled with I. O is used to label all other tokens. Post-processing consisted of joining the B and I labels for the subword tokens produced by WordPiece[30] (BERT) and SentencePiece[31] (XLNet). The output of MTI required special processing which will be explained in the next section.

C Evaluation

For each system, micro averaged F1-score, recall, and precision were calculated. For this study, we merge all classes of annotations in order to understand which tool will give us the best performance overall. To simplify the diverse outputs produced by the tools tested, we take the set of all entities identified per article, similarly to the chemical document indexing sub-task in BC4CHEMD.

Matches between predictions and manual annotations were measured in two ways: First we used exact matches between the set of system annotations and the manual annotations to count true positives. In addition to the exact matching criteria, we reasoned that the machine learning methods which do not rely on dictionary lookups would benefit more from using relaxed match criteria than the other tools. We therefore calculated inexact matches with the Levenshtein distance,[‡] the least number of insertions, deletions or replacements of single characters required to make one string equal to another.[32] In order to avoid rewarding the editing of short acronyms and element abbreviations (three characters or fewer) as true positives, we normalized the Levenshtein distance by the length of the predicted span, using

$$L < 1/3$$

as a threshold to count true positives. Here L is the normalized Levenshtein distance, the numerator refers to the number of edits, and the denominator refers to the predicted span. For example, the entity *EBV encoded small RNA*

[†]The code for the LSTM-CRF is available at `https://github.com/guillaumegenthial/tf_ner`

[‡]The leven Python package is available at `https://github.com/semanticize/leven`

could be edited to *EBV encoded small RNA 1*, resulting in a measurement of 1/21 after normalization and thus counting as a true positive. Further explanation of use of this metric is provided in the discussion section.

As stated earlier, the output of MTI is MeSH, and the MeSH vocabulary does not necessarily exactly match the strings occurring in the text. This means that we could not directly compare the output of MTI to the manually annotated text. Take, for example, the MeSH term *IL6 protein, human*. This term is unlikely to be found in the text, but it will be the output of MTI when triggered by the string *Il-6*. It is possible, however, to access the spans that trigger MTI's recommendation. We used these to retrieve the chemical strings for the evaluation of MTI.

Results

Inter-annotator agreement (F-score) between annotator 1 and 2 was determined to be 99.52%. Between annotator 1 and the harmonized mentions, agreement was 98.43%, and between annotator 2 and the harmonized mentions, 98.40%

Micro F1, precision, recall, and standard error of all tools can be seen in Table 2. MTI achieves 39.69% F1-score, 57.52% precision and 30.30% recall. ChemDataExtractor achieves the highest precision of 82.02%, and the SciBERT-Ensemble achieves the highest F1-score and recall of 75.09% and 79.41%, respectively. Both SciBERT and BioBERT ensembles perform significantly better than the BERT ensemble. By taking the standard deviation of micro-averaged metrics on 1000 bootstrapped samples of all articles in the collection, we were able to calculate standard error for each measurement, as performed in the BC4CHEMD evaluation.

Table 2: Micro-averaged performance of tools evaluated on ChEMFAM Corpus

	F1 (%)	**Precision (%)**	**Recall (%)**
MTI	39.69 ± 1.83	57.52 ± 2.24	30.30 ± 1.73
MML	42.95 ± 1.59	39.45 ± 1.57	47.13 ± 2.02
PubTator Central	65.34 ± 1.72	79.55 ± 1.55	55.45 ± 2.01
ChemDataExtractor	58.51 ± 2.04	$\mathbf{82.02 \pm 1.49}$	45.48 ± 2.27
LSTM-CRF	42.26 ± 1.90	64.80 ± 1.95	31.35 ± 1.85
ChemListem	48.35 ± 2.24	81.41 ± 1.79	34.39 ± 2.15
BERT-BC4CHEMD	53.95 ± 2.19	73.56 ± 1.68	42.95 ± 2.35
BERT-BC2GM	45.44 ± 2.07	60.17 ± 2.20	36.50 ± 2.13
BERT-Ensemble	71.40 ± 1.42	69.21 ± 1.58	73.73 ± 1.68
SciBERT-Ensemble	$\mathbf{75.09 \pm 1.31}$	71.22 ± 1.54	$\mathbf{79.41 \pm 1.40}$
BioBERT-Ensemble	74.19 ± 1.38	70.60 ± 1.55	78.15 ± 1.55
XLNet-Ensemble	57.26 ± 1.41	52.69 ± 1.53	62.71 ± 1.67

Table 3 shows MTI, PubTator Central, and the deep learning ensembles' performance with the relaxed metric. The XLNet-ensemble benefited the most, 8.3%, using this method of comparison. The highest performing ensemble, SciBERT, improved 1.65% more than did the highest performing publicly available tool, PubTator Central.

