Sentence Segmentation in Clinical Notes via Sequence Labeling and a Dynamic Sliding Window

Dongfang Xu[1*], Davy Weissenbacher[1*], Karen O’Connor[2], Siddharth Rawal[2], Graciela Gonzalez-Her
[1] Cedars-Sinai Medical Center, Los Angeles, CA, USA
[2] University of Pennsylvania, Philadelphia, PA, USA
{dongfang.xu, davy.weissenbacher}@cshs.org, karoc@pennmedicine.upenn.edu, graciela.gonzalezhernande

Abstract

Sentence segmentation is a linguistic task and is widely used as a pre-processing step in many NLP applications. The need for sentence segmentation is particularly pronounced in clinical notes, where ungrammatical and fragmented texts are common. We propose a straightforward and effective sequence labeling classifier to predict sentence spans using a dynamic sliding window based on the prediction of each input sequence. This sliding window algorithm allows our approach to segment long text sequences on the fly. To evaluate our approach, we annotated 90 clinical notes from the MIMIC-III dataset. Additionally, we tested our approach on five other datasets to assess its generalizability and compared its performance against state-of-the-art systems on these datasets. Our approach outperformed all the systems, achieving an F1 score that is 15% higher than the next best-performing system on the clinical dataset.

1 Introduction

Sentence segmentation is the task of automatically identifying the boundaries of sentences in a written document, where a sentence is commonly defined as a sequence of grammatically linked words ending with a punctuation mark (PM). It is often the first preprocessing step for other natural language processing (NLP) tasks such as sentiment analysis (Medhat et al., 2014), information extraction (Angeli et al., 2015; Xu et al., 2020; Zhang and Bethard, 2023; Zhang et al., 2024), semantic textual similarity (Agirre et al., 2013), question answering ((Zhang et al., 2021b), and machine translation (Liu et al., 2020). Even tasks that operate at the paragraph or document level, such as coreference resolution (Stylianou and Vlahavas, 2021) or summarization (Pilault et al., 2020), often make use of sentences internally. Errors in segmentation could have detrimental effects on downstream task performance, e.g., in machine translation (Minixhofer et al., 2023), language modeling (Ek et al., 2020), and simultaneous speech translations (Wang et al., 2019). Detecting sentence boundaries is especially crucial for processing and understanding clinical text, as most clinical NLP tasks depend on this information for annotation and model training (Fan et al., 2013; Gao et al., 2022).

Despite its importance, sentence segmentation has received much less attention in the last few decades than other linguistic tasks. For non-clinical text, high-performing baseline systems use simple rule-based (Jurafsky and Martin, 2000; Manning et al., 2014) or machine learning-based (Gillick, 2009; Schweter and Ahmed, 2019) approaches that capture obvious and frequent sentence ending PMs (EPMs) such as [..]”. Such baselines leave little room for further improvement on traditional benchmarks derived from formal news(wire) sources or published articles. The focus on formal or edited text assumes EPMs as sentence boundaries, which is not directly applicable to real-world data such as clinical text (Read et al., 2012) or web text. These type of texts often contain fragmented and incomplete sentences, complex graphemic devices (e.g. abbreviations, and acronyms), and markups, which present challenges even for state-of-the-art sentence segmentation approaches, e.g., 70 − 85% F1 score on English Web Treebank (Straka, 2018; Qi et al., 2020). Another comprehensive evaluation of sentence segmentation in the clinical domain reveals that four standard sentence segmentation tools perform 20 − 30% worse on clinical texts compared to general-domain texts (Griffis et al., 2016).

Here, we present a sentence segmentation approach specifically tailored for real-world data, particularly clinical notes. Our method uses a sequence labeling classifier to predict sentence spans over a sliding window. During inference, we dynamically slide the window based on the prediction of each input sequence, such that the window always starts with a complete predicted sentence. This allows our approach to segment long text sequences on the fly without needing to pre-split the text. Moreover, the sequence labeling classifier does not rely on PMs for segmentation. To evaluate our approach on real-world clinical texts that can be shared, we annotated 90 clinical notes from MIMIC-III. Additionally, we extensively tested our method on five other datasets to assess its generalizability. Unlike other studies (Wicks and Post, 2021; Udagawa et al., 2023) that have modified datasets for sentence segmentation, we retained the original raw text, preserving their form and document structure.

Our work makes the following contributions:

- We propose a sentence segmentation approach capable of handling texts from diverse genres and domains without relying on specific text formats or EPMs. Our sliding-window algorithm segments long sequence texts on the fly, eliminating the need for pre-processing.
- We release a new sentence segmentation dataset based on MIMIC-III corpus. To the best of our knowledge, this is the first manually annotated sentence segmentation dataset using clinical notes.
- We comprehensively compare our approach against seven widely used off-the-shelf tools across six datasets. Our approach outperforms all these tools on five datasets, with particularly large margins on clinical datasets.

The code for our proposed approach and the new dataset are available at `https://bitbucket.org/hlpgonzalezlab/hlp_segmenter`.

2 Related Work

Existing sentence segmentation approaches can be categorized into rule- and learning-based approaches. Rule-based approaches (Aberdeen et al., 1995; Koehn et al., 2007; Dridan and Open, 2012; Sadvilkar and Neumann, 2020) utilize handcrafted rules, abbreviation lexicons, and linguistic features to decide whether a PM belongs to a token (an abbreviation or a number), or indicates the end of a sentence. For instance, Stanford CoreNLP toolkit (Manning et al., 2014) utilizes rules such as sentence ending PMs, or two consecutive line breaks to segment text. However, one major limitation of rule-based approaches is that the handcrafted rules are language- or domain-specific, making them difficult to maintain and adapt to new texts.

