A Generate-and-Rank Framework with Semantic Type Regularization for Biomedical Concept Normalization

Dongfang Xu and **Zeyu Zhang** and **Steven Bethard**
School of Information
University of Arizona
Tucson, AZ
{dongfangxu9,zeyuzhang,bethard}@email.arizona.edu

Abstract

Concept normalization, the task of linking textual mentions of concepts to concepts in an ontology, is challenging because ontologies are large. In most cases, annotated datasets cover only a small sample of the concepts, yet concept normalizers are expected to predict all concepts in the ontology. In this paper, we propose an architecture consisting of a candidate generator and a list-wise ranker based on BERT. The ranker considers pairings of concept mentions and candidate concepts, allowing it to make predictions for any concept, not just those seen during training. We further enhance this list-wise approach with a semantic type regularizer that allows the model to incorporate semantic type information from the ontology during training. Our proposed concept normalization framework achieves state-of-the-art performance on multiple datasets.

1 Introduction

Mining and analyzing the constantly-growing unstructured text in the bio-medical domain offers great opportunities to advance scientific discovery (Gonzalez et al., 2015; Fleuren and Alkema, 2015) and improve the clinical care (Rumshisky et al., 2016; Liu et al., 2019). However, lexical and grammatical variations are pervasive in such text, posing key challenges for data interoperability and the development of natural language processing (NLP) techniques. For instance, *heart attack*, *MI*, *myocardial infarction*, and *cardiovascular stroke* all refer to the same concept. It is critical to disambiguate these terms by linking them with their corresponding concepts in an ontology or knowledge base. Such linking allows downstream tasks (relation extraction, information retrieval, text classification, etc.) to access the ontology's rich knowledge about biomedical entities, their synonyms, semantic types and mutual relationships.

Concept normalization is a task that maps *concept mentions*, the in-text natural-language mentions of ontological concepts, to *concept entries* in a standardized ontology or knowledge base. Techniques for concept normalization have been advancing, thanks in part to recent shared tasks including clinical disorder normalization in 2013 ShARe/CLEF (Suominen et al., 2013) and 2014 SemEval Task 7 Analysis of Clinical Text (Pradhan et al., 2014), and adverse drug event normalization in Social Media Mining for Health (SMM4H) (Sarker et al., 2018; Weissenbacher et al., 2019). Most existing systems use a string-matching or dictionary look-up approach (Leal et al., 2015; D'Souza and Ng, 2015; Lee et al., 2016), which are limited to matching morphologically similar terms, or supervised multi-class classifiers (Belousov et al., 2017; Tutubalina et al., 2018; Niu et al., 2019; Luo et al., 2019a), which may not generalize well when there are many concepts in the ontology and the concept types that must be predicted do not all appear in the training data.

We propose an architecture (shown in Figure 1) that is able to consider both morphological and semantic information. We first apply a candidate generator to generate a list of candidate concepts, and then use a BERT-based list-wise classifier to rank the candidate concepts. This two-step architecture allows unlikely concept candidates to be filtered out prior to the final classification, a necessary step when dealing with ontologies with millions of concepts. In contrast to previous list-wise classifiers (Murty et al., 2018) which only take the concept mention as input, our BERT-based list-wise classifier takes both the concept mention and the candidate concept name as input, and is thus able to handle concepts that never appear in the training data. We further enhance this list-wise approach with a semantic type regularizer that allows our ranker to leverage semantic type information from

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8452–8464
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

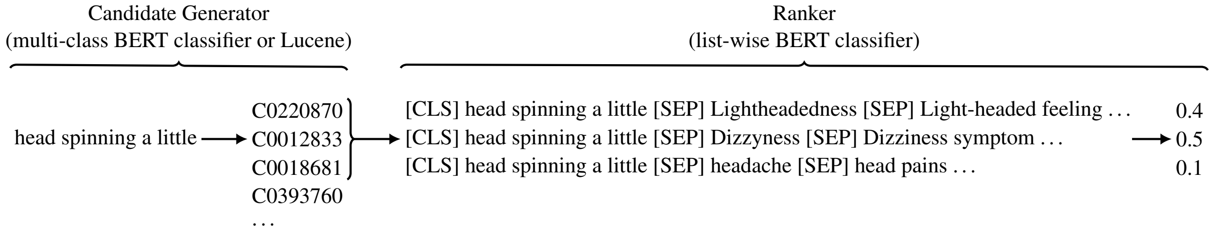
Candidate Generator
(multi-class BERT classifier or Lucene)
Ranker
(list-wise BERT classifier)
head spinning a little
C0220870
C0012833
C0018681
C0393760
...
[CLS] head spinning a little [SEP] Lightheadedness [SEP] Light-headed feeling ...
[CLS] head spinning a little [SEP] Dizzyness [SEP] Dizziness symptom ...
[CLS] head spinning a little [SEP] headache [SEP] head pains ...
0.4
0.5
0.1

Figure 1: Proposed architecture for concept normalization: candidate generation and ranking.

the ontology during training.

Our work makes the following contributions:

- Our proposed concept normalization framework achieves state-of-the-art performance on multiple datasets.

- We propose a concept normalization framework consisting of a candidate generator and a list-wise classifier. Our framework is easier to train and the list-wise classifier is able to predict concepts never seen during training.

- We introduce a semantic type regularizer which encourages the model to consider the semantic type information of the candidate concepts. This semantic type regularizer improves performance over the BERT-based list-wise classifier on multiple datasets.

The code for our proposed generate-and-rank framework is available at `https://github.com/dongfang91/Generate-and-Rank-ConNorm`.

