BENNERD: A Neural Named Entity Linking System for COVID-19

Mohammad Golam Sohrab[†,*], **Khoa N. A. Duong**[†,*], **Makoto Miwa**[†,‡]
, **Goran Topić**[†], **Masami Ikeda**[†], and **Hiroya Takamura**[†]
[†]Artificial Intelligence Research Center (AIRC)
National Institute of Advanced Industrial Science and Technology (AIST), Japan
[‡]Toyota Technological Institute, Japan
{sohrab.mohammad, khoa.duong, goran.topic}@aist.go.jp,
{ikeda-masami, takamura.hiroya}@aist.go.jp,
makoto-miwa@toyota-ti.ac.jp

Abstract

We present a biomedical entity linking (EL) system BENNERD that detects named entities in text and links them to the unified medical language system (UMLS) knowledge base (KB) entries to facilitate the corona virus disease 2019 (COVID-19) research. BENNERD mainly covers biomedical domain, especially new entity types (e.g., coronavirus, viral proteins, immune responses) by addressing CORD-NER dataset. It includes several NLP tools to process biomedical texts including tokenization, flat and nested entity recognition, and candidate generation and ranking for EL that have been pre-trained using the CORD-NER corpus. To the best of our knowledge, this is the first attempt that addresses NER and EL on COVID-19-related entities, such as COVID-19 virus, potential vaccines, and spreading mechanism, that may benefit research on COVID-19. We release an online system to enable real-time entity annotation with linking for end users. We also release the manually annotated test set and CORD-NERD dataset for leveraging EL task. The BENNERD system is available at https://aistairc.github.io/BENNERD/.

1 Introduction

In response to the coronavirus disease 2019 (COVID-19) for global research community to apply recent advances in natural language processing (NLP), COVID-19 Open Research Dataset (CORD-19)[1] is an emerging research challenge with a resource of over 181,000 scholarly articles that are related to the infectious disease COVID-19 caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). To facilitate COVID-19 studies, since NER is considered a fundamental step in text mining system, Xuan et al. (2020b) created CORD-NER dataset with comprehensive NE annotations. The annotations are based on distant or weak supervision. The dataset includes 29,500 documents from the CORD-19 corpus. The CORD-NER dataset gives a shed on NER, but they do not address linking task which is important to address COVID-19 research. For example, in the example sentence in Figure 1, the mention SARS-CoV-2 needs to be disambiguated. Since the term SARS-CoV-2 in this sentence refers to a virus, it should be linked to an entry of a virus in the knowledge base, not to an entry of 'SARS-CoV-2 vaccination', which corresponds to therapeutic or preventive procedure to prevent a disease.

We present a BERT-based Exhaustive Neural Named Entity Recognition and Disambiguation (BENNERD) system. The system is composed of four models: **NER model** (Sohrab and Miwa, 2018) that enumerates all possible spans as potential entity mentions and classifies them into entity types, **masked language model** BERT (Devlin et al., 2019), **candidate generation model** to find a list of candidate entities in the unified medical language system (UMLS) knowledge base (KB) for entity linking (EL) and **candidate ranking model** to disambiguate the entity for concept indexing. The BENNERD system provides a web interface to facilitate the process of text annotation and its disambiguation without any training for end users. In addition, we introduce CORD-NERD (COVID-19 Open Research Dataset for Named Entity Recognition and Disambiguation) dataset an extended version of CORD-NER as for leveraging EL task.

2 System Description

The main objective of this work is to address recent pandemic of COVID-19 research. To facilitate COVID-19 studies, we introduce the BEN-

*Both authors contributed equally.