As an alternative, other systems aim to automatically learn segmentation rules through machine learning algorithms. When working with unlabeled data, unsupervised approaches (Mikheev, 2002; Kiss and Strunk, 2006) automatically curate information about abbreviations and proper names from large corpora and use them to determine whether the token preceding a period is an abbreviation and whether the token following a period is a proper name. One representative algorithm of the approach is in the Punkt system (Kiss and Strunk, 2006), as it computes the likelihood ratio of the truncated words and the following periods to identify abbreviations. An implementation of Punkt is bundled with the NLTK tool (Bird and Loper, 2004). Although these unsupervised approaches do not require extensive lexical resources or manual annotations and are easily adaptable to new domains, they can only segment sentential units (SUs) that use periods as sentence boundaries.

With the increasing availability of annotated corpora, supervised learning approaches have become predominant. One type of supervised approach combines a regular-expression-based detector to generate candidate SUs with a binary classifier. For generating candidate SUs, researchers have focused on only periods (Riley, 1989; Gillick, 2009), multiple EPMs (Reynar and Ratnaparkhi, 1997; Palmer and Hearst, 1997; Schweter and Ahmed, 2019), or more complex regular expressions (Wicks and Post, 2021). For classifying candidate SUs, most approaches employ binary classifiers with various features, e.g., a feedforward neural network with POS tags features (Palmer and Hearst, 1997), an SVM classifier with features such as length and the case of the words occurring before and after the PMs (Gillick, 2009), deep neural models using characters from the surrounding context (Schweter and Ahmed, 2019) of candidate SUs, or a two-layer Transformer encoder using the surrounding context words (Wicks and Post, 2021). However, all these approaches focus on proofread and edited documents, always assuming the existence of EPMs in all SUs. This assumption does not hold for informal, user-generated text or clinical notes with minimal proofreading and post-editing. As a consequence, several studies noted a substantial decline in performance when these systems move to texts with less formal language (Read et al., 2012; Rudra- pal et al., 2015).

Another competing supervised approach treats sentence segmentation as a sequence labeling task, assigning a tag to each input unit to mark sentence boundaries (Evang et al., 2013; Toleu et al., 2017; Du et al., 2019; Geng, 2022). This approach has the advantage of not relying on EPMs and can segment ungrammatical and fragmented texts. For example, Elephant (Evang et al., 2013) uses a CRF classifier to jointly segment tokens and sentences. By tagging each character in the input sequence, their classifier can identify SUs ending with various characters. Several works (Du et al., 2019; Rehbein et al., 2020; Udagawa et al., 2023) apply BERT-based sequence labeling classifiers for sentence segmentation. Due to the sequence length constraint of BERT models, these approaches split the original documents/texts into smaller sequences as inputs for BERT. This splitting is achieved either through domain knowledge, such as identifying pauses, speaker turns, or discourse markers from spoken language transcripts (Du et al., 2019), or by using an existing sentence segmentation tool (Udagawa et al., 2023). In contrast, our approach employs a sliding window to segment long sequence text on the fly, requiring no domain knowledge or off-the-shelf tools for pre-processing, which makes it easily applicable to texts from different domains and genres.

The approach proposed by Udagawa et al. (2023) shares similarities with ours in extending sentence segmentation beyond formal, standardized text using BERT-based sequence labeling classifier. Their method involves a two-step process: firstly, applying ERSATZ (Wicks and Post, 2021) - a segmentation tool based on punctuations - to the raw text; and secondly, using a classifier on the segmented text to detect sentence boundaries. However, in their evaluation, they ignore the boundaries of fragmented sentences generated by ERSATZ. Additionally, instead of directly identifying sentence boundaries during the sequence labeling step, as in our approach, they use a dynamic programming algorithm to infer labels for the entire document.

3 Methods

We approach sentence segmentation as a sequence labeling task using a BIO tagging scheme (shown in Figure 1). In this scheme, each token in an input sequence is assigned a tag to mark sentence boundaries: B indicates the Beginning of a sentence, I represents Inside of a sentence, and O denotes Outside of a sentence. We chose this tagging schema as it allows not only to segment sentences from a document but also to differentiate SUs (labelled as B and I) from non-SUs (labelled as O), also known as sentence identification task (Udagawa et al., 2023). Non-SUs typically include metadata from email attachments, markups in web text, irregular series of nouns, repetition of symbols for separating texts, and plain text tables in clinical notes, among other examples. All these non-SUs require additional text cleaning for downstream tasks. Unless otherwise specified, we do not differentiate between sentence identification and sentence segmentation in the following sections.

Formally, let $T = [t_0, t_1, \ldots, t_{n-1}]$ represent an input sequence that consists of n tokens; $Y = [y_0, y_1, \ldots, y_{n-1}]$ represent a sequence of BIO labels. So the goal of sentence segmentation task is to find a label sequence Y which satisfies:

$y_i = B$, when t_i is the first token of a SU. $y_i = I$, when t_i is any token within a SU except for the first token. $y_i = O$, when t_i is any token outside of a SU.

Pre-trained language models (PLM) (Edunov et al., 2019) have shown great improvements in NLP tasks, encompassing text classification, named entity recognition, or question answering, among others. Here, we use BERT (Devlin et al., 2019) in a sequence labelling configuration, where we feed a list of input tokens T to BERT, followed by a Softmax classification layer to predict the conditional probability of $P(Y|T)$.

3.1 Sliding window algorithm

Because of the quadratic computational cost along with the sequence length of the self-attention in transformer architecture (Vaswani et al., 2017), and the pre-training configuration of BERT-style PLMs, BERT models can only take input sequences with up to 512 tokens. Although the development of sparse attention mechanisms in transformer networks has improved the capability of PLMs for long sequence text (Beltagy et al., 2020), it is still challenging to take an entire clinical note as one input sequence. To segment long sequence text using BERT models, we propose a sliding window algorithm to process the input text, and then repetitively tag the text within a smaller sliding window (shown in Figure 1).