Table 3: Micro-averaged performance of tools using relaxed criteria

	F1 (%)	**Precision (%)**	**Recall (%)**
MTI	42.39 ± 1.83	61.40 ± 2.16	32.36 ± 1.77
PubTator Central	68.10 ± 1.62	$\mathbf{82.77 \pm 1.38}$	57.84 ± 1.98
BERT-Ensemble	75.61 ± 1.31	73.17 ± 1.51	78.21 ± 1.60
SciBERT-Ensemble	$\mathbf{79.50 \pm 1.15}$	75.19 ± 1.46	$\mathbf{84.33 \pm 1.22}$
BioBERT-Ensemble	79.20 ± 1.16	75.13 ± 1.42	83.72 ± 1.30
XLNet-Ensemble	65.56 ± 1.31	60.23 ± 1.53	71.92 ± 1.55

Discussion

Our intention for this study was twofold: First, to create a manually annotated corpus that could be used to assess the performance of CER systems in a setting more similar to the indexing environment than previously annotated corpora;

and second, to identify a tool that not only provides more comprehensive chemical recognition coverage than MTI, but also performs well on CER in general, without a set of curated, task-specific resources.

We show that MTI achieves lower F-score than all tools evaluated here. The reason for this is simple: MTI was never designed to be a chemical recognition tool. Using in-house rules and pattern recognition, and taking advantage of MetaMap's ability to recognize chemicals within the UMLS vocabulary, MTI has been able to identify chemicals to partially meet the needs of the indexers. Though its contribution to the indexers is not necessarily reflected in this evaluation, it is clear that the CER field has matured substantially since the development of MTI. We find that the BERT models pretrained on biomedical data not only perform better than the publicly available chemical tools and other machine learning methods we tested, but they did so without supplemental lexical features, lookup vocabularies, and tuning of hyperparameters. Of all the methods, the SciBERT-Ensemble has the potential to provide the most comprehensive list of chemicals to MTI. However, before we can implement any system in the indexing pipeline, the output must first be mapped to MeSH and tested in the context of that vocabulary. Further research will therefore focus on evaluating the extent to which MeSH recommendations change when MTI is supplemented with a CER system.

MTI uses numerous lexical mapping approaches to map strings in text to MeSH. It is for this reason that using the Levenshtein distance as a method to compute inexact matches is informative for the future implementation of a CER tool in MTI: The spans of strings that are not automatically annotated perfectly may still be mapped to the correct set of MeSH. In addition, the Levenshtein metric allows us to measure performance on the nested and fragmented entities to some extent. For example, the corpus includes the entity *interleukin (Il)-6*. This is manually annotated as fragments: *interleukin-6* and *Il-6*. However, some tools identify only *interleukin (Il)-6*. Using exact matching criteria, the entity would be counted as a false positive and the true entities as false negatives. If adjusted using the normalized Levenshtein distance, *Il-6* could be deleted from the predicted string, generating a true positive. The drawback of using this approach is that the systems can occasionally be rewarded for false negatives that were truly missed. Though this adds noise to the evaluation, the increased performance of approximately 3-5% for the tools shown in Table 3 indicates that some of the inexact predicted spans will still be useful to us.

To develop a CER tool to best fit our needs, we are primarily limited by the lack of training data. While many datasets annotated with chemical entities exist, there is no unifying annotation scheme, and combining the datasets that ostensibly annotate the same type of entities has been shown to decrease performance.[33] Since the deep learning models do not rely on resources other than the distribution and context of tokens, the amount of training data provided in the fine-tuning phase is particularly important. One possible solution to this problem is the implementation of a multitask model, such as MT-DNN,[34] where loss can be minimized on multiple objectives on multiple datasets during training, maximizing the available training data.

Future work will primarily focus on integrating the SciBERT-Ensemble chemical predictions into MTI's MeSH recommendations. Once the chemical annotations are mapped to MeSH and added to the recommendations, MTI can be evaluated on articles with MeSH indexing previously assigned. If the CER system's predictions are useful, MTI will be able to recommend an increased amount of chemical MeSH, as well as flag potentially novel chemicals. Additionally, performance on entity classes was not assessed for this study. This will also be considered in future work, particularly when conducting error analysis of the indexing recommendations.

Conclusion

This study presents the ChEMFAM corpus, in which entity mentions of inorganic and organic chemicals, genes, proteins, and other biological molecules have been annotated. To our knowledge, it is the only existing corpus in which all of these entities are annotated in titles and abstracts. Though we used this corpus to assess CER systems for the purpose of identifying chemicals for indexing in MEDLINE, it can be easily downloaded and used to assess any CER system.

Our results indicate that all tools tested here achieve better CER coverage than MTI. We show that the SciBERT-Ensemble may provide the greatest contribution to MTI in the indexing pipeline, and that in general, BERT architecture pre-trained on biomedical data and fine-tuned on chemical entity mentions outperforms other approaches.

Acknowledgements

This research was supported by the Intramural Research Program of the National Institutes of Health, National Library of Medicine, and Lister Hill National Center for Biomedical Communications.

References

[1] Mork JG, Jimeno-Yepes A, Aronson AR. The NLM Medical Text Indexer System for Indexing Biomedical Literature. In: Proceedings of the first Workshop on Bio-Medical Semantic Indexing and Question Answering, a Post-Conference Workshop of Conference and Labs of the Evaluation Forum 2013 (CLEF 2013) , Valencia, Spain, September 27th, 2013; 2013. p. 1–6.