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

Proceedings of the 2020 EMNLP (Systems Demonstrations), pages 182–188
November 16-20, 2020. ©2020 Association for Computational Linguistics

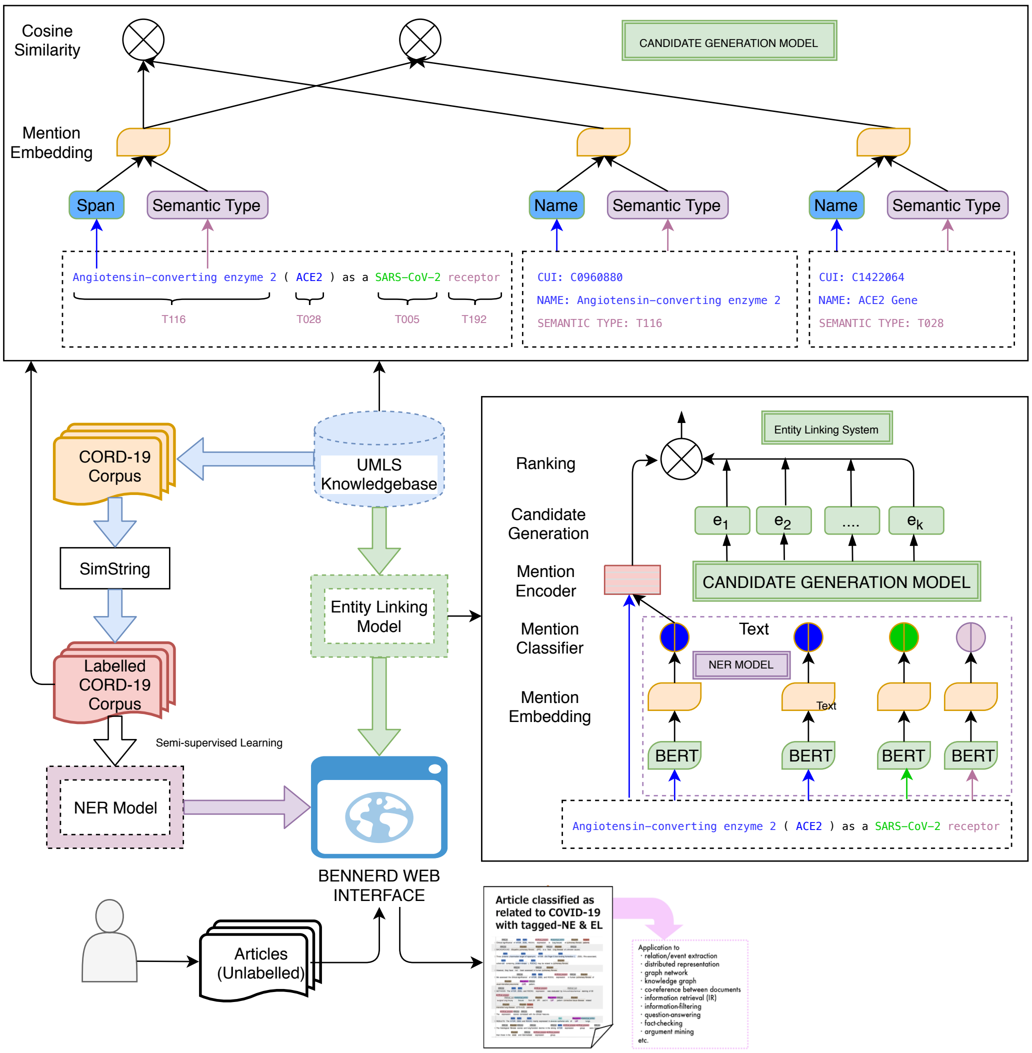


Figure 1: Workflow of BENNERD System

NERD system that finds nested named entities and links them to a UMLS knowledge base (KB). BENNERD mainly comprises two platforms: a web interface and a back-end server. The overall workflow of the BENNERD system is illustrated in Figure 1.

2.1 BENNERD Web Interface

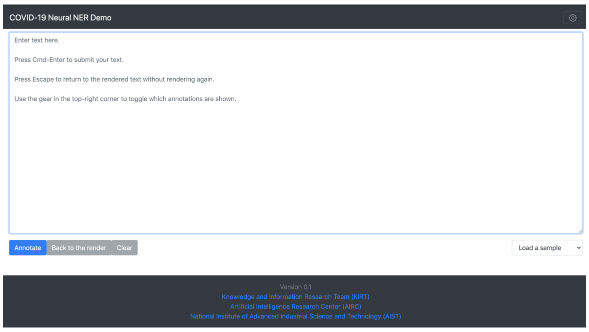
The user interface of our BENNERD system is a web application with input panel, load a sample tab, annotation tab, gear box tab, and .TXT and .ANN tabs. Figure 2 shows an users' input interface of BENNERD. For a given text from users or loading a sample text from a sample list, the annotation tab will show the annotations with the text based on best NER- and EL-based training model. Figure 3 shows an example of text annotation based on the BENNERD's NER model. Different colors represent different entity types and, when the cursor floats over a coloured box representing an entity above text, the corresponding concept unique identifier (CUI) on the UMLS is shown. Figure 3 also shows an example where entity mention SARS-CoV-2 links to its corresponding CUI. Users can save the machine readable text in txt format and annotation files in the ann format where the ann annotation file provides standoff annotation output in brat (Stenetorp et al., 2012)[2] format.

