Biomedical and clinical English model packages for the Stanza Python NLP library

Yuhao Zhang¹, Yuhui Zhang², Peng Qi², Christopher D. Manning³, and Curtis P. Langlotz⁴

¹Biomedical Informatics Training Program, Stanford University, Stanford, California, USA; ²Computer Science Department, Stanford University, Stanford, California, USA; ³Computer Science and Linguistics Departments, Stanford University, Stanford, California, USA; and ⁴Department of Radiology, Stanford University, Stanford, California, USA

Corresponding Author: Yuhao Zhang, PhD, Biomedical Informatics Training Program, Stanford University, Stanford, CA 94305, USA; yuhao.zhang@stanford.edu

Received 5 January 2021; Revised 5 April 2021; Editorial Decision 22 April 2021; Accepted 3 May 2021

ABSTRACT

Objective: The study sought to develop and evaluate neural natural language processing (NLP) packages for the syntactic analysis and named entity recognition of biomedical and clinical English text.

Materials and Methods: We implement and train biomedical and clinical English NLP pipelines by extending the widely used Stanza library originally designed for general NLP tasks. Our models are trained with a mix of public datasets such as the CRAFT treebank as well as with a private corpus of radiology reports annotated with 5 radiology-domain entities. The resulting pipelines are fully based on neural networks, and are able to perform tokenization, part-of-speech tagging, lemmatization, dependency parsing, and named entity recognition for both biomedical and clinical text. We compare our systems against popular open-source NLP libraries such as CoreNLP and scispaCy, state-of-the-art models such as the BioBERT models, and winning systems from the BioNLP CRAFT shared task.

Results: For syntactic analysis, our systems achieve much better performance compared with the released scispaCy models and CoreNLP models retrained on the same treebanks, and are on par with the winning system from the CRAFT shared task. For NER, our systems substantially outperform scispaCy, and are better or on par with the state-of-the-art performance from BioBERT, while being much more computationally efficient.

Conclusions: We introduce biomedical and clinical NLP packages built for the Stanza library. These packages offer performance that is similar to the state of the art, and are also optimized for ease of use. To facilitate research, we make all our models publicly available. We also provide an online demonstration (http://stanza.run/bio).

Key words: natural language processing, machine learning, syntactic analysis, dependency parsing, named entity recognition

INTRODUCTION

A large portion of biomedical knowledge and clinical communication is encoded in free-text biomedical literature or clinical notes.¹,² The biomedical and clinical natural language processing (NLP) communities have made substantial efforts to unlock this knowledge, by building systems that are able to extract information,³,⁴ answer questions,⁵,⁶ or understand conversations⁷ from biomedical and clinical text.

NLP toolkits that are able to understand the linguistic structure of biomedical and clinical text and extract information from it are often used as the first step of building such systems.⁸,⁹ Existing
general-purpose NLP toolkits are optimized for high performance and ease of use but are not easily adapted to the biomedical domain with state-of-the-art performance. For example, the Stanford CoreNLP library 10 and the spaCy library (https://spacy.io/), despite being widely used by the NLP community, do not provide customized models for biomedical language processing. The recent scispacy toolkit 11 extends spaCy’s coverage to the biomedical domain, yet it does not provide state-of-the-art performance on syntactic analysis or entity recognition tasks, and does not offer models customized to clinical text processing.

In addition to general-purpose NLP toolkits, several NLP toolkits specialized for processing biomedical or clinical text are available. For example, Mayo Clinic’s cTAKES (Clinical Text Analysis and Knowledge Extraction System) provides a dictionary-based named-entity recognizer to find Unified Medical Language System Metathesaurus terms 12 in text, in addition to other NLP functionalities, such as tokenization, part of speech tagging, and parsing. 13 Other similar packages include the Health Information Text Extraction (HiTEx) library, 14 the MetaMap toolkit, 15 and the CLAMP clinical NLP toolkit. 16 These packages often integrate sophisticated domain-specific features crafted by experts, yet they fall short of integrating modern deep learning-based models that offer much more accurate performance than traditional rule-based or machine learning methods. Moreover, as Python becomes a common language of choice in the biomedical data science community, 17 the lack of native Python support has significantly limited users’ ability to adopt these toolkits and integrate them with modern computational libraries such as the deep learning libraries.

The recently introduced Stanza NLP library 18 offers state-of-the-art syntactic analysis and NER functionality with native Python support. Its fully neural pipeline design enables extension of its language processing capabilities to the biomedical and clinical domain. In this study, we present biomedical and clinical English model packages for the Stanza library (Figure 1). These packages are built on top of Stanza’s neural system, and offer syntactic analysis support for biomedical and clinical text, including tokenization, lemmatization, part-of-speech (POS) tagging, and dependency parsing, based on the Universal Dependencies v2 (UDv2) formalism, 19 and highly accurate named entity recognition (NER) capabilities covering a wide variety of domains.

