Learning Entity-Likeness with Multiple Approximate Matches for Biomedical NER

An Nguyen Le, Hajime Morita and **Tomoya Iwakura**
Artificial Intelligence Laboratory, Fujitsu Research, Fujitsu Ltd.
4 Chome-1-1 Kamikodanaka, Nakahara Ward, Kawasaki, Kanagawa, Japan
{nguyenle.an,hmorita,iwakura.tomoya}@fujitsu.com

Abstract

Biomedical Named Entities are complex, so approximate matching has been used to improve entity coverage. However, the usual approximate matching approach fetches only one matching result, which is often noisy. In this work, we propose a method for biomedical NER that fetches multiple approximate matches for a given phrase to leverage their variations to estimate entity-likeness. The model uses pooling to discard the unnecessary information from the noisy matching results, and learn the entity-likeness of the phrase with multiple approximate matches. Experimental results on three benchmark datasets from the biomedical domain, BC2GM, NCBI-disease, and BC4CHEMD, demonstrate the effectiveness. Our model improves the average F-measures by up to 0.21 percentage points compared to a BioBERT-based NER.

1 Introduction

In the biomedical field, obtaining labelled data is very costly. Biomedical Named Entities (NEs) are complex and new NEs are continuously increasing in significant numbers, leading to unknown-word issues in Biomedical Named Entity Recognition (BioNER) tasks. One reason why biomedical NEs are complex is that they have many variations with the interchangeability of Roman numbers and Latin characters, spaces and hyphens, etc. The number of new biomedical research papers is increasing, wherein approximately two papers per minute, resulting in more than 1 million papers each year, are added to the PubMed database (Landhuis, 2016). With this number of publications, new NEs are constantly being reported.

In the last few years, NER using pre-trained language models (LMs), such as BERT (Devlin et al., 2018), ELMo (Peters et al., 2018), and Flair (Akbik et al., 2019), has shown state-of-the-art performance. In the biomedical domain, pre-trained LMs such as BioBERT (Lee et al., 2019a) and BioELMo (Jin et al., 2019), which are BERT and ELMo trained on a biomedical domain text, have achieved the state-of-the-art performance in many biomedical natural language processing tasks including NER. However, only using previously trained LMs cannot cover the continuously increasing new entities due to complex characteristics of biomedical NEs, lead to unknown words problem. Despite being used as approaches to avoid unknown words problem, subword segmentation (Sennrich et al., 2015; Kudo and Richardson, 2018) methods consider subwords represented as unique IDs, but not words or their synonyms. therefore, it is difficult for subword or character based LMs to cover biomedical NEs , which are complex and contain various of expression described in section 3. Moreover, LM pre-training is costly, time-consuming, and computationally expensive. Training BioBERT on biomedical corpora based on the BERT model requires 10 to 23 days on eight NVIDIA V100 GPUs (Lee et al., 2019a).

To deal with the complex and continuously increasing entities, the use of dictionary-based approaches can be an effective approach in previous works (Collobert et al., 2011; Rijhwani et al., 2020). In contrast to pre-training models, we can cover new NEs by adding entries to the dictionary, without needing time-consuming pre-training. There are two types of dictionary application methods: exact matching and approximate matching. Exact matching has been incorporated into neural NER (Collobert et al., 2011; Chiu and Nichols, 2016; Wu et al., 2018) and non-neural NER methods (Uchimoto et al., 2000) to improve accuracy.

Exact matching cannot totally cover all of the complex and newly-created NEs. In the biomedical domain, new NEs are created by modifying the endings of the existing one. For example, the new gene TAAR7P was named by modifying the

Proceedings of Recent Advances in Natural Language Processing, pages 1040–1049
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_117

ending of the existing gene TAAR8. To improve the coverage of entities, approximate matching has been used to manage new NEs in non-neural NER (Cohen and Sarawagi, 2004). However, the approximate matching approach fetches only one matching result, which cannot cover all variations of NEs. For example, NEs "Type-1 angiotensin II receptor-associated protein" have many variations such as "Type-1 angiotensin II receptor associated protein", "Type-1 angiotensin 2 receptor associated protein", and "Type 1 angiotensin II receptor-associated protein". Also, approximate matching results are often noisy.

In this paper, we propose a method to improve neural BioNER **by learning the entity-likeness of a given input sentence using multiple approximate matches of the input sentence with a dictionary**. We define the entity-likeness as the degree to which a certain input sentence is likely to appear in the dictionary. It is estimated from matching results between the input sentence and entities in the dictionary.

We evaluated our method with three biomedical domain benchmarks, i.e., BC2GM, NCBI-disease, and BC4CHEMD dataset. The experimental results show the effectiveness of our approach. It improves F-measures by up to +0.21 points on the biomedical benchmark, and +2.2 points when probing the biomedical ELMo (Jin et al., 2019), which is a recent state-of-the-art pre-training method.