2.1.1 Data Flow of Web Interface

We provide a quick inside look of our BENNERD web interface (BWI). The data flow of BWI is presented as follows:

Server-side initialization (a) The BWI configuration, concept embeddings, and NER and EL models are loaded (b) GENIA sentence splitter and BERT basic tokenizer instances are initialized (c)

[2] https://brat.nlplab.org



COVID-19 Neural NER Demo

Enter text here.

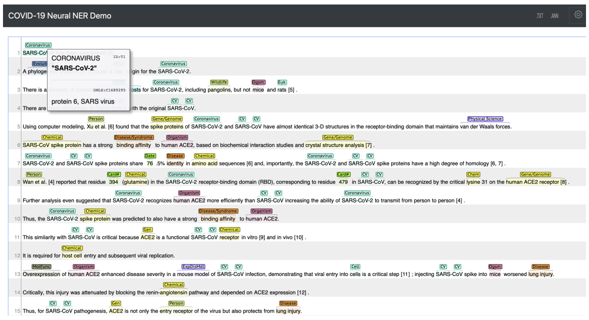
Press Cmd-Enter to submit your text.

Press Escape to return to the rendered text without rendering again.

Use the gear in the top-right corner to toggle which annotations are shown.

Version 0.1
Knowledge and Information Research Team (KIRT)
Artificial Intelligence Research Center (AIRC)
National Institute of Advanced Industrial Science and Technology (AIST)

Figure 2: BENNERD Users' Input Interface



COVID-19 Neural NER Demo

CORONAVIRUS ID:T1
"SARS-CoV-2"
UMLS:C1689295
protein 6, SARS virus

1 SARS-CoV-2 has been sequenced [3] .
2 A phylogenetic analysis [3, 4] found a bat origin for the SARS-CoV-2.
3 There is a diversity of possibl [illegible] osts for SARS-CoV-2, including pangolins, but not mice and rats [5] .
4 There are many similarities of SARS-CoV-2 with the original SARS-CoV.
5 Using computer modeling, Xu et al. [6] found that the spike proteins of SARS-CoV-2 and SARS-CoV have almost identical 3-D structures in the receptor-binding domain that maintains van der Waals forces.
6 SARS-CoV spike protein has a strong binding affinity to human ACE2, based on biochemical interaction studies and crystal structure analysis [7] .
7 SARS-CoV-2 and SARS-CoV spike proteins share 76 .5% identity in amino acid sequences [6] and, importantly, the SARS-CoV-2 and SARS-CoV spike proteins have a high degree of homology [6, 7] .
8 Wan et al. [4] reported that residue 394 (glutamine) in the SARS-CoV-2 receptor-binding domain (RBD), corresponding to residue 479 in SARS-CoV, can be recognized by the critical lysine 31 on the human ACE2 receptor [8] .
9 Further analysis even suggested that SARS-CoV-2 recognizes human ACE2 more efficiently than SARS-CoV increasing the ability of SARS-CoV-2 to transmit from person to person [4] .
10 Thus, the SARS-CoV-2 spike protein was predicted to also have a strong binding affinity to human ACE2.
11 This similarity with SARS-CoV is critical because ACE2 is a functional SARS-CoV receptor in vitro [9] and in vivo [10] .
12 It is required for host cell entry and subsequent viral replication.
13 Overexpression of human ACE2 enhanced disease severity in a mouse model of SARS-CoV infection, demonstrating that viral entry into cells is a critical step [11] ; injecting SARS-CoV spike into mice worsened lung injury.
14 Critically, this injury was attenuated by blocking the renin-angiotensin pathway and depended on ACE2 expression [12] .
15 Thus, for SARS-CoV pathogenesis, ACE2 is not only the entry receptor of the virus but also protects from lung injury.

