Keyphrase Identification Using Minimal Labeled Data with Hierarchical Context and Transfer Learning

Rohan Goli, MS[1]; Nina Hubig, PhD[1]; Hua Min, PhD[2]; Yang Gong, PhD[3]; Dean F. Sittig, PhD[3]; Lior Rennert, PhD[4]; David Robinson, MD[5]; Paul Biondich, MD[6]; Adam Wright, PhD[7]; Christian Nøhr, PhD[8]; Timothy Law, DO[9]; Arild Faxvaag, PhD[10]; Aneesa Weaver, BS[4]; Ronald Gimbel, PhD[4]; Xia Jing, PhD[4]

[1]School of Computing, College of Engineering, Computing and Applied Science, Clemson University, Clemson, SC, USA; [2]Department of Health Administration and Policy, College of Public Health, George Mason University, Fairfax, VA, USA; [3]School of Biomedical Informatics, The University of Texas Health Science Center at Houston, Houston, TX, USA; [4]Department of Public Health Sciences, College of Behavioral, Social, and Health Sciences, Clemson University, Clemson, SC, USA; [5]General Practitioner/Independent Consultant, Cumbria, UK; [6]Clem McDonald Biomedical Informatics Center, Regenstrief Institute, Department of Pediatrics, Indiana University School of Medicine, Indianapolis, IN, USA; [7]Vanderbilt University Medical Center, Nashville, TN, USA; [8]Department of Planning, Faculty of Engineering, Aalborg University, Aalborg, Denmark; [9]Ohio Musculoskeletal and Neurologic Institute, Ohio University, Athens, OH, USA; [10]Department of Neuromedicine and Movement Science, Faculty of Medicine and Health Sciences, Norwegian University of Science and Technology, Trondheim, Norway

ABSTRACT

Interoperable clinical decision support system (CDSS) rules provide a pathway to interoperability, a well-recognized challenge in health information technology. Building an ontology facilitates creating interoperable CDSS rules, which can be achieved by identifying the keyphrases (KP) from the existing literature. However, KP identification for data labeling requires human expertise, consensus, and contextual understanding. This paper aims to present a semi-supervised KP identification framework using minimal labeled data based on hierarchical attention over the documents and domain adaptation. Our method outperforms the prior neural architectures by learning through synthetic labels for initial training, document-level contextual learning, language modeling, and fine-tuning with limited gold standard label data. To the best of our knowledge, this is the first functional framework for the CDSS sub-domain to identify KPs, which is trained on limited labeled data. It contributes to the general natural language processing (NLP) architectures in areas such as clinical NLP, where manual data labeling is challenging, and light-weighted deep learning models play a role in real-time KP identification as a complementary approach to human experts' effort.

KEYWORDS

Clinical Decision Support System, Minimal labeled data, Hierarchical context, Semi-supervised learning, Domain adaptation, Natural language processing

Abbreviations:

NLP: Natural language processing
CDSS: Clinical decision support system
HDE: Human domain expert
BiLSTM: Bidirectional long short-term memory
BiLM: Bidirectional language model
CRF: Conditional random field

The code is available on GitHub: https://github.com/xjing16/cdss4pcp_nlpml_pipeline.

Correspondence Author: Xia Jing, Email: xjing@clemson.edu

GS: Gold standard
KP: Keyphrase

1. Introduction

Interoperability [1,2] is a well-recognized barrier in health informatics. For example, a lack of interoperability can create chaos when transmitting patients' health records between institutions. Despite good progress in interoperability in healthcare, it is not yet a common reality. Clinical decision support systems (CDSSs), especially rule-based CDSSs have been effective in improving the quality of healthcare and preventive services [3,4]. However, developing and maintaining CDSS rules are resource-demanding, and it is difficult to share such rules across multiple healthcare settings. Having interoperable CDSS rules can facilitate interoperability via an ontology [5,6] that uses unambiguous concepts and their relationships.

In an article, these concepts are an orderly sequence of words or N-grams, namely, keyphrases (KP). These KPs contribute to their meanings and the contextual understanding of the text. A KP is a gold standard (GS) if it is selected by a human domain expert (HDE) after careful review and with consensus among multiple HDEs.

An ontology can be constructed using GS terms and their relationships, which in turn provide foundations for interoperable and generic CDSS rules [7]. However, ontology construction is usually a manual process with the experts' input and curators' deep understanding of the domain and application contexts, where KP identification is one of the steps. Automatic KP identification is a critical complement to the process of manual construction and curation of an ontology.

We aim to build a system using natural language processing (NLP), which speeds up KP identification by humans. When the system design involves reviewing a text corpus and finding the data patterns to determine the N-grams, it can be driven by NLP neural network architectures [8, 9], which can automate identifying possible GS terms.

Given any text, some classic NLP algorithms (supervised and rule-based approaches) [8, 9] require human-labeled data as the GS terms aligning with the HDE interests in CDSS concepts. However, only HDE can provide labeled data. Unsupervised algorithms [8,9] work with text similarity or semantic relatedness and do not need labels. With the growing corpus and increased contextual complexity, previous approaches do not align with our goal in identifying the terms using contextual awareness.

Although Transformer models [10, 11] have been quite popular in accomplishing such a task using the context information with attention, they are computationally intense and require labeled data to fine-tune or to adapt from the biomedical domain to the CDSS domain. Additionally, generating high-quality human-labeled data can be a challenge. Currently, 1.2% of the total data corpus from the CDSS literature is labeled (GS). We propose generating synthetic CDSS labels to bootstrap the machine learning (ML) model and later fine-tune it with GS labels. We will discuss how to solve the challenges of domain adaptation, which usually requires high-quality labels with this technique.

To avoid the above-mentioned challenges, inferior neural architectures (compared to the Transformer [10,11] and other [12,13] models) can help us identify the possible N-gram combination of tokens as a valid GS candidate term. For example, long short-term memory (LSTM)-based encoders [14] and the conditional random fields (CRF)-based decoder models [15] (a statistical modeling method for text pattern recognition, where current prediction is affected by neighbors). This encoder-decoder network

accommodates the customization of text features and various attention levels over the text while recognizing the candidate GS terms from the CDSS literature.

Bidirectional attention for LSTM enhances the prediction of KP [16], focusing more on contextual understanding. Our approach is based on the NLP architectures (attention-based BiLSTM-CRF model) presented by Zichao and Guohai et al. [17, 18] to create a hybrid approach by augmenting document-level attention layer, preserving its light-weighted heritage, and adding context awareness. It leverages the broader understanding of the underlying concepts in the text during KP Identification.

We describe harnessing the power of the newly augmented framework with a minimally labeled dataset for KP identification in our domain of interest, CDSS, and the challenges we faced. Furthermore, the main objectives of this paper are to describe the following:

- Identifying KPs with long-range contextual dependencies with a hierarchical attention-based encoder (Hier-Attn-BiLSTM) neural network architecture, incorporating document, word, and sentence-level attentions.
- Creating high-quality synthetic labels in CDSS to bootstrap an ML model with a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model in the biomedical domain.
- Harnessing and optimizing the fine-tuning process when the ML model is limited by labeled data in a semi-supervised approach.

The article will discuss the related research in Section 2; formulate the task and describe the method and architecture in Section 3; outline the CDSS dataset, training procedure, and the experiments, and results in Section 4; discuss the analysis and the challenges identified during this project in Section 5; and conclude in Section 6.

2. Background

2.1. CDSS Ontology

CDSS has been broadly adopted in healthcare settings due to its effectiveness in improving healthcare quality and adherence to prescriptions and other clinical orders [3,4]. CDSS is usually a part of an electronic health record system. Its rules are created by incorporating the clinical domain knowledge and contextual information, which affect its operational behavior. However, creating and maintaining these rules are tedious, resource-demanding, and challenging in resource-constrained settings.

Creating an ontology facilitates the interoperability of CDSS rules [4,7]. Traditionally, ontology construction is an expert-driven manual process. As the field and science constantly evolve, identifying new terms automatically and promptly is critical to building a comprehensive ontology. This work contributes to the efficiency of ontology construction by automatically identifying KP within the process.

2.2. Similarity with other NLP problems

The NLP-based ML approaches deal with unstructured text data sources to extract structured information, understand patterns, and identify the KP. Identifying a KP involves: (1) extracting N-grams, limiting to noun phrases only, and (2) ranking the N-grams to find the best and mark them as KP. Some of the popular methods of KP extraction, as given by Zhiyong He et al. [8] will be summarized in the forthcoming sections. The differences between the problem we focus on and others are discussed.

2.2.1. Statistical and unsupervised methods

For limited labeled data, ML methods involving no labeled data, statistical, or unsupervised, would be an ideal use case, as proposed by Kazim et al. [9]. Some statistical features, such as term frequency-inverse document frequency (TF-IDF) [19, 20] and Best Match 25 (BM25) [21], differentiate the candidate terms into good or bad categories. But they fail to deal with the unseen data distribution, as the statistics are drawn from the existing corpus.

In unsupervised methods, the KPs are determined using semantic similarity, assuming that the more essential candidates cover all the important topics of the document. A graph is created using the KP as nodes and their semantic similarity as the relations. Such graphs may be used by ranking algorithms such as Google's PageRank [22], MultiPartiteRank [23], PositionRank [24], and TopicRank [25] algorithms to retrieve the KP by scoring the terms across the relations drawn. However, the relation is given by the similarity between N-gram tokens, and it does not consider the document's context to truly identify a KP.

2.2.2. Supervised methods

Considering the KP identification as a classification task, labeled data helps the ML model align the predictions toward human interest. It can be coupled with hand-crafted features to hold up to improve term identification, to classify a term as either KP or non-KP, recasting it as a binary classifier. Furthermore, the popular choices among supervised algorithms are Naïve Bayes [26], Decision Trees [27], and Support Vector Machines (SVM) [28], which can be used to solve binary classification. As the KPs are not independent entities and are always an N-gram combination, they create chaos in the conceptual formulation of the problem.