2 Related Work

For the NER task, previous studies have examined the application of dictionaries in machine learning. Dictionary matching was employed in SVM-based NER (Ratinov and Roth, 2009) and partial matching computed by distance feature between a token and entity in dictionary was considered in semi-Markov extraction processes (Cohen and Sarawagi, 2004).

Dictionary matching is also used in Neural NER approaches. Liu et al. added a pre-trained module that softly matches the gazetteers to the semi-Markov CRF-based segmental NER task. Soft matching of gazetteers is also used in the work of Rijhwani et al. (2020) for low-resource NER. Exact matching was used by Collobert et al. (2011) ; they use a network layer to map words of dictionary into feature vectors by a lookup table operation and train the features as input in their model. Chiu and Nichols proposed the use of the longest matching, including partial lexicon matching in neural networks. Each word vector has dimensions to express dictionary matching.

In the CRF-based sequence labeling model for NER, the clustering results of phrases in the search engine query logs were used as features by Lin and Wu (2009). To improve word representation, a word embedding learning method that leverages information from relevant lexicons to phrase embedding was proposed by Passos et al. (2014). Handcrafting features obtained from gazetteers were also incorporated to model additional information in the named entity (Wu et al., 2018; Shang et al., 2018).

Related to approaches employing approximate string matching in Biomedical NER, Tsuruoka and Tsujii proposed a method to recognize entity candidates by approximate searching and filtering out false positives using a binary classifier. Yang et al. used approximate string matching and added pre- and post-keywords for each bio-entity name to expand the coverage of the dictionary. Xu et al. constructed a dictionary attention layer to incorporate exact dictionary matching and a document-level attention mechanism to improve disease NER.

Approaches based on neural network were also applied for Biomedical NER (Habibi et al., 2017; Crichton et al., 2017; Wang et al., 2018). For a transformer-based approach, Khan et al. used a shared transformer encoder to capture the embedding vector of each token in input sentence and task specific linear layers to generate representations of multi-tasks including Biomedical NER.

Differing from these works, we propose a method to learn the entity-likeness of a sentence by leveraging multiple approximate matches of the sentence with one or multiple dictionaries. Recent approaches based on pre-training for specific domains, such as biomedical (Lee et al., 2019a; Jin et al., 2019), clinical (Huang et al., 2019) and scientific (Beltagy et al., 2019), have shown high levels of accuracy; our method is complementary to these approaches.

3 NEs in Biomedical Domain

Biomedical NEs are complex and ambiguous due to the following characteristics:

Variation of Expression Biomedical NEs have various synonyms, including abbreviations, interchangeability of Roman numbers and Latin characters, insertions and deletions of hyphens and spaces, and changes in word order. For example, the gene

"Angiotensin II Receptor Type 1" has the official name "AGTR1", as well as more than ten other names, e.g., AGTR -1, Type -1 Angiotensin II Receptor, Angiotensin Receptor 1B, and AT1 Receptor. Even if the dictionary is further expanded, exact matching cannot entirely cover all possible variations of NEs.

Composite Mentions NEs in the biomedical domain are frequently connected by "and," "or" in a single span which refers to more than one entity. For example, "alpha and beta globin" refers to "alpha globin" and "beta globin".

Nested NEs Nested NEs (Kim et al., 2003; Ringland et al., 2019), where one NE is completely contained by the other, are also commonly used in biomedical data. For example, both "adenylate cyclase activating polypeptide 1" and "adenylate cyclase" are the names of proteins.

Entity ambiguity The same mention may often refer to many different entities depending on context. For example, "VHL" can be either a disease name "Von Hippel–Lindau (VHL) disease" or a gene name "VHL gene" depending on context.

NEs in the biomedical domain are continuously increasing in number every year. When using exact matching or pre-trained LMs for BioNER, it is difficult to sufficiently cover all possible combinations of NEs, leading to the omission of NE recognition.

4 Learning Entity-likeness with Multiple Approximate Matches

The concept of our approach is that the **entity-likeness of a given input sentence** can be estimated by its maximal similarity to entities in a dictionary. Our motivation is to **assign the entity-likeness to each word** of the input sentence.

The overall flow of the proposed approach is as follows:

1. Given an input sentence, we first fetch matching results between the input sentence and a specified dictionary.
2. We create matching patterns based on the matching results, and assign them to each word in the input sentence. The matching pattern is a label that indicates how each word matches with the dictionary.
3. For each word in the input sentence, we build a vector for predicting entity-likeness from the multiple matching patterns by a pooling operation.
4. We build an NER model learning both vector of entity-likeness and contextual embedding derived from pre-trained LMs.