[2] Aronson AR, Lang FM. An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association : JAMIA. 2010;17.

[3] Beltagy I, Cohan A, Lo K. SciBERT: Pretrained Contextualized Embeddings for Scientific Text; [Preprint] 2019.

[4] Smith L, Tanabe LK, Ando R, Kuo CJ, Chung IF, Hsu CN, et al. Overview of BioCreative II gene mention recognition. Genome Biology. 2008;9(2):S2.

[5] Krallinger M, Leitner F, Rabal O, Vazquez M, Oyarzabal J, Valencia A. CHEMDNER: The drugs and chemical names extraction challenge. Journal of Cheminformatics. 2015;7(1):S1.

[6] Wei CH, Peng Y, Leaman R, Davis AP, Mattingly CJ, Li J, et al. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. Database. 2016;2016.

[7] Krallinger M, Rabal O, Lourenço A, Perez MP, Rodriguez GP, Vazquez M, et al. Overview of the CHEMDNER patents task. In: Proceedings of the Fifth BioCreative Challenge Evaluation Workshop; 2015. p. 63–75.

[8] Kim JD, Ohta T, Tsuruoka Y, Tateisi Y, Collier N. Introduction to the bio-entity recognition task at JNLPBA. In: Proceedings of the international joint workshop on natural language processing in biomedicine and its applications; 2004. p. 70–75.

[9] Crichton G, Pyysalo S, Chiu B, Korhonen A. A neural network multi-task learning approach to biomedical named entity recognition. BMC Bioinformatics. 2017;18(1):368.

[10] Hawizy L, Jessop DM, Adams N, Murray-Rust P. ChemicalTagger: A tool for semantic text-mining in chemistry. Journal of Cheminformatics. 2011;3(1):17.

[11] Rocktäschel T, Weidlich M, Leser U. Chemspot: A hybrid system for chemical named entity recognition. Bioinformatics. 2012;28(12):1633–1640.

[12] Swain M, Cole J. ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature. Journal of Chemical Information and Modeling. 2016;56(10):1894–1904.

[13] Lowe DM, Sayle RA. LeadMine: A grammar and dictionary driven approach to entity recognition. Journal of Cheminformatics. 2015;7(1):S5.

[14] Krallinger M, Rabal O, Lourenço A, Oyarzabal J, Valencia A. Information retrieval and text mining technologies for chemistry. Chemical Reviews. 2017;117(12):7673–7761.

[15] Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. Distributional Semantics Resources for Biomedical Text Processing. In: Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, 12-13 December; 2013. p. 39–44.

[16] Habibi M, Weber L, Neves M, Wiegandt DL, Leser U. Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics. 2017;33(14):i37–i48.

[17] Corbett P, Boyle J. Chemlistem: Chemical named entity recognition using recurrent neural networks. Journal of Cheminformatics. 2018;10(1):59.

[18] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017;p. 6000–6010.

[19] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding; [Preprint] 2018.

[20] Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al.. BioBERT: a pre-trained biomedical language representation model for biomedical text mining; [Preprint] 2019.

[21] Stenetorp P, Pyysalo S, Topic G, Ohta T, Ananiadou S, Tsujii J. BRAT: a Web-based Tool for NLP-Assisted Text Annotation. Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012;p. 102–107.

[22] Corbett P, Batchelor C, Teufel S. Annotation of chemical named entities. In: BioNLP '07 Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing; 2007. p. 57–64.

[23] Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: An evaluation of a new Java implementation of MetaMap. Journal of the American Medical Informatics Association. 2017;24(4):841–844.

[24] Demner-Fushman D, Rogers WJ. CTB: A Custom Taxonomy Builder for Named Entity Extraction; 2017. Poster Presented at American Medical Informatics Association 2017 Annual Symposium.

[25] Wei CH, Allot A, Leaman R, Lu Z. PubTator central: automated concept annotation for biomedical full text articles. Nucleic Acids Research. 2019;47(W1):W587–W593.

[26] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. 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; 2016. p. 260–270.

[27] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics; 2014. p. 1532–1543.

[28] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: Generalized Autoregressive Pretraining for Language Understanding; [Preprint] 2019.

[29] Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics; 2016. p. 2383–2392. Available from: `https://www.aclweb.org/anthology/D16-1264`.

[30] Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, et al.. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation; [Preprint] 2016.

[31] Kudo T, Richardson J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing; [Preprint] 2018.

[32] Levenshtein V. Binary codes capable of correcting spurious insertions and deletions of ones. Problems of Information Transmission. 1965;1:8–17.

[33] Wang Y, Kim JD, Sætre R, Pyysalo S, Tsujii J. Investigating heterogeneous protein annotations toward cross-corpora utilization. BMC Bioinformatics. 2009;10:403.

[34] Liu X, He P, Chen W, Gao J. Multi-Task Deep Neural Networks for Natural Language Understanding; [Preprint] 2019.