2 Related work

Traditional approaches for concept normalization involve string match and dictionary look-up. These approaches differ in how they construct dictionaries, such as collecting concept mentions from the labeled data as extra synonyms (Leal et al., 2015; Lee et al., 2016), and in different string matching techniques, such as string overlap and edit distance (Kate, 2016). Two of the most commonly used knowledge-intensive concept normalization tools, MetaMap (Aronson, 2001) and cTAKES (Savova et al., 2010) both employ rules to first generate lexical variants for each noun phrase and then conduct dictionary look-up for each variant. Several systems (D'Souza and Ng, 2015; Jonnagaddala et al., 2016) have demonstrated that rule-based concept normalization systems achieve performance competitive with other approaches in a *sieve-based approach* that carefully selects combinations and orders of dictionaries, exact and partial matching, and heuristic rules. However, such rule-based approaches struggle when there are great variations between concept mention and concept, which is common, for example, when comparing social media text to medical ontologies.

Due to the availability of shared tasks and annotated data, the field has shifted toward machine learning techniques. We divide the machine learning approaches into two categories, classification (Savova et al., 2008; Stevenson et al., 2009; Limsopatham and Collier, 2016; Yepes, 2017; Festag and Spreckelsen, 2017; Lee et al., 2017; Tutubalina et al., 2018; Niu et al., 2019) and learning to rank (Leaman et al., 2013; Liu and Xu, 2017; Li et al., 2017; Nguyen et al., 2018; Murty et al., 2018).

Most classification-based approaches using deep neural networks have shown strong performance. They differ in using different architectures, such as Gated Recurrent Units (GRU) with attention mechanisms (Tutubalina et al., 2018), multi-task learning with auxiliary tasks to generate attention weights (Niu et al., 2019), or pre-trained transformer networks (Li et al., 2019; Miftahutdinov and Tutubalina, 2019); different sources for training word embeddings, such as Google News (Limsopatham and Collier, 2016) or concept definitions from the Unified Medical Language System (UMLS) Metathesaurus (Festag and Spreckelsen, 2017); and different input representations, such as using character embeddings (Niu et al., 2019). All classification approaches share the disadvantage that the output space must be the same size as the number of concepts to be predicted, and thus the output space tends to be small such as 2,200 concepts in (Limsopatham and Collier, 2016) and around 22,500 concepts in (Weissenbacher et al., 2019). Classification approaches also struggle with concepts that have only a few example mentions in the training data.

Researchers have applied point-wise learning to rank (Liu and Xu, 2017; Li et al., 2017), pair-wise learning to rank (Leaman et al., 2013; Nguyen et al., 2018), and list-wise learning to rank (Murty et al., 2018; Ji et al., 2019) on concept normalization. Generally, the learning-to-rank approach has the advantage of reducing the output space by first obtaining a smaller list of possible candidate concepts via a candidate generator and then ranking them. DNorm (Leaman et al., 2013), based on a pair-wise learning-to-rank model where both mentions and concept names were represented as TF-IDF vectors, was the first to use learning-to-rank for concept normalization and achieved the best performance in the ShARe/CLEF eHealth 2013 shared task. List-wise learning-to-rank approaches are both computationally more efficient than pair-wise learning-to-rank (Cao et al., 2007) and empirically outperform both point-wise and pair-wise approaches (Xia et al., 2008). There are two implementations of list-wise classifiers using neural networks for concept normalization: Murty et al. (2018) treat the selection of the best candidate concept as a flat classification problem, losing the ability to handle concepts not seen during training; Ji et al. (2019) take a generate-and-rank approach similar to ours, but they do not leverage resources such as synonyms or semantic type information from UMLS in their BERT-based ranker.

3 Proposed methods

3.1 Concept normalization framework

We define a concept mention m as an abbreviation such as “MI”, a noun phrase such as “heart attack”, or even a short text such as “an obstruction of the blood supply to the heart”. The goal is then to assign m with a concept c. Formally, given a list of pre-identified concept mentions $M = \{m_1, m_2, ..., m_n\}$ in the text and an ontology or knowledge base with a set of concepts $C = \{c_1, c_2, ..., c_t\}$, the goal of concept normalization is to find a mapping function $c_j = f(m_i)$ that maps each textual mention to its correct concept.

We approach concept normalization in two steps: we first use a candidate generator $G(m, C) \rightarrow C_m$ to generate a list of candidate concepts C_m for each mention m, where $C_m \subseteq C$ and $|C_m| \ll |C|$. We then use a candidate ranker $R(m, C_m) \rightarrow \hat{C_m}$, where $\hat{C_m}$ is a re-ranked list of candidate concepts sorted by their relevance, preference, or importance. But unlike information retrieval tasks where the order of candidate concepts in the sorted list $\hat{C_m}$ is crucial, in concept normalization we care only that the one true concept is at the top of the list.

The main idea of the two-step approach is that we first use a simple and fast system with high recall to generate candidates, and then a more precise system with more discriminative input to rank the candidates.

3.2 Candidate generator

We implement two kinds of candidate generators: a BERT-based multi-class classifier when the number of concepts in the ontology is small, and a Lucene-based[1] dictionary look-up when there are hundreds of thousands of concepts in the ontology.

3.2.1 BERT-based multi-class classifier

BERT (Devlin et al., 2019) is a contextualized word representation model that has shown great performance in many NLP tasks. Here, we use BERT in a multi-class text-classification configuration as our candidate concept generator. We use the final hidden vector $V_m \in \mathbb{R}^H$ corresponding to the first input token (`[CLS]`) generated from $BERT(m)$ and a classification layer with weights $W \in \mathbb{R}^{|C| \times H}$, and train the model using a standard classification loss:

$$L_G = y * log(softmax(V_m W^T)) \qquad (1)$$

where y is a one-hot vector, and $|y| = |C|$. The score for all concepts is calculated as:

$$p(C) = softmax(V_m W^T) \qquad (2)$$

We select the top k most probable concepts in $p(C)$ and feed that list C_m to the ranker.

