# Biomedical Entity Representations with Synonym Marginalization

**Mujeen Sung** **Hwisang Jeon** **Jinhyuk Lee**[†] **Jaewoo Kang**[†]
Korea University
{mujeensung,j_hs,jinhyuk_lee,kangj}@korea.ac.kr

## Abstract

Biomedical named entities often play important roles in many biomedical text mining tools. However, due to the incompleteness of provided synonyms and numerous variations in their surface forms, normalization of biomedical entities is very challenging. In this paper, we focus on learning representations of biomedical entities *solely based on the synonyms of entities*. To learn from the incomplete synonyms, we use a model-based candidate selection and maximize the marginal likelihood of the synonyms present in top candidates. Our model-based candidates are iteratively updated to contain more difficult negative samples as our model evolves. In this way, we avoid the explicit pre-selection of negative samples from more than 400K candidates. On four biomedical entity normalization datasets having three different entity types (disease, chemical, adverse reaction), our model BIOSYN consistently outperforms previous state-of-the-art models almost reaching the upper bound on each dataset.

## 1 Introduction

Biomedical named entities are frequently used as key features in biomedical text mining. From biomedical relation extraction (Xu et al., 2016; Li et al., 2017a) to literature search engines (Lee et al., 2016), many studies are utilizing biomedical named entities as a basic building block of their methodologies. While the extraction of the biomedical named entities is studied extensively (Sahu and Anand, 2016; Habibi et al., 2017), the normalization of extracted named entities is also crucial for improving the precision of downstream tasks (Leaman et al., 2013; Wei et al., 2015).

Unlike named entities from general domain text, typical biomedical entities have several different surface forms, making the normalization of biomedical entities very challenging. For instance, while two chemical entities '*motrin*' and '*ibuprofen*' belong to the same concept ID (MeSH:D007052), they have completely different surface forms. On the other hand, mentions having similar surface forms could also have different meanings (e.g. '*dystrophinopathy*' (MeSH:D009136) and '*bestrophinopathy*' (MeSH:C567518)). These examples show a strong need for building latent representations of biomedical entities that capture semantic information of the mentions.

In this paper, we propose a novel framework for learning biomedical entity representations based on the synonyms of entities. Previous works on entity normalization mostly train binary classifiers that decide whether the two input entities are the same (positive) or different (negative) (Leaman et al., 2013; Li et al., 2017b; Fakhraei et al., 2019; Phan et al., 2019). Our framework called BIOSYN uses the synonym marginalization technique, which maximizes the probability of all synonym representations in top candidates. We represent each biomedical entity using both sparse and dense representations to capture morphological and semantic information, respectively. The candidates are iteratively updated based on our model's representations removing the need for an explicit negative sampling from a large number of candidates. Also, the model-based candidates help our model learn from more difficult negative samples. Through extensive experiments on four biomedical entity normalization datasets, we show that BIOSYN achieves new state-of-the-art performance on all datasets, outperforming previous models by 0.8%∼2.6% top1 accuracy. Further analysis shows that our model's performance has almost reached the performance upper bound of each dataset.

[†]Corresponding authors



*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650

The contributions of our paper are as follows: First, we introduce BIOSYN for biomedical entity representation learning, which uses synonym marginalization dispensing with the explicit needs of negative training pairs. Second, we show that the iterative candidate selection based on our model's representations is crucial for improving the performance together with synonym marginalization. Finally, our model outperforms strong state-of-the-art models up to 2.6% on four biomedical normalization datasets.[1]

## 2 Related Works

Biomedical entity representations have largely relied on biomedical word representations. Right after the introduction of Word2vec (Mikolov et al., 2013), Pyysalo et al. (2013) trained Word2Vec on biomedical corpora such as PubMed. Their biomedical version of Word2Vec has been widely used for various biomedical natural language processing tasks (Habibi et al., 2017; Wang et al., 2018; Giorgi and Bader, 2018; Li et al., 2017a) including the biomedical normalization task (Mondal et al., 2019). Most recently, BioBERT (Lee et al., 2019) has been introduced for contextualized biomedical word representations. BioBERT is pre-trained on biomedical corpora using BERT (Devlin et al., 2019) and numerous studies are utilizing BioBERT for building state-of-the-art biomedical NLP models (Lin et al., 2019; Jin et al., 2019; Alsentzer et al., 2019; Sousa et al., 2019). Our model also uses pre-trained BioBERT for learning biomedical entity representations.

The intrinsic evaluation of the quality of biomedical entity representations is often verified by the biomedical entity normalization task (Leaman et al., 2013; Phan et al., 2019). The goal of the biomedical entity normalization task is to map an input mention from a biomedical text to its associated CUI (Concept Unique ID) in a dictionary. The task is also referred to as the entity linking or the entity grounding (D'Souza and Ng, 2015; Leaman and Lu, 2016). However, the normalization of biomedical entities is more challenging than the normalization of general domain entities due to a large number of synonyms. Also, the variations of synonyms depend on their entity types, which makes building type-agnostic normalization model difficult (Leaman et al., 2013; Li et al., 2017b; Mondal et al., 2019). Our work is generally applicable to any type of entity and evaluated on four datasets having three different biomedical entity types.