Let l be the maximal sequence length of any PLMs, and T_w be a sliding window of l tokens from the text input. The main idea of our algorithm is to tag each token within a sliding window, and then slide the text window based on the predicted sentence boundary. Specifically, for each sliding window, we find the start index of the first sentence b_i by locating the first B label in Y (line 9 of algorithm 1), the start index of the second sentence b_{i+1} by locating the second B label in Y (line 10), and the end index of the first sentence e_i by locating the last I label preceding b_{i+1} in Y (line 11). We then slide the input window to the start of the second sentence b_{i+1}. If there is no second sentence from the current sliding window (line 5), we slide the window by l tokens (line 12), and predict the labels for the new sliding window. We then concatenate the labels of multiple text windows to find the second sentence. During the training, since we already know all the sentence boundary indices beforehand, we generate the training instance by directly moving the sliding window along each sentence, where each text window always starts with the first token of a sentence, and has a length of l tokens.

Algorithm 1 SEGMENT_TEXT

```
function SEGMENT_TEXT($T$, $l$)
    $S \leftarrow []$
    $w_i \leftarrow 0$
    repeat
        $Y \leftarrow []$
        $e_i \leftarrow$ None, $b_{i+1} \leftarrow$ None
        while not_found $(e_i, b_{i+1})$ do
            $T_w \leftarrow T[w_i : w_i + l]$
            $Y_w \leftarrow$ Sequence_Labeller$(T_w)$
            Concatenate $Y_w$ to $Y$
            $b_i \leftarrow$ find_start_index$(Y, B, 0)$
            $b_{i+1} \leftarrow$ find_start_index$(Y, B, 1)$
            $e_i \leftarrow$ find_end_index$(Y, I, b_{i+1})$
            $w_i \leftarrow w_i + l$
        end while
        $w_i \leftarrow b_{i+1}$
        Append $(b_i, e_i)$ to $S$
    until $w_i \geq$ len$(T)$
    return $S$
end function
```

IMAGE NOT PROVIDED

Figure 1: Figure 1: Sliding window algorithm for sentence segmentation. We segment the text using three sliding windows sequentially (SW-1, SW-2, and SW-3). Each sliding window contains up to 8 tokens. The final sentence segmentation tags are at the top (Pred) of the diagram.

IMAGE NOT PROVIDED

Figure 2: Figure 2: Sentence boundary annotation from a small portion of a discharge summary note. We use B and E to mark the beginning and end of a sentence, respectively; "_" to mark an empty space between sentences; "⟨⟨n⟩⟩" to mark a newline character from the original note.

4 Datasets

4.1 MIMIC-III dataset annotation

To the best of our knowledge, there is no manually annotated sentence segmentation dataset in clinical domain. Zhang et al. (2021a) created a silver-standard treebank from clinical notes in the MIMIC-III using the default CoreNLP tokenizer (Manning et al., 2014), and later train and evaluate the Stanza (Qi et al., 2020) on such treebank for syntactic analysis. However, their treebank dataset was not reviewed by domain experts, and the evaluation on their treebank basically reflects how well other sentence segmentation approaches master the segmentation rules in Stanford CoreNLP library. There are also other clinical datasets (Uzuner et al., 2007, 2011, 2012; Sun et al., 2013) containing sentence boundary information, where the clinical notes have already been pre-processed with each sentence placed on a separate line. However, this modified structure does not reflect the format of real-world clinical notes. To address this gap, we collected a subset of clinical notes from the MIMIC-III corpus (Johnson et al., 2016), and manually annotated sentence boundaries without changing the original structure of clinical notes.

MIMIC-III contains de-identified clinical notes from 38,597 distinct patients admitted to a Beth Israel Deaconess Medical Center between 2001 and 2012. It covers 15 note types including discharge summary, physician note, radiology report, social work, among others. We randomly sampled 6 notes for each note type for annotation, yielding 90 notes in total. We stratified the notes into training, development, and test sets (57/15/18), respectively.

Clinical text presents unique challenges for syntactic annotation due to the irregular usage of punctuation, incomplete or fragmented sentences, and a blend of structured and narrative text formats, as illustrated in Figure 2. Guidelines designed for syntactic annotation in texts following typical structural and writing conventions might not be suitable for detecting sentence boundaries within the clinical domain. To mitigate these challenges, we developed a detailed annotation guideline and summarized what constitutes a sentence in the clinical note genre (more details in appendix A.1):

- Grammatically linked words written in an uninterrupted sequence that follow the conventional rules of a sentence in English, with or without an appropriate EPM.

- A text fragment that conveys a complete thought, e.g., a section header, or each item in a form or bulleted list, such as "Lab Test", "Results", or "Diagnosis", among many others.

One major challenge in our annotation is to distinguish a table from a list in clinical notes. Table text typically contains column headers, row labels, and texts from individual cells. We can not simply separate table text into multiple sentences by rows or cells because interpreting each cell requires an understanding of the original tabular structure, which is not typically included (and usually cannot be included due to technical limitations) in a data export from electronic health record systems such as EPIC. Thus, we assign *O* labels to the entire table text and leave parsing table text into sentences for future work.

Two annotators independently annotated each note, with the lead annotator being an expert in annotating clinical notes. At the first iteration, the annotators independently annotated the entire 90 notes, and notes without complete agreement were discussed until resolution during the second iteration. During the first iteration (on 15 notes), it took an average of 5.7 minutes to annotate each note. Before resolution, the inter-annotator agreement was 0.89 F1 (Hripcsak and Rothschild, 2005) on sentence boundary annotation which is considered moderate to strong agreement (McHugh, 2012).