These packages include 2 UD-compatible biomedical syntactic analysis pipelines trained on the publicly available CRAFT 20 and GENIA 8 treebanks, respectively; a UD-compatible clinical syntactic analysis pipeline, trained with a silver-standard treebank created from clinical notes in the MIMIC-III (Medical Information Mart for Intensive Care-III) database. 21; 8 accurate biomedical NER models augmented with contextualized representations, achieving near state-of-the-art performance; and 2 clinical NER models, including a newly introduced model specialized in recognizing entities in clinical radiology reports.

We show through a variety of experiments that these packages achieve performance that meets or surpasses state-of-the-art results. We further show via examples and benchmarking that these packages are easy to use and do not compromise speed, especially when GPU acceleration is available. We hope that our packages will facilitate future research to analyze and understand biomedical and clinical text.

MATERIALS AND METHODS

Syntactic analysis modules and implementations

Stanza’s syntactic analysis pipeline consists of modules for tokenization, sentence segmentation, POS tagging, lemmatization, and dependency parsing. All modules are implemented as neural network models. We briefly introduce each component in turn and refer readers to the Stanza system paper 18 for details.

Tokenization and sentence splitting

The first step of text analysis is usually tokenization and sentence segmentation. In Stanza, these 2 tasks are jointly modeled as a tagging problem over character sequences, in which the model predicts whether a given character is the end of a token, a sentence, or neither. This joint task is realized with a lightweight recurrent neural network. We choose to combine these tasks because they are usually context-sensitive and can benefit from joint inference to reduce ambiguity.

POS tagging

Once the text is tokenized, Stanza predicts the POS tags for each word in each sentence.

We adopt a bidirectional long short-term memory network (BiLSTM) as the basic architecture to predict both the language-specific POS (XPOS) tags and the universal POS (UPOS) tags.

We further adapt the biaffine scoring mechanism from the biaffine neural parser 22 to condition XPOS prediction on that of UPOS, which improves the prediction consistency between XPOS and UPOS tags. 23

Lemmatization

In many practical downstream applications, it is useful to recover the canonical form of a word by lemmatizing it (eg, recovering the lemma form do from the word did) for better pattern matching. Stanza’s lemmatizer is implemented as an ensemble of a dictionary-based lemmatizer and a neural sequence-to-sequence lemmatizer that operate on character sequences. An additional classifier is built on the encoder output of the seq2seq model, to predict shortcut operations such as lowercasing the input word or using an exact copy of the input word as lemma. These shortcut operations improve the robustness of the neural lemmatizer on long input character sequences such as URLs by avoiding the unnecessary generation of very long sequences.

Dependency parsing

To analyze the syntactic structure of each sentence, Stanza parses it into the UD format, 19 in which each word in a sentence is assigned a syntactic head that is either another word in the sentence, or in the case of the root word, an artificial root symbol. The dependency parser in Stanza is a variant of the BiLSTM-based deep biaffine neural dependency parser 22 that Qi et al. 23 have modified for improved accuracy.

Biomedical syntactic analysis pipeline

We provide 2 separate syntactic analysis pipelines for biomedical text by training Stanza’s neural syntactic pipeline on 2 publicly available biomedical treebanks: the CRAFT treebank 20 and the GENIA treebank. 8, 24 The 2 treebanks differ in 2 main ways. First, while GENIA is collected from PubMed abstracts related to “human,” “blood cells,” and “transcription factors,” CRAFT is collected from full-text articles related to the Mouse Genome Informatics database. Second, while the CRAFT treebank tokenizes segments of hyphenated words separately (eg, up-regulation tokenized into up-regulation), the GENIA treebank treats hyphenated words as single tokens.
Because both treebanks provide only Penn Treebank annotations in their original releases, to train our neural pipeline, we first convert both of them into UDv2 format annotations, using the UD converter in the Stanford CoreNLP library. To facilitate future research we have made the converted files publicly available (https://nlp.stanford.edu/projects/stanza/bio/).

**Treebank combination**

Because the tokenization in the CRAFT treebank is fully compatible with that in the general UD English treebanks, in practice we found it beneficial to combine the English Web Treebank (EWT) with the CRAFT treebank for training the CRAFT syntactic analysis pipeline. We show later via experiments that this treebank combination improves the robustness of the resulting pipeline on both general and in-domain text.

**Clinical syntactic analysis pipeline**

Unlike the biomedical domain, no large annotated treebanks for clinical text are publicly available.