4.1 Creating Multiple Approximate Matches

Given an input sentence, we first fetch the matching results between the input sentence and entities in a dictionary. Since we cannot specify which part of the input sentence contains the entity, we **calculate the string similarity of all continuous word level N-grams ($N \leq 5$)** in the input sentence with all dictionary entries. The matching returns entries whose similarity with the N-gram is larger than a specified threshold [1]. We regard a match of N-gram with an entity with threshold 1.0 as an exact matching.

By employing the multiple approximate matchings of N-gram with the dictionary, it is possible to obtain useful information about the multiple matches for estimating the entity-likeness of the N-grams, especially in the case of predicting a new NE which is similar to the existing one. For example, we can obtain information on the interchangeability of Greek or Roman characters in NEs from dictionary entries "beta-1 Adrenergic Receptor", "β-1 Adrenergic Receptor" and other synonyms. The information is useful for recognizing the unknown NE "α-1 Adrenergic Receptor".

4.2 Creating Dictionary Matching Patterns

Based on the matching results of N-grams ($N \leq 5$) with a dictionary obtained in section 4.1, we create a set of dictionary matching patterns that includes the information of **the dictionary that is used, the types of matching, and the matching position**; this information is assigned to each word in the input sentence. The type of matching is set to "Exact" if the N-gram exactly matches the dictionary entry, otherwise it is set to "Approximate". There are three types of matching positions (B (Beginning), I (Inside), and E (Ending)) which indicate the position of the word in the N-gram.

For example, as shown in Figure 1, the input sentence "*EGFR is epidermal growth factor receptor*" is matched with a gene/protein dictionary. The gene/protein dictionary includes entries such as "*epidermal growth factor receptor substrate*

[1] Note that an N-gram can be matched with one or multiple dictionaries when we have two or more dictionaries.

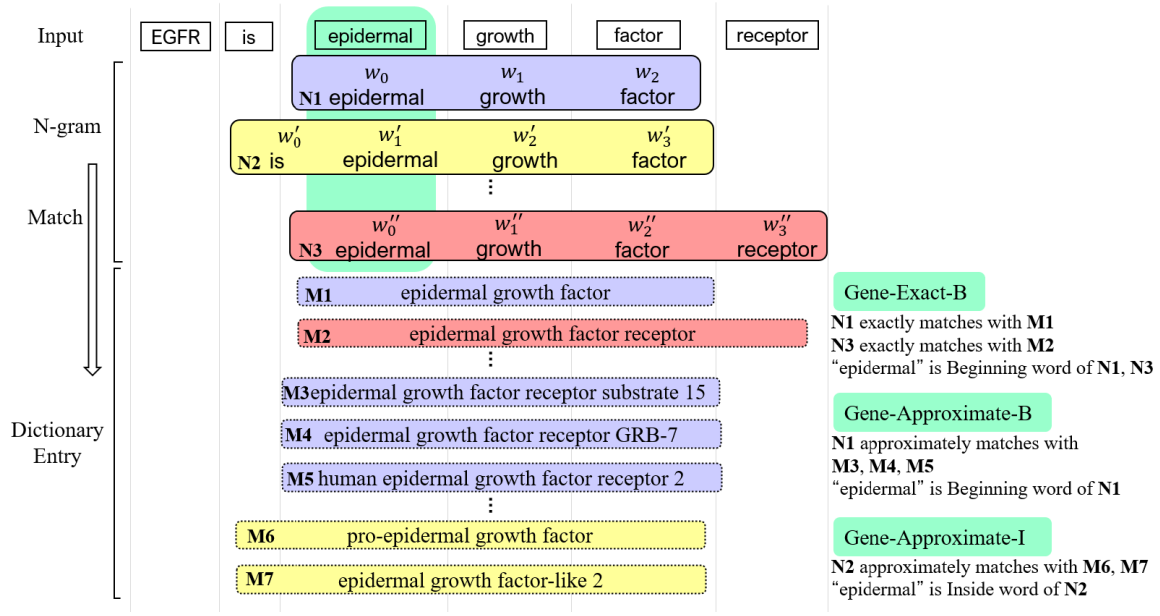

Figure 1: Method to create matching patterns using a gene/protein dictionary. The blue markers represent N-grams of the input sentence, and the purple, yellow, and red markers represent N-grams matching with the corresponding dictionary entries. The green marker describes the current word and corresponding matching patterns that are created and assigned to the current word.

15," "*epidermal growth factor receptor GRB-7*," etc. As shown in Figure 1, 3-gram **N1** with the beginning word w_0 "epidermal" exactly matches with gene/protein dictionary entry **M1** and it approximately matches with entries **M3**, **M4** and **M5**. The matching result of the 3-gram **N1** assigns matching patterns: "Gene-Exact-B" and "Gene-Approximate-B" to w_0.

In the same way, 4-gram **N2** approximately matches with dictionary entries **M6** and **M7**. In this case, the word "epidermal" is inside the **N2** and therefore the matching result of the **N2** assigns matching patterns:"Gene-Approximate-I' to w_0.