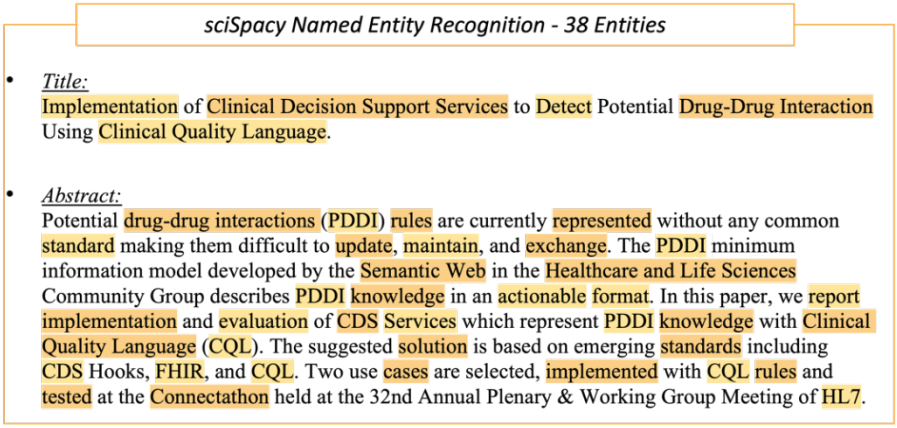
Using ranking, and marking the top N entities as the KP, Witten Ian et al. [29] developed a popular approach, Keyword Extraction Algorithm, which uses statistical features like TF-IDF and Word's First Occurrence Position (WFOP). Chengzhi Zhang et al. [30] included additional features such as the length of the token and linguistic features such as Part of Speech (POS) [31] tags to normalize the position and occurrence of the KP. A linear ranking SVM was used to rank the KP [32]. The BiLSTM-CRF model [33] considers it a sequence tagging problem and extracts the KP with superior performance [34]. However, the direct implementation of supervised methods does not solve the problem of labeled data.

2.2.3. Named Entity Recognition (NER)

Any information extraction and retrieval sub-task classifying the N-gram entities into predefined categories, such as name, drug, gene, disease, organization, quantity, numeric values, location, and data, is known as NER [35]. It can be handled as two problems: entity identification, and entity classification, like KP extraction. The identification phase is N-gram segmentation, where the N-gram can be the sequence of tokens. The classification phase is like creating categories in which each entity can be distinct and exist independently.

NER can be achieved by grammar-based (hand-crafted rules, defined entity structures, requires manual intervention) or statistics-based (a comprehensive set of rules derived from large-labeled dataset) methods. NER and KP Identification are subsets of text annotation which need an annotated corpus and are mutually exclusive in their objective functions. Although NER is based on a contextual understanding of the text, it is often comprehended by confidence in classifying an entity into one of the predefined categories.

As shown in Figure 1, the entities identified by NER are mostly nouns. Other grammatical entities are disregarded. Objectively, it differs from KP identification, which includes an N-gram combination of all grammatical entities based on context. Although the document understanding is homogenous in both, the identified entities would be different. The methods have fundamental differences, but it is very easy to see KP identification erroneously as similar to traditional NER.



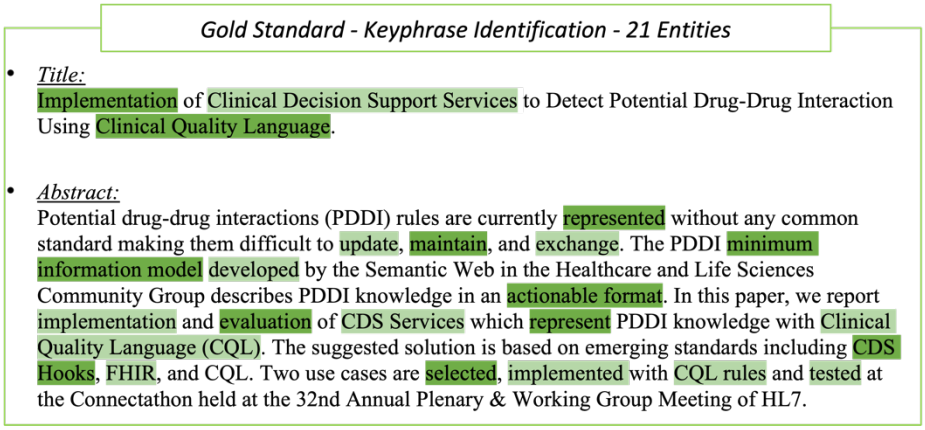


Figure 1. Entities identified on sample CDSS abstract demonstrate differences between human labeled Gold Standards and NER by sciSpacy.

2.3. Domain adaptation

Entity recognition or KP identification is a fundamental task for NLP applications, such as building an ontology, recommendation systems, and knowledge graph construction. Training such deep learning models is challenging and needs significant resources, time, and data. Domain adaptation [36,37] can help to adapt the pre-trained ML models in the parent/similar domain, fine-tuning them to the sub-domain with minimal use of labeled data.

A popular model in entity identification, Spacy [38], based on BERT [10,11], is trained on OntoNotes5 [39] and WordNet [40] open-source corpora. Although it works very well with English language modeling and text comprehension, it fails to identify the entities in biomedical and clinical informatics sub-domains). Therefore, Spacy, a large language model (LLM), has been domain-adapted with a 785 K vocabulary and 600-word vectors into sciSpacy, specializing in identification of biomedical entities [42]. We further fine-tuned the sciSpacy model to the CDSS sub-domain and strengthened the transfer learning approaches to achieve our goals.

2.4. Language Model (LM)

The LM is a critical aspect of NLP architectures [43]. It is a statistical and probabilistic technique to determine the conditional probability of each word's occurrence in a given sentence based on the hidden Markov model (HMM) [44]. It is often used when the labeled training data is limited.

To create such an LM, all the sentences in the document are unified into one, and punctuation is removed. Then, we slide over the word windows to train the LM without using labeled data to understand the context of the words and their characteristics. The ML model needs to understand domain-specific language and the distribution of words, the CDSS domain in our case. We can use this trained LM to transfer its neural network parameters to the actual model, helping it learn the language distribution for the CDSS domain [45].

3. Methods

As shown in Figure 1, our task is similar to an entity or sequence labeling but clearly diverges from NER. It should be able to identify concepts new to the CDSS ontology based on contextual understanding of the text. Our task can be implemented using Recurrent Neural Networks (RNN) with additional text features (TF-IDF, Length of token, POS tags, WFOF). In general, a bidirectional long short-term memory (BiLSTM) with these features acts as the encoder with a CRF layer as the decoder. It can learn the N-gram entity patterns and their occurrence over the sequence of words with context over the current sentence [16, 17].

Word embeddings (WE) play a significant role in transforming text information into mathematical representation to provide input data for deep learning models. We propose a hierarchical attention-driven context added to each word to improve the inference and learn a variety of text patterns with minimum labels to bridge the gap of contextual understanding for word representations. The details are presented in the following sections.

3.1. Overview

3.1.1. Defining the task

KP identification is a typical sequence labeling task where we find the N-gram KP from the document. For a document with m sentences, $d = (s_1, s_2, \dots, s_m)$. Each sentence containing n tokens or words, $s_i = (w_{i1}, w_{i2}, \dots, w_{in})$ is the input to the model and output $z_i = (z_{i1}, z_{i2}, \dots, z_{in})$ would be a sequence of tags in BIO token tagging representation.

In BIO token tagging [46], the first N-gram phrase word is labeled B-KP, the rest are labeled I-KP, and the non-KP tokens are marked as O. Figure 2 presents a BIO token tagging example with an input document. The model can output the sequence tag (B-KP/I-KP/O) where the keyphrases can be generated by decoding the output tags (electronic patient records, implicit source, clinical behavior, problem-solving knowledge).

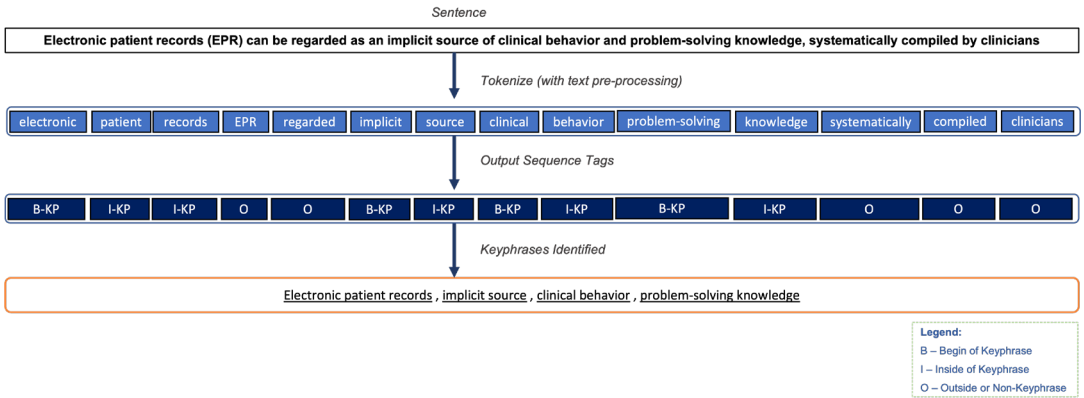


Figure 2. Flow of labeling Keyphrases (KP) from a sentence

We provide the document-level context by combining hierarchical attention (i.e., adding word-level and sentence-level attentions in a hierarchical fashion to create the document vector), improving the performance. Thus, all the sentences in the form of embeddings and their corresponding attentions are used to complement the

understanding of the current sentence. That is, the input to our model will be all the sentences from a single document, and for each sentence, we find its relevance compared to other sentences and their words to calculate hierarchical attention in understanding the context.

3.1.2. High-level design

Our approach to ML model architecture includes creating synthetic labels for unlabeled data, pre-training, BiLSTM-CRF model, and fine-tuning with HDE (GS) labeled data as illustrated in Figure 3. Based on the research of Guohai et al. [18] and Saad et al. [47], first, we train the word embedding model and bidirectional language model (BiLM) using unlabeled data. Then we transfer their knowledge into the actual model's initial layers for embedding and LSTM, respectively. Second, all the sentences from a single document are fed into the model in batches, one document at a time. Each word in the sentence is transformed into a vector with the WE model. Then, we introduce the abstraction of hierarchical attention, attention at word and sentence levels, to aggregate them into sentence and document vectors, respectively.