Therefore, to build a syntactic analysis pipeline that generalizes well to the clinical domain, we created a silver-standard treebank by making use of the publicly available clinical notes in the MIMIC-III database. The creation of this treebank is based on 2 main observations made via qualitative analysis over sampled clinical notes from the MIMIC-III database. First, we find that Stanza’s neural syntactic analysis pipeline trained on general English treebanks generalizes reasonably well to well-formatted text in the clinical domain. Second, the highly optimized rule-based tokenizer in the Stanford CoreNLP library produces more accurate and consistent tokenization and sentence segmentation on clinical text than the neural tokenizer in Stanza trained on a single treebank. For example, while the neural tokenizer trained on a general English treebank tends to produce inconsistent sentence segmentations in the presence of consecutive punctuation marks or spaces in a sentence, the CoreNLP tokenizer handles these cases in a much more consistent and accurate manner.

Based on these observations, we create a silver-standard MIMIC treebank with the following procedure. First, we randomly sample 800 clinical notes of all types from the MIMIC-III database, and stratify the notes into training/dev/test splits with 600/100/100 clinical notes, respectively. These numbers are chosen to create a treebank of similar size to the general English EWT treebank. Second, we tokenize and sentence-segment the sampled notes with the default CoreNLP tokenizer. Third, we pretrain Stanza’s general English syntactic analysis pipeline on the EWT treebank, then run it on the pretokenized notes, and produce syntactic annotations following the UDv2 format. Fourth, to improve the robustness of the resulting models trained on this treebank, similar to the CRAFT pipeline, we concatenate the training split of the original EWT treebank with this silver-standard MIMIC treebank. We show later via experiments that this treebank combination again improves the robustness of the resulting pipeline on syntactic analysis tasks. A diagram that illustrates this whole training procedure is shown in Figure 2.

**NER models**

Stanza’s NER component adopts the architecture of the contextualized string representation-based sequence tagger. For each domain, we train a forward and a backward LSTM character-level...
language model (CharLM) to supplement the word representation in each sentence. At tagging time, we concatenate the representations from these CharLMs at each word position with a word embedding, and feed the result into a standard 1-layer BiLSTM sequence tagger with a conditional random field–based decoder. The pretrained CharLMs provide rich domain-specific representations that notably improve the accuracy of the NER models.

For the biomedical domain, we provide 8 individual NER models trained on 8 publicly available biomedical NER datasets: AnatomEM, BC5CDR, BC4CHEMD, BioNLP13CG, JNLPBA, Linnaeus, NCBI-Disease, and S800. These models cover a wide variety of entity types in domains ranging from anatomical analysis to genetics and cellular biology. For training, we use preprocessed versions of these datasets provided by Wang et al.

For the clinical NER models, we provide 2 individually trained models. First, we provide a general-purpose NER model trained on the 2010 i2b2/VA dataset that extracts problem, test, and treatment entities from various types of clinical notes. Second, we also provide a new radiology NER model, which extracts 5 types of entities from radiology reports: anatomy, observation, anatomy modifier, observation modifier, and uncertainty. The training dataset of this NER model consists of 150 chest computed tomography radiology reports collected from 3 individual hospitals. Two radiologists were trained to annotate the reports with 5 entity types with an estimated Cohen’s kappa interannotator agreement of 0.75. For full details of the entity types and corpora used in this dataset, we refer the readers to Hassanpour and Langlotz.

For computational efficiency, we sampled about half of the 2020 PubMed Baseline dump as our training corpus, which includes about 2.1 billion tokens. For the clinical NER models, we pretrain the CharLMs on all types of the MimIC-III clinical notes. During preprocessing of these notes, we exclude sentences in which at least 1 anonymization mask is applied (e.g., First Name\([\text{NamePattern2}]\)), to prevent the prevalence of such masks from polluting the representations learned by the CharLMs. The final corpus for training the clinical CharLMs includes about 0.4 billion tokens.

RESULTS

Syntactic analysis performance

We compare Stanza’s syntactic analysis performance mainly with CoreNLP and scispaCy, and present the results in Table 1. We focus on evaluating the end-to-end performance of all toolkits starting from raw text. In this evaluation setup, a system takes raw text as input, and each module makes predictions by taking outputs from its previous modules. This setup is more challenging than using gold tokenized text and other annotations as input to downstream modules, as used in a lot of previous evaluations. For quantitative evaluation of the syntactic pipeline, we adopt the official evaluation metrics used in the CoNLL 2018 Universal Dependencies Shared Task. We include detailed descriptions of our metrics in Supplementary Appendix A, and refer the readers to the shared task official website for in-depth introductions (https://universaldependencies.org/conll18/evaluation.html).

For fair comparisons, for both CoreNLP and scispaCy, we present their results by retraining their pipelines on the corresponding treebanks using the official training scripts. scispaCy results are generated by retraining the scispacy-large models. For the MimIC treebank, we do not include a comparison with scispaCy, mainly because we observed a severely degraded performance when applying it to tokenizing and sentence-segmenting clinical notes.