While traditional biomedical entity normalization models are based on hand-crafted rules (D'Souza and Ng, 2015; Leaman et al., 2015), recent approaches for the biomedical entity normalization have been significantly improved with various machine learning techniques. DNorm (Leaman et al., 2013) is one of the first machine learning-based entity normalization models, which learns pair-wise similarity using tf-idf vectors. Another machine learning-based study is CNN-based ranking method (Li et al., 2017b), which learns entity representations using a convolutional neural network. The most similar works to ours are NSEEN (Fakhraei et al., 2019) and BNE (Phan et al., 2019), which map mentions and concept names in dictionaries to a latent space using LSTM models and refines the embedding using the negative sampling technique. However, most previous works adopt a pair-wise training procedure that explicitly requires making negative pairs. Our work is based on marginalizing positive samples (i.e., synonyms) from iteratively updated candidates and avoids the problem of choosing a single negative sample.

In our framework, we represent each entity with sparse and dense vectors which is largely motivated by techniques used in information retrieval. Models in information retrieval often utilize both sparse and dense representations (Ramos et al., 2003; Palangi et al., 2016; Mitra et al., 2017) to retrieve relevant documents given a query. Similarly, we can think of the biomedical entity normalization task as retrieving relevant concepts given a mention (Li et al., 2017b; Mondal et al., 2019). In our work, we use maximum inner product search (MIPS) for retrieving the concepts represented as sparse and dense vectors, whereas previous models could suffer from error propagation of the pipeline approach.

## 3 Methodology

### 3.1 Problem Definition

We define an input mention $m$ as an entity string in a biomedical corpus. Each input mention has its own CUI $c$ and each CUI has one or more synonyms defined in the dictionary. The set of synonyms for a CUI is also called as a synset. We denote the union of all synonyms in a dictionary as $N = [n_1, n_2, \dots]$ where $n \in N$ is a single syn-

[1] Code available at https://github.com/dmis-lab/BioSyn.

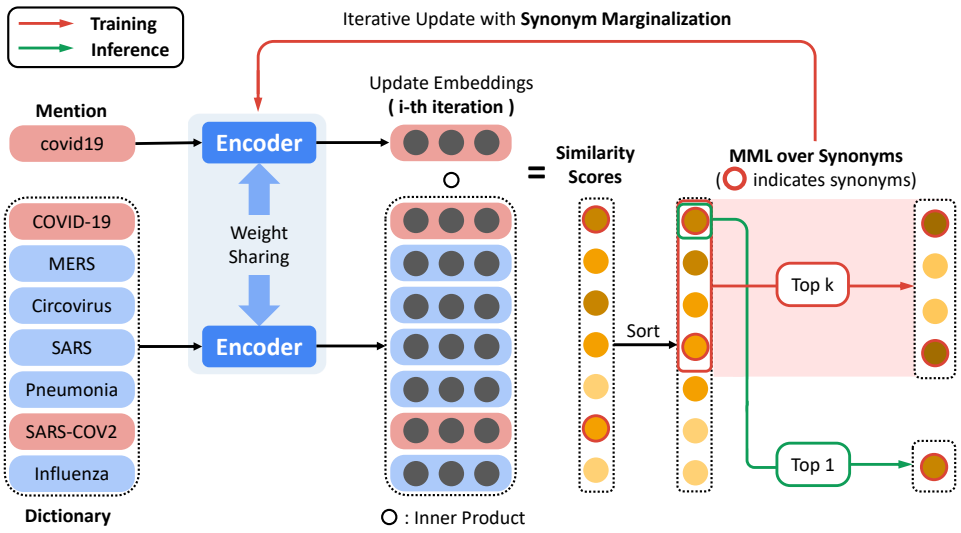



Figure 1: The overview of BIOSYN. An input mention and all synonyms in a dictionary are embedded by a shared encoder and the nearest synonym is retrieved by the inner-product. Top candidates used for training are iteratively updated as we train our encoders.

onym string. Our goal is to predict the gold CUI $c^*$ of the input mention $m$ as follows:

$$c^* = \texttt{CUI}(\text{argmax}_{n \in N} P(n|m;\theta)) \quad (1)$$

where $\texttt{CUI}(\cdot)$ returns the CUI of the synonym $n$ and $\theta$ denotes a trainable parameter of our model.

### 3.2 Model Description

The overview of our framework is illustrated in Figure 1. We first represent each input mention $m$ and each synonym $n$ in a dictionary using sparse and dense representations. We treat $m$ and $n$ equally and use a shared encoder for both strings. During training, we iteratively update top candidates and calculate the marginal probability of the synonyms based on their representations. At inference time, we find the nearest synonym by performing MIPS over all synonym representations.