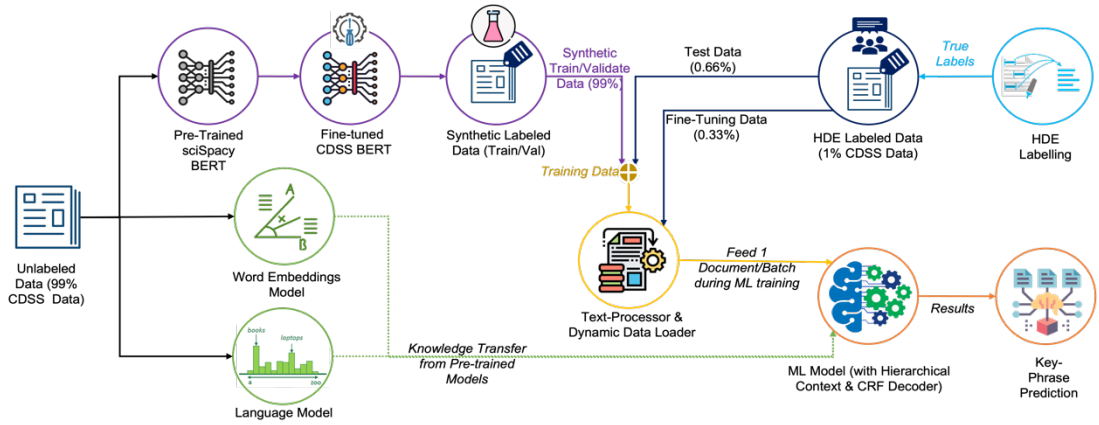


Figure 3. High-level design of the proposed method

Using these embedding and attention vectors, we calculate the hierarchical attention for any given word using the second LSTM, which is further deduced into the final LSTM network along with the outputs of the first LSTM (BiLM) network completely, encoding one document at-a-time. Lastly, we use the encoded information to feed the conditional random fields (CRF) layer which decodes the best probable sequence decisions to mark the output labels with BIO token tag representations which are used to group the tokens and identify the KP. Later, we fine-tune the model to enhance its performance using a minimally labeled dataset.

3.2. Synthetic labels

Due to the shortage of labeled datasets, domain-adapted or fine-tuned models can be used to generate the synthetic labels which helps us to bootstrap the ML model's training process as shown in Figure 3. Later, the BiLSTM-CRF model can be fine-tuned

with the HDE labels to identify the actual keyphrases, avoiding the exhaust of HDE generated labels.

To achieve this, we performed *domain adaptation* of a sciSpacy BERT model [42] by generating the KPs (intermediate) on the CDSS dataset and using them to fine-tune the sciSpacy BERT to the CDSS sub-domain. Then, we generated the **KP (synthetic)** on the CDSS dataset and marked the labels in the BIO format on the texts without HDE labels, namely the synthetic dataset with labels for the CDSS sub-domain. We used this dataset to train and test our BiLSTM-CRF model.

3.3. Pre-training

3.3.1. Word Embedding (WE) model

A WE is a mathematical vector representation of a given word, which ensures minimal distance between the vectors with words of similar meaning. These embeddings capture the language semantics and syntactic information using the Word2Vec [48] *skip-gram* approach and are used as input to train deep learning models. We have also experimented with creating fastText [49, 50] and GloVe [51] embeddings as alternative embedding models to compare their performances in our approach.

While Word2Vec and GloVe work on semantic similarity between vectors, the latter uses word-pair co-occurrence with reduced dimensions and works faster on large volumes of data. Conversely, fastText considers sub-words to generalize unseen vocabulary, and works using syntactic analogy. Due to the overhead of character N-grams in constructing sub-word information, it is slower than others. Also, both GloVe and fastText have dynamic and reduced dimensions in the embedding matrix, creating a hiatus for knowledge transfer between them and BiLSTM-CRF. Therefore, we only focused on creating a Word2Vec embedding model for our approach.

3.3.2. Bi-Directional Language Modeling (BiLM)

To learn the probability distribution over sequences of words, we use a shallow layered bidirectional RNN [52] (e.g., LSTM and GRU) to learn the joint probabilities represented by WE. To ensure the network learns such a distribution, we evaluate its perplexity as a metric. A network that learns the word distribution is known as the BiLM [43]. It computes the conditional probability of occurrence of the next word (w_i) based on the previous ($w_1, \dots, w_{i-1}$) and future words ($w_{i+1}, \dots, w_n$) in a sentence (s) as shown in Eq. (A. 1), (A. 2) [43], where each sentence (s) is represented by the last word's context (given by LSTM's cell state) in both left ($\overleftarrow{c}_n^{LM}$) and right ($\overrightarrow{c}_n^{LM}$) directions. Here Eq. (A. 2) is the probability of LM in the reversal order when compared with the Eq. (A. 1).

$$p(w_1, w_2, \dots, w_n) = p(w_2|w_1) \dots p(w_n|w_{n-1}) = \prod_{i=2}^{n} p(w_i|w_1, w_2, \dots, w_{i-1}) \quad (A.\ 1)$$

$$p(w_n, w_{n-1}, \dots, w_1) = p(w_{n-1}|w_n) \dots p(w_1|w_2) = \prod_{i=n-1}^{1} p(w_i|w_n, w_{n-1}, \dots, w_{i+1}) \quad (A.\ 2)$$

$$s = [\overleftarrow{c}_n^{LM}; \overrightarrow{c}_1^{LM}] \quad (A.\ 3)$$

For a given word (w_i), the forward and backward LSTMs encode the history of previous tokens in each direction into fixed dimensional vectors ($\overleftarrow{h}_{i-1}^{LM}$, $\overrightarrow{h}_{i-1}^{LM}$), where a soft-max layer maximizes the likelihood (p) of the word (w_i) in the given sentence (s) in the corpus. After training, a BiLM can represent a sentence of a document by concatenating the last cell (i.e., the last word of the sentence) state carrying the context in either direction to represent the input sentence as shown in Eq. (A. 3).

3.4. Hierarchical-Attention-BiLSTM-CRF Model

3.4.1. Encoder

This architecture is adopted from Zichao Yang, Guohai Xu, and Luo L et al. (Figure 4) [17, 18, 53]. To construct the context of the text, we encode one document at-a-time to capture document-level context with a stacked BiLSTM [11]. Here, the rudimentary layers of stacked BiLSTM are initiated with a transfer strategy from pre-trained WE and BiLM models' weights.

The embedding and first LSTM layers in our encoder share the architecture of the pre-trained models, and can seamlessly *transfer the model parameters or weights* between the models [18]. Using the transfer strategy, our model can efficiently initialize and understand the CDSS-domain language distribution. It enables the model to focus on learning to identify the KP from a text rather than understanding the language.

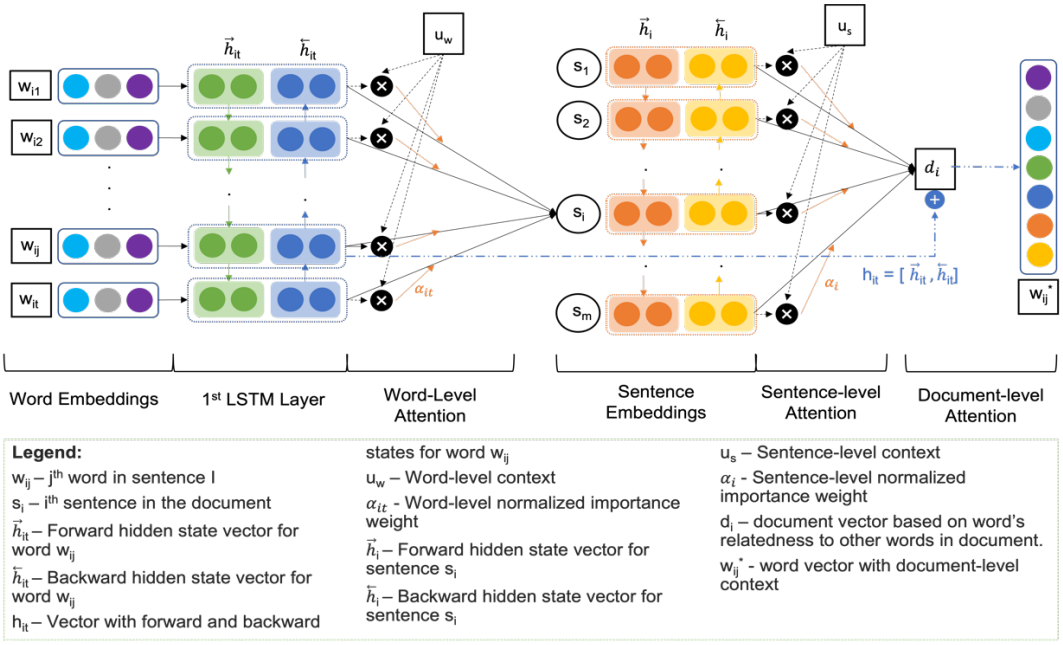


Figure 4. Detailed encoder - word encoding with Hierarchical-Attention-BiLSTM with document-level context.