Based on matching results between all N-grams of the input sentence and the dictionary, we can obtain a set of matching patterns for each word in the input sentence. The possible matching patterns for each word are {Number of dictionaries} $\times$ {Exact, Approximate} $\times$ {B, I, E}. For example, in Figure 1, a set of matching patterns with the Gene dictionary for the third word "epidermal" are {"Gene-Exact-B," "Gene-Approximate-B," "Gene-Approximate-I"}.

4.3 Representation of Multiple Matching Patterns

After creating sets of dictionary matching patterns corresponding to each word, we build a representation for the dictionary matching patterns.

Suppose each word w_i corresponds to a subset of matching patterns $S_i \subset \mathbf{S}$, where $\mathbf{S}$ is the possible matching patterns, S_i is obtained in section 4.2. Here, S_i represents the likeliness of that the word forms a part of entities. E_i corresponds to embeddings of S_i:

$$E_i = \{emb(s)|s \in S_i\} \quad (1)$$

where $emb(\cdot)$ indicates an embedding operation. In experiments, embedding $emb(s)$ is randomly initialized from a normal distribution but not fine-tuned.

Next, we build a vector representation D_i of entity-likeness by pooling the embeddings E_i; D_i has the same dimension as E_i:

$$D_i = f_{pool}(E_i) \quad (2)$$

where f_{pool} is a pooling operation.

The aim of the pooling is to aggregate information for learning from various matching patterns. In order to investigate the effect of various pooling functions, we consider four types of pooling: Sum, Max, Average and Convolution.

Sum Pooling It is expected that summarizing all features of the possible matching pattern embeddings gives information for estimating entity-likeness of words.

$$f_{sum}(E_i) = \sum_{v \in E_i} v \quad (3)$$

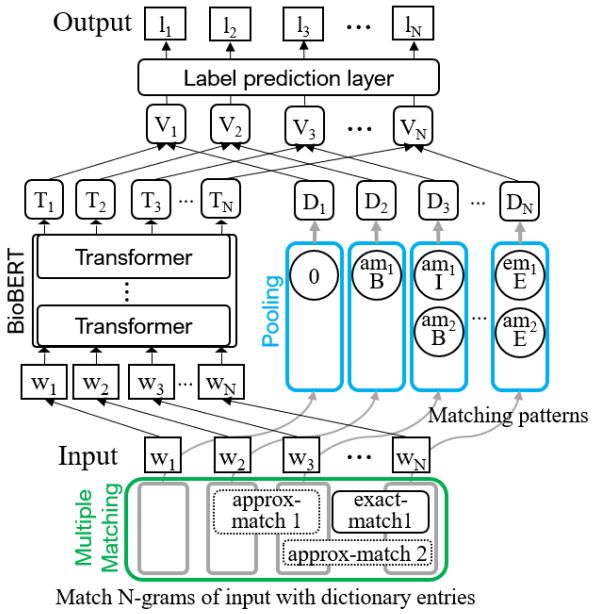

Figure 2: Illustration of the proposed model architecture. T_i and D_i are the corresponding contextual word embedding module and dictionary matching pattern module for each word w_i in the input sentence, respectively. V_i represents interaction between each word and its entity-likeness . The model predicts the token-level NE label, l_i. am_1B, am_1I,... are embeddings of matching patterns.

Max Pooling Instead of sum pooling, we use max pooling to compose the set of matching pattern embeddings:

$$f_{max}(E_i) = \max(E_i) \tag{4}$$

Average Pooling In the same way, we consider the average variation of the pooling method:

$$f_{avg}(E_i) = \text{avg}(E_i) \tag{5}$$

Convolution As a way to combine embeddings, we apply 1-D convolution over the set of matching pattern embeddings to build the dictionary matching embedding:

$$f_{conv}(E_i) = \text{Conv1d}(E_i) \tag{6}$$

4.4 Learning Representations of Entity-likeness with NER

Figure 2 shows the overview of our method. Given the output of the contextual word embedding T_i, and vector representation of entity-likeness D_i, the label prediction module predicts the IOB2 labels of input sentence w_i. By learning T_i and D_i together, it is possible to recognize new NEs which were not in the dictionary or training data of LMs. For the pre-trained LMs, we use BioBERT (Lee et al., 2019a) or BioELMo (Jin et al., 2019) depending on experiments.

The layer numbers and the internal details of the label prediction layer vary depending on the used pre-trained LMs. We follow the settings of the original studies (Lee et al., 2019a; Jin et al., 2019). In the case of BioBERT, we use a single linear layer to compute token level IOB2 probabilities. In the case of BioELMo, we follow the probing settings in the work of Jin et al. (2019). We use several linear layers to compute the probabilities.