4.2 Other datasets

To check whether our proposed approach is daagnostic, we extensively evaluated our approach on other standard corpora from different domains and genres, including 1) biomedical domain with clinical notes (i2b2- 2010), and abstracts of biomedical articles (Genia); and 2) the general domain, including various sources of English texts (Brown and WSJ) and web text (EWT). We summarize the dataset statistics in Table 1. Specifically, we examined whether the dataset format had any modifications during preprocessing or remained in its original form. For the general domain corpora, they assume each document is a disjoint union of sentences (no document information and no *O* tokens). However, since WSJ and EWT provide the original documents where each sentence belongs, we processed their annotations, and mapped each sentence into its original document (Original row in Table 1). We also analyzed statistics related to different sentence structures, such as sentences ending with EPMs, alphanumeric characters, or PMs other than EPMs (OPM). These sentence characteristics contribute to the complexity faced by different sentence segmentation approaches.

i2b2- 2010 The i2b2- 2010 corpus (Uzuner et al., 2011) consists of 426 labeled clinical notes (43,940 sentences). The corpus was released in 2010 i2b2 shared task focused on identifying concepts, assertions, and relations in discharge summaries and progress reports. This corpus had already been preprocessed, with each sentence placed on a separate line for each note. This pre-processing step simplifies both the original i2b2 shared task and the sentence segmentation task, as original clinical texts typically contain multiple newline characters within a sentence and multiple sentences within a single line. For our experiments, we maintain the same train/dev/test splits as in the 2010 i2b2 challenge.

Genia The Genia corpus (Kim et al., 2003) is a collection of 1,999 MEDLINE abstracts with 16,479 sentences related to transcription factors in human blood cells. These abstracts are unstructured text, and meticulously edited to include complete sentences. We use the split in Griffis et al. (2016) and randomly sample 400 and 200 documents for the development and test sets, respectively.

EWT The English Web Treebank (Silveira et al., 2014) comprises 1174 samples of web text sourced from five distinct genres: blog posts, newsgroup threads, emails, product reviews and answers from question-answer websites. Similar to the clinical corpus, EWT contains incomplete and fragmented sentences, but in general domain English language. We use the standard train/dev/test splits.

Brown The Brown corpus (Francis and Kucera, 1964) contains 500 samples of running text of edited American-English prose. Each sample begins at the beginning of a sentence but not necessarily of a paragraph or other larger division, and it ends at the first sentence ending after 2000 words. The text is drawn from a variety of sources such as books, newspapers, magazines, and transcripts of spoken language. Thus, this corpus have much formal sentence units. In our experiments, we load the corpus from the NLTK library (Bird and Loper, 2004), where sentences from each document are separated by empty spaces. We randomly sample 10% and 20% files for the development and test sets, respectively.

WSJ The WSJ corpus (Paul and Baker, 1992) contains 2312 samples of running text primarily sourced from the Wall Street Journal newspaper, covering a wide range of topics related to business, finance, economics, and current affairs. We preprocess this corpus to keep the original format of each running text based on their raw text file. We follow the configuration in Bird and Loper (2004) to keep section 24 for validation, and sections 03-06 for test.

A major difference between these datasets is their sentence structure. For clinical notes, MIMICIII and i2b2-2010 have only around 39% and 52% of sentences

Table 1: Dataset statistics. Original indicates that a dataset has its original format ($Y = Yes$). Sentence-EPM indicates the percentage of sentences ending with a EPM. Sentence-Alphanum indicates the percentage of sentences ending with an alphanumeric character. Sentence-OPM indicates the percentage of sentences ending with a PM other than an EPM. Sentence- Sep- N indicates the percentage of sentences separated by at least one newline character.

Dataset	#Sentences	#Documents	Original	Sentence-EPM (%)	Sentence-Alphanum (%)	Sentence-(
Sentence-Sep-N (%)						
MIMIC-III	4,520	90	Y	39.4	42.3	18
27.7						
i2b2-2010	43,940	426	N	52.1	31.5	16
3.9						
Genia	16,479	1,999	Y	99.2	0.4	0.
0.0						
EWT	12,283	1,174	Y	72.1	20.3	7.
5.9						
Brown	50,592	500	Y	90.5	6.8	2.
0.0						
WSJ	49,208	2,312	Y	91.0	6.5	2.
0.0						

end with EPMs (Sentence- EPM), respectively, compared against around 90% of sentences with EPMs in Brown and WSJ, and 99% of sentences in Genia. For approaches that purely rely on EPMs for sentence segmentation, they could only detect up to 52% of sentences for clinical notes, while 90% for general domain texts. This indicates the limitation of purely using EPM information for sentence segmentation. Clinical notes and web texts (EWT) have more sentences ending with alphanumeric characters (Sentence- Alphanum) or non- sentence ending PMs (Sentence- OPM) than the general domain texts or biomedical articles; they also often use newline characters to separate sentence. This indicates the importance of understanding text contents and text formats for sentence segmentation, especially for clinical notes and web texts.

5 Experiments

5.1 Comparisons with related approaches

We compared our proposed approach against seven off-the-shelf sentence segmentation systems: NLTK (Bird and Loper, 2004), CoreNLP (Manning et al., 2014), cTAKES (Savova et al., 2010), Syntok[1], spaCy[2], Stanza (Qi et al., 2020), Trankit (Nguyen et al., 2021). We selected these segmenters because they are state-of-the-art and easy-to-run standard NLP tools, and therefore widely used ”as is” by the community when processing text data. We provide a detailed description of each tool in appendix A.2.