We use all the sentences in a document, $d = (s_1, s_2, \dots, s_m)$, where each sentence is represented by $s_i = (w_{i1}, w_{i2}, \dots, w_{in})$ and its words by $w_{it} \forall t \in [1, n]$. We embed the words into a vector (x_{it}) through an *embedding matrix* (W_e). BiLSTM summarizes the bidirectional context information as shown in Eq. (B. 1)(B. 2)(B. 3) where each word vector's hidden state (h_{it}) is obtained by concatenating the forward ($\overleftarrow{h}_{it}$) and

backward ($\overrightarrow{h}_{it}$) hidden state vectors, i.e., $h_{it} = [\overleftarrow{h}_{it}; \overrightarrow{h}_{it}]$. The hidden state vector provides sentence-level context to each word [12].

$$x_{it} = W_e . w_{it} \forall t \in [1, n] \quad (B.\ 1)$$

$$\overrightarrow{h}_{it} = \overrightarrow{LSTM}(x_{it}) \forall t \in [1, n] \quad (B.\ 2)$$

$$\overleftarrow{h}_{it} = \overleftarrow{LSTM}(x_{it}) \forall t \in [1, n] \quad (B.\ 3)$$

We calculate word similarity (u_{it}) using a neural network's parameter for weighted matrix (W_w) and word representation (h_{it}) given by BiLSTM along with bias (b_w) [54]. Then we calculate the **word-level attention** by aggregating the h_{it} and u_{it} using a word-level context vector (u_w) [19, 20] to get a word-level normalized importance weight (α_{it}). Finally, we compute the sentence vector (s_i) as a weighted sum of word representations as shown in Eq. (C. 1)(C. 2)(C. 3). Initially, u_w is the neural network parameter with random initialization, learned during the training process.

$$u_{it} = \tanh(W_w . h_{it} + b_w) \forall t \in [1, n] \quad (C.\ 1)$$

$$\alpha_{it} = softmax(u_{it}^T . u_w) = \frac{\exp(u_{it}^T . u_w)}{\sum_t \exp(u_{it}^T . u_w)} \forall t \in [1, n] \quad (C.\ 2)$$

$$s_i = \sum_t \alpha_{it} . h_{it} \forall t \in [1, n] \quad (C.\ 3)$$

Similarly, a document vector can be computed using **sentence-level attention** over the sentence vectors (s_i) [19, 20] using a second BiLSTM network and thereby concatenating the forward ($\overleftarrow{h}_i$) and backward ($\overrightarrow{h}_i$) states to encode a sentence, $h_i = [\overleftarrow{h}_i; \overrightarrow{h}_i]$ based on neighbor sentences as shown in Eq. (D. 1)(D. 2).

$$\overrightarrow{h}_i = \overrightarrow{LSTM}(s_i) \forall i \in [1, m] \quad (D.\ 1)$$

$$\overleftarrow{h}_i = \overleftarrow{LSTM}(s_i) \forall i \in [1, m] \quad (D.\ 2)$$

As shown in Eq. (E. 1)(E. 2)(E. 3), to estimate the sentence-level context vector (u_s), first, we use neural network parameter for weighted matrix (W_s), sentence representation (h_i) and bias (b_s) to calculate sentence similarity (u_i). Second, we randomly initialize u_s, to calculate the sentence-level normalized importance weight (α_i), which yields a **document vector(d_i)** for each word representing the sentences that are important to consider for a given word while identifying it as a KP as provided [17].

$$u_i = \tanh(W_s . h_i + b_s) \forall i \in [1, m] \quad (E.\ 1)$$

$$\alpha_i = softmax\left(u_i^T . u_s\right) = \frac{\exp\left(u_i^T . u_s\right)}{\sum_i \exp\left(u_i^T . u_s\right)} \forall i \in [1, m] \quad (E.\ 2)$$

$$d_i = \alpha_i . h_i \forall i \in [1, m] \quad (E.\ 3)$$

Unlike the previous work as proposed by Guohai Xu et al. [18], we then concatenate the first LSTM's hidden local state (h_{it}) with the document vector (d_i) into a new vector $[h_{it}; d_i] \forall t \in [1, n]$, given the word's relatedness to other words in the document. That is, providing document-level context to each word. Next, the extended representation will be further used by the final LSTM layer to identify the labels.

3.4.2. Decoder

As described by Ling Luo et al. [53], we use the CRF [15] layer as the decoder producing the confidence scores for the words with each label (B-KP/ I-KP/ O) as the output score of the decoder. Given the transition and network scores, we make tagging decisions independently, considering P, the matrix of scores of the network output.

The score of sentence (s_i), with a sequence of predictions $y_i = \left(y_{i1}, \ldots, y_{ij}, \ldots, y_{it}\right)$, is given by the sum of transition scores and network scores as shown in Eq.(F. 1). Here each $P_{a,b}$ represents the matrix of scores of b^{th} tag of the a^{th} word in the sentence. Furthermore, the tagging transformation matrix (T) is trained as the model parameter. Here $T_{a,b}$ represents the transition score from tag a to tag b through successive words where $T_{0,b}$ is the initial score starting from tag b.

To yield the conditional probability of the path (y), we normalize the score for all possible paths ($\tilde{y}$) using a soft-max function using Eq. (F. 2). Then, we maximize the log probability of valid tag sequences. We obtain the maximum score using the dynamic programming approach of Viterbi decoding [55] for the best tag path given by Eq. (F. 3).

$$score(s_i, y_i) = \sum_{k=0}^{n} \left(T_{y(i,k-1),y(i,k)} + P_{(k,y(i,k))}\right) \quad \forall i \in [m, 1] \quad (F.\ 1)$$

$$p(y_i|s_i) = \frac{\exp\left(score(s_i, y_i)\right)}{\sum_{\tilde{y}_i} score(s_i, \tilde{y}_i)} \quad \forall i \in [m, 1] \quad (F.\ 2)$$

$$z_i = \underset{\tilde{y}_i}{argmax}\left(score(s_i, \tilde{y}_i)\right) \ \forall i \in [m, 1] \quad (F.\ 3)$$

4. Experiments and results

4.1. Dataset

The text corpus was obtained from PubMed by filtering the CDSS literature in the MEDLINE format. The articles with a valid PubMed Identifier (PMID) for a unique article were retrieved and we retained the articles filtered using the CDSS as the Medical Subject Headings (MeSH) term (Tables 1 and 2). Appendix A shows detailed information on the dataset at the various levels of text preprocessing.

Of the total dataset retrieved from PubMed (3545 articles), 3326 articles were left after XML parsing. There contained some articles from the GS dataset. During the preprocessing, we removed the articles with abstracts of three or fewer sentences, treating the title as one sentence.

Table 1. Details of the CDSS dataset

	FC	FC with PMIDs	GS (+8 ACM)	No/little abs.	Final/total dataset
Articles	3545	3326	133	99	3281

FC – Full CDSS dataset

No/little abs. - Abstracts having less than 3 sentences including the title.

Table 2. CDSS Dataset: train, validate, and test

	Total	Unlabeled (with Synthetic KP)	Train	Validate	Test (GS91)*	GS(GS42)*
Articles	3281	3148	1049	2099	91	42

* GS91 and GS42 are 2 sets of HDE-labeled datasets.

In addition, we had 2 sets of human-labeled datasets (GS91 and GS42). One set, GS91, was unseen data to compare with the model's performance, and the other set, GS42, was later used to fine-tune the ML model. Cohen's kappa rates for the first 42 (GS42) abstracts were 0.93 (annotators 1 and 2) and 0.73 (annotators 1 and 3) [41]. For the second set of abstracts (GS91), Cohen's kappa rates were 0.87 (annotators 1 and 2) and 0.97 (annotators 1 and 3).

There were 3148 unlabeled articles remaining, and we created synthetic KP and marked the labels to create a synthetic labeled dataset for the CDSS domain with a 1:2 train-validation split.

4.2. Synthetic label creation

To maximize the quality of the synthetic labels created, we experimented with different unsupervised algorithms (namely, PositionRank, MultiPartiteRank, and TopicRank) and NER (i.e., sciSpacy). They identified the KP from a given text and we compared their performance with the manual labels. As shown in Table 3, BERT-based sciSpacy [42] outperformed other unsupervised methods. Further, we fine-tuned the sciSpacy model over the CDSS domain to enhance the quality of the generated synthetic labels.

Table 3. Evaluation of synthetic KP generated with different approaches

Approach	Accuracy	Misclassification	Precision	Recall	Specificity	F1-Score
sciSpacy	0.69	0.31	0.36	**0.81**	0.66	**0.50**
PositionRank	0.76	0.23	**0.39**	0.36	0.86	0.38
MultiPartiteRank	0.76	0.24	0.38	0.36	0.86	0.37
TopicRank	0.77	0.23	**0.39**	0.36	**0.87**	0.37

4.3. Preparation

4.3.1. CDSS domain adaptation

To optimize the sciSpacy model for identifying relevant entities in the CDSS context, we performed domain adaptation [36,37] for the sciSpacy BERT model. We adapted it from biomedical entities to CDSS entities by fine-tuning it in a semi-supervised approach as proposed by Syed et al. [56]. We generated synthetic labels from the base sciSpacy models on the unlabeled CDSS corpus and thereafter used them to adapt the model iteratively over the CDSS domain.

To compare the quality of labels generated from the fine-tuned model, we tested fine-tuning at different levels and dataset combinations to find the best (Table 4 and Figure 8). We found that Level 1 fine-tuning of the sciSpacy model with the synthetic dataset yielded better results (Figure 5) than the base sciSpacy model's prediction shown in Figure 1, and further fine-tuning overfitted the model's predictions. Appendix B contains consolidated pictures of predictions Figures 1, 5, and 7 to aid understanding.

Table 4. Evaluation of fine-tuning sciSpacy model for CDSS

Fine-Tune	Base	Model	Train Dataset	GS Dataset	Precision	Recall	Accuracy	F1-Score
Level 0	sciSpacy	sciSpacy (en_core_sci_lg)	PubMed	42	0.61	0.18	0.93	0.27
				91	0.59	0.23	0.97	0.33
				133	0.62	0.22	0.96	0.33
Level 1	**sciSpacy**	**cdssSciSpacy**	Synthetic CDSS (1866 Train / 622 Val)	42	0.70	0.38	0.97	0.5
				91	0.73	**0.64**	**0.99**	**0.68**
				133	**0.74**	0.59	**0.99**	0.66
Level 2	cdssSciSpacy	cdssSciSpacy GS42	42 GS (33 Train / 9 Val)	91	0.57	**0.64**	**0.99**	0.60
Level 2	cdssSciSpacy	cdssSciSpacy GS91	91 GS (72 Train / 19 Val)	42	0.66	0.38	0.97	0.48
Level 2	sciSpacy	sciSpacy GS42	42 GS (33 Train / 9 Val)	91	0.57	0.54	**0.99**	0.55
Level 2	cdssSciSpacy	cdssSciSpacy GS66 [1]	66 GS (52 Train / 14 Val)	67	0.63	0.62	**0.99**	0.62

[1]Repeated experiment 50 times on random samples of GS 133.