5 Experiments

In this section, we conduct three experiments. Experiment 1 confirms the effectiveness of learning both entity-likeness and contextual embedding for BioNER. Also, we want to confirm if applying appropriate pooling operations can reduce noise in the case of approximate matching. Experiment 2 confirms portability by using our method with different pre-trained LMs. Experiment 3 confirms the effectiveness of our method not only in the biomedical domain but also in the general domain. For pre-trained LMs, we employed BioBERT and BioELMo trained on PubMed and PMC biomedical articles. For experiments on a general domain dataset, we applied the pre-trained BERT base cased LMs.

5.1 Datasets

In this study, the results were obtained by adopting the proposed and BioBERT-based methods to three benchmark biomedical datasets, BC2GM, NCBI-disease, and BC4CHEMD, which are exclusively annotated with protein, disease, and chemical entities [2], respectively. For the general domain, we used the CoNLL 2003 dataset (Tjong Kim Sang and De Meulder, 2003). Table 1 shows the size of the datasets. All datasets are publicly available.

5.2 Dictionary

We consider the dictionary as a set of names including synonyms of the entities, e.g., Gene, Disease, and Drug. In the biomedical domain, there are several publicly available databases that can be used to create dictionaries. The dictionaries are built from

[2] https://github.com/cambridgeltl/MTL-Bioinformatics-2016

Dataset	train	dev	test
BC2GM	12,574	2,519	5,038
NCBI-disease	5,424	923	940
BC4CHEMD	30,682	30,639	26,364
CoNLL 2003	14,987	3,466	3,684

Table 1: Size of NER datasets used in the experiments. The numbers are in sentences.

the databases. Therefore, we do not need to create and maintain dictionaries from scratch.

We construct dictionaries for genes/proteins, diseases, and drugs, to train the proposed model on the BC2GM, NCBI-disease, and BC4CHEMD datasets, respectively. Further, three dictionaries of person (PER), location (LOC), and organization (ORG) are built to train the proposed model on the CoNLL 2003 dataset.

Gene/protein dictionary We created a gene/protein dictionary from public databases: Human Gene Nomenclature (HGNC) and NCBI Entrez Gene (Maglott et al., 2019). HGNC is a database containing unique names and alias names for human genes. NCBI Entrez Gene is the National Center for Biotechnology Information (NCBI)'s database for gene-specific information (Maglott et al., 2011). We extracted gene names, their symbols, alias symbols, and alias names to build our gene/protein dictionary. The dictionary contains 292,853 gene entity surfaces.

Disease dictionary We built a disease dictionary based on Human Disease Ontology (LM et al., 2019). Our disease dictionary is built from disease names and their synonyms based on the ontology with 30,426 disease entities.

Drug dictionary For the drug dictionary, we used DrugBank Vocabulary [3] from DrugBank (DS et al., 2019). We entered common names and synonyms as drug names into the dictionary. The dictionary contains 26,235 drug entities.

PER, LOC, and ORG dictionaries We constructed three dictionaries on person (PER), location (LOC), and organization (ORG) from the DBpedia database [4] to train the proposed model on the CoNLL 2003 dataset. We used categories from the 2019-8-30 Version and extracted categories that include keywords such as "Person," "Organization," and "Places" to construct the dictionaries. The dictionary consists of 710,492 PER, 37,687 ORG, and 69,028 LOC entities.

5.3 Experimental Setting

To obtain multiple approximate matches of the input sentence and dictionary, we used Simstring (Okazaki and Tsujii, 2010), an approximate string matching library that searches for similarities between a set of characters (e.g., "cosine," "jaccard") with a query string length exceeding a specified threshold. Simstring is known as a fast and efficient algorithm for approximate dictionary matching.

We used Simstring to obtain matching results for N-gram ($N \leq 5$) with the dictionary. The cosine similarity threshold between N-grams of the input sentence and dictionary entries was empirically set to 0.8. This is because the threshold value of 0.8 revealed good results during preliminary experiments. Next, we created a set of matching patterns based on the matching results.

For hyperparameter tuning, entity-likeness representation dimension sizes of 50, 100, and 300, and batch sizes of 16 and 32, were selected. Therein, we decided the parameter for entity-likeness representation and batch size are 100 and 32, respectively. Contextual word embedding derived from the pre-trained model is concatenated with 100-dimensional entity-likeness representation embeddings, and then fed into a label prediction layer. We applied four types of pooling: Sum, Max, Average, and Convolution. We trained for 20 epochs and the NER results were averaged over five seeds.

All experiments were conducted using a single NVIDIA GeForce RTX 16 GB GPU. Pytorch version was 1.4.0. We used the HuggingFace PyTorch implementation of (Wolf et al., 2019) [5] to conduct the experiments.