Figure 3: Entity Annotation and Linking with BENNERD

Concept embeddings are indexed by Faiss (Johnson et al., 2019)

When a text is submitted (a) The text is split into sentences and tokens (b) Token and sentence standoffs are identified (c) NER model is run on tokenized sentences (d) EL model is run on the result (e) The identified token spans are translated into text standoffs (f) The identified concepts' names are looked up in the UMLS database (g) A brat document is created (h) The brat document is translated into JSON, and sent to the client side (i) The brat visualizer renders the document

2.2 BENNERD Back-end

The BENNERD back-end implements a pipeline of tools (e.g., NER, EL), following the data flow described in Section 2.1.1. This section provides implementation details of our back-end modules for NER and EL.

2.2.1 Neural Named Entity Recognition

We build the mention detection, a.k.a NER, based on the BERT model (Devlin et al., 2019). The layer receives subword sequences and assigns contextual representations to the subwords via BERT. We denote a sentence by $S = (x_1, ..., x_n)$, where x_i is the i-th word, and x_i consists of s_i subwords. This layer assigns a vector $\boldsymbol{v}_{i,j}$ to the j-th subword of the i-th word. Then, we generate the vector embedding $\boldsymbol{v}_i$ for each word x_i by computing the unweighted average of its subword embeddings $\boldsymbol{v}_{i,j}$. We generate mention candidates based on the same idea as the span-based model (Lee et al., 2017; Sohrab and Miwa, 2018; Sohrab et al., 2019a,b), in which all continuous word sequences are generated given a maximal size L_x (span width). The representation $\boldsymbol{x}_{b,e} \in R^{d_x}$ for the span from the b-th word to the e-th word in a sentence is calculated from the embeddings of the first word, the last word, and the weighted average of all words in the span as follows:

$$\boldsymbol{x}_{b,e} = \left[\boldsymbol{v}_b ; \sum_{i=b}^{e} \boldsymbol{\alpha}_{b,e,i} \boldsymbol{v}_i ; \boldsymbol{v}_e \right], \tag{1}$$

where $\boldsymbol{\alpha}_{b,e,i}$ denotes the attention value of the i-th word in a span from the b-th word to the e-th word, and $[;;]$ denotes concatenation.

2.2.2 Entity Linking

In our EL component, for every mention span $\boldsymbol{x}_{b,e}$ of a concept in a document, we are supposed to identify its ID in the target KB.[3] Let us call the ID a concept unique identifier (CUI). The input is all predicted mention spans $M = \{m_1, m_2, \ldots, m_n\}$, where m_i denotes the i-th mention and n denotes the total number of predicted mentions. The list of entity mentions $\{m_i\}_{i=1,...,n}$ needs to be mapped to a list of corresponding CUIs $\{c_i\}_{i=1,...,n}$. We decompose EL into two subtasks: candidate generation and candidate ranking.

Candidate Generation To find a list of candidate entities in KB to link with a given mention, we build a candidate generation layer adapting a dual-encoders model (Gillick et al., 2019). Instead of normalizing entity definition to disambiguate entities, we simply normalize the semantic types in both mention and entity sides from UMLS.

The representation of a mention m in a document by the semantic type $\boldsymbol{t}_m$, can be denoted as:

$$\boldsymbol{v}_m = [\boldsymbol{w}_m ; \boldsymbol{t}_m], \tag{2}$$

where $\boldsymbol{t}_m \in R^{d_{t_m}}$ is the mention type embedding. For the entity (concept) side with semantic type information, the representation $\boldsymbol{a}_e$, and its entity type embedding $\boldsymbol{t}_e \in R^{d_{t_e}}$ can be computed as:

$$\boldsymbol{v}_e = [\boldsymbol{a}_e ; \boldsymbol{t}_e]. \tag{3}$$

[3] We used the UMLS KB in the experiments.