5.2 Experiment details

As our MIMIC-III dataset contains non-sentential tokens (tagged as O) such as table text, for a fair comparison between these tools and our approach on the MIMIC-III dataset, we created an alternative evaluation, MIMIC-III$_p$ (shown in table 2). Specifically, we post-process the segmented output from off-the-shelf tools with six rules that take into account the text structures, such as removing multiple empty spaces or newline characters from the sentence boundary if they are at the end of a sentence. We also remove non-sentential tokens before segmentation during evaluation.

For clinical notes (MIMIC-III and i2b2-2010), and biomedical articles (Genia), we chose PubMedBERT (Gu et al., 2021) for our sequence labeling classifier. PubMed-BERT is a domain-specific language model pre-trained on biomedical text from scratch; it has achieved state-of-the-art performances on multiple biomedical NLP tasks. While for the general domain corpus (EWT, Brown, and WSJ), we chose RoBERTa-base (Liu et al., 2019). One limitation of BERT-style PLMs is that their tokenizers remove newline characters from input, which makes it challenging to segment text when newline characters are the only sentence separators. To mitigate this issue,

[1] [ILLEGIBLE]

[2] [ILLEGIBLE]

we insert the newline character as a special token in the tokenizer to keep the text format signal. Training details are illustrated in appendix A.3.

We trained two types of models: 1) Segmenter-Data, where we trained one model on each dataset (six models in total); 2) Segmenter-Domain, where we combined datasets from each domain, and train one model on the biomedical domain, and one model on the general domain.

5.3 Evaluation

We evaluated each system by comparing the predicted sentence spans against the gold annotations in the test sets. We measured the performance using the standard F1 evaluation metric, consistent with the evaluation adopted in the 2018 UD Shared Task for sentence boundary detection (Zeman et al., 2018). A sentence span is defined as a pair of offsets representing the first and last characters of a sentence. A predicted sentence span is considered accurate only if both offsets in the predicted pair match those in the gold annotation pair.

6 Results

On the MIMIC-III dataset, table 2 shows that our models outperform off-the-shelf tools by large margins (p = 0^3), ranging from 35.99% to 71.12% of F1. For a fair comparison, after post-processing the segmented outputs from all the tools and removing non-sentential tokens, we improve the performances of each tool by up to 32.86% of F1 (see column MIMIC$_p$), but they are still lower than our best model (Segmenter- Data) with 88.34% of F1.

Across five other standard benchmark datasets, table 2 also shows that our two type of models, Segmenter- Data and Segmenter- Domain, consistently achieve the best F1 on four datasets (except the WSJ dataset), for an average rank of 1.6 and 2, respectively. Trankit achieves the best performance on the WSJ dataset, with an average rank of 3.5. Compared against Segmenter- Data models that are trained on each individual datasets, Segmenter- Domain models that are trained on the combination of datasets from each domain, achieves nearly identical performances. This suggests that instead of maintaining six separate models, we can effectively use just two models for the segmentation task.

On another clinical dataset - i2b2- 2010, all tools except cTAKES achieve less than 58.68% of F1; while on a well-formed dataset - Genia, all tools except cTAKES achieve more than 96.93% of F1. Along with the evaluation on MIMIC- III dataset, we find that tools developed on the general domain texts struggle with clinical texts; however, they still achieve great performances on biomedical articles. This indicates that sentence segmentation is influenced not only by domain-specific language, such as terminology and abbreviations, but also by sentence structure and text form. Surprisingly, comparing the performances of cTAKES on i2b2- 2010 and MIMIC- III, we see a big performance drop. This is probably because the training data used in cTAKES is more similar to the i2b2- 2010 corpus.

Following the rankings of our models, only Trankit and Stanza achieve competitive performances on all three general domain datasets, with results exceeding 89.31% on EWT, 86.43% on Brown, and 93.78% on WSJ. Both CoreNLP and Syntok achieve slightly worse on Brown and WSJ, while much worse performances on EWT (around 66%). This is likely because both CoreNLP and Syntok fail to account for characteristics of web language, such as fragmented text and the absence of EPMs. Besides cTAKES, which is designed specifically for the clinical domain, both NLTK and Spacy achieve the worst performance on one of the three general domain datasets. We analyzed the sentence segmentation experiments with Spacy on the WSJ corpus, and found that newline characters caused many segmentation errors. After removing newline characters from the documents, we achieved an F1 score of nearly 96%.

IMAGE NOT PROVIDED

Figure 3: Figure 3: Cross-domain evaluation of our two Segmenter-Domain models. The blue bars show the performance of the segmenter trained on a combination of three biomedical corpora, while the red bars represent the performance of the segmenter trained on a combination of three general domain corpora.

7 Discussion

From the evaluation of off-the-shelf tools, we can see inconsistent performances on different datasets. This is expected because of language variation, sentence structures, and text form. To check whether such a phenomenon also exists in our approach, we conducted a cross-domain evaluation for our Segmenter-Domain models, i.e., evaluating models trained on biomedical domain datasets on the general domain datasets, and vice versa. Figure 3 shows similar findings as other tools: except on Genia and WSJ, there are around 27% of F1 drop on biomedical datasets, and around 20% of F1 drop on general

Table 2: Comparison of our proposed approach against off-the-shelf sentence segmenters. MIMIC- III$_p$ is an alternative evaluation on MIMIC- III dataset, where we post-processed the segmented outputs from all the off-the- shelf tools, and removed non- sentential tokens for a fair comparison. The last column Avg. Rank shows the average rank of each segmentation system across the datasets. We excluded the MIMIC- III$_p$ column when computing Avg. Rank, as it is not the real-world setting. The system with the best average rank is highlighted in grey; the best F1 scores on each dataset are bolded.