Experiment 1: Learning Entity-likeness with BioBERT We followed the recipe of Lee et al. (2019a) to train the model with the following hyperparameters: learning rates of 1e-5; batch sizes of 32; and weight-decay of 0.001. We used the pre-trained model BioBERT v1.0 (Wiki + Books + PubMed 200K + PMC 270K) [6] as a contextual word embedding.

[3] https://www.drugbank.ca/releases/5-1-4/downloads/all-drugbank-vocabulary

[4] https://downloads.dbpedia.org/repo/lts/generic/

[5] https://github.com/huggingface/transformers

[6] https://github.com/naver/biobert-pretrained

For the approach using the approximate matching result, we compared our method with Liu et al. (2019). They proposed a pre-training sub-tagger *softdict* that softly matches a sentence with gazetteers for NER. This sub-tagger plays the role of an approximate dictionary look-up. Softdict is trained on gazetteers and non-entity N-grams sampled from the corpus.

They sampled 1 million non-entity N-grams from 14,987 sentences in the CoNLL 2003 training data. For each dataset, we sampled non-entity N-grams using the same ratio of data size and sample size. Following the settings in their work, we used pre-trained 50-dimensional Glove word embedding (Pennington et al., 2014), contextualized ELMo embedding, a convolutional character encoder and the pre-trained softdict to train the NER model.

Experiment 2: Learning Entity-likeness with BioELMo and Bio_word2vec We confirmed the performance of the proposed method with other pre-trained LMs. We conducted experiments using contextual embeddings from pre-trained models BioELMo (Jin et al., 2019) [7] and Bio_word2vec (Pyysalo et al., 2013) [8]. We kept the default hyper-parameters settings in Jin et al.'s work, with a batch size of 32, Adam learning rate of 0.002, and training for 10 epochs. The embedding derived from BioELMo or Bio_word2vec is concatenated with 100-dimensional entity-likeness representation embeddings and then are fed to four feed-forward layers and a CRF output layer.

Experiment 3: Learning Entity-likeness with BERT For experiments on the CoNLL 2003 dataset, a pre-trained BERT-base-cased model was used instead of BioBERT. Hyperparameters were set the same as for learning entity-likeness with Experiment 1.

5.4 Results

For learning entity-likeness with BioBERT, we evaluated the accuracy of the results with an entity-level F-measures. For learning entity-likeness with BioELMo and Bio_word2vec, we used the official evaluation codes of BC2GM, which contain multiple ground-truth tags to calculate F-measures, following the work of Jin et al. (2019).

The experimental results are presented in Tables 2, 3 and 4. In Table 2, the F-measures were obtained in the experiments conducted based on the Pytorch implementation library of (Wolf et al., 2019); the best scores are denoted in bold. The scores are almost the same with scores reported in (Lee et al., 2019b), which are not the scores reported in the original BioBERT papers (Lee et al., 2019a).

The difference in scores of the original paper (Lee et al., 2019a) and (Lee et al., 2019b) is due to the neural network implementation library (Pytorch-based or TensorFlow-based), the implementation framework (HuggingFace, etc.), and the GPU architecture and setting of the random seed.

Model	P	R	F
BC2GM			
BioBERT	82.34 ±0.02	84.82 ±0.02	83.56 ±0.02
Liu et al.	79.63 ±0.002	81.09 ±0.009	80.35 ±0.004
Exa-Sum	82.58 ±0.05	84.65 ±0.05	83.60 ±0.02
Exa-Max	82.54 ±0.01	84.61 ±0.02	83.56 ±0.00
Exa-Avg	82.52 ±0.01	84.61 ±0.02	83.55 ±0.00
Exa-Conv	82.56 ±0.03	84.61 ±0.06	83.57 ±0.04
App-Sum	**82.69** ±0.01	**84.71** ±0.01	**83.69** ±0.02
App-Max	82.57 ±0.01	84.66 ±0.03	83.65 ±0.03
App-Avg	82.51 ±0.04	84.60 ±0.02	83.58 ±0.00
App-Conv	82.54 ±0.04	84.66 ±0.01	83.58 ±0.02
NCBI-disease			
BioBERT	86.67 ±0.06	90.28 ±0.02	88.44 ±0.03
Liu et al.	85.21 ±0.006	87.01 ±0.005	86.10 ±0.003
Exa-Sum	86.40 ±0.02	90.37 ±0.02	88.34 ±0.03
Exa-Max	86.67 ±0.06	90.30 ±0.06	88.44 ±0.02
Exa-Avg	86.68 ±0.04	90.38 ±0.05	88.49 ±0.10
Exa-Con	86.57 ±0.05	90.26 ±0.07	88.38 ±0.06
App-Sum	86.74 ±0.06	**90.64** ±0.06	**88.65** ±0.05
App-Max	86.39 ±0.02	90.58 ±0.02	88.43 ±0.03
App-Avg	**86.73** ±0.04	90.51 ±0.05	88.58 ±0.01
App-Con	86.46 ±0.06	90.51 ±0.12	88.49 ±0.08
BC4CHEMD			
BioBERT	91.89 ±0.06	90.95 ±0.04	91.41 ±0.02
Liu et al.	88.78 ±0.06	89.02 ±0.02	88.89 ±0.03
Exa-Sum	91.79 ±0.10	91.08 ±0.05	91.43 ±0.02
Exa-Max	91.92 ±0.06	90.93 ±0.10	91.43 ±0.05
Exa-Avg	91.90 ±0.08	91.00 ±0.10	91.44 ±0.00
Exa-Con	91.86 ±0.06	91.04 ±0.03	91.45 ±0.01
App-Sum	91.81 ±0.10	**91.11** ±0.05	91.45 ±0.02
App-Max	**91.94** ±0.06	91.01 ±0.08	**91.47** ±0.01
App-Avg	91.88 ±0.10	91.06 ±0.10	**91.47** ±0.00
App-Con	91.85 ±0.10	91.03 ±0.08	91.44 ±0.00