We use cosine similarity to compute the similarity score between a mention m and an entity e and feed it into a linear layer (LL) to transform the score into an unbounded logit as:

$$\begin{aligned} \text{sim}\,(m, e) &= \cos\left(\boldsymbol{v}_m, \boldsymbol{v}_e\right), \quad (4) \\ \text{score}\,(m, e) &= \text{LL}(\text{sim}\,(m, e)). \quad (5) \end{aligned}$$

We employ the in-batch random negatives technique as described in the previous work (Gillick et al., 2019). For evaluating the performance of the model during training, we use the in-batch recall@1 metric (Gillick et al., 2019) on the development set to track and save the best model.

We calculate the embedding of each detected mention from the mention detection layer and each of all entities in KB and then using an approximate nearest neighbor search algorithm in Faiss (Johnson et al., 2019) to retrieve the top k entities as candidates for the ranking layer.

Candidate Ranking The cosine similarity score in the candidate generation is insufficient to disambiguate the entities in which the correct entity should be assigned the highest score which is comparable from the k candidate entities. We employed a fully-connected neural network model to aim at ranking the entity candidate list to select the best entity linked to the mention. Given a mention m and a set of candidate entities $\{e_1, e_2, ..., e_k\}$, we concatenate the embedding of m in Equation (2) with the embedding of each entity e_i in Equation (3) to form a vector $\boldsymbol{v}_{m,e_i}$. Then the vector $\boldsymbol{v}_{m,e_i}$ is fed into a LL to compute the ranking score:

$$\text{score}(m, e_i) = \text{LL}(\boldsymbol{v}_{m,e_i}). \quad (6)$$

The model is then trained using a softmax loss to maximize the score of the correct entity compared with other incorrect entities retrieved from the trained candidate generation model.

3 Experimental Settings

In this section, we evaluate our toolkit on CORD-NER and CORD-NERD datasets.

3.1 CORD-NER Dataset

We carry out our experiments on CORD-NER, a distant or weak supervision-based large-scale dataset that includes 29,500 documents, 2,533,485 sentences, and 10,388,642 mentions. In our experiment, CORD-NER covers 63 fine-grained entity types[4]. CORD-NER mainly supports four sources including 18 biomedical entity types[5], 18 general entity types[6], knowledge base entity types, and nine[7] seed-guided new entity types. We split the CORD-NER dataset into three subsets: train, development, and test, which respectively contain 20,000, 4,500, and 5,000 documents.

3.2 CORD-NERD Dataset

CORD-NER dataset comprises only NER task. To solve the EL task, we expand this dataset by leveraging a CUI for each mention in the CORD-NER dataset, we call this CORD-NERD dataset. We use the most recent UMLS version 2020AA release that includes coronavirus-related concepts. To create CORD-NERD dataset, we use a dictionary matching approach based on exact match using UMLS KB. CORD-NERD includes 10,470,248 mentions, among which 6,794,126 and 3,676,122 mentions are respectively present and absent in the UMLS. Therefore, the entity coverage ratio of CORD-NERD over the UMLS is 64.89%. We annotate the entity mentions that are not found in the UMLS with CUI_LESS. To evaluate the EL performances on CORD-NERD, 302,166 mentions are assigned for 5,000 test set, we call this UMLS-based test set. The train and development sets of CORD-NERD dataset, we simply calls UMLS-based train- and UMLS-based dev-set respectively. Besides, we assigned a biologist to annotate 1,000 random sentences based on chemical, disease, and gene types to create a manually annotated test set. This test set includes 311 disease mentions for the NER task and 946 mentions[8] with their corresponding CUIs for the EL task.

3.3 Data Prepossessing

Each text and the corresponding annotation file are processed by BERT's basic tokenizer. After tokenization, each text and its corresponding annotation file was directly passed to the deep neural approach for mention detection and classification.

[4] In the original CORD-NER paper (Xuan et al., 2020b), the authors reported 75 fine-grained entity types, but we found only 63 types.

[5] https://uofi.app.box.com/s/k8pw7d5kozzpoum2jwfaqdaey1oij93x/file/637866394186

[6] https://spacy.io/api/annotation#named-entities

[7] Coronavirus, Viral Protein, Livestock, Wildlife, Evolution, Physical Science, Substrate, Material, Immune Response

[8] Among them, 38, 311, and 597 mentions are of chemical, disease, and gene entity types respectively.