	MIMIC-III	MIMIC-III$_p$	i2b2-2010	Genia	EWT	Brown	WSJ	A
NLTK	21.81	40.83	23.66	95.74	74.58	85.90	88.97	
CoreNLP	17.22	45.58	28.06	97.17	66.29	89.73	95.31	
cTAKES	52.84	52.84	84.17	96.93	[ILLEGIBLE]	[ILLEGIBLE]	[ILLEGIBLE]	[II
Syntok	18.15	43.82	35.16	96.90	66.41	88.32	94.84	
spaCy	29.58	62.44	58.68	97.96	89.31	86.43	82.13	
Stanza	28.88	61.74	56.48	99.37	91.49	91.10	93.78	
Trankit	41.22	66.89	52.75	99.02	92.14	92.35	**98.32**	
Segmenter-Data	**88.34**	**88.34**	**92.10**	**99.71**	**95.28**	**96.73**	97.05	
Segmenter-Domain	86.57	86.57	91.24	99.58	94.84	96.34	97.16	

domain datasets. We also performed cross-dataset evaluation (models that are trained on one dataset and then evaluated on other datasets) for Segmenter-Data models, but the decline in performance was even more pronounced. We posit that Segmenter- Domain models hold better applicability in real-world scenarios due to their ability to generalize across multiple datasets.

To understand how each tool and our approach work on different text form, we compute the recall of top 4 off-the-shelf tools (based on their average rank in table 2) and our domain models on different forms of sentences (see table 1). We combine texts from test sets of multiple corpora including MIMIC-III, i2b2-2010, EWT, and WSJ to balance the amount of sentences in each subset. Table 3 shows the performances on each sentence subset. Firstly, sentences ending with alphanumerics are the most challenging for off-the-shelf tools, while our models successfully detect more than 95% of them. Although most tools particularly target on sentence ending with PMs, but they still miss 10% to 25% of such sentences. Lastly, as a notable feature for sentence segmentation task, we can see newline characters are not effectively utilized in off-the-shelf tools.

8 Conclusion

In conclusion, our proposed sentence segmentation approach addresses the challenges posed by real-world, ungrammatical, and fragmented text used in the daily, often harried and hectic hospital environment when typing clinical notes. Utilizing a sequence labeling classifier with a dynamic sliding window, our approach effectively segments long text sequences on the fly without requiring pre-splitting or relying on PMs. Additionally, we contribute a new sentence segmentation dataset derived from the MIMIC-III corpus, providing a valuable resource for future research in this domain. The evaluation on our annotated clinical notes, along with extensive testing on five additional datasets, demonstrated the generalizability and effectiveness of our approach over seven commonly used tools.

Table 3: Comparison of our Segmenter-Domain models against top 4 off-the-shelf-tools on different forms of sentences: Sentence-EPM, Sentence-Alphanum, and Sentence-Sep-NI. These sentences are from the test sets of MIMIC-III, i2b2-2010, EWT, and WSJ

Approach	EPM	Alphanum	NI
CoreNLP	75.78	0.97	45.67
Syntok	76.51	1.25	46.55
Stanza	82.63	9.73	53.22
Trankit	87.49	30.31	63.08
Segmenter-Domain	97.73	95.93	98.00

9 Limitations and future work

Similar to other sentence segmentation approaches using BERT-style PLMs (Nguyen et al., 2021; Udagawa et al., 2023), our method faces the limitation of high computational cost. The primary reason for this is the self-attention mechanism in BERT models, which causes the computational cost to increase quadratically with the input sequence length. Additionally, the inference time scales linearly with the number of times we slide the input window over the sequence. To address these challenges, future work could explore more efficient PLMs. Potential alternatives include ALBERT (Lan et al., 2019), which reduces model size and improves efficiency through parameter-sharing techniques; and DistilBERT (Sanh et al., 2020), which is a smaller, faster, and lighter version of BERT achieved through knowledge distillation.

Appendix

A Comprehensive guidelines for annotating sentences in clinical notes

The guidelines for annotating sentences within section headers, text forms, text lists, and text tables in clinical notes are as follows.

A.1 Section header

Section headers may be in all capital letters and may be followed by a colon or hyphen. If a header is followed by a colon or hyphen and is immediately followed by text that directly relates to the header, both the header and its corresponding text should be considered part of the same sentence. These elements may span across separate lines but should remain within the same sentence annotation. However, if a header followed by a colon or hyphen is succeeded by a different structure, such as a form, the header itself should be annotated as a separate sentence.

A.2 Text form

Text forms should appear within a sentence that includes only the label and its response (if provided). These forms can be identified as phrases that are not entirely capitalized and are always immediately followed by a colon. Both the label and its corresponding response should be part of the same sentence. If there is no response and another form begins immediately after the colon or on a new line, the label and colon should form a separate sentence.

When there is no clear indication of the end of a label/response (such as a period, new line, or semicolon), annotators should extend the sentence until the next distinct idea, fragment, or text structure. A label without a response may resemble an uncapitalized section header; however, both structures should be annotated similarly.

Nested forms can occur if the response to a label includes a list separated by commas or semicolons. In such cases, only the outer label and its direct response should be considered part of the annotated sentence, encompassing all nested forms within it. Forms separated by different characters, such as new lines, should not be treated as nested.

A.3 Text list

Numbered or bulleted lists should be annotated so that each list item, including its number or bullet, is treated as a separate sentence. List items may appear on a single line or be separated by newline characters. In cases where a list item's number or bullet is on one line and its text on the next, both should be included in the same sentence annotation. If a list item contains multiple sentences, the bullet or number should be associated with the first sentence, while subsequent sentences are annotated normally.