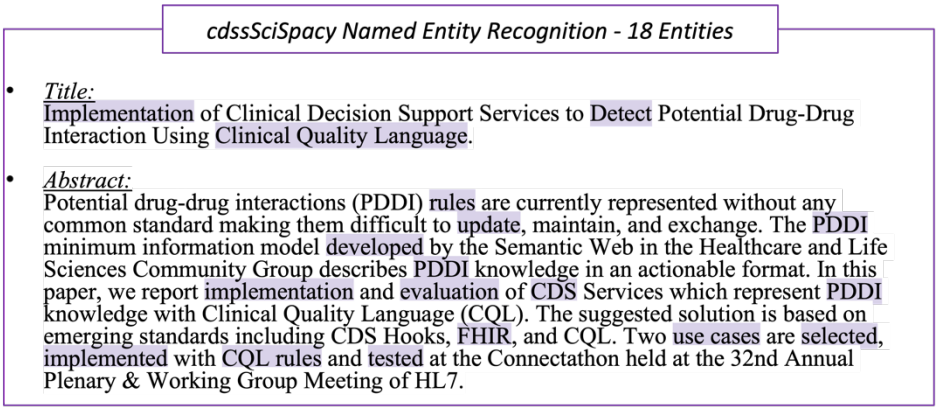


Figure 5. Entities identified on sample CDSS article context with CDSS-domain adapted sciSpacy NER.

4.3.2. Token tagging representation

To identify an N-gram sequence, we used token tagging representation where each token in the text is marked with either the BIO or BILOU encoding schema [46] to represent the KP chunks (Figure 2). We tested both schemas to determine which fit the CDSS corpus best, and the results are shown in Table 5. Both had similar performance metrics; therefore, we chose the BIO token tagging for the label marking as it slightly outperformed BILOU in F1-Scores.

Table 5. Entity-level metric evaluation - token tagging

Encoding Schema	Dataset	Precision	Recall	Accuracy	F1-Score
BIO	Validation Dataset (Synthetic) Labels	0.75	**0.68**	**0.92**	**0.71**
	GS42 Labels	0.60	**0.50**	**0.88**	**0.54**
	GS91 Labels	0.61	**0.50**	**0.88**	**0.55**
BILOU	Validation Dataset (Synthetic) Labels	**0.76**	0.60	0.92	0.69
	GS42 Labels	**0.60**	0.41	0.87	0.49
	GS91 Labels	**0.65**	0.42	0.86	0.51

4.3.3. Stemming vs. non-stemming

Stemming the vocabulary is a normalization technique used for text pre-processing before feeding it to ML models. It represents the morphological structure of the language and although it benefits the document indexing, it can sometimes worsen the topic understanding [57]. To analyze the effect of stemming on CDSS corpora, we experimented with the performance of KP identification on stemmed and non-stemmed KP on the synthetic and GS-labeled data (Tables 6, 7). The performance of the ML models deteriorated with stemming, so we opted for non-stemming in the text preprocessing steps.

Table 6. Comparison of stemming evaluation on the validation dataset (Synthetic)

Metrics	Validation Data Labels

	Non-Stemming			Stemming		
	B-KP	I-KP	O	B-KP	I-KP	O
Accuracy	0.87	0.92	0.92	0.85	0.91	0.90
Misclassification	0.13	0.09	0.08	0.16	0.09	0.10
Precision	0.85	0.80	0.91	0.83	0.74	0.87
Recall	0.92	0.76	0.81	0.92	0.55	0.77
Specificity	0.83	0.96	0.96	0.74	0.97	0.95
F1-Score	**0.88**	**0.78**	**0.86**	0.87	0.63	0.82

Table 7. Comparison of stemming evaluation on the GS42 Dataset

Metrics	GS42 Labels					
	Non-Stemming			Stemming		
	B-KP	I-KP	O	B-KP	I-KP	O
Accuracy	0.57	0.86	0.51	0.52	0.86	0.53
Misclassification	0.44	0.15	0.49	0.48	0.14	0.47
Precision	0.30	0.40	0.83	0.32	0.41	0.81
Recall	0.74	0.67	0.35	0.80	0.36	0.33
Specificity	0.52	0.88	0.85	0.42	0.93	0.47
F1-Score	0.43	**0.50**	**0.49**	**0.45**	0.38	0.47

4.3.4. Loading pre-trained models

As discussed in Section 3.4.1, initially a WE model and the BiLM were trained separately on the unlabeled corpus (Figures 3 and 4). Then, we transferred the parameter weights into the BiLSTM encoder's initial layers, to bootstrap the CDSS language distribution before we started training the BiLSTM-CRF model.

4.3.5. Training

After obtaining synthetic labels generated from the best performing domain-adapted model (see Sections 4.2 and 4.3.1), we labeled the KP with the BIO token tagging schema [46] to start the ML model training procedure for 30 epochs. Then, we evaluated the sequence-level entity metrics using standard ML metrics (i.e., precision, recall/sensitivity, F1-Score, and accuracy). The parameters and configurations of the Hier-Attn-BiLSTM-CRF neural network model are as follows:

- WE Dimension: 300
- LSTM hidden layer dimension: 256
- Dropout Ratio: 0.2
- Epoch: 30 (number of times every document is shown to the ML model)
- Batch Size: 1 (one document at a time is shown to the model, to calculate the context, with documents having varying sentences numbers, up to a maximum of 52 for an abstract)
- Max sentence length: 128 (For CDSS corpus, the maximum words per sentence is 105)
- WE Type: Word2Vec
- Text pre-processing: remove stop words and punctuation.
- Stemming: no
- Train-validation split: 1:2
- Pre-trained sciSpacy BERT model: en_core_sci_lg

4.4. Evaluation

4.4.1. Leveraging document-level context

Understanding the context during KP identification is important for predicting relevant candidate terms from a given text [16, 58]. To reinforce this philosophy, we experimented with the different encoding combinations at word and character-level embeddings, and CNN-based text features (length, POS tag, text rank, TF-IDF score and Position of First Occurrence [59]). We compared them with our proposed method (BiLSTM-CRF with Hierarchical-Attention and sentence-level embedding working at the document-level context). The results are shown in Table 8, Figures 7 and 8. Our method, which included a hierarchical context-driven model, had better metrics than the base BiLSTM-CRF model and performed as well as the other models with character embedding and CNN-based text features, but with lesser recall values. Appendix B contains consolidated pictures of predictions (Figures 1, 5, and 7) for easier understanding.

Table 8. Comparison of evaluations on different contextual level attention

Model	Encoder Details	Experiment Runs	Train Dataset	Test Dataset	Precision	Recall	Accuracy	F1-Score
BiLSTM-CRF	BiLSTM (Word Embd's)	1	1049 Synthetic	2099 Synthetic	0.72	0.66	**0.92**	0.69
				42 GS	0.54	0.46	0.86	0.49
				91 GS	0.59	0.48	**0.88**	0.53
BiLSTM-CRF	BiLSTM (Word Embd's) + BiLSTM (Char Embd's)	1	1049 Synthetic	2099 Synthetic	0.70	0.70	0.85	0.70
				42 GS	0.52	**0.56**	0.78	0.54
				91 GS	0.58	0.53	0.77	**0.55**
BiLSTM-CRF	BiLSTM (Word Embd's) + BiLSTM (Char Embd's) + CNN (Text Features)	1	1049 Synthetic	2099 Synthetic	0.73	**0.71**	0.85	**0.72**
				42 GS	0.56	0.55	0.78	0.55
				91 GS	0.58	**0.55**	0.78	0.57
Hier-Attn-BiLSTM-CRF (our method)	BiLSTM (Word Embd's) + Hierarchical Context (Word-level sentence-level attentions)	1	1049 Synthetic	2099 Synthetic	**0.75**	0.68	**0.92**	**0.71**
				42 GS	**0.6**	0.5	**0.88**	0.54
				91 GS	**0.61**	0.5	**0.88**	**0.55**

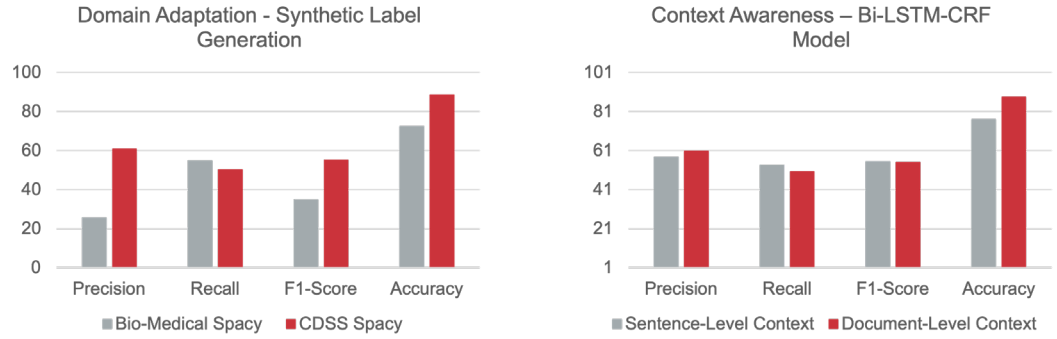


Figure 6. Comparison of results for domain adaptation and hierarchical context (document context through word-level and sentence-level attention).

Our Proposal: Hier-Attn-BiLSTM-CRF Model - 18 Entities

- *Title:*
  Implementation of Clinical Decision Support Services to Detect Potential Drug-Drug Interaction Using Clinical Quality Language.
- *Abstract:*
  Potential drug-drug interactions (PDDI) rules are currently represented without any common standard making them difficult to update, maintain, and exchange. The PDDI minimum information model developed by the Semantic Web in the Healthcare and Life Sciences Community Group describes PDDI knowledge in an actionable format. In this paper, we report implementation and evaluation of CDS Services which represent PDDI knowledge with Clinical Quality Language (CQL). The suggested solution is based on emerging standards including CDS Hooks, FHIR, and CQL. Two use cases are selected, implemented with CQL rules and tested at the Connectathon held at the 32nd Annual Plenary & Working Group Meeting of HL7.

Figure 7. Entities identified on sample CDSS article context with our Hier-Attn-BiLSTM-CRF model.

4.4.2. Fine-tuning with Gold Standard (GS) labels

As discussed previously, we harnessed the semi-supervised learning approach and further fine-tuned the Hierarchical-Attention based BiLSTM-CRF model to strengthen its predictions [56]. The experiment included adding the truly labeled samples to synthetic labeled samples in different proportions, i.e., 0, 2, 4, 6, and 8 GS labeled documents are sampled for every batch of 100 synthetic labeled documents, respectively. It helps us measure the learning performance with human feedback over the ML training iterations, by running independent experiments 10 and 50 times. As shown in Figure 8, exposing 2–4 true or GS labeled samples to 100 synthetic samples enabled the model to learn more efficiently from the minimum labeled dataset. The tabulated performance metrics are presented in Appendices C and D.