Model	**Gene**			**Chemical**			**Disease**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F** (%)
SciSpacy(BIONLP13CG)	91.48	82.06	86.51	64.66	39.81	49.28	8.11	2.75	4.11
SciSpacy(BC5CDR)	-	-	-	86.97	51.86	64.69	80.31	59.65	68.46
CORD-NER System	82.14	74.68	72.23	82.93	75.22	78.89	75.73	68.42	71.89

Table 1: Performance comparison of baseline systems on three biomedical entity types in CORD-NER corpus.

Model	**Development set**			**Test set**		
	P	**R**	**F**	**P**	**R**	**F** (%)
BENNERD + ClinicalCovid BERT (CCB)	**84.62**	86.43	**85.52**	82.83	**83.23**	**83.03**
BENNERD + SciBERT	84.03	**87.05**	85.51	82.16	**83.81**	82.98
BENNERD + Covid BERT Base	78.31	66.80	72.10	77.44	66.80	71.73

Table 2: NER Performances using different pre-trained BERT models.

Model	**Gene**			**Chemical**			**Disease**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F** (%)
BENNERD + CCB	76.07	74.83	75.45	83.55	84.60	84.07	84.85	84.99	84.92

Table 3: Performance comparison of BENNERD on three major biomedical entity types in CORD-NER corpus. CCB denotes ClinicalCovid BERT.

Model	**P**	**R**	**F** (%)
SciSpacy(BC5CDR)	36.01	**56.27**	43.91
BENNERD	**49.16**	47.27	**48.20**

Table 4: Performance comparison of BENNERD with pre-trained SciSpacy over the disease entity types on the manually annotated test set.

4 Results

4.1 NER Performances on Baseline Model

Table 1 shows the performance of SciSpacy on CORD-NER dataset. In this table, the results are based on randomly picked 1,000 manually annotated sentences as the test set.

4.2 NER Performances on BENNERD Model

Table 2 shows the performance comparison of our BENNERD with different pre-trained BERT models based on our test set. Since the manually annotated CORD-NER test set is not publicly available, we cannot directly compare our system performance. Instead, in Table 3, we show the performance of gene, chemical, and disease based on our UMLS-based test set. Besides, in Table 4, we also show the NER performances comparison of BENNERD with BC5CDR corpus-based SciSpacy model on the manually annotated disease entities.

4.3 Candidate Ranking Performance

As we are the first to perform EL task on CORD-19 dataset, we present different scenarios to evaluate our candidate ranking performance. The results of EL are depicted in Table 5. In this table, we evaluate our candidate ranking performances based on two experiment settings. In setting1, we train the CUIs based on manually annotated MedMention (Murty et al., 2018) dataset. In setting2, the BENNERD model is trained on automatically annotated CORD-NERD dataset. Table 5 also shows that our BENNERD model with setting2 is outperformed in compare to setting1 in every cases in terms of accuracy@(1, 10, 20, 30, 40, 50). Table 6 shows the EL performance on the manually annotated test set. In this table, it also shows that our system with setting2 is outperformed in compare to setting1. Besides, we also evaluate the manually annotated test set simply with string matching approach where the results of the top 10, 20, 30, 40 or 50 predictions for a gold candidate are unchanged.

4.4 Performances on COVID-19 Entity Types

Finally, in Table 7, we show the performance of nine new entity types discussed in Section 3.1 related to COVID-19 studies, which may benefit research on COVID-19 virus, spreading mechanism, and potential vaccines.

Model	UMLS-based Test set					
	A@1	A@10	A@20	A@30	A@40	A@50 (%)
BENNERD + NER's Pred. + Setting1	27.61	44.56	49.74	51.88	53.08	54.19
BENNERD + Gold NEs + Setting1	29.78	48.33	53.89	56.22	57.53	58.74
BENNERD + NER's TP + Setting1	30.31	48.91	54.60	56.95	58.27	59.49
BENNERD + NER's Pred. + Setting2	47.46	64.32	67.70	69.87	71.12	72.07
BENNERD + Gold NEs + Setting2	50.73	69.31	73.10	75.58	77.03	78.13
BENNERD + NER's TP + Setting2	53.90	73.06	76.90	79.36	80.79	81.87

Table 5: EL performance on test set. We report Accuracy@n, where $n = 1, 10, 20, 30, 40, 50$. Accuracy@1, gold candidate was ranked highest. Accuracy@$\{10, 20, 30, 40, 50\}$ indicates, gold candidate was in top 10, 20, 30, 40 or in 50 predictions of the ranker. Pred., NEs, and TP refers to predictions, named entities, and true positive respectively. Setting1 and 2 denotes model is trained on MEDMention and CORD-NERD datasets respectively.