Bullets can consist of various symbols such as '-', '#', or '*'. Some lists, like those detailing drugs or tests performed, may not be explicitly bulleted or numbered. However, when annotating, these should be treated similarly to standard bulleted or numbered lists, with each item in the list annotated as a separate sentence.

A.4 Text table

Text formatted in a table typically cannot be segmented into individual sentences. Therefore, the entire contents of the table should be labeled as Non-SUs. If there is a section header that marks the beginning of the table, the header should also be included in the Non-SU annotation.

B Off-the-shelf NLP tools

NLTK The Natural Language Toolkit contains the Punkt sentence tokenizer (Kiss and Strunk, 2006) for sentence segmentation – an unsupervised system that uses frequency of occurrences of input features such as casing, punctuation, and length, to identify whether a period is from an abbreviation or a sentence ending PM. Punkt was trained on the WSJ corpus.

CoreNLP The Stanford CoreNLP toolkit uses a rule-based splitter: it first tokenizes the entire document into tokens, and then identifies whether a sentence-ending PM serves as sentence boundaries.

The rules of the system were developed using WSJ, GENIA, and other general domain English text. We evaluated the same system on all our datasets.

cTAKES The Apache cTAKES, a toolkit for analyzing electronic medical record clinical free-text, contains a sentence segmentation component that extends the OpenNLP's supervised ME sentence detector tool. It predicts whether a period, question mark, or exclamation mark ends a sentence. This model was trained on three corpora: Penn Treebank, Genia, and a corpus of clinical notes sampled from Mayo Clinic EMR.

Syntok The syntok package provides rule-based modules for tokenization and sentence segmentation. Similar to CoreNLP, the sentence segmentation module takes a token stream from the tokenizer as input, and split the token stream into sentences by checking whether a token is a sentence terminal marker.

spaCy The current version of spaCy library features transformer-based models for sentence segmentation, where it uses a sequence labeller to identify the first token of each sentence. In our experiments, we evaluated on the EWT, Brown, and WSJ, the default labeller of the pipeline, a RoBERTa-based model trained on blogs, news and comments. We evaluated on the MIMIC-III, i2b2-2010, and Genia corpora the labeller of the biomedical pipeline, a scibert-base model trained on biomedical text.

Stanza Stanza combines tokenization and sentence segmentation from raw text into a single module. It provides trained neural network models to perform tagging tasks over character sequences, where the models predict whether a given character is the end of a token, end of a sentence, or end of a multi-word token. Similar to spaCy, we evaluated three different Stanza models on our corpora: on EWT, Brown, and WSJ, the default English model trained on the English portion of the Universal Dependencies v2.5 treebanks; on Genia, the default biomedical model trained on the Genia treebank; on MIMIC-III and i2b2-2010, the default clinical model trained on EWT and a silver-standard corpus collected from the MIMIC-III database.

Trankit Trankit is a light-weight transformer-based toolkit for multilingual NLP. It provides a trainable pipeline that jointly perform tokenization and sentence segmentation over word-piece based input, where the model predict whether a wordpiece is the end of a single-word token, end of a sentence, or end of a multi-word token. Trankit utilizes the state-of-the-art multilingual pretrained transformer XLM-Roberta (Conneau et al., 2020), and is further trained on 90 Universal Dependencies treebanks. We evaluated the multilingual model on all our datasets.

C Training details

Unless specifically noted otherwise, we kept the default hyper-parameters as in huggingface's pytorch implementation across all datasets. For all the datasets, we kept the same hyper-parameters: learning rate = 3e − 5, sequence length = 512, the batch size = 32, epoch size = 10. We selected the best models based on the performances on the development set in a single run. We trained our models on one A100 GPU.

References

[1] John Aberdeen, John Burger, David Day, Lynette Hirschman, Patricia Robinson, and Marc Vilain. 1995. MITRE: Description of the Alembic system used for MUC-6. In Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995.

[2] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, pages 32-43, Atlanta, Georgia, USA. Association for Computational Linguistics.

[3] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 344-354, Beijing, China. Association for Computational Linguistics.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

[5] Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In Proceedings of the ACL Interactive Poster and Demonstration Sessions, pages 214-217, Barcelona, Spain. Association for Computational Linguistics.

[6] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440-8451, Online. Association for Computational Linguistics.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT 2019, pages 4171-4186.

[8] Rebecca Dridan and [ILLEGIBLE] Open. 2012. [ILLEGIBLE].

[9] [ILLEGIBLE] Du, [ILLEGIBLE]. 2019. [ILLEGIBLE].

[10] Sergey Edunov, [ILLEGIBLE]. 2019. [ILLEGIBLE].

[11] [ILLEGIBLE] Ek, [ILLEGIBLE]. 2020. [ILLEGIBLE].

[12] [ILLEGIBLE] Evang, [ILLEGIBLE]. 2013. [ILLEGIBLE].

[13] [ILLEGIBLE] Fan, [ILLEGIBLE]. 2013. [ILLEGIBLE].

[14] W. Nelson Francis and Henry Kucera. 1964. A Standard Corpus of Present-Day Edited American English, for use with Digital Computers. Brown University.

[15] [ILLEGIBLE] Gao, [ILLEGIBLE]. 2022. [ILLEGIBLE].

[16] [ILLEGIBLE] Geng, [ILLEGIBLE]. 2022. [ILLEGIBLE].

[17] [ILLEGIBLE] Gillick, [ILLEGIBLE]. 2009. [ILLEGIBLE].

[18] [ILLEGIBLE] Griffis, [ILLEGIBLE]. 2016. [ILLEGIBLE].

[19] [ILLEGIBLE] Gu, [ILLEGIBLE]. 2021. [ILLEGIBLE].