3.2.2 Lucene-based dictionary look-up system

Multi-pass sieve rule based systems (D'Souza and Ng, 2015; Jonnagaddala et al., 2016; Luo et al., 2019b) achieve competitive performance when used with the right combinations and orders of different dictionaries, exact and partial matching, and heuristic rules. Such systems relying on basic lexical matching algorithms are simple and fast to implement, but they are only able to generate candidate concepts which are morphologically similar to a given mention.

Inspired by the work of Luo et al. (2019b), we implement a Lucene-based sieve normalization system which consists of the following components (see Appendix A.1 for details):

[1] https://lucene.apache.org/

a. Lucene index over the training data finds all mentions that exactly match m.

b. Lucene index over ontology finds concepts whose preferred name exactly matches m.

c. Lucene index over ontology finds concepts where at least one synonym of the concept exactly matches m.

d. Lucene index over ontology finds concepts where at least one synonym of the concept has high character overlap with m.

The ranked list C_m generated by this system is fed as input to the candidate ranker.

3.3 Candidate ranker

After the candidate generator produces a list of concepts, we use a BERT-based list-wise classifier to select the most likely candidate. BERT allows us to match morphologically dissimilar (but semantically similar) mentions and concepts, and the list-wise classifier takes both mention and candidate concepts as input, allowing us to handle concepts that appear infrequently (or never) in the training data.

Here, we use BERT similar to a question answering configuration, where given a concept mention m, the task is to choose the most likely candidate concept c_m from all candidate concepts C_m. As shown in Figure 1, our classifier input includes the text of the mention m and all synonyms of the candidate concept c_m, and takes the form `[CLS]` m `[SEP]` $syn_1(c_m)$ `[SEP]` ... `[SEP]` $syn_s(c_m)$ `[SEP]`, where $syn_i(c_m)$ is the i^{th} synonym of concept c_m[2]. We calculate the final hidden vector $V_{(m,c_m)} \in \mathbb{R}^H$ corresponding to the first input token (`[CLS]`) generated from BERT for each such input, and then concatenate the hidden vectors of all candidate concepts to form a matrix $V_{(m,C_m)} \in \mathbb{R}^{|C_m| \times H}$. We use this matrix and classification layer weights $W \in \mathbb{R}^H$, and compute a standard classification loss:

$$L_R = y * log(softmax(V_{(m,C_m)} W^T)). \quad (3)$$

where y is a one-hot vector, and $|y| = |C_m|$.

3.4 Semantic type regularizer

To encourage the list-wise classifier towards a more informative ranking than just getting the correct concept at the top of the list, we propose a semantic type regularizer that is optimized when candidate concepts with the correct semantic type are ranked above candidate concepts with incorrect types. The semantic type of the candidate concept is assumed correct only if it exactly matches the semantic type of the gold truth concept. If the concept has multiple semantic types, all must match. Our semantic type regularizer consists of two components:

$$R_p(\hat{y_t}, \hat{y_p}) = \sum_{p \in P(y)} (m_1 + \hat{y_p} - \hat{y_t}) \quad (4)$$

$$R_n(\hat{y_p}, \hat{y_n}) = \sum_{p \in P(y)} \max_{n \in N(y)} (m_2 + \hat{y_n} - \hat{y_p}) \quad (5)$$

where $\hat{y} = V_{(m,c_m)} W^T$, $N(y)$ is the set of indexes of candidate concepts with incorrect semantic types (negative candidates), $P(y)$ (positive candidates) is the complement of $N(y)$, $\hat{y_t}$ is the score of the gold truth candidate concept, and thus $t \in P(y)$. The margins m_1 and m_2 are hyper-parameters for controlling the minimal distances between $\hat{y_t}$ and $\hat{y_p}$ and between $\hat{y_p}$ and $\hat{y_n}$, respectively. Intuitively, R_p tries to push the score of the gold truth concept above all positive candidates at least by m_1, and R_n tries to push the best scored negative candidate below all positive candidates by m_2.

The final loss function we optimize for the BERT-based list-wise classifier is:

$$L = L_R + \lambda R_p(\hat{y_t}, \hat{y_p}) + \mu R_n(\hat{y_p}, \hat{y_n}) \quad (6)$$

where λ and μ are hyper-parameters to control the tradeoff between standard classification loss and the semantic type regularizer.

4 Experiments

4.1 Datasets

Our experiments are conducted on three social media datasets, AskAPatient (Limsopatham and Collier, 2016), TwADR-L (Limsopatham and Collier, 2016), and SMM4H-17 (Sarker et al., 2018), and one clinical notes dataset, MCN (Luo et al., 2019b). We summarize dataset characteristics in Table 1.

AskAPatient The AskAPatient dataset[3] contains 17,324 adverse drug reaction (ADR) annotations collected from blog posts. The mentions are mapped to 1,036 medical concepts with

[2] In preliminary experiments, we tried only the concept's preferred term and several other ways of separating synonyms, but none of these resulted in better performance.

[3] http://dx.doi.org/10.5281/zenodo.55013

Dataset	AskAPatient	TwADR-L	SMM4H-17	MCN
Ontology	SNOMED-CT & AMT	MedDRA	MedDRA (PT)	SNOMED-CT & RxNorm
Subset	Y	Y	N	N
$\|C_{ontology}\|$	1,036	2,220	22,500	434,056
$\|ST_{ontology}\|$	22	18	61	125
$\|C_{dataset}\|$	1,036	2,220	513	3,792
$\|M\|$	17,324	5,074	9,149	13,609
$\|M_{train}\|$	15665.2	4805.7	5,319	5,334
$\|M_{test}\|$	866.2	142.7	2,500	6,925
$\|M\|/\|C_{dataset}\|$	16.72	2.29	17.83	3.59
$\|C_{test} - C_{train}\|$	0	0	43	2,256
$\|M_{test} - M_{train}\|/M_{test}$	39.7%	39.5%	34.7%	53.9%
$\|M_{ambiguous}\|/\|M\|$	1.2%	12.8%	0.8%	4.5%

Table 1: Dataset statistics, where C is a set of concepts, ST is a set of semantic types, and M is a set of mentions.