Notably, we find that Stanza’s neural pipeline generalizes well to all treebanks we evaluate on, and achieves the best results for all components on all treebanks.
POS and parsing with gold input
The much lower tokenization performance of scispaCy on the CRAFT treebank is due to different tokenization rules adopted: the tokenizer in scispaCy is originally developed for the GENIA treebank and therefore segments hyphenated words differently from the CRAFT treebank annotations (see Biomedical Pipeline), leading to lower tokenization performance. To understand the underlying syntactic analysis performance without this tokenization difference, we run an individual evaluation on the CRAFT treebank with gold tokenization results provided to the POS tagger and parser at test time. We find that under this gold tokenization setup, Stanza is able to achieve an XPOS F1 score of 98.40 and a parsing labeled attachment score (LAS) score of 92.10, while CoreNLP achieves 97.67 and 86.17, and scispaCy achieves 97.85 and 87.52 for XPOS and parsing LAS, respectively. Therefore, even with gold tokenization input (and gold POS tags for the parser), Stanza’s neural pipeline still leads to substantially better performance for both POS tagging and UD parsing, with a notable gain of 5.93 and 4.58 LAS compared with CoreNLP and scispaCy, respectively. Our findings are in line with previous observations that a neural biaffine architecture outperforms other models on biomedical syntactic analysis tasks.

Comparisons with CRAFT shared tasks 2019 systems
We further compare our end-to-end syntactic analysis results with the state-of-the-art system in the CRAFT Shared Tasks 2019, for which CRAFT is also used as the evaluation treebank. For all systems, we also report results for the official morphology-aware LAS (MLAS) and bi-lexical dependency score (BLEM) metrics, which, apart from dependency predictions, also take POS tags and lemma outputs into account.

Under this setting, we find that the CRAFT shared task 2019 baseline system, which uses a combination of the NLTK tokenizer\(^40\) and the SyntaxNet neural parser\(^31\) retrained with the CRAFT treebank, achieves limited performance with LAS = 56.68 and MLAS = 44.22 (no BLEX score due to missing lemma outputs), while our syntactic pipeline trained on the CRAFT dataset achieves a much better performance: LAS = 89.67, MLAS = 86.06, and BLEX = 86.47. For comparisons, the shared task winning system\(^42\) reports similar performance, with LAS = 89.70, MLAS = 85.56, and BLEX = 86.63. We note that the results from our system are not directly comparable to those from the shared task, owing to the different dependency parsing formalisms used (i.e., while we use UDv2 parse trees, the shared task used a parsing formalism similar to the older Stanford Dependencies formalism). Nevertheless, these results suggest that the accuracy of our pipeline is on par with that of the CRAFT shared task 2019 winning system, and substantially outperforms the shared task baseline system.

Effects of using combined treebanks
To evaluate the effect of using combined treebanks, we train Stanza’s biomedical and clinical syntactic analysis pipeline on each individual treebank as well as the combined treebanks and evaluate their performance on the test set of each individual treebank. We present the results in Table 2. We find that by combining the biomedical or clinical treebanks with the general English EWT treebank, the resulting model not only is able to preserve its high performance on processing general-domain text, but also achieves marginally better in-domain performance compared with using the biomedical and clinical treebanks alone. For example, while the pipeline trained on the EWT treebank alone is only able to achieve an LAS score of 68.99 on the CRAFT test set, the pipeline trained on the combined dataset achieves the overall best LAS score of 89.57 on the CRAFT test set, with only a drop in LAS of 1.2 on the EWT test set. These results suggest that compared with using the in-domain treebank alone, using the combined treebanks improves the robustness of Stanza’s pipeline on both in-domain and general English text.

NER performance
We mainly compare Stanza’s NER performance to BioBERT, which achieves state-of-the-art performance on most of the datasets tested, and scispaCy in Table 3. For both toolkits, we compare with their official reported results.\(^4\)\(^,\)\(^1\)\(^1\) We find that on most datasets tested, Stanza’s NER performance is on par with or superior to the strong performance achieved by BioBERT, despite using considerably more compact models. A substantial difference is observed on the BC4CHEMD and NCBI-Disease datasets, where BioBERT leads by 2.71 and 2.22 in F1, respectively, and on the S800 dataset, in which Stanza leads by 2.29 in F1 score. Compared with scispaCy, Stanza achieves substantially higher performance on all datasets tested. On the newly introduced Radiology dataset, Stanza achieves an overall F1 score of 84.80 micro-averaged over 5 entity types.

In addition to BioBERT, we also compare Stanza’s performance with SciBERT,\(^43\) which achieves F1 scores of 90.01, 77.28, and 88.57 on the BC5CDR, JNLPBA, and NCBI-Disease datasets, respectively, and ClinicalBERT,\(^44\) which achieves an F1 score of 86.4 on the i2b2 dataset. We find that Stanza’s performance is on par with or better than the strong performance offered by SciBERT and ClinicalBERT, too.

