ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing

Mark Neumann, Daniel King, Iz Beltagy, Waleed Ammar
Allen Institute for Artificial Intelligence, Seattle, WA, USA
{markn,daniel,beltagy,waleeda}@allenai.org

Abstract

Despite recent advances in natural language processing, many statistical models for processing text perform extremely poorly under domain shift. Processing biomedical and clinical text is a critically important application area of natural language processing, for which there are few robust, practical, publicly available models. This paper describes scispaCy, a new Python library and models for practical biomedical/scientific