Table 2: Experimental results of the proposed method with BioBERT-base model on three biomedical datasets BC2GM, NCBI-disease, and BC4CHEMD. Cells represent Precision, Recall and F-measure with standard deviation on each test set, respectively. Exa and App denote Exact and Approximate, respectively.

[7] https://github.com/Andy-jqa/bioelmo
[8] http://bio.nlplab.org

Model	P	R	F
BioELMo			
BioELMo	-	-	88.4
Exa-Sum	89.6 ±0.61	**90.4** ±0.56	90.0 ±0.06
Exa-Max	90.4 ±0.18	89.9 ±0.33	90.1 ±0.24
Exa-Avg	90.6 ±0.38	89.7 ±0.53	90.2 ±0.35
Exa-Con	89.5 ±0.72	89.9 ±0.78	89.7 ±0.02
App-Sum	**91.1** ±0.68	90.0 ±0.67	**90.6** ±0.25
App-Max	90.0 ±0.34	**90.4** ±0.50	90.2 ±0.29
App-Avg	89.3 ±0.55	90.3 ±0.18	89.8 ±0.19
App-Con	89.1 ±0.10	90.0 ±0.30	89.5 ±0.10
Bio_word2vec			
Bio_w2v	-	-	78.5
Exa-Sum	**86.3** ±0.30	80.3 ±0.24	83.2 ±0.16
Exa-Max	86.2 ±0.28	79.9 ±0.11	82.9 ±0.10
Exa-Avg	86.2 ±0.06	80.2 ±0.30	83.1 ±0.07
Exa-Con	84.9 ±0.36	80.7 ±0.38	82.7 ±0.06
App-Sum	85.7 ±0.10	**81.3** ±0.33	**83.4** ±0.09
App-Max	85.4 ±0.51	80.7 ±0.13	83.0 ±0.11
App-Avg	85.9 ±0.28	80.9 ±0.51	83.3 ±0.13
App-Con	85.2 ±0.29	81.2 ±0.07	83.2 ±0.10

Table 3: Results of learning entity-likeness by probing BioELMo and Bio_word2vec on the BC2GM dataset. Cells represent Precision, Recall and F-measure with standard deviation. Exa and App denote Exact and Approximate, respectively.

As listed in Tables 2, 3 and 4, learning both exact matching and approximate matching outperforms BioBERT-based methods and improves F-measures by up to +0.13, +0.21 and +0.06 points on the three biomedical benchmarks BC2GM, NCBI-disease and BC4CHEMD, respectively; BioELMo and Bio_word2vec improve F-measures by up to +2.2 and +4.9 points on BC2GM; BERT-based methods improve F-measures by up to +0.25 points on CoNLL 2003.

6 Discussion

The experimental results indicate that, in the case of exact matching, F-measures are not highly different for the four types of pooling. As shown in Table 2, 3 and 4, sum pooling obtains the best results in the case of approximate matching. It is considered to be more informative for summarizing all features of the possible approximate matching patterns to estimate entity-likeness. Precision is improved in exact matching while recall is improved in approximate matching. In approximate matching, even though the matching results are noisy, tuning to select the appropriate pooling can help minimize noise. Our approach has effectiveness for small datasets such as NCBI-disease, and multi-category datasets such as CoNLL 2003, where F-measures improved by up to +0.21 and +0.25 points, respectively.