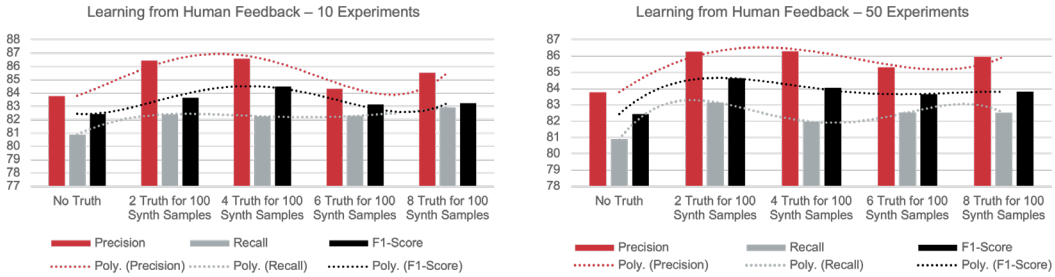


Figure 8. Results for fine-tuning with Gold Standard (GS) labels.

5. Discussion

Our work demonstrates an ML model, which can be used to identify KPs by leveraging limited expert-labeled data, a common challenge in clinical NLP. Identifying the higher significance KP in summarizing text is a different task from ours. KPs identified in the literature are reviewed by human experts before they can be added to the CDSS ontology.

Even though it appears similar on the surface, it is completely different underneath because the KPs are not a fixed set of terms that can be objectively marked as correct. Despite its importance to text understanding, spanning the best coverage, and relevance to the given text as shown in Figure 1, Appendix B, manual labeling is expensive. However, automating human understanding of the text is challenging, especially in specialized fields, such as medicine.

Although various ML approaches have evolved to solve some of the common challenges, they fail to understand the context behind the annotation because of the long-range dependencies of natural language. To solve this problem, we used a semi-supervised approach with hierarchical attention over text to provide a larger, but still focused, context (one document) to the model while working with a word.

5.1. Results interpretation

The lack of labeled data is one of the key challenges that this project aimed to overcome. The task can be presented as a simple yes or no labeling of a sequence of words, but identification is more complicated than a binary task. The HDE uses rich background knowledge to make judgments and understands the context of the domain. Manual curation of entities and marking labels are laborious and expensive. We used a semi-supervised approach to generate synthetic labels and trained our BiLSTM-CRF model with them.

Assessing the quality of the generated synthetic labels is crucial because it impacts the BiLSTM-CRF model's learning. Therefore, we experimented with unsupervised ranking approaches [22–25] and the pre-trained spacy, sciSpacy models based on the Transformer neural architectures [10, 11]. We found that sciSpacy [42] (BERT model) outperformed the others in matching the synthetic labels to the GS candidate terms (Table 3).

Although the metrics shown in Table 3 (such as F1-Score ≤ 0.5) could be better, the models compared are trained for a different task as shown in Figure 1, Appendix B and the predicted labels seem to be imperfect. Due to the correlation of NER and KP Identification, the task might look similar; however, what we tried to achieve was different from the NER trained on the bio-medical corpus. Therefore, we decided to perform domain adaptation on the sciSpacy NER model using CDSS corpora to improve the quality of the generated synthetic labels.

We found that domain adaptation provided us with much better labels than the sciSpacy NER model, limiting the entities identified outside of the domain (Figure 5, Appendix B and Table 4). However, due to its initial training on a bio-medical corpus, the sciSpacy model inherently identifies entities irrelevant to the CDSS domain, and noisy labels were generated (Appendix B).

To examine it more closely, we performed supplementary fine-tuning with different combinations (42 GS, 91 GS, and 67 GS, different train-validate-test data subsets each time from 133 GS) of the labeled dataset, where each combination is newly trained at the respective fine-tuning level (Table 4). We understood that incremental fine-tuning introduced variance into the LM and increased perplexity, dropping its performance further as the fine-tuning levels grew. Therefore, we opted to use the CDSS domain-adapted sciSpacy model, which has 2-fold improvement in F1-Score from 0.33 to 0.66 on GS133 dataset (Table 4) and avoid other fine-tuned models.

Once we have labels for ML model training, we need to represent the token to the model efficiently. Therefore, we experimented with the BIO and BILOU encoding schemas' token tagging representations to compare their performances on the CDSS corpus. The BIO encoding schema performed slightly better (Table 5). Although standard approaches in NLP pre-processing include either stemming or lemmatization resulting in high performance, in our case, it reduced the topical understanding and inference of the ML model. We experimented with the ML model's performance on

stemmed vs. non-stemmed tokens, and the results (Tables 6, 7) aligned with our conceptual understanding.

Once the words are tokenized, we need embeddings to bind the token information to a vector to feed it to the model. Most WE models work with vocabulary from the existing text corpus and fail to handle Out of Vocabulary (OOV) words. To solve the OOV problem, we could use sub-word information with character N-grams using fastText. This reduces the length of the vocabulary as it remembers sub-word information. We did not use this, as the reduced dimensions of the matrices create conflict in transferring the weights between fastText and BiLSTM-CRF layers.

The reason for this irregularity in matrix dimensions is that the total vocabulary with Word2Vec is around 15.8 K, whereas fastText has only 4.7 K sub-words. Also, it only shows a 0.5%–1% improvement, as reported by Benedict et al. [60]. Therefore, we reverted to the older Word2Vec approach for pre-training the WE model as it is easier to transfer the embedding matrix weights between pre-trained and actual models. Our method uses index-to-token and token-to-index mapping while encoding the words. The length of the vocabulary(L) is further used as square matrix dimensions of the WE (W_e) matrix, which helps us find the similarity between any two words.

We then introduced the word-level attention mechanism, as not all words contribute equally to the meaning of the sentence. We aggregate the word representations to form a sentence vector, which enables us to further create a document vector for each word in the broader context of the document and its sentences. We conducted experiments on the different encoders (word-level attention, character-level attention, and text-based CNN) against our hierarchical attention-based encoder (Table 8) and a generic CRF decoder to all the models. Furthermore, we evaluated the performance of our model with input word representations bearing the document-level context to understand the long-range dependencies of the text (Table 8 and Figure 6). Although the metrics are on-par with the other models, our model had no hand-crafted features except the pre-training for WE and LM. Therefore, we can infer that our model can understands the CDSS language distribution effectively while identifying the KPs, compared with GS labels identified (Appendix B).

While the visual representation for KPs identified on a sample text (Figure 7, Appendix B) looks closer to the GS labels (Figure 1, Appendix B), most of the entities spanning N-grams do not have an exact match. Although it looks convincing, the complexity of the evaluation will be difficult and will no longer work at the token level. Nevertheless, it needs to be evaluated on the entity level with an exact GS label match, as proposed by Nancy et al. [61]. Therefore, we used the sequence labeling evaluation given by Hiroki et al. [62] to decipher the results. As shown in Figure 6, our model with **hierarchical attention improved accuracy** by **10%,** from 78% to 88%, compared to the sentence-level attention model. The complete match for GS labels brings down the metrics. Without any character-level and textual features, the model's results (Table 8) infer an improvement in the overall performance metrics due to the added hierarchical context provided to each word representation. In particular, we noted that it has better precision than the remaining models and allowed us to maintain the F1-Score ($55\% \pm 2\%$) even with the decline in recall values (Table 8).

The hierarchical context requires all the sentences of a single document at-a-time to calculate the attention for words and sentences and to create sentence-level and document-level vectors, respectively. Usually, a static batch-size ($32/64/128$) is chosen for the data-loader, which yields sentences from different documents grouped together

as a batch. Therefore, the sentences of a document could span different batches, which creates complexity in calculating hierarchical attention. To overcome this, we used the non-conventional technique of dynamic batches for the data-loader, i.e., each time a document with a different number of sentences was sent into the encoder-decoder during the ML model training process. This means that the number of iterations for the ML model training equaled the number of documents shown, ultimately increasing the training time, making it 2–3 times slower than models without a hierarchical context.

To further strengthen the ML model, we fine-tuned it with the GS labels to align the model's predictions from synthetic to GS labeling. To evaluate the model's performance after training, we reserved 91 GS dataset as unseen data and only fine-tuned the model with 42 GS labeled documents from the CDSS corpus. During the training, we varied the true labeled documents shown (0/2/4/6/8 GS) for every 100 labeled documents during the model's iteration, marking the essence of minimal true labels shown. A poly-fit curve over the scores demonstrates that showing 2–4 true samples for 100 synthetic samples during ML model training demonstrates better performance. The **F1-Score** improved **from 55% to 84%**, **accuracy from 88% to 96%, precision from 61% to 86% and recall from 50% to 82%**. The results guide us to optimize our model and settings for the operation, and we hope the results can be a reference point for others in planning their NLP tasks.

5.2. Challenges

The generation of manually labeled data is an expert-intensive process in fields like medicine. To surmount this problem, HDE can label a small set of samples. During the HDE labeling process, it is important to pick the samples from different areas of the CDSS sub-domain, which helps the model to learn efficiently from the diversified samples. To avoid selection bias, we picked random diverse samples for human annotation from CDSS corpora. However, the current data loader randomly picks samples without any diversification, which exposes us to selection bias. The same problem occurs in the selection of data samples or documents for the fine-tuning process.

To use the context effectively, fastText works better with sub-word level information. However, we faced challenges in adapting it to pre-processing (splitting words into sub-words), post-processing (combining sub-words into actual words), feature engineering (POS tags, TF-IDF, TextRank etc.,), calculating word-level attention over sub-words, and using the pre-trained WE model.

6. Conclusion

This paper proposes a novel KP identification method using minimal labeled data and hierarchical attention by retaining longer contextual dependencies. It incrementally builds the context to understand the word and its relationship with other words across the document to understand the long-range context. The proposed model (Hier-Attn-BiLSTM-CRF) demonstrated 10% improved accuracy, from 78% to 88%, for KP identification by adding document-level context through experimentation.

Further, the domain adaptation in a semi-supervised approach improved the overall performance of the model by contributing to creating high-quality synthetic labels, to solve the challenges of minimal labeled data. We also found that the custom batch-loader yielding 2–4 true samples for every 100 synthetic samples helps the fine-tuning process with the limited labeled data. In addition, it facilitates optimizing the quantity of GS labeled-data required.