22 semantic types from the subset of Systematized Nomenclature Of Medicine-Clinical Term (SNOMED-CT) and the Australian Medicines Terminology (AMT). We follow the 10-fold cross validation (CV) configuration in Limsopatham and Collier (2016) which provides 10 sets of train/dev/test splits.

TwADR-L The TwADR-L dataset[3] contains 5,074 ADR expressions from social media. The mentions are mapped to 2,220 Medical Dictionary for Regulatory Activities (MedDRA) concepts with 18 semantic types. We again follow the 10-fold cross validation configuration defined by Limsopatham and Collier (2016).

SMM4H-17 The SMM4H-17 dataset [4] consists of 9,149 manually curated ADR expressions from tweets. The mentions are mapped to 22,500 concepts with 61 semantic types from MedDRA Preferred Terms (PTs). We use the 5,319 mentions from the released set as our training data, and keep the 2,500 mentions from the original test set as evaluation.

MCN The MCN dataset consists of 13,609 concept mentions drawn from 100 discharge summaries from the fourth i2b2/VA shared task (Uzuner et al., 2011). The mentions are mapped to 3792 unique concepts out of 434,056 possible concepts with 125 semantic types in SNOMED-CT and RxNorm. We take 40 clinical notes from the released data as training, consisting of 5,334 mentions, and the standard evaluation data with 6,925 mentions as our test set. Around 2.7% of mentions in MCN could not be mapped to any concepts in the terminology, and are assigned the *CUI-less* label.

A major difference between the datasets is the space of concepts that systems must consider. For AskAPatient and TwADR-L, all concepts in the test data are also in the training data, and in both cases only a couple thousand concepts have to be considered. Both SMM4H-17 and MCN define a much larger concept space: SMM4H-17 considers 22,500 concepts (though only 513 appear in the data) and MCN considers 434,056 (though only 3,792 appear in the data). AskAPatient and TwADR-L have no unseen concepts in their test data, SMM4H-17 has a few (43), while MCN has a huge number (2,256). Even a classifier that perfectly learned all concepts in the training data could achieve only 70.15% accuracy on MCN. MCN also has more unseen mentions: 53.9%, where the other datasets have less than 40%. The MCN dataset is thus harder to memorize, as systems must consider many mentions and concepts never seen in training.

Unlike the clinical MCN dataset, in the three social media datasets – AskAPatient, TwADR-L, and SMM4H-17 – it is common for the ADR expressions to share no words with their target medical concepts. For instance, the ADR expression "makes me like a zombie" is assigned the concept "C1443060" with preferred term "feeling abnormal". The social media datasets do not include context, only the mentions themselves, while the MCN dataset provides the entire note surrounding each mention. Since only 4.5% of mentions in the MCN dataset are ambiguous, for the current experiments we ignore this additional context information.

4.2 Unified Medical Language System

The UMLS Metathesaurus (Bodenreider, 2004) links similar names for the same concept

[4] http://dx.doi.org/10.17632/rxwfb3tysd.1

from nearly 200 different vocabularies such as SNOMED-CT, MedDRA, RxNorm, etc. There are over 3.5 million concepts in UMLS, and for each concept, UMLS also provides the definition, preferred term, synonyms, semantic type, relationships with other concepts, etc.

In our experiments, we make use of synonyms and semantic type information from UMLS. We restrict our concepts to the three vocabularies, MedDRA, SNOMED-CT, and RxNorm in the UMLS version 2017AB. For each concept in the ontologies of the four datasets, we first find its concept unique identifier (CUI) in UMLS. We then extract synonyms and semantic type information according to the CUI. Synonyms (English only) are collected from level 0 terminologies containing vocabulary sources for which no additional license agreements are necessary.

4.3 Evaluation metrics

For all four datasets, the standard evaluation of concept normalization systems is accuracy. For the AskAPatient and TwADR-L datasets, which use 10-fold cross validation, the accuracy metrics are averaged over 10 folds.

4.4 Implementation details

We use the BERT-based multi-class classifier as the candidate generator on the three social media datasets AskAPatient, TwADR-L, and SMM4H-17, and the Lucene-based candidate generator for the MCN dataset. In the social media datasets, the number of concepts in the data is small, few test concepts are unseen in the training data, and there is a greater need to match expressions that are morphologically dissimilar from medical concepts. In the clinical MCN dataset, the opposites are true.

For all experiments, we use BioBERT-base (Lee et al., 2019), which further pre-trains BERT on PubMed abstracts (PubMed) and PubMed Central full-text articles (PMC). We use huggingface's pytorch implementation of BERT[5]. We select the best hyper-parameters based on the performance on dev set. See Appendix A.2 for hyperparameter settings.

4.5 Comparisons with related methods

We compare our proposed architecture with the following state-of-the-art systems.

WordCNN Limsopatham and Collier (2016) use convolutional neural networks over pre-trained word embeddings to generate a vector representation for each mention, and then feed these into a softmax layer for multi-class classification.

WordGRU+Attend+TF-IDF Tutubalina et al. (2018) use a bidirectional GRU with attention over pre-trained word embeddings to generate a vector representation for each mention, concatenate such vector representations with the cosine similarities of the TF-IDF vectors between the mention and all other concept names, and then feed the concatenated vector to a softmax layer for multi-class classification.

BERT+TF-IDF Miftahutdinov and Tutubalina (2019) take similar approach as Tutubalina et al. (2018), but use BERT to generate a vector representation for each mention. They concatenate the vector representations with the cosine similarities of the TF-IDF vectors between the mention and all other concept names, and then feed the concatenated vector to a softmax layer for multi-class classification.