Model	P	R	F
CoNLL 2003			
BERT	90.73 ±0.06	92.00 ±0.05	91.36 ±0.03
Exa-Sum	90.96 ±0.02	92.16 ±0.02	91.56 ±0.01
Exa-Max	90.90 ±0.06	92.10 ±0.06	91.50 ±0.00
Exa-Avg	90.89 ±0.04	92.17 ±0.05	91.52 ±0.03
Exa-Con	90.87 ±0.05	92.09 ±0.07	91.48 ±0.03
App-Sum	**91.01** ±0.02	**92.23** ±0.02	**91.61** ±0.01
App-Max	90.91 ±0.06	92.12 ±0.06	91.51 ±0.00
App-Avg	90.91 ±0.04	92.17 ±0.05	91.53 ±0.03
App-Con	90.86 ±0.06	92.11 ±0.12	91.48 ±0.03

Table 4: Experimental results of the proposed method with BERT on CoNLL 2003.

In Table 2, the improvement of F-measures is not significant on the BC4CHEMD dataset. It is thought that this is because approximate matching of N-gram ($N \leq 5$) returns only dictionary entries which approximately match with N-gram only up to 5-words, while there are drug names whose length are much longer than 5-gram in BC4CHEMD dataset. For datasets containing long NEs, it is necessary to set N-grams with larger values.

Our approach has effectiveness for small datasets with complicated NEs. In reality, obtaining large-scale domain specific data like BC2GM and BC4CHEMD is very costly, while NEs in the biomedical domain are complex and continuously increasing every year.

7 Conclusion

In this paper, we proposed a new approach: learning the entity-likeness of phrases in sentences by using multiple approximate matching results. The experiments show three properties. The approach has portability with various pre-trained LMs. Our Sum pooling methods efficiently filter noisy approximate matching results for learning entity-likeness. Our approach effectively works particularly on small datasets, not only in the biomedical area but also in more general domains. Moreover, our approach does not require expensive computation. We hope that the proposed approach can contribute to identifying NEs in such cases.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3613–3618.

Jason P. C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4:357–370.

William W. Cohen and Sunita Sarawagi. 2004. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 89–98.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Wishart DS, Knox C, Guo AC, Shrivastava S, Hassanali M, Stothard P, Chang Z, and Woolsey J. 2019. Drugbank: a comprehensive resource for in silico drug discovery and exploration.

Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48.

HUGO Gene Nomenclature Committee (HGNC), European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, and United Kingdom www.genenames.org. 2019. Hgnc gene name database.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *CoRR*, abs/1904.05342.

Qiao Jin, Bhuwan Dhingra, William W. Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. *CoRR*, abs/1904.02181.

Muhammad Raza Khan, Morteza Ziyadi, and Mohamed AbdelHady. 2020. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, 19 Suppl 1:i180–2.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Esther Landhuis. 2016. Scientific literature: Information overload. *Nature*, 535:457–458.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019a. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019b. Biobert: a pre-trained biomedical language representation model for biomedical text mining.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1030–1038, Suntec, Singapore. Association for Computational Linguistics.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy. Association for Computational Linguistics.

Schriml LM, Mitraka E, Munro J, Tauber B, Schor M, Nickle L, Felix V, Jeng L, Bearer C, and Lichenstein R. 2019. Human disease ontology 2018 update: classification, content and workflow expansion.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. 2011. Entrez gene: gene-centered information at ncbi. *Nucleic acids research*, 39(Database issue), D52–D57.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. 2019. Entrez gene: gene-centered information at ncbi.

Naoaki Okazaki and Jun'ichi Tsujii. 2010. Simple and efficient algorithm for approximate dictionary matching. In *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pages 851–859.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 78–86.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Sampo Pyysalo, Filip Ginter, Hans Moen, T. Salakoski, and S. Ananiadou. 2013. Distributional semantics resources for biomedical text processing.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime Carbonell. 2020. Soft gazetteers for low-resource named entity recognition.

Nicky Ringland, Xiang Dai, Ben Hachey, Sarvnaz Karimi, Cécile Paris, and James R. Curran. 2019. NNE: A dataset for nested named entity recognition in english newswire. *CoRR*, abs/1906.01359.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. 2018. Learning named entity tagger using domain-specific dictionary. *CoRR*, abs/1809.03599.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2004. Improving the performance of dictionary-based approaches in protein name recognition. *Journal of Biomedical Informatics*, 37(6):461 – 470. Named Entity Recognition in Biomedicine.

Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. 2000. Named entity extraction based on A maximum entropy model and transformation rules. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000*.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *CoRR*, abs/1801.09851.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Minghao Wu, Fei Liu, and Trevor Cohn. 2018. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856, Brussels, Belgium. Association for Computational Linguistics.

Kai Xu, Zhenguo Yang, Peipei Kang, Qi Wang, and Wenyin Liu. 2019. Document-level attention-based bilstm-crf incorporating disease dictionary for disease named entity recognition. *Computers in Biology and Medicine*, 108:122 – 132.

Zhihao Yang, Hongfei Lin, and Yanpeng Li. 2008. Exploiting the performance of dictionary-based bio-entity name recognition in biomedical literature. *Computational Biology and Chemistry*, 32(4):287 – 291.