[20] George Hripcsak and Adam S. Rothschild. 2005. Agreement, the F-Measure, and Reliability in Information Retrieval. Journal of the American Medical Informatics Association, 12(3):296-298.

[21] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. Scientific Data, 3:160035.

[22] Daniel Jurafsky and James H. Martin. 2000. Speech and Language Processing. Prentice Hall.

[23] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. Bioinformatics, 19(suppl_1):i180–i182.

[24] Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. Computational Linguistics, 32(4):485–525.

[25] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, pages 177–180.

[26] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations.

[27] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8:726-742.

[28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[29] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In Proceedings

of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55-60, Baltimore, Maryland. Association for Computational Linguistics.

[30] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. Biochemia medica, 22(3):276-282.

[31] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4):1093-1113.

[32] Andrei Mikheev. 2002. Periods, capitalized words, etc. Computational Linguistics, 28(3):289-318.

[33] Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulic. 2023. Where's the point? self-supervised multilingual punctuation-agnostic sentence segmentation. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7215-7235, Toronto, Canada. Association for Computational Linguistics.

[34] Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A lightweight transformer-based toolkit for multilingual natural language processing. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 80-90, Online. Association for Computational Linguistics.

[35] David D. Palmer and Marti A. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. Computational Linguistics, 23(2):241-267.

[36] Douglas B. Paul and Janet M. Baker. 1992. The design for the Wall Street Journal-based CSR corpus. In Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992.

[37] Jonathan Pilaul, Raymond Li, Sandeep Subramanian, and Chris Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9308-9319, Online. Association for Computational Linguistics.

[38] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 101-108, Online. Association for Computational Linguistics.

[39] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jorgen Solberg. 2012. Sentence boundary detection: A long solved problem? In Proceedings of COLING 2012: Posters, pages 985-994, Mumbai, India. The COLING 2012 Organizing Committee.

[40] Ines Rehbein, Josef Ruppenhofer, and Thomas Schmidt. 2020. Improving sentence boundary detection for spoken language transcripts. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 7102-7111, Marseille, France. European Language Resources Association.

[41] Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In Fifth Conference on Applied Natural Language Processing, pages 16-19, Washington, DC, USA. Association for Computational Linguistics.

[42] Michael D. Riley. 1989. Some applications of tree-based modelling to speech and language. In Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989.

[43] Dwijen Rudrapal, Anupam Jamatia, Kunal Chakma, Amitava Das, and Bjorn Gamback. 2015. Sentence boundary detection for social media text. In Proceedings of the 12th International Conference on Natural Language Processing, pages 254-260, Trivandrum, India. NLP Association of India.

[44] Nipun Sadvikar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), pages 110-114, Online. Association for Computational Linguistics.

[45] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[46] 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.

[47] Stefan Schweter and Sajawel Ahmed. 2019. Deep-eos: General-purpose neural networks for sentence boundary detection. In Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019).

[48] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 2897-2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

[49] Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 197-207, Brussels, Belgium. Association for Computational Linguistics.

[50] Nikolaos Stylianou and Ioannis Vlahavas. 2021. A neural entity coreference resolution review. Expert Systems with Applications, 168:114466.

[51] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Annotating temporal information in clinical narratives. Journal of Biomedical Informatics, 46:S5-S12. Supplement: 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data.

[52] Alymzhan Toleu, Gulmira Tolegen, and Aibek Makazhanov. 2017. Character-based deep learning models for token and sentence segmentation. In Proceedings of the 5th International Conference on Turkic Languages Processing (TurkLang 2017).

[53] Takuma Udagawa, Hiroshi Kanayama, and Issei Yoshida. 2023. Sentence identification with BOS and EOS label combinations. In Findings of the Association for Computational Linguistics: EACL 2023, pages 343-358, Dubrovnik, Croatia. Association for Computational Linguistics.

[54] Ozlem Uzuner, Yuan Luo, and Peter Szolovits. 2007. Evaluating the State-of-the-Art in Automatic Deidentification. Journal of the American Medical Informatics Association, 14(5):550-563.

[55] Ozlem 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.

[56] Ozlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R South. 2012. Evaluating the state of the art in coreference resolution for electronic medical records. Journal of the American Medical Informatics Association, 19(5):786-791.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

[58] Xiaolin Wang, Masao Utiyama, and Eiichiro Sumita. 2019. Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network. In Proceedings of Machine Translation Summit XVII: Research Track, pages 1-11, Dublin, Ireland. European Association for Machine Translation.

[59] Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3995-4007, Online. Association for Computational Linguistics.

[60] Dongfang Xu, Zeyu Zhang, and Steven Bethard. 2020. A generate-and-rank framework with semantic type regularization for biomedical concept normalization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8452-8464, Online. Association for Computational Linguistics.

[61] Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing

from Raw Text to Universal Dependencies, pages 1-21, Brussels, Belgium. Association for Computational Linguistics.

[62] Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D Manning, and Curtis P Langlotz. 2021a. Biomedical and clinical English model packages for the Stanza Python NLP library. Journal of the American Medical Informatics Association, 28(9):1892-1899.

[63] Zeyu Zhang and Steven Bethard. 2023. Improving top-norm resolution with better candidate generation, transformer-based reranking, and two-stage resolution. In Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023), pages 48-60, Toronto, Canada. Association for Computational Linguistics.

[64] Zeyu Zhang, Egoitz Laparra, and Steven Bethard. 2024. Improving top-norm resolution by predicting attributes to constrain geographical ontology entries. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 35-44, Mexico City, Mexico. Association for Computational Linguistics.

[65] Zeyu Zhang, Thuy Vu, and Alessandro Moschitti. 2021b. Joint models for answer verification in question answering systems. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3252-3262, Online. Association for Computational Linguistics.