CharCNN+Attend+MT Niu et al. (2019) use a multi-task attentional character-level convolution neural network. They first convert the mention into a character embedding matrix. The auxiliary task network takes the embedding matrix as input for a CNN to learn to generate character-level domain-related importance weights. Such learned importance weights are concatenated with the character embedding matrix and fed as input to another CNN model with a softmax layer for multi-class classification.

CharLSTM+WordLSTM Han et al. (2017) first use a forward LSTM over each character of the mention and its corresponding character class such as lowercase or uppercase to generate a character-level vector representation, then use another bi-directional LSTM over each word of the mention to generate a word-level representation. They concatenate character-level and word-level representations and feed them as input to a softmax layer for multi-class classification.

LR+MeanEmbedding Belousov et al. (2017) calculate the mean of three different weighted word embeddings pre-trained on GoogleNews, Twitter and DrugTwitter as vector representations for

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

	TwADR-L		AskAPatient		SMM4H-17	
Approach	Dev	Test	Dev	Test	Dev	Test
WordCNN (Limsopatham and Collier, 2016)	-	44.78	-	81.41	-	-
WordGRU+Attend+TF-IDF (Tutubalina et al., 2018)	-	-	-	85.71	-	-
BERT+TF-IDF (Miftahutdinov and Tutubalina, 2019)	-	-	-	-	-	**89.64**
CharCNN+Attend+MT (Niu et al., 2019)	-	46.46	-	84.65	-	-
CharLSTM+WordLSTM (Han et al., 2017)	-	-	-	-	-	87.20
LR+MeanEmbedding (Belousov et al., 2017)	-	-	-	-	-	87.70
BERT	47.08	44.05	88.63	**87.52**	84.74	87.36
BERT + BERT-rank	48.07	46.32	88.14	87.10	84.44	87.66
BERT + BERT-rank + ST-reg	47.98	**47.02**	88.26	87.46	84.66	88.24
BERT + gold + BERT-rank	52.70	49.69	89.06	87.92	88.57	90.16
BERT + gold + BERT-rank + ST-reg	52.84	50.81	89.68	88.51	88.87	91.08

Table 2: Comparisons of our proposed concept normalization architecture against the current state-of-the-art performances on *TwADR-L*, *AskAPatient*, and *SMM4H-17* datasets.

the mention, where word weights are calculated as inverse document frequency. Such vector representations are fed as input to a multinomial logistic regression (LR) model for multi-class classification.

Sieve-based Luo et al. (2019b) build a sieve-based normalization model which contains exact-match and MetaMap (Aronson, 2001) modules. Given a mention as input, the exact-match module first looks for mentions in the training data that exactly match the input, and then looks for concepts from the ontology whose synonyms exactly match the input. If no concepts are found, the mention is fed into MetaMap. They run this sieve-based normalization model twice. In the first round, the model lower-cases the mentions and includes acronym/abbreviation tokens during dictionary lookup. In the second round, the model lower-cases the mentions spans and also removes special tokens such as "&apos;s", "&quot;", etc.

Since our focus is individual systems, not ensembles, we compare only to other non-ensembles[6].

4.6 Models

We separate out the different contributions from the following components of our architecture.

BERT The BERT-based multi-class classifier. When used alone, we select the most probable concept as the prediction.

	MCN	
Approach	Dev	Test
Sieve-based (Luo et al., 2019b)	-	76.35
Lucene		79.25
Lucene+BERT-rank	83.56	82.75
Lucene+BERT-rank+ST-reg	**84.44**	**83.56**
Lucene+gold+BERT-rank	86.89	84.77
Lucene+gold+BERT-rank+ST-reg	88.59	86.56

Table 3: Accuracy of our proposed concept normalization architecture on *MCN* dataset.

Lucene The Lucene-based dictionary look-up. When used alone, we take the top-ranked candidate concept as the prediction.

+BERT-rank The BERT-based list-wise classifier, always used in combination with either BERT or Lucene as a canddiate generator

+ST-reg The semantic type regularizer, always used in combination with BERT-ranker.

We also consider the case (**+gold**) where we artificially inject the correct concept into the candidate generator's list if it was not already there.

5 Results

Table 2 shows that our complete model, BERT + BERT-rank + ST-reg, achieves a new state-of-the-art on two of the social media test sets, and Table 3 shows that Lucene + BERT-rank + ST-reg achieves a new state-of-the-art on the clinical MCN test set. The TwADR-L dataset is the most difficult, with our complete model achieving 47.02% accuracy. In the other datasets, performance of our complete

[6] An ensemble of three systems (including CharLSTM+WordLSTM and LR+MeanEmbedding) achieved 88.7% accuracy on the SMM4H-17 dataset (Sarker et al., 2018).

model is much higher: 87.46% for AskAPatient, 88.24% for SMM4H-17[7].

On the TwADR-L, SMM4H-17, and MCN test sets, adding the BERT-based ranker improves performance over the candidate generator alone, and adding the semantic type regularization further improves performance. For example, Lucene alone achieves 79.25% accuracy on the MCN data, adding the BERT ranker increases this to 82.75%, and adding the semantic type regularizer increases this to 83.56%. On AskAPatient, performance of the full model is similar to just the BERT multi-class classifier, perhaps because in this case BERT alone already successfully improves the state-of-the-art from 85.71% to 87.52%. The +gold setting allows us to answer how well our ranker would perform if our candidate generator made no mistakes. First, we can see that if the correct concept is always in the candidate list, our list-based ranker (+BERT-rank) outperforms the multi-class classifier (BERT) on all test sets. We also see in this setting that the benefits of the semantic type regularizer are amplified, with test sets of TwADR-L and MCN showing more than 1.00% gain in accuracy from using the regularizer. These findings suggest that improving the quality of the candidate generator should be a fruitful future direction.