**Sparse Entity Representation** We use tf-idf to obtain a sparse representation of $m$ and $n$. We denote each sparse representation as $e^s_m$ and $e^s_n$ for the input mention and the synonym, respectively. tf-idf is calculated based on the character-level n-grams statistics computed over all synonyms $n \in N$. We define the sparse scoring function of a mention-synonym pair $(m, n)$ as follows:

$$S_{\text{sparse}}(m, n) = f(e^s_m, e^s_n) \in \mathbb{R} \quad (2)$$

where $f$ denotes a similarity function. We use the inner product between two vectors as $f$.

**Dense Entity Representation** While the sparse representation encodes the morphological information of given strings, the dense representation encodes the semantic information. Learning effective dense representations is the key challenge in the biomedical entity normalization task (Li et al., 2017b; Mondal et al., 2019; Phan et al., 2019; Fakhraei et al., 2019). We use pre-trained BioBERT (Lee et al., 2019) to encode dense representations and fine-tune BioBERT with our synonym marginalization algorithm.[2] We share the same BioBERT model for encoding mention and synonym representations. We compute the dense representation of the mention $m$ as follows:

$$e^d_m = \texttt{BioBERT}(\overline{m})[\texttt{CLS}] \in \mathbb{R}^h \quad (3)$$

where $\overline{m} = \{\overline{m}_1, ..., \overline{m}_l\}$ is a sequence of sub-tokens of the mention $m$ segmented by the WordPiece tokenizer (Wu et al., 2016) and $h$ denotes the hidden dimension of BioBERT (i.e., $h = 768$). [CLS] denotes the special token that BERT-style models use to compute a single representative vector of an input. The synonym representation $e^d_n \in \mathbb{R}^h$ is computed similarly. We denote the dense scoring function of a mention-synonym pair $(m, n)$ using the dense representations as follows:

$$S_{\text{dense}}(m, n) = f(e^d_m, e^d_n) \in \mathbb{R} \quad (4)$$

where we again used the inner product for $f$.

[2]We used BioBERT v1.1 (+ PubMed) in our work.

**Similarity Function** Based on the two similarity functions $S_{\text{sparse}}(m, n)$ and $S_{\text{dense}}(m, n)$, we now define the final similarity function $S(m, n)$ indicating the similarity between an input mention $m$ and a synonym $n$:

$$S(m, n) = S_{\text{dense}}(m, n) + \lambda S_{\text{sparse}}(m, n) \in \mathbb{R} \quad (5)$$

where $\lambda$ is a trainable scalar weight for the sparse score. Using $\lambda$, our model learns to balance the importance between the sparse similarity and the dense similarity.

### 3.3 Training

The most common way to train the entity representation model is to build a pair-wise training dataset. While it is relatively convenient to sample positive pairs using synonyms, sampling negative pairs are trickier than sampling positive pairs as there are a vast number of negative candidates. For instance, the mention '*alpha conotoxin*' (MeSH:D020916) has 6 positive synonyms while its dictionary has 407,247 synonyms each of which can be a negative sampling candidate. Models trained on these pair-wise datasets often rely on the quality of the negative sampling (Leaman et al., 2013; Li et al., 2017b; Phan et al., 2019; Fakhraei et al., 2019). On the other hand, we use a model-based candidate retrieval and maximize the marginal probability of positive synonyms in the candidates.

**Iterative Candidate Retrieval** Due to a large number of candidates present in the dictionary, we need to retrieve a smaller number of candidates for training. In our framework, we use our entity encoder to update the top candidates iteratively. Let $k$ be the number of top candidates to be retrieved for training and $\alpha$ ($0 \leq \alpha \leq 1$) be the ratio of candidates retrieved from $S_{\text{dense}}$. We call $\alpha$ as the dense ratio and $\alpha = 1$ means consisting the candidates with $S_{\text{dense}}$ only. First, we compute the sparse scores $S_{\text{sparse}}$ and the dense scores $S_{\text{dense}}$ for all $n \in N$. Then we retrieve the $k - \lfloor \alpha k \rfloor$ highest candidates using $S_{\text{sparse}}$, which we call as sparse candidates. Likewise, we retrieve the $\lfloor \alpha k \rfloor$ highest candidates using $S_{\text{dense}}$, which we call as dense candidates. Whenever the dense and sparse candidates overlap, we add more dense candidates to match the number of candidates as $k$. While the sparse candidates for a mention will always be the same as they are based on the static tf-idf representation, the dense candidates change every epoch as our model learns better dense representations.

Our iterative candidate retrieval method has the following benefits. First, it makes top candidates to have more difficult negative samples as our model is trained, hence helping our model represent a more accurate dense representation of each entity. Also, it increases the chances of retrieving previously unseen positive samples in the top candidates. As we will see, comprising the candidates purely with sparse candidates have a strict upper bound while ours with dense candidates can maximize the upper bound.