Finally, our method contributes to the general architecture of NLP in effectively creating ML models with limited labeled domain data by leveraging domain adaptation techniques, document-level context, pre-trained LM, and pre-trained WE. Adding character-level, text-based features to the model's encoder, and confidence scores to the model's inference, would further strengthen our results. These will be the basis of our future studies.

Acknowledgement(s)

The work was supported by the National Institute of General Medical Sciences of the National Institutes of Health under Award Number R01GM138589 and partially under P20GM121342. This work has also benefited from research training resources and the intellectual environment enabled by the NIH/NLM T15 South Carolina Biomedical Informatics and Data Science for Health Equity (SC BIDS4Health) research training program (T15LM013977). The content is solely the authors' responsibility and does not necessarily represent the official views of the National Institutes of Health.

We acknowledge Clemson University for the generous allotment of compute time on the Palmetto cluster for our experimentation. Also, we thank the anonymous reviewers for their detailed and insightful comments on earlier drafts of this paper.

Disclosure statement

There is no conflict of interest to disclose the details.

7. References

[1] Marco-Ruiz L, Bellika JG. Semantic Interoperability in Clinical Decision Support Systems: A Systematic Review. Stud Health Technol Inform 2015;216:958.

[2] Fernández-Breis JT, Vivancos-Vicente PJ, Menárguez-Tortosa M, et al. Using semantic technologies to promote interoperability between electronic healthcare records' information models. Conf Proc IEEE Eng Med Biol Soc 2006;2006:2614-7. doi: 10.1109/iembs.2006.259686.

[3] Lobach D, Sanders G, Bright T, et al. Enabling Health Care Decision making Through Clinical Decision Support and Knowledge Management. Evidence Report No. 203. (Prepared by the Duke Evidence-based Practice Center under Contract No. 290-2007-10066-I.) AHRQ Publication No. 12-E001-EF.; 2012; Rockville, MD.

[4] Jing X, Himawan L, Law T. Availability and usage of clinical decision support systems (CDSS) in office-based primary care settings in the United States of America. BMJ Health & Care Informatics (under revision) 2019. https://doi.org/10.1136/bmjhci-2019-100015

[5] Gruber T. What is an ontology? Knowledge Acquisition 1993;5:199-220.

[6] Rector A. Foundations of the Semantic Web: Ontology Engineering. 2005;2006

[7] Jing X, Min H, Gong Y, et al. A systematic review of ontology-based clinical decision support system rules: usage, management, and interoperability. medRxiv; 2022. https://doi.org/10.1101/2022.05.11.22274984.

[8] He Zhiyong, Wang Zanbo, Wei Wei, Feng Shanshan, Mao Xianling, Jiang Sheng. (2020). A Survey on Recent Advances in Sequence Labeling from Deep Learning Models. https://doi.org/10.48550/arXiv.2011.06727.

[9] Kazi Saidul Hasan, Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1262–1273, Baltimore, Maryland. Association for Computational Linguistics. http://dx.doi.org/10.3115/v1/P14-1119.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010. https://dl.acm.org/doi/10.5555/3295222.3295349.

[11] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding https://doi.org/10.48550/arXiv.1810.04805.

[12] Zeyer Albert, Bahar Parnia, Irie Kazuki, Schluter Ralf, Ney Hermann. (2019). A Comparison of Transformer and LSTM Encoder Decoder Models for ASR. 8-15. https://doi.org/10.1109/ASRU46091.2019.9004025.

[13] Imran Ahamad Sheikh, Emmanuel Vincent, Irina Illina. Transformer versus LSTM Language Models Trained on Uncertain ASR Hypotheses in Limited Data Scenarios. LREC 2022 - 13th Language Resources and Evaluation Conference, Jun 2022, Marseille, France. hal-03362828v2

[14] Hochreiter S., Schmidhuber J"urgen. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[15] Lafferty J. D., McCallum A., Pereira F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the Eighteenth International Conference on Machine Learning (p./pp. 282–289), San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1-55860-778-1. https://dl.acm.org/doi/proceedings/10.5555/645530

[16] Sahrawat, D. et al. (2020). Keyphrase Extraction as Sequence Labeling Using Contextualized Embeddings. In:, et al. Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science (), vol 12036. Springer, Cham. https://doi.org/10.1007/978-3030-45442-5 41.

[17] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/N16-1174

[18] Xu, Guohai & Wang, Chengyu & He, Xiaofeng. (2018). Improving Clinical Named Entity Recognition with Global Neural Attention: Second International Joint Conference, APWeb-WAIM 2018, Macau, China, July 23–25, 2018, Proceedings, Part II. 10.1007/9783-319-96893-3 20.

[19] Salton, G; McGill, M. J. (1986). Introduction to modern information retrieval. McGrawHill. ISBN 978-0-07-054484-0. 10.5555/576628

[20] Hasan K. S., Ng V. (2010). Conundrums in unsupervised keyphrase extraction: Making sense of the state-of-the-art. In: Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, pp. 365–373. https://dl.acm.org/doi/proceedings/10.5555/1944566.

[21] Amati, G. (2009). BM25. In: LIU, L., OZSU, M.T. (eds) Encyclopedia of Database Sys-¨ tems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9 921

[22] Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1–7), 107–117. https://doi.org/10.1016/s0169-7552(98)00110-x

[23] Florian Boudin. 2018. Unsupervised Keyphrase Extraction with Multipartite Graphs. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 667–672, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.48550/arXiv.1803.08721

[24] Corina Florescu and Cornelia Caragea. 2017. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/P17-1102

[25] Adrien Bougouin, Florian Boudin, and B´eatrice Daille. 2013. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

[26] Raschka, S. (2014). Naive Bayes and Text Classification I - Introduction and Theory. ArXiv, abs/1410.5329.

[27] Quinlan, J.R. Induction of decisiontrees. Mach Learn 1, 81–106 (1986). https://doi.org/10.1007/BF00116251

[28] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7 12.

[29] Witten, Ian & Paynter, Gordon & Frank, Eibe & Gutwin, Carl & Nevill-Manning, Craig. (1999). KEA: Practical Automatic Keyphrase Extraction. ACM DL. 254-255. 10.1145/313238.313437.

[30] Chengzhi Zhang, Lei Zhao, Mengyuan Zhao, Yingyi Zhang. Enhancing Keyphrase Extraction from Academic Articles with their Reference Information. Scientometrics, 2022, 127(2): 703–731. https://doi.org/10.48550/arXiv.2111.14106

[31] Liu F., Pennell D., Liu F., Liu Y. (2009) Unsupervised approaches for automatic keyphrase extraction using meeting transcripts. In: Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, pp. 620–628. https://dl.acm.org/doi/proceedings/10.5555/1620754

[32] Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09). Association for Computing Machinery, New York, NY, USA, 756–757. https://doi.org/10.1145/1571941.1572113

[33] Huang, Z.H., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991 (2015)

[34] Jiang Y, Zhao T, Chai Y, et al. (2020) Bidirectional LSTM-CRF models for keyword extraction in Chinese sport news. In: MIPPR 2019: Pattern Recognition and Computer Vision. International Society for Optics and Photonics, 11430: 114300H. https://doi.org/10.48550/arXiv.1508.01991

[35] J. Li, A. Sun, J. Han and C. Li," A Survey on Deep Learning for Named Entity Recognition," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50-70, 1 Jan. 2022, doi: 10.1109/TKDE.2020.2981314. https://doi.org/10.1109/TKDE.2020.2981314

[36] Mikhailov, Vladislav & Shavrina, Tatiana. (2020). Domain-Transferable Method for Named Entity Recognition Task. 83-92. 10.5121/csit.2020.101407. https://doi.org/10.48550/arXiv.2011.12170

[37] Kulkarni, Vivek & Mehdad, Yashar & Chevalier, Troy. (2016). Domain Adaptation for Named Entity Recognition in Online Media with Word Embeddings. https://doi.org/10.48550/arXiv.1612.00148

[38] Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

[39] Weischedel, Ralph, et al. OntoNotes Release 5.0 LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium, 2013. https://doi.org/10.35111/xmhb-2b84

[40] Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

[41] Jing X, Indani A, Hubig NC, et al. A systematic approach to configuring MetaMap for optimal performance. Methods Inf Med 2022 doi: 10.1055/a-1862-0421

[42] Neumann, M., King, D., Beltagy, I., & Ammar, W. (2019). ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. ArXiv, abs/1902.07669.

[43] Yoshua Bengio, R´ejean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, null (3/1/2003), 1137–1155. https://dl.acm.org/toc/jmlr/2003/3/null

[44] L. Rabiner and B. Juang," An introduction to hidden Markov models," in IEEE ASSP Magazine, vol. 3, no. 1, pp. 4-16, Jan 1986, doi: 10.1109/MASSP.1986.1165342.

[45] Sachan, D.S., Xie, P., Sachan, M., & Xing, E.P. (2017). Effective Use of Bidirectional Language Modeling for Transfer Learning in Biomedical Named Entity Recognition. Machine Learning in Health Care. https://doi.org/10.48550/arXiv.1711.07908

[46] 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, Boulder, Colorado. Association for Computational Linguistics. https://dl.acm.org/doi/proceedings/10.5555/1596374

[47] Saad, F., Aras, H., Hackl-Sommer, R. (2020). Improving Named Entity Recognition for Biomedical and Patent Data Using Bi-LSTM Deep Neural Network Models. In: M´etais, E., Meziane, F., Horacek, H., Cimiano, P. (eds) Natural Language Processing and Information Systems. NLDB 2020. Lecture Notes in Computer Science (), vol 12089. Springer, Cham. https://doi.org/10.1007/978-3-030-51310-8 3

[48] Mikolov, Tomas et al. "Efficient Estimation of Word Representations in Vector Space." International Conference on Learning Representations (2013). https://doi.org/10.48550/arXiv.1301.3781

[49] Enriching Word Vectors with Subword Information, Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, 2016. https://doi.org/10.48550/arXiv.1607.04606

[50] Bag of Tricks for Efficient Text Classification, Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, 2016. https://doi.org/10.48550/arXiv.1607.01759

[51] 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. http://dx.doi.org/10.3115/v1/D14-1162

[52] M. Schuster and K. K. Paliwal," Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov. 1997, doi: 10.1109/78.650093.