Overall, we see the biggest performance gains from our proposed generate-and-rank architecture in the MCN dataset. This is the most realistic setting, where the number of candidate concepts is large and many test concepts were never seen during training. In such cases, we cannot use a multi-class classifier as a candidate generator since it would never generate unseen concepts. Thus, our ranker shines in its ability to sort through the long list of possible concepts.

6 Qualitative analysis

Candidates	L	BR
Repair of abdominal wall hernia	1	3
Repair of anterior abdominal wall hernia	2	4
Obstructed hernia of anterior abdominal wall	3	5
Hernia of abdominal wall	4	1
Abdominal wall hernia procedure	5	2

Table 4: Predicted candidate concepts for mention *An abdominal wall hernia* and their rankings among the outputs of Lucene (L) and BERT-Ranker (BR). Gold concept is *Hernia of abdominal wall.*

Candidates	BR	STR	ST
Influenza-like illness	1	2	DS
Influenza	2	4	DS
Influenza-like symptoms	3	1	SS
Feeling tired	4	5	F
Muscle cramps in feet	5	3	SS

Table 5: Predicted candidate concepts for mention *felt like I was coming down with flu* and their rankings among the outputs of BERT-Ranker (BR) and BERT-Ranker + semantic type regularizer (STR). Gold concept is *flu-like symptoms*. Semantic types (ST) of the candidates include: disease or syndrome (DS), sign or symptom (SS), finding (F)

Table 4 shows an example that is impossible for the multi-class classifier approach to concept normalization. The concept mention "an abdominal wall hernia" in the clinical MCN dataset needs to be mapped to the concept with the preferred name "Hernia of abdominal wall", but that concept never appeared in the training data. The Lucene-based candidate generator finds this concept, but only through character overlap (step d.) and several other concepts have high overlap as well. Thus Lucene ranks the correct concept 4th in its list. The BERT ranker is able to compare "an abdominal wall hernia" to "Hernia of abdominal wall" and recognize that as a better match than the other options, re-assigning it to rank 1.

Table 5 shows an example that illustrates why the semantic type regularizer helps. The mention "felt like I was coming down with flu" in the social media AskAPatient dataset needs to be mapped to the concept with the preferred name "influenza-like symptoms", which has the semantic type of a sign or symptom. The BERT ranker ranks two disease or syndromes higher, placing the correct concept at rank 3. After the semantic type regularizer is added, the system recognizes that the mention should be mapped to a sign or symptom, and correctly ranks it above the disease or syndromes. Note that this happens even though the ranker does not get to see the semantic type of the input mention at prediction time.

7 Limitations and future research

The available concept normalization datasets are somewhat limited. Lee et al. (2017) notes that AskAPatient and TwADR-L have issues including

[7] Miftahutdinov and Tutubalina (2019) use the same architecture as our BERT-based multi-class classifier (row 7), but they achieve 89.28% of accuracy on SMM4H-17. We were unable to replicate this result as their code and parameter settings were unavailable.

duplicate instances, which can lead to bias in the system; many phrases have multiple valid mappings to concepts but the context necessary to disambiguate is not part of the dataset; and the 10-fold cross-validation makes training complex models unnecessarily expensive. These datasets are also unrealistic in that all concepts in the test data are seen during training. Future research should focus on more realistic datasets that follow the approach of MCN in annotating mentions of concepts from a large ontology and including the full context.

Our ability to explore the size of the candidate list was limited by our available computational resources. As the size of the candidate list increases, the true concept is more likely to be included, but the number of training instances also increases, making the computational cost larger, especially for the datasets using 10-fold cross-validation. We chose candidate list sizes as large as we could afford, but there are likely further gains possible with larger candidate lists.

Our semantic type regularizer is limited to exact matching: it checks only whether the semantic type of a candidate exactly matches the semantic type of the true concept. The UMLS ontology includes many other relations, such as is-a and part-of relations, and extending our regularizer to encode such rich semantic knowledge may yield further improvements in the BERT-based ranker.

8 Conclusion

We propose a concept normalization framework consisting of a candidate generator and a list-wise classifier based on BERT.

Because the candidate ranker makes predictions over pairs of concept mentions and candidate concepts, it is able to predict concepts never seen during training. Our proposed semantic type regularizer allows the ranker to incorporate semantic type information into its predictions without requiring semantic types at prediction time. This generate-and-rank framework achieves state-of-the-art performance on multiple concept normalization datasets.

Acknowledgments

We thank the anonymous reviewers for their insightful comments on an earlier draft of this paper. This work was supported in part by National Institutes of Health grant R01LM012918 from the National Library of Medicine (NLM) and grant R01GM114355 from the National Institute of General Medical Sciences (NIGMS). The computations were done in systems supported by the National Science Foundation under Grant No. 1228509. This research was supported in part by an appointment to the Oak Ridge National Laboratory Advanced Short-Term Research Opportunity (ASTRO) Program, sponsored by the U.S. Department of Energy and administered by the Oak Ridge Institute for Science and Education. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, National Science Foundation, or Department of Energy.

References

Alan R. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, pages 17–21. American Medical Informatics Association.

Maksim Belousov, William Dixon, and Goran Nenadic. 2017. Using an Ensemble of Generalised Linear and Deep Learning Models in the SMM4H 2017 Medical Concept Normalisation Task. In *CEUR Workshop Proceedings*, volume 1996, pages 54–58.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136. Association for Computing Machinery.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 297–302, Beijing, China. Association for Computational Linguistics.

Sven Festag and Cord Spreckelsen. 2017. Word Sense Disambiguation of Medical Terms via Recurrent Convolutional Neural Networks. In *Health Informatics Meets EHealth: Digital InsightInformation-Driven Health & Care. Proceedings of the 11th EHealth2017 Conference*, volume 236, pages 8–15.

Wilco W.M. Fleuren and Wynand Alkema. 2015. Application of text mining in the biomedical domain. *Methods*, 74:97–106.