**Synonym Marginalization** Given the top candidates from iterative candidate retrieval, we maximize the marginal probability of positive synonyms, which we call as synonym marginalization. Given the top candidates $N_{1:k}$ computed from our model, the probability of each synonym is obtained as:

$$P(n|m; \theta) = \frac{\exp(S(n, m))}{\sum_{n' \in N_{1:k}} \exp(S(n', m))} \quad (6)$$

where the summation in the denominator is over the top candidates $N_{1:k}$. Then, the marginal probability of the positive synonyms of a mention $m$ is defined as follows:

$$P'(m, N_{1:k}) = \sum_{\substack{n \in N_{1:k} \\ \text{EQUAL}(m,n)=1}} P(n|m; \theta) \quad (7)$$

where $\text{EQUAL}(m, n)$ is 1 when $\texttt{CUI}(m)$ is equivalent to $\texttt{CUI}(n)$ and 0 otherwise. Finally, we minimize the negative marginal log-likelihood of synonyms. We define the loss function of our model as follows:

$$\mathcal{L} = -\frac{1}{M} \sum_{i=1}^{M} \log P'(m_i, N_{1:k}) \quad (8)$$

where $M$ is the number of training mentions in our dataset. We use mini-batch for the training and use Adam optimizer (Kingma and Ba, 2015) to minimize the loss.

### 3.4 Inference

At inference time, we retrieve the nearest synonym of a mention representation using MIPS. We compute the similarity score $S(m, n)$ between the input mention $m$ and all synonyms $n \in N$ using the inner product and return the CUI of the nearest candidate. Note that it is computationally cheap to find the nearest neighbors once we pre-compute the dense and sparse representations of all synonyms.

## 4 Experimental Setup

### 4.1 Implementation Details

We perform basic pre-processings such as lowercasing all characters and removing the punctuation for both mentions and synonyms. To resolve the typo issues in mentions from NCBI disease, we apply the spelling check algorithm following the previous work (D'Souza and Ng, 2015). Abbreviations of entities are widely used in biomedical entities for an efficient notation which makes the normalization task more challenging. Therefore, we use the abbreviation resolution module called Ab3P[3] to detect the local abbreviations and expand it to its definition from the context (Sohn et al., 2008). We also split composite mentions (e.g. 'breast and ovarian cancer') into separate mentions (e.g. 'breast cancer' and 'ovarian cancer') using heuristic rules described in the previous work (D'Souza and Ng, 2015). We also merge mentions in the training set to the dictionary to increase the coverage following the previous work (D'Souza and Ng, 2015).

For sparse representations, we use character-level uni-, bi-grams for tf-idf. The maximum sequence length of BioBERT is set to 25[4] and any string over the maximum length is truncated to 25. The number of top candidates $k$ is 20 and the dense ratio $\alpha$ for the candidate retrieval is set to 0.5. We set the learning rate to 1e-5, weight decay to 1e-2, and the mini-batch size to 16. We found that the trainable scalar $\lambda$ converges to different values between 2 to 4 on each dataset. We train BIOSYN for 10 epochs for NCBI Disease, BC5CDR Disease, and TAC2017 ADR and 5 epochs for BC5CDR Chemical due to its large dictionary size. Except the number of epochs, we use the same hyperparameters for all datasets and experiments.

We use the top $k$ accuracy as an evaluation metric following the previous works in biomedical entity normalization tasks (D'Souza and Ng, 2015; Li et al., 2017b; Wright, 2019; Phan et al., 2019; Ji et al., 2019; Mondal et al., 2019). We define Acc@$k$ as 1 if a correct CUI is included in the top $k$ predictions, otherwise 0. We evaluate our models using Acc@1 and Acc@5. Note that we treat predictions for composite entities as correct if every prediction for each separate mention is correct.

[3] https://github.com/ncbi-nlp/Ab3P

[4] This covers 99.9% of strings in all datasets.

| Dataset | Documents | | | Mentions | | |
|---|---|---|---|---|---|---|
| | Train | Dev | Test | Train | Dev | Test |
| NCBI Disease | 592 | 100 | 100 | 5,134 | 787 | 960 |
| BC5CDR Disease | 500 | 500 | 500 | 4,182 | 4,244 | 4,424 |
| BC5CDR Chemical | 500 | 500 | 500 | 5,203 | 5,347 | 5,385 |
| TAC2017ADR | 101 | - | 99 | 7,038 | - | 6,343 |

Table 1: Data statistics of four biomedical entity normalization datasets. See Section 4.2 for more details.

### 4.2 Datasets

We use four biomedical entity normalization datasets having three different biomedical entity types (disease, chemical, adverse reaction). The statistics of each dataset is described in Table 1.

**NCBI Disease Corpus** NCBI Disease Corpus (Doğan et al., 2014)[5] provides manually annotated disease mentions in each document with each CUI mapped into the MEDIC dictionary (Davis et al., 2012). In this work, we use the July 6, 2012 version of MEDIC containing 11,915 CUIs and 71,923 synonyms included in MeSH and/or OMIM ontologies.