Effects of pretrained character LMs
To understand the effect of using the domain-specific pretrained CharLMs in NER models, on each dataset we also trained a baseline NER model in which the pretrained LM is replaced by a randomly initialized character-level BiLSTM, which is fine-tuned with other components during training. We compare Stanza’s full NER performance with this baseline model in Table 4. We find that by pretrain-

---

**Table 1. Neural syntactic analysis pipeline performance**

| Treebank System | Tokens Sents. | UPOS XPOS Lemmas UAS | LAS  |
|-----------------|--------------|----------------------|------|
| CRAFT           | 99.66        | 99.16                | 98.18 | 97.95 | 98.92 | 91.09 | 89.67 |
| GenIA           | 99.81        | 99.78                | 98.81 | 98.76 | 99.58 | 91.01 | 89.48 |
| MIMIC           | 99.18        | 97.11                | 95.64 | 95.25 | 97.37 | 85.44 | 82.81 |
| CoreNLP         | 98.22        | 97.20                | 93.40 | 96.98 | 97.97 | 84.75 | 83.16 |
| scispaCy        | 98.84        | 97.18                | 89.84 | 97.55 | 97.02 | 88.15 | 86.57 |
| CoreNLP         | 98.80        | 98.45                | 93.65 | 96.56 | 97.99 | 83.59 | 81.81 |
| scispaCy        | 91.49        | 97.47                | 83.81 | 89.67 | 89.39 | 79.08 | 77.74 |

---

All results are F1 scores produced by the 2018 UD Shared Task official evaluation script. All CoreNLP (>0.0) and scispaCy (>0.5) results are from models pretrained on the corresponding treebanks. UPOS results for scispaCy are generated by manually converting XPOS predictions to UPOS tags with the conversion script provided by spaCy. For scispaCy results, the *scispaCy-large* models are used. Note that the MIMIC results are based on silver-standard training and evaluation data as described previously.

LAS: labeled attachment score; MIMIC: Medical Information Mart for Intensive Care; UAS: unlabeled attachment score; UPOS: universal part of speech; XPOS: language part of speech.
ing Stanza’s CharLMs on large corpora, we are able to achieve an average gain of in F1 score of 2.91 and 1.94 on the biomedical and clinical NER datasets, respectively.

**Speed comparisons**

We compare the speed of Stanza with CoreNLP and scispaCy on syntactic analysis tasks, and with scispaCy and BioBERT on the NER task (for BioBERT, we implemented our own code to run inference on the test data, as an inference API is not provided in the BioBERT official repository). We use the CRAFT test set, which contains about 1.2 million raw characters, for benchmarking the syntactic analysis pipeline, and the test split of the JNLPBA NER dataset, which contains about 101k tokens, for benchmarking the NER task. Apart from CPU speed, we also benchmark a toolkit’s speed on GPU whenever GPU acceleration is available. Experiments are run on a machine with 2 Intel Xeon Gold 5222 CPUs (14 cores each). For GPU tests, we use a single NVIDIA Titan RTX card.

For each of the tasks, we focus on comparing the runtime of each toolkit relative to scispaCy. We find that for syntactic analysis, Stanza’s speed is on par with scispaCy when a GPU is used (1.42x runtime), although it is much slower when only a CPU is available (6.83x runtime vs scispaCy). Even in the CPU setting, Stanza’s biomedical syntactic analysis pipeline is still slightly faster than CoreNLP, which uses 7.23x runtime compared with scispaCy. For NER with GPU acceleration, Stanza’s biomedical models are marginally faster than scispaCy (0.95x runtime vs scispaCy) and are considerably faster than BioBERT (4.59x runtime vs scispaCy). When only CPU is available, Stanza’s biomedical models take much longer time to process text than scispaCy (14.8x runtime) but remain much faster than BioBERT which uses 121x runtime compared with scispaCy.

**DISCUSSION**

**System usage**

We provide a fully unified Python interface for using Stanza’s biomedical/clinical models and general NLP models. The biomedical and clinical syntactic analysis pipelines can be specified with a `package` keyword. We demonstrate how to download the CRAFT biomedical package and run syntactic analysis for an example sentence in Figure 3. For NER, Stanza’s biomedical and clinical models can be specified with a `processors` keyword. We demonstrate how to download the i2b2 clinical NER model along with the MIMIC clinical pipeline, and run NER annotation over an example clinical text in Figure 3. To easily integrate with external tokenization libraries, Stanza’s biomedical and clinical pipelines also support annotating text that is pretokenized and sentence-segmented. This can be easily specified with a `tokenize_pretokenized` keyword when initializing the pipelines.