Model	Manually Annotated Test set					
	A@1	A@10	A@20	A@30	A@40	A@50 (%)
BENNERD + Setting1	24.27	42.95	47.07	48.81	50.00	50.92
BENNERD + Setting2	31.84	50.25	54.53	56.87	58.39	60.12
BENNERD + String Matching	30.21	41.00	41.00	41.00	41.00	41.00

Table 6: EL performance on our manually annotated test set.

Model	UMLS-based Test set		
	P	R	F (%)
Coronovirus	98.46	98.94	98.70
Viral Protein	89.39	91.09	90.23
Livestock	96.67	97.26	96.96
Wildlife	98.43	97.56	97.99
Evolution	97.16	98.46	97.80
Physical Science	96.80	93.08	94.90
Substrate	95.99	98.46	97.21
Material	94.80	90.46	92.58
Immune Response	97.29	99.42	98.35

Table 7: Performances on nine types of COVID-19

5 Related Work

To facilitate the biomedical text mining research on COVID-19, recently a few works have reported to address text mining tasks. Xuan et al. (2020b) created CORD-NER dataset with distant or weak supervision and reported first NER performances on different NER models. Motivated by this work, we presented a first web-based toolkit that addresses both NER and EL. In addition, we also extend the CORD-NER dataset to solve EL task.

Xuan et al. (2020a) created EvidenceMiner system that retrieves sentence-level textual evidence from CORD-NER dataset. Tonia et al. (2020) developed an NLP pipeline to extract drug and vaccine information about SARS-CoV-2 and other viruses to help biomedical experts to easily track the latest scientific publications. To the best of our knowledge, this work is our first effort to solve both NER and EL models in a pipeline manner.

6 Conclusion

We presented the BENNERD system for entity linking, hoping that we can bring insights for the COVID-19 studies on making scientific discoveries. To the best of our knowledge, BENNERD represents the first web-based workflow of NER and EL for NLP research that addresses CORD-19 dataset that leads to create CORD-NERD dataset to facilitate COVID-19 work. The online system is available for meeting real-time extraction for end users. The BENNERD system is continually evolving; we will continue to improve the system as well as to implement new functions such as relation extraction to further facilitate COVID-19 research. We refer to visit https://aistairc.github.io/BENNERD/ to know more about BENNERD and CORD-NERD.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. This work is based on results obtained from a project commissioned by the Public/Private R&D Investment Strategic Expansion PrograM (PRISM).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

J. Johnson, M. Douze, and H. Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 97–109.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Mohammad Golam Sohrab, Minh Thang Pham, Makoto Miwa, and Hiroya Takamura. 2019a. A neural pipeline approach for the pharmaconer shared task using contextual exhaustive models. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 47–55.

Mohammad Golam Sohrab, Pham Minh Thang, and Makoto Miwa. 2019b. A generic neural exhaustive approach for entity recognition and sensitive span detect. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, pages 735–743, Span. IberLEF 2019.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. 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*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Korves Tonia, Peterson Matthew, Garay Christopher, Read Tom, Chang Wenling, Bartlett Marta, Quattrochi Lauren, and Hirschman Lynette. 2020. Nlp for extracting covid-19 drug and vaccine information from scientific literature. In *Proceedings of the 28th Conference on Intelligent Systems for Moleculer Biology*.

Wang Xuan, Liu Weili, Chauhan Aabhas, Guan Yingjun, and Han Jiawei. 2020a. Automatic textual evidence mining in COVID-19 literature. *2020 Intelligent Systems for Molecular Biology (ISMB'20)*.

Wang Xuan, Song Xiangchen, Li Bangzheng, Guan Yingjun, and Han Jiawei. 2020b. Comprehensive named entity recognition on CORD-19 with distant or weak supervision. *2020 Intelligent Systems for Molecular Biology (ISMB'20)*.