**Biocreative V CDR** BioCreative V CDR (Li et al., 2016)[6] is a challenge for the tasks of chemical-induced disease (CID) relation extraction. It provides disease and chemical type entities. The annotated disease mentions in the dataset are mapped into the MEDIC dictionary like the NCBI disease corpus. The annotated chemical mentions in the dataset are mapped into the Comparative Toxicogenomics Database (CTD) (Davis et al., 2018) chemical dictionary. In this work, we use the November 4, 2019 version of the CTD chemical dictionary containing 171,203 CUIs and 407,247 synonyms included in MeSH ontologies. Following the previous work (Phan et al., 2019), we filter out mentions whose CUIs do not exist in the dictionary.

**TAC2017ADR** TAC2017ADR (Roberts et al., 2017)[7] is a challenge whose purpose of the task is to extract information on adverse reactions found in structured product labels. It provides manually annotated mentions of adverse reactions that are mapped into the MedDRA dictionary (Brown et al., 1999). In this work, we use MedDRA v18.1 which contains 23,668 CUIs and 76,817 synonyms.

[5] https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE

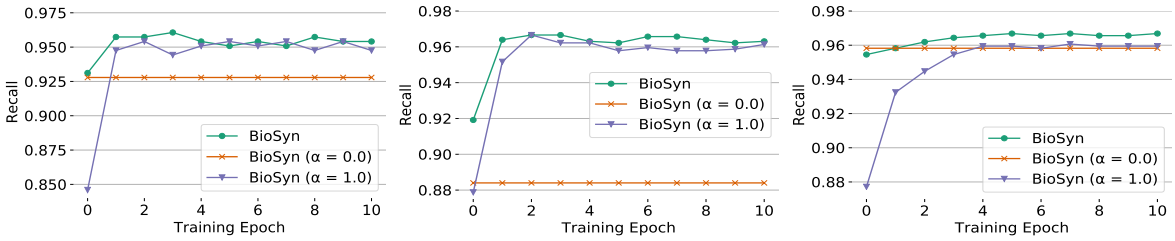
[6] https://biocreative.bioinformatics.udel.edu/tasks/biocreative-v/track-3-cdr

[7] https://bionlp.nlm.nih.gov/tac2017adversereactions

| Models | NCBI Disease | | BC5CDR Disease | | BC5CDR Chemical | | TAC2017ADR | |
|---|---|---|---|---|---|---|---|---|
| | Acc@1 | Acc@5 | Acc@1 | Acc@5 | Acc@1 | Acc@5 | Acc@1 | Acc@5 |
| Sieve-Based (D'Souza and Ng, 2015) | 84.7 | - | 84.1 | - | 90.7† | - | 84.3† | - |
| Taggerone (Leaman and Lu, 2016) | 87.7 | - | 88.9 | - | 94.1 | - | - | - |
| CNN Ranking (Li et al., 2017b) | 86.1 | - | - | - | - | - | - | - |
| NormCo (Wright, 2019) | 87.8 | - | 88.0 | - | - | - | - | - |
| BNE (Phan et al., 2019) | 87.7 | - | 90.6 | - | 95.8 | - | - | - |
| BERT Ranking (Ji et al., 2019) | 89.1 | - | - | - | - | - | 93.2 | - |
| TripletNet (Mondal et al., 2019) | 90.0 | - | - | - | - | - | - | - |
| BioSyn (s-score) | 87.6 | 90.5 | 92.4 | 95.7 | 95.9 | 96.8 | 91.4 | 94.5 |
| BioSyn (d-score) | 90.7 | 93.5 | 92.9 | 96.5 | **96.6** | 97.2 | 95.5 | 97.5 |
| BioSyn ($\alpha = 0.0$) | 89.9 | 93.3 | 92.2 | 94.9 | 96.3 | 97.2 | 95.3 | 97.6 |
| BioSyn ($\alpha = 1.0$) | 90.5 | **94.5** | 92.8 | **96.0** | 96.4 | **97.3** | **95.8** | **97.9** |
| BioSyn (Ours) | **91.1** | 93.9 | **93.2** | **96.0** | **96.6** | 97.2 | 95.6 | 97.5 |

† We used the author's provided implementation to evaluate the model on these datasets.

Table 2: Experimental results on four biomedical entity normalization datasets




(a) NCBI Disease (b) BC5CDR Disease (c) BC5CDR Chemical

Figure 2: Effect of iterative candidate retrieval on the development sets of NCBI Disease, BC5CDR Disease, and BC5CDR Chemical. We show the recall of top candidates of each model.

## 5 Experimental Results

We use five different versions of our model to see the effect of each module in our framework. First, BioSyn denotes our proposed model with default hyperparameters described in Section 4.1. BioSyn (s-score) and BioSyn (d-score) use only sparse scores or dense scores for the predictions at inference time, respectively. To see the effect of different dense ratios, BioSyn ($\alpha = 0.0$) uses only sparse candidates and BioSyn ($\alpha = 1.0$) uses only dense candidates during training.