Graciela H. Gonzalez, Tasnia Tahsin, Britton C. Goodale, Anna C. Greene, and Casey S. Greene. 2015. Recent Advances and Emerging Applications in Text and Data Mining for Biomedical Discovery. *Briefings in Bioinformatics*, 17(1):33–42.

Sifei Han, Tung Tran, Anthony Rios, and Ramakanth Kavuluru. 2017. Team UKNLP: Detecting ADRs, Classifying Medication Intake Messages, and Normalizing ADR Mentions on Twitter. In *CEUR Workshop Proceedings*, volume 1996, pages 49–53.

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

Jitendra Jonnagaddala, Toni Rose Jue, Nai-Wen Chang, and Hong-Jie Dai. 2016. Improving the dictionary lookup approach for disease normalization using enhanced dictionary and query expansion. *Database*, 2016:baw112.

Rohit J. Kate. 2016. Normalizing clinical terms using learned edit distance patterns. *Journal of the American Medical Informatics Association*, 23(2):380–386.

André Leal, Bruno Martins, and Francisco Couto. 2015. ULisboa: Recognition and normalization of medical concepts. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 406–411, Denver, Colorado. Association for Computational Linguistics.

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

Hsin-Chun Lee, Yi-Yu Hsu, and Hung-Yu Kao. 2016. AuDis: an automatic CRF-enhanced disease normalization in biomedical text. *Database*, 2016. Baw091.

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*. Btz682.

Kathy Lee, Sadid A. Hasan, Oladimeji Farri, Alok Choudhary, and Ankit Agrawal. 2017. Medical Concept Normalization for Online User-Generated Texts. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 462–469. IEEE.

Fei Li, Yonghao Jin, Weisong Liu, Bhanu Pratap Singh Rawat, Pengshan Cai, and Hong Yu. 2019. Fine-Tuning Bidirectional Encoder Representations From Transformers (BERT)–Based Models on Large-Scale Electronic Health Record Notes: An Empirical Study. *JMIR Med Inform*, 7(3):e14830.

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

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1014–1023, Berlin, Germany. Association for Computational Linguistics.

Feifan Liu, Abhyuday Jagannatha, and Hong Yu. 2019. Towards drug safety surveillance and pharmacovigilance: Current progress in detecting medication and adverse drug events from electronic health records. *Drug Saf*, 42:95–97.

Hongwei Liu and Yun Xu. 2017. A Deep Learning Way for Disease Name Representation and Normalization. In *Natural Language Processing and Chinese Computing*, pages 151–157. Springer International Publishing.

Yen-Fu Luo, Weiyi Sun, and Anna Rumshisky. 2019a. A Hybrid Normalization Method for Medical Concepts in Clinical Narrative using Semantic Matching. In *AMIA Joint Summits on Translational Science proceedings*, volume 2019, pages 732–740. American Medical Informatics Association.

Yen-Fu Luo, Weiyi Sun, and Anna Rumshisky. 2019b. MCN: A comprehensive corpus for medical concept normalization. *Journal of Biomedical Informatics*, pages 103–132.

Zulfat Miftahutdinov and Elena Tutubalina. 2019. Deep neural models for medical concept normalization in user-generated texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 393–399, Florence, Italy. 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, Melbourne, Australia. Association for Computational Linguistics.

Thanh Ngan Nguyen, Minh Trang Nguyen, and Thanh Hai Dang. 2018. Disease Named Entity Normalization Using Pairwise Learning To Rank and Deep Learning. Technical report, VNU University of Engineering and Technology.

Jinghao Niu, Yehui Yang, Siheng Zhang, Zhengya Sun, and Wensheng Zhang. 2019. Multi-task Character-Level Attentional Networks for Medical Concept Normalization. *Neural Process Lett*, 49(3):1239–1256.

Sameer Pradhan, Noémie Elhadad, Wendy Chapman, Suresh Manandhar, and Guergana Savova. 2014. SemEval-2014 task 7: Analysis of clinical text. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 54–62, Dublin, Ireland. Association for Computational Linguistics.

Anna Rumshisky, Marzyeh Ghassemi, Tristan Naumann, Peter Szolovits, V.M. Castro, T.H. McCoy, and R.H. Perlis. 2016. Predicting early psychiatric readmission with natural language processing of narrative discharge summaries. *Transl Psychiatry*, 6(10):e921–e921.

Abeed Sarker, Maksim Belousov, Jasper Friedrichs, Kai Hakala, Svetlana Kiritchenko, Farrokh Mehryary, Sifei Han, Tung Tran, Anthony Rios, Ramakanth Kavuluru, Berry de Bruijn, Filip Ginter, Debanjan Mahata, Saif M. Mohammad, Goran Nenadic, and Graciela Gonzalez-Hernandez. 2018. Data and systems for medication-related text classification and concept normalization from Twitter: insights from the Social Media Mining for Health (SMM4H)-2017 shared task. *Journal of the American Medical Informatics Association*, 25(10):1274–1283.

Guergana K. Savova, Anni R. Coden, Igor L. Sominsky, Rie Johnson, Philip V. Ogren, Piet C. De Groen, and Christopher G. Chute. 2008. Word sense disambiguation across two domains: Biomedical literature and clinical notes. *Journal of Biomedical Informatics*, 41(6):1088–1100.

Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C. Kipper-Schuler, and Christopher G. Chute. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Mark Stevenson, Yikun Guo, Abdulaziz Alamri, and Robert Gaizauskas. 2009. Disambiguation of biomedical abbreviations. In *Proceedings of the BioNLP 2009 Workshop*, pages 71–79, Boulder, Colorado. Association for Computational Linguistics.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J.F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon. 2013. Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 212–231. Springer Berlin Heidelberg.