We provide full details on how to use the biomedical and clinical models via online documentation (https://stanfordnlp.github.io/stanza/).

| Biomedical Syntactic Analysis Pipelines | EWT Test | CRAFT Test |
|----------------------------------------|----------|------------|
| Training Corpus                        | Token F1  | LAS        |
| EWT                                    | 99.01    | 83.59      |
| CRAFT                                  | 93.67    | 60.42      |
| Combined                               | 98.99    | 82.37      |

| Clinical Syntactic Analysis Pipelines  | EWT Test | MIMIC Test |
|---------------------------------------|----------|------------|
| Training Corpus                       | Token F1  | LAS        |
| EWT                                   | 99.01    | 83.59      |
| MIMIC                                 | 94.39    | 66.63      |
| Combined                              | 98.84    | 82.57      |

For the biomedical pipeline we show results for the English EWT treebank and the CRAFT treebank; for the clinical pipeline we show results for the English EWT treebank and a silver-standard MIMIC treebank. For each test set, tokenization F1 and LAS scores are shown for models trained with each treebank alone and a combined treebank.

EWT: English Web Treebank; LAS: labeled attachment score; MIMIC: Medical Information Mart for Intensive Care.

![Figure 3. Example code for using the biomedical syntactic analysis and named entity recognition pipelines in Stanza.](image-url)
Table 3. Named entity recognition performance across different datasets in the biomedical and clinical domains

| Category | Dataset            | Domain (# of Entities) | Stanza | BioBERT | scispacy-medium |
|----------|--------------------|------------------------|--------|---------|-----------------|
| Bio      | AnatEM             | Anatomy (1)            | 88.18  | –       | 84.14           |
|         | BC3CDR             | Chemical, Disease (2)  | 88.08  | –       | 83.92           |
|         | BC4CHEMD           | Chemical (1)           | 89.65  | 92.36   | 84.55           |
|         | BioNLP13CG         | Cancer Genetics (16)   | 84.34  | –       | 77.60           |
|         | JNLPBA             | Protein, DNA, RNA, Cell line, Cell type (5) | 76.09 | 77.49 | 73.21 |
|         | Linnaeus           | Species (1)            | 88.27  | 88.24   | 81.74           |
|         | NCBI-Disease       | Disease (1)            | 87.49  | 89.71   | 81.65           |
|         | S800               | Species (1)            | 76.35  | 74.06   | –               |
| Clinical | i2b2               | Problem, Test, Treatment (3) | 88.13 | 86.73 | –               |
|         | Radiology          | Radiology Report (5)   | 84.80  | –       | –               |

All scores reported are entity micro-averaged test F1. For each dataset, we also list the domain and number of its entity types. BioBERT results are from the v1.1 models reported by Lee et al.; scispacy results are from the scispacy-medium models reported by Neumann et al.

Table 4. Named entity recognition performance comparison between Stanza and a baseline BiLSTM-CRF model without character language models pretrained on large corpora

| Category | Dataset | Baseline | Stanza | Δ    |
|----------|---------|----------|--------|------|
| Bio      | AnatEM  | 85.14    | 88.18  | +3.04|
|         | BC3CDR  | 86.14    | 88.08  | +1.94|
|         | BC4CHEMD| 87.45    | 89.65  | +2.20|
|         | BioNLP13CG| 82.09  | 84.34  | +2.25|
|         | JNLPBA  | 75.29    | 76.09  | +0.80|
|         | Linnaeus| 83.74    | 88.27  | +4.53|
|         | NCBI-Disease | 84.04 | 87.49 | +3.45 |
|         | S800    | 71.30    | 76.35  | +5.05|
|         | Average (8 datasets) | 81.90 | 84.81 | +2.91 |
| Clinical | i2b2    | 86.04    | 88.08  | +2.04|
|         | Radiology| 83.01   | 84.80  | +1.79|
|         | Average (2 datasets) | 84.53 | 86.47 | +1.94 |

For both the biomedical and clinical models, an average difference over 8 models and 2 models are shown in the last column, respectively.

BiLSTM-CRF: bidirectional long short-term memory network conditional random field.

CONCLUSION

We present the biomedical and clinical model packages in the Stanza Python NLP toolkit. We show that Stanza’s biomedical and clinical packages offer highly accurate syntactic analysis and named entity recognition capabilities, while maintaining competitive speed with existing toolkits, especially when GPU acceleration is available. These packages are highly integrated with Stanza’s existing Python NLP interface, and require no additional effort to use. We hope to continuously maintain and expand these packages as new resources become available.

FUNDING

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

AUTHOR CONTRIBUTIONS

YuhaZ, YuhuZ, and PQ implemented the models used in this article. YuhaZ and YuhuZ performed the data collection, data processing and experiments. YuhaZ created the figures and tables, and drafted the manuscript. All authors conceived of the idea for the article. All authors were involved in the design of the methodologies and experiments, and in the preparation of the final manuscript.

SUPPLEMENTARY MATERIAL

Supplementary material is available at Journal of the American Medical Informatics Association online.

CONFLICT OF INTEREST STATEMENT

Curtis P. Langlotz is in the board of directors and a shareholder of Bunker Hill Health and is an advisor and option holder of whiterabbit.ai, Nines, GalileoCDS, and Sirona Medical.