### 5.1 Main Results

Table 2 shows our main results on the four datasets. Our model outperforms all previous models on the four datasets and achieves new state-of-the-art performance. The Acc@1 improvement on NCBI Disease, BC5CDR Disease, BC5CDR Chemical and TAC2017ADR are 1.1%, 2.6%, 0.8% and 2.4%, respectively. Training with only dense candidates ($\alpha = 1.0$) often achieves higher Acc@5 than BioSyn showing the effectiveness of dense candidates.

### 5.2 Effect of Iterative Candidate Retrieval

In Figure 2, we show the effect of the iterative candidate retrieval method. We plot the recall of top candidates used in each model on the development sets. The recall is 1 if any top candidate has the gold CUI. BioSyn ($\alpha = 1$) uses only dense candidates while BioSyn ($\alpha = 0$) uses sparse candidates. BioSyn utilizes both dense and sparse candidates. Compared to the fixed recall of BioSyn ($\alpha = 0$), we observe a consistent improvement in BioSyn ($\alpha = 1$) and BioSyn. This proves that our proposed model can increase the upper bound of candidate retrieval using dense representations.

### 5.3 Effect of the Number of Candidates

We perform experiments by varying the number of top candidates used for training. Figure 3 shows that a model with 20 candidates performs reasonably well in terms of both Acc@1 and Acc@5. It shows that more candidates do not guarantee higher performance, and considering the training complexity, we choose $k = 20$ for all experiments.

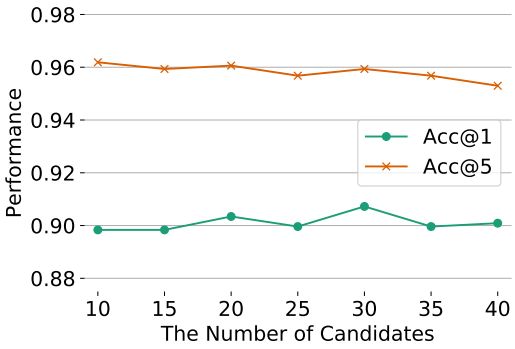



Figure 3: Performance of BIOSYN on the development set of NCBI Disease with different numbers of candidates

### 5.4 Effect of Synonym Marginalization

Our synonym marginalization method uses marginal maximum likelihood (MML) as the objective function. To verify the effectiveness of our proposed method, we compare our method with two different strategies: hard EM (Liang et al., 2018) and the standard pair-wise training (Leaman et al., 2013). The difference between hard EM and MML is that hard EM maximizes the probability of a single positive candidate having the highest probability. In contrast, MML maximizes marginalized probabilities of all synonyms in the top candidates. For hard EM, we first obtain a target $\tilde{n}$ as follows:

$$\tilde{n} = \text{argmax}_{n \in N_{1:k}} P(n|m;\theta) \quad (9)$$

where most notations are the same as Equation 1. The loss function of hard EM is computed as follows:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M} \log P(\tilde{n}|m_i;\theta). \quad (10)$$

The pair-wise training requires a binary classification model. For the pair-wise training, we minimize the binary cross-entropy loss using samples created by pairing each positive and negative candidate in the top candidates with the input mention. Table 3 shows the results of applying three different loss functions on BC5CDR Disease and BC5CDR Chemical. The results show that MML used in our framework learns better semantic representations than other methods.

## 6 Analysis

### 6.1 Iterative Candidate Samples

In Table 4, we list top candidates of BIOSYN from the NCBI Disease development set. Although the initial candidates did not have positive samples due to the limitation of sparse representations, candidates at epoch 1 begin to include more positive candidates. Candidates at epoch 5 include many positive samples, while negative samples are also closely related to each mention.

| Methods | BC5CDR D. | | BC5CDR C. | |
|---|---|---|---|---|
| | Acc@1 | Acc@5 | Acc@1 | Acc@5 |
| MML | **91.1** | 95.4 | **96.7** | **97.7** |
| Hard EM | 91.0 | **95.8** | 96.5 | 97.5 |
| Pair-Wise Training | 90.7 | 94.4 | 96.3 | 97.2 |

Table 3: Comparison of two different training methods on the development sets of BC5CDR Disease, BC5CDR Chemical

### 6.2 Error Analysis

In Table 5, we analyze the error cases of our model on the test set of NCBI Disease. We manually inspected all failure cases and defined the following error cases in the biomedical entity normalization task: Incomplete Synset, Contextual Entity, Overlapped Entity, Abbreviation, Hypernym, and Hyponym. Remaining failures that are difficult to categorize are grouped as Others.