Elena Tutubalina, Zulfat Miftahutdinov, Sergey Nikolenko, and Valentin Malykh. 2018. Medical concept normalization in social media posts with recurrent neural networks. *Journal of Biomedical Informatics*, 84:93–102.

Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael J. Paul, and Graciela Gonzalez-Hernandez. 2019. Overview of the fourth social media mining for health (SMM4H) shared tasks at ACL 2019. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 21–30, Florence, Italy. Association for Computational Linguistics.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1192–1199. Association for Computing Machinery.

Antonio Jimeno Yepes. 2017. Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. *Journal of Biomedical Informatics*, 73:137–147.

A Appendices

A.1 Lucene-based dictionary look-up system

The lucene-based dictionary look-up system consists of the following components:

(a) Lucene index over the training data finds all CUI-less mentions that exactly match mention m.

(b) Lucene index over the training data finds CUIs of all training mentions that exactly match mention m.

(c) Lucene index over UMLS finds CUIs whose preferred name exactly matches mention m.

(d) Lucene index over UMLS finds CUIs where at least one synonym of the CUI exactly matches mention m.

(e) Lucene index over UMLS finds CUIs where at least one synonym of the CUI has high character overlap with mention m. To check the character overlap, we run the following three rules sequentially: token-level matching, fuzzy string matching with a maximum edit distance of 2, and character 3-gram matching..

See Figure A1 for the flow of execution across the components. Whenever there are multiple CUIs generated from a component **(a)** to **(e)**, they are fed, along with the concept mention, to the BERT-based reranker (f).

During training, we used component **(e)** alone instead of the combination of components **(b)**-**(e)** to generate training instances for the BERT-based reranker (f) as it generated many more training examples and resulted in better performance on the dev set. During evaluation, we used the whole pipeline.

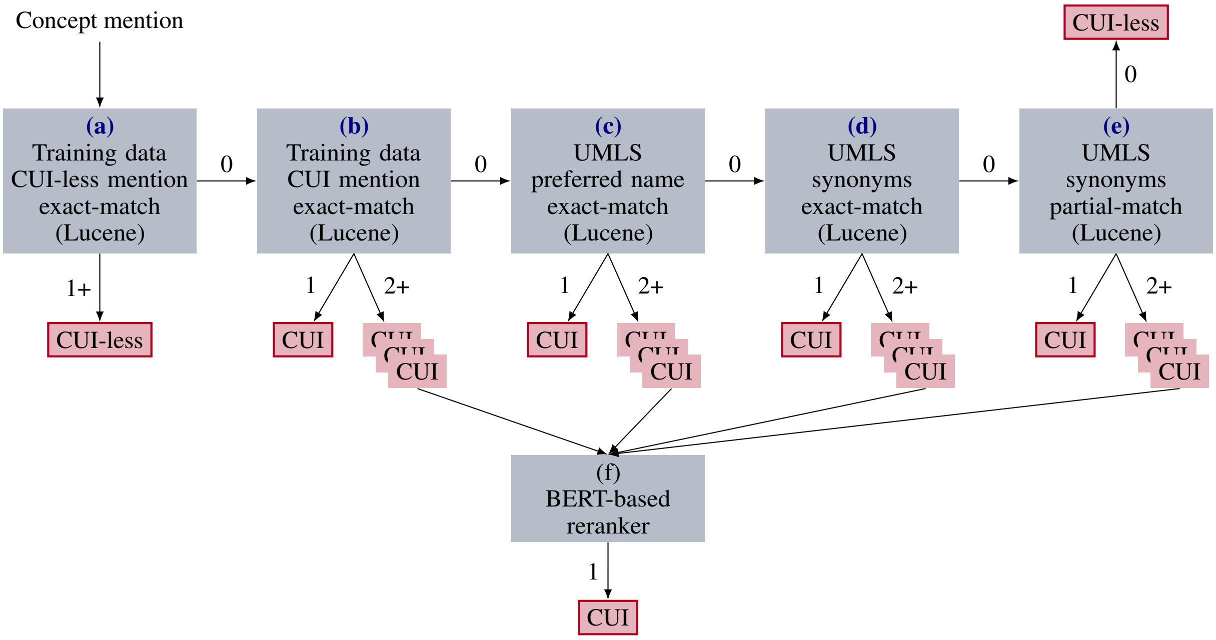

Figure A1: Architecture of the lucene-based dictionary look-up system. The edges out of a search process indicate the number of matches necessary to follow the edge. Outlined nodes are terminal states that represent the predictions of the system.

	Multi-class			List-wise			
	AAP	TwADR-L	SMM4H-17	AAP	TwADR-L	SMM4H-17	MCN
learning_rate	1e-4	5e-5	5e-5	5e-5	5e-5	3e-5	3e-5
num_train_epochs	30	30	40	10	10	20	30
per_gpu_train_batch_size	32	16	32	16	16	16	8
save_steps	487	301	166	976	301	333	250
warmup_steps	1463	903	664	976	301	666	750
list size (k)	-	-	-	10	20	10	30
m_1	-	-	-	0.0	0.0	0.0	0.1
m_2	-	-	-	0.2	0.2	0.2	0.2
λ	-	-	-	0.6	0.4	0.4	0.4
μ	-	-	-	0.6	0.4	0.4	0.8

Table A1: Hyper-parameters for BERT-based multi-class and list-wise classifiers. AAP=AskAPatient. Terms with underscores are hyper-parameters in huggingface's pytorch implementation of BERT.

A.2 Hyper-parameters

Table A1 shows the hyper-parameters for our models. We use huggingface's pytorch implementation of BERT. We tune the hyperparameters via grid search, and select the best BERT hyper-parameters based on the performance on the dev set.

To keep the size of the candidate list equal to k for every mention, we apply the following rules: if the list does not contain the gold concept and is already of length k, we inject the correct one and remove an incorrect candidate; if the list is not length of k, we inject the gold concept and the most frequent concepts in the training set to reach k.