DATA AVAILABILITY STATEMENT

The source code used in this paper is available at https://github.com/stanfordnlp/stanza. All pretrained models used in this paper can be downloaded by following the instructions at: https://stanfordnlp.github.io/stanza/biomed. html. An online demo of the models is available at: http://stanza.run/bio. The preprocessed biomedical treebanks used in this paper are available at: https://nlp.stanford.edu/projects/.

REFERENCES

1. Hunter L, Bretonnel Cohen K. Biomedical language processing: what’s beyond PubMed? Mol Cell 2006; 21 (5): 589–94.
2. Jha AK, DesRoches CM, Campbell EG, et al. Use of electronic health records in U.S. hospitals. N Engl J Med 2009; 360 (16): 1628–38.
3. Poon H, Quirk C, DeZiel C, et al. Literome: PubMed-scale genomic knowledge base in the cloud. Bioinformatics 2014; 30 (19): 2840–2.
4. Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2020; 36 (4): 1234–40.
5. Cao Y, Liu F, Simpson P, et al. AskHERMES: An online question answering system for complex clinical questions. J Biomed Inform 2011; 44 (2): 277–88.
6. Jin Q, Dhingra B, Liu Z, et al. PubMedQA: a dataset for biomedical research question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP); 2019: 2567–77. doi:10.18653/v1/d19-1259.
7. Du N, Chen K, Kannan A, et al. Extracting symptoms and their status from clinical conversations. In: Proceedings of the 57th Annual Meeting
of the Association for Computational Linguistics; 2019; 915–25.
doi:10.18653/v1/p19-1087.
8. McCosky D, Charniak E. Self-training for biomedical parsing. In:
Proceedings of the 46th Annual Meeting of the Association for
Computational Linguistics on Human Language Technologies Short Papers - HLT
’08; 2008: 101–4. doi:10.3115/1557690.1557717.
9. Baumgartner W, Bada M, Pyysalo S, et al. CRAFT shared tasks 2019 over-
view—integrated structure, semantics, and coreference. In: Proceedings of
the 5th Workshop on BioNLP Open Shared Tasks; 2019; 174–84.
doi:10.18653/v1/d19-5725.
10. Manning C, Surdeanu M, Bauer J, et al. The Stanford CoreNLP natural
language processing toolkit. In: Proceedings of 52nd Annual Meeting of
the Association for Computational Linguistics: System Demonstrations;
2014; 55–60. doi:10.3115/1p14-3010.
11. Neummann M, King D, Belagly I, et al. Scipacy: fast and robust models
for biomedical natural language processing. In: Proceedings of the 18th
BioNLP Workshop and Shared Task; 2019; 319–27. doi:10.18653/v1/
w19-5034.
12. Bodenreider O, The Unified Medical Language System (UMLS): inte-
grating biomedical terminology. Nucleic Acids Res 2004; 32: D267–70.
13. Savova GK, Masanz JJ, Ogren PV, et al. Mayo clinical Text Analysis and
Knowledge Extraction System (cTAKES): architecture, component evalua-
tion and applications. J Am Med Inform Assoc 2010; 17 (5): 507–13.
14. Zeng Q, Goryachev S, Weiss S, et al. Extracting principal diagnosis, co-
morbidity and smoking status for asthma research: evaluation of a natural
language processing system. BMC Med Inform Decis Mak 2006; 6: 30.
15. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective
and recent advances. J Am Med Inform Assoc 2010; 17 (3): 229–36.
16. Soysal E, Wang J, Jiang M, et al. CLAMPA—a toolkit for efficiently build-
ing customized clinical natural language processing pipelines. J Am Med
Inform Assoc 2018; 25 (3): 331–6.
17. Deardorff A. Why do biomedical researchers learn to program? An ex-
ploratory investigation. J Med Libr Assoc 2020; 108 (1): 29–35.
18. Qi P, Zhang Y, Zhang Y, et al. Stanza: A Python Natural Language Pro-
cessing Toolkit for Many Human Languages. In: Proceedings of the 38th
Annual Meeting of the Association for Computational Linguistics: System
Demonstrations; 2020; 101–8. doi:10.18653/v1/2020.acl-demos.14.
19. Neve J, de Marneffe M-C, Ginter F, et al. Universal dependencies v2: an
evergrowing multilingual treebank collection. In: Proceedings of the
Twelfth International Conference on Language Resources and Evaluation
(LREC’20); 2020: 4034–43.
20. Verspoor K, Cohen KB, Lanfranchi A, et al. A corpus of full-text journal
articles is a robust evaluation tool for revealing differences in performance of
biomedical natural language processing tools. BMC Bioinform 2012; 13: 207.
21. Johnson AEW, Pollard TJ, Shen L, et al. MIMIC-III, a freely accessible
critical care database. Sci Data 2016; 3: 160035.
22. Dozat T, Manning CD. Deep Biaffine attention for neural dependency