**Incomplete Synset** is the case when the surface form of an input mention is very different from the provided synonyms of a gold CUI and requires the external knowledge for the normalization. **Contextual Entity** denotes an error case where an input mention and the predicted synonym are exactly the same but have different CUIs. This type of error could be due to an annotation error or happen when the same mention can be interpreted differently depending on its context. **Overlapped Entity** is an error where there is an overlap between the words of input mention and the predicted candidate. This includes nested entities. **Abbrevation** is an error where an input mention is in an abbreviated form but the resolution has failed even with the external module Ab3P. **Hypernym** and **Hyponym** are the cases when an input mention is a hypernym or a hyponym of the annotated entity.

Based on our analyses, errors are mostly due to ambiguous annotations (Contextual Entity, Overlapped Entity, Hypernym, Hyponym) or failure of pre-processings (Abbreviation). Incomplete Synset can be resolved with a better dictionary having richer synonym sets. Given the limitations in annotations, we conclude that the performance of BIOSYN has almost reached the upper bound.

| Rank | tf-idf | Epoch 0 | Epoch 1 | Epoch 5 |
|---|---|---|---|---|
| | | prostate carcinomas (MeSH:D011471) | | |
| 1 | carcinomas | **prostatic cancers*** | **prostate cancers*** | **prostate cancers*** |
| 2 | teratocarcinomas | **prostate cancers*** | **prostatic cancers*** | **prostate cancer*** |
| 3 | pancreatic carcinomas ... | glioblastomas | **prostate neoplasms*** | **prostatic cancers*** |
| 4 | carcinomatoses | carcinomas | **prostate cancer*** | **prostate neoplasms*** |
| 5 | carcinomatosis | renal cell cancers | **prostate neoplasm*** | **prostatic cancer*** |
| 6 | breast carcinomas | renal cancers | **prostatic cancer*** | **cancers prostate*** |
| 7 | teratocarcinoma | retinoblastomas | **prostatic neoplasms*** | **prostate neoplasm*** |
| 8 | carcinoma | cholangiocarcinomas | **advanced prostate cancers*** | **cancer of prostate*** |
| 9 | breast carcinoma | pulmonary cancers | **prostatic neoplasm*** | **cancer of the prostate*** |
| 10 | carcinosarcomas | gonadoblastomas | prostatic adenomas | **cancer prostate*** |
| | | brain abnormalities (MeSH:D001927) | | |
| 1 | nail abnormalities | brain dysfunction minimal | **brain pathology*** | **brain disorders*** |
| 2 | abnormalities nail | **brain pathology*** | **brain disorders*** | **brain disorder*** |
| 3 | facial abnormalities | deficits memory | white matter abnormalities | **brain diseases*** |
| 4 | torsion abnormalities | memory deficits | **brain disease*** | **brain disease*** |
| 5 | spinal abnormalities | neurobehavioral manifestations | **brain diseases*** | abnormalities in brain dev... |
| 6 | skin abnormalities | white matter diseases | **brain disorder*** | nervous system abnormalities |
| 7 | genital abnormalities | brain disease metabolic | neuropathological abnormalities | white matter abnormalities |
| 8 | nail abnormality | neuropathological abnormalities | brain dysfunction minimal | metabolic brain disorders |
| 9 | clinical abnormalities | neurobehavioral manifestation | white matter lesions | brain metabolic disorders |
| 10 | abnormalities in brain dev... | **brain disease*** | brain injuries | **brain pathology*** |
| | | type ii deficiency (OMIM:217000) | | |
| 1 | mat i iii deficiency | deficiency disease | **type ii c2 deficient*** | factor ii deficiency |
| 2 | naga deficiency type iii ... | type 1 citrullinemia | deficiency disease | **type ii c2 deficient*** |
| 3 | properdin deficiency type iii ... | cmo ii deficiency | deficiency diseases | factor ii deficiencies |
| 4 | properdin deficiency type i ... | mitochondrial trifunctional ... | **type ii c2d deficiency*** | **type ii c2d deficiency*** |
| 5 | naga deficiency type iii | **type ii c2 deficient*** | factor ii deficiency | diabetes mellitus type ii |
| 6 | naga deficiency type ii | deficiency aga | deficiency protein | deficiency factor ii |
| 7 | properdin deficiency type iii | sodium channel myotonia | deficiency vitamin | **c2 deficiency*** |
| 8 | properdin deficiency type ii | deficiency diseases | deficiency factor ii | t2 deficiency |
| 9 | tc ii deficiency | tuftsin deficiency | deficiency arsa | tc ii deficiency |
| 10 | si deficiency | triosephosphate isomerase ... | class ii angle | mitochondrial complex ii ... |

Table 4: Changes in the top 10 candidates given the two input mentions from the NCBI Disease development set. Synonyms having correct CUIs are indicated in boldface with an asterisk.