[53] Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, Wang J. An attention-based BiLSTMCRF approach to document-level chemical named entity recognition. Bioinformatics. 2018 Apr 15;34(8):1381-1388. doi: 10.1093/bioinformatics/btx761. PMID: 29186323.

[54] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. arXiv preprint arXiv:1503.08895.

[55] A. Viterbi," Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," in IEEE Transactions on Information Theory, vol. 13, no. 2, pp. 260-269, April 1967, doi: 10.1109/TIT.1967.1054010.

[56] Syed, Muzamil Hussain, and Sun-Tae Chung. 2021." MenuNER: Domain-Adapted BERT Based NER Approach for a Domain with Limited Dataset and Its Application to Food

Menu Domain" Applied Sciences 11, no. 13: 6007. https://doi.org/10.3390/app11136007

[57] Alexandra Schofield and David Mimno. 2016. Comparing Apples to Apple: The Effects of Stemmers on Topic Models. Transactions of the Association for Computational Linguistics, 4:287–300. http://dx.doi.org/10.1162/tacl_a_00099

[58] Benoit Favre. Contextual language understanding Thoughts on Machine Learning in Natural Language Processing. Computation and Language [cs.CL]. Aix-Marseille Universite, 2019. tel-02470185

[59] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. 12, 2493– 2537 (2011). https://doi.org/10.48550/arXiv.1103.0398

[60] Benedict A. Rabut, Arnel C. Fajardo, and Ruji P. Medina. 2019. Multi-class Document Classification Using Improved Word Embeddings. In Proceedings of the 2nd International Conference on Computing and Big Data (ICCBD 2019). Association for Computing Machinery, New York, NY, USA, 42–46. https://doi.org/10.1145/3366650.3366661

[61] Nancy Chinchor and Beth Sundheim. 1993. MUC-5 Evaluation Metrics. In Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993.

[62] Hiroki Nakayama." A Python framework for sequence labeling evaluation" 2018. https: //github.com/chakki – works/seqeval

8. Appendices

Appendix A. Dataset Details

Table A1. Showing explicit details of the CDSS dataset during preprocessing

Type	Abstracts Number
Total after parsing PubMed XML	3326
HDE-labeled Set 1 (GS42)	42
ACM abstracts [8] + HDE-labeled Set 2 (PMIDs not in XML) [4]	8+4 = 12
New total with duplicates (Some articles from GS42 are in full text XML)	3380
Abstracts (<3 sentences ~little/no abstract)	99
New total with duplicates (After removing abstracts with <3 sentences)	3281 (1093 train + 2188 test)
HDE-labeled Set 2 (GS91) (ACM 8 + PubMed 83)	83 + 8 = 91
Total GS	91 + 42 = 133
Final total (Synthetic-labeled dataset) (After removing GS 133 from full dataset)	3148 (1049 train + 2099 test)

Appendix B. Entities identified on sample CDSS abstract

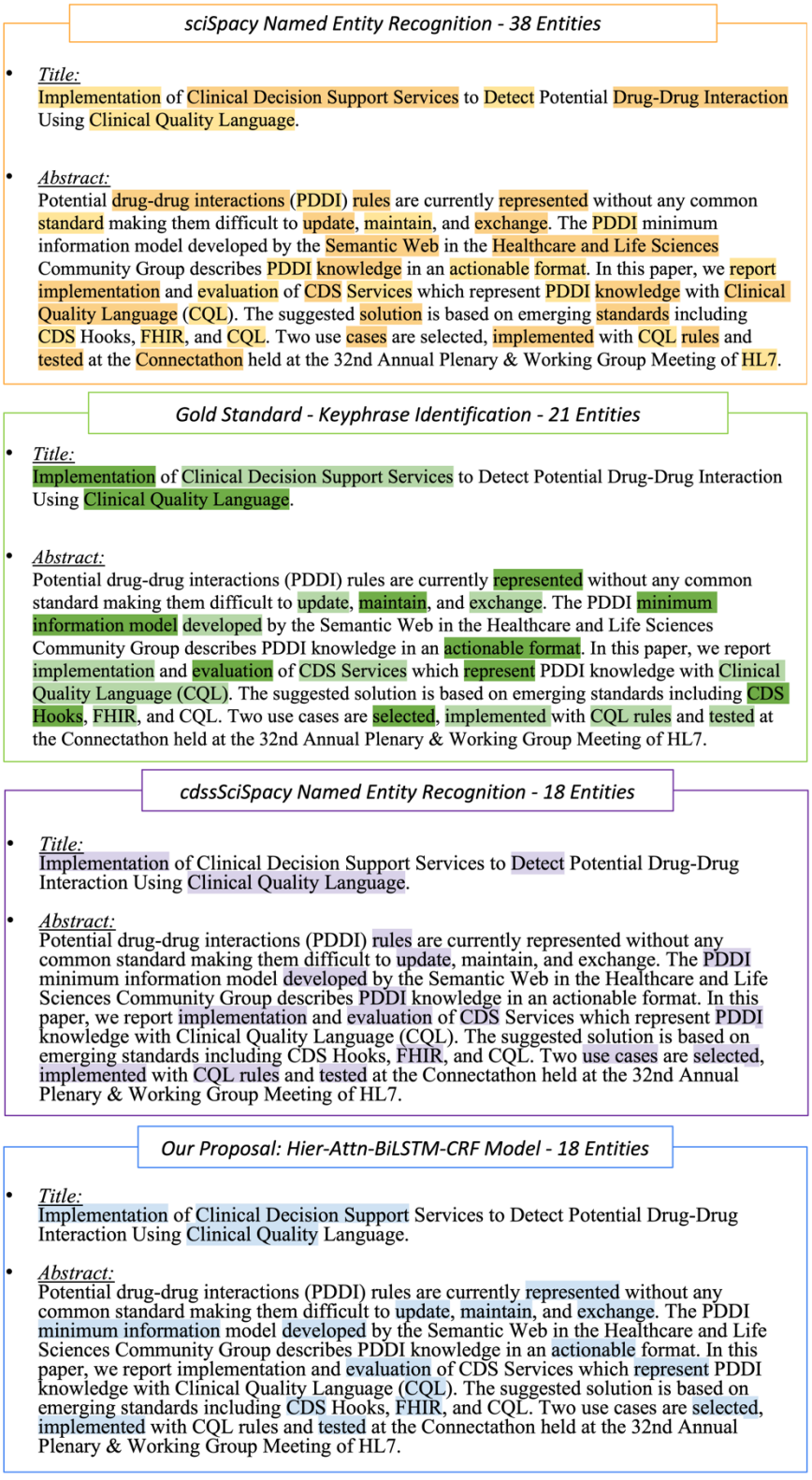


Figure B1. Entities identified on sample CDSS abstract demonstrate differences between human-labeled Gold Standards and different models (sciSpacy, cdssSciSpacy, and Hier-Attn-BiLSTM-CRF).

Appendix C. Metrics for fine-tuning on GS

Table C1. Fine-tuning with GS labels - 10 experiments

GS	0	2	4	6	8	10	12
Precision	83.78 ± 11.12	86.43 ± 5.78	86.56 ± 9.86	84.32 ± 4.99	85.54 ± 6.94	84.35 ± 9.92	85.99 ± 10.75
Recall	80.88 ± 4.89	82.41 ± 7.85	82.26 ± 5.69	82.28 ± 2.21	82.91 ± 9.41	81.80 ± 15.32	82.99 ± 7.20
Accuracy	95.62 ± 0.93	96.21 ± 0.51	96.24 ± 1.04	95.65 ± 0.58	95.73 ± 1.12	95.88 ± 0.56	96.13 ± 0.80
F1-Score	82.44 ± 4.82	83.66 ± 3.79	84.48 ± 5.68	83.14 ± 2.05	83.35 ± 7.21	82.81 ± 4.38	84.22 ± 7.42

Table C2. Fine-tuning with GS labels - 50 experiments

GS	0	2	4	6	8	10	12
Precision	83.78 ± 10.21	86.27 ± 8.28	86.29 ± 9.15	85.30 ± 8.01	85.95 ± 9.16	85.91 ± 8.70	86.22 ± 11.35
Recall	80.88 ± 4.49	83.16 ± 8.14	81.97 ± 11.91	82.53 ± 6.66	82.50 ± 7.83	82.45 ± 11.26	82.85 ± 8.20
Accuracy	95.62 ± 0.85	96.24 ± 0.74	96.16 ± 1.04	95.92 ± 0.72	96.06 ± 0.82	96.08 ± 0.87	96.27 ± 0.85
F1-Score	82.44 ± 4.43	84.65 ± 5.42	84.05 ± 7.36	83.65 ± 4.54	83.81 ± 6.05	83.91 ± 6.61	84.43 ± 6.59

Appendix D. Plots for fine-tuning on GS

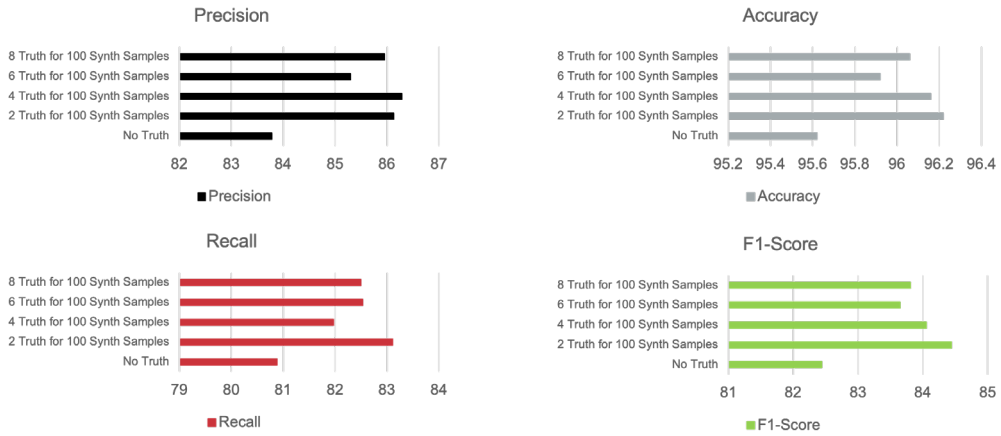


Figure D1. Plot evaluation metrics for fine-tuning with GS labels - 50 experiments

Figure D2. Plot evaluation metrics for fine-tuning with GS labels - 10 experiments