parsing. In: International Conference on Learning Representations (ICLR);
2017.
23. Qi P, Dozat T, Zhang Y, et al. Universal dependency parsing from scratch.
In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from
Raw Text to Universal Dependencies; 2018: 160–70.
24. Kim JD, Ohta T, Tateisi Y, et al. GENIA corpus—a semantically annotated
corpus for bio-textmining. Bioinformatics 2003; 19 (Suppl 1): i180–2.
25. Schuster S, Manning CD. Enhanced English universal dependencies: An
improved representation for natural language understanding tasks. Pro-
cedings of the Tenth International Conference on Language Resources and
Evaluation (LREC’16); 2016. 2371–8.
26. Silveira N, Dozat T, de Marneffe M-C, et al. A gold standard dependency
corpus for English. In: Proceedings of the Ninth International Conference
on Language Resources and Evaluation (LREC-2014); 2014: 2897–904.
27. Abkh A, Blythe D, Vollgraf R. Contextual string embeddings for se-
quence labeling. In: Proceedings of the 27th International Conference on
Computational Linguistics; 2018: 1638–49.
28. Pyysalo S, Ananiadou S. Anatomical entity mention recognition at litera-
ture scale. Bioinformatics 2014; 30 (6): 868–75.
29. Li J, Sun Y, Johnson RJ, et al. BioCreative V CDR task corpus: a resource
for chemical disease relation extraction. Database (Oxford) 2016; 2016:
baw068.
30. Krallinger M, Rabol O, Leitner F, et al. The CHEMDNER corpus of
chemicals and drugs and its annotation principles. J Cheminform 2015; 7
(Suppl 1): S2.
31. Pyysalo S, Ohta T, Rak R, et al. Overview of the cancer genetics and path-
way curation tasks of BioNLP shared task 2013. BMC Bioinform 2015;
16 (S10): S2.
32. Kim J-D, Ohta T, Tsuruoka Y, et al. Introduction to the bio-entity recog-
nition task at JNLPBA. In: Proceedings of the International Joint Work-
shop on Natural Language Processing in Biomedicine and its Applications;
2004: 73–8.
33. Gerner M, Nenadic G, Bergman CM. LINNAEUS: a species name iden-
tification system for biomedical literature. BMC Bioinform 2010; 11:
85.
34. Doğan RL, Leaman R, Lu Z. NCRI disease corpus: a resource for disease
name recognition and concept normalization. J Biomed Inform 2014; 47:
1–10.
35. Païfis E, Frankild S, Faninì L, et al. The SPECIES and ORGANISMS
resources for fast and accurate identification of taxonomic names in text.
PloS One 2013; 8 (6): e65390.
36. Wang X, Zhang Y, Ren X, et al. Cross-type biomedical named entity rec-
ognition with deep multi-task learning. Bioinformatics 2019; 35 (10):
1743–52.
37. Uzuner O, South BR, Shen S, et al. 2010 i2b2/VA challenge on concepts,
assertions, and relations in clinical text. J Am Med Inform Assoc 2011; 18
(5): 552–6.
38. Hassanpour S, Langlotz CP. Information extraction from multi-
institutional radiology reports. Arti Intell Med 2016; 66: 29–39.
39. Nguyen DQ, Verspoor K. From POS tagging to dependency parsing for
biomedical event extraction. BMC Bioinform 2019; 20 (1): 72.
40. Bird S, Klein E, Loper E. Natural Language Processing with Python: An-
alyzing Text with the Natural Language Toolkit. Newton, MA: O’Reilly
Media; 2009.
41. Andor D, Alberti C, Weiss D, et al. Globally normalized transition-based
neural networks. In: Proceedings of the 54th Annual Meeting of the Asso-
ciation for Computational Linguistics; 2016: 2442–52.
42. Ngo TM, Kanerva J, Ginter F, et al. Neural dependency parsing of
biomedical text: TurkuNLP entry in the CRAFT structural annotation task.
In: Proceedings of the 5th Workshop on BioNLP Open Shared Tasks;
2019: 206–15.
43. Beltya I, Lo K, Cohan A. SciBERT: a pretrained language model for sci-
centific text. In: Proceedings of the 2019 Conference on Empirical Methods
in Natural Language Processing and the 9th International Joint Confer-
ence on Natural Language Processing (EMNLP-IJCNLP); 2019;
3615–20.
44. Alsentzer E, Murphy J, Boag W, et al. Publicly available clinical BERT
embeddings. In: Proceedings of the 2nd Clinical Natural Language Pro-
cessing Workshop; 2019; 72–8.
45. Moen SP, Ananiadou TS. Distributional semantics resources for biomedi-
ical text processing. In: Proceedings of Languages in Biology and Medi-
cine on Language Resources and Evaluation (LREC-2014);
2016; 3: 160035.
46. Zhang Y, Chen Q, Yang Z, et al. BioWordVec, improving biomedical
word embeddings with subword information and MeSH. Sci Data 2019; 6
(1): 52.