| Error Type | Input | Predicted | Annotated | Statistics |
|---|---|---|---|---|
| Incomplete Synset | hypomania | hypodermyiasis | mood disorders | 25 (29.4%) |
| Contextual Entity | colorectal adenomas | colorectal adenomas | polyps adenomatous | 3 (3.5%) |
| Overlapped Entity | desmoid tumors | desmoid tumor | desmoids | 11 (12.9%) |
| Abbreviation | sca1 | oca1 | spinocerebellar ataxia 1 | 10 (11.8%) |
| Hypernym | campomelia | campomelic syndrome | campomelia cumming type | 10 (11.8%) |
| Hyponym | eye movement abnormalities | eye movement disorder | eye abnormalities | 23 (27.1%) |
| Others | hamartoma syndromes | hamartomas | multiple hamartoma syndromes | 3 (3.5%) |

Table 5: Examples and statistics of error cases on the NCBI Disease test set

## 7 Conclusion

In this study, we introduce BIOSYN that utilizes the synonym marginalization technique and the iterative candidate retrieval for learning biomedical entity representations. On four biomedical entity normalization datasets, our experiment shows that our model achieves state-of-the-art performance on all datasets, improving previous scores up to 2.6%. Although the datasets used in our experiments are in English, we expect that our methodology would work in any language as long as there is a synonym dictionary for the language. For future work, an extrinsic evaluation of our methods is needed to prove the effectiveness of learned biomedical entity representations and to prove the quality of the entity normalization in downstream tasks.

## Acknowledgments

This research was supported by National Research Foundation of Korea (NRF-2016M3A9A7916996, NRF-2014M3C9A3063541). We thank the members of Korea University, and the anonymous reviewers for their insightful comments.

## References


Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, WA Redmond, and Matthew BA McDermott. 2019. Publicly available clinical bert embeddings. In *NAACL-HLT*.

Elliot G Brown, Louise Wood, and Sue Wood. 1999. The medical dictionary for regulatory activities (meddra). *Drug safety*, 20(2).

Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. 2018. The comparative toxicogenomics database: update 2019. *NAR*, 47(D1):D948–D954.

Allan Peter Davis, Thomas C Wiegers, Michael C Rosenstein, and Carolyn J Mattingly. 2012. Medic: a practical disease vocabulary used at the comparative toxicogenomics database. *Database*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In *ACL*.

Shobeir Fakhraei, Joel Mathew, and Jose-Luis Ambite. 2019. Nseen: Neural semantic embedding for entity normalization. In *ECML-PKDD*.

John M Giorgi and Gary D Bader. 2018. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics*, 34(23):4087–4094.

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.

Zongcheng Ji, Qiang Wei, and Hua Xu. 2019. Bert-based ranking for biomedical entity normalization. *arXiv preprint arXiv:1908.03548*.

Qiao Jin, Bhuwan Dhingra, William Cohen, and Xinghua Lu. 2019. Probing biomedical embeddings from language models. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Robert Leaman and Zhiyong Lu. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics*, 32(18):2839–2846.

Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. 2015. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics*, 7(Suppl 1).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Sunwon Lee, Donghyeon Kim, Kyubum Lee, Jaehoon Choi, Seongsoon Kim, Minji Jeon, Sangrak Lim, Donghee Choi, Sunkyu Kim, Aik-Choon Tan, et al. 2016. Best: next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PloS one*, 11(10).

Fei Li, Meishan Zhang, Guohong Fu, and Donghong Ji. 2017a. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, 18(1):198.

Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang. 2017b. Cnn-based ranking for biomedical entity normalization. *BMC bioinformatics*, 18(Suppl 11):385.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*.

Chen Liang, Mohammad Norouzi, Jonathan Berant, Quoc V Le, and Ni Lao. 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In *NIPS*.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A bert-based universal model for both within-and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*.

Ishani Mondal, Sukannya Purkayastha, Sudeshna Sarkar, Pawan Goyal, Jitesh Pillai, Amitava Bhattacharyya, and Mahanandeeshwar Gattu. 2019. Medical entity linking using triplet network. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *TASLP*, 24(4).

Minh C Phan, Aixin Sun, and Yi Tay. 2019. Robust representation learning of biomedical names. In *ACL*.

Sampo Pyysalo, F Ginter, Hans Moen, T Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. *Proceedings of Languages in Biology and Medicine*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

Kirk Roberts, Dina Demner-Fushman, and Joseph M Tonning. 2017. Overview of the tac 2017 adverse reaction extraction from drug labels track. In *TAC*.

Sunil Sahu and Ashish Anand. 2016. Recurrent neural network models for disease name recognition using domain invariant features. In *ACL*.

Sunghwan Sohn, Donald C Comeau, Won Kim, and W John Wilbur. 2008. Abbreviation definition identification based on automatic precision estimates. *BMC bioinformatics*, 9:402.

Diana Sousa, André Lamúrias, and Francisco M Couto. 2019. A silver standard corpus of human phenotype-gene relations. In *NAACL-HLT*.

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. *Bioinformatics*, 35(10):1745–1752.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2015. Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*.

Dustin Wright. 2019. *NormCo: Deep disease normalization for biomedical knowledge base construction*. Ph.D. thesis, UC San Diego.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*.

Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. 2016. Cd-rest: a system for extracting chemical-induced disease relation in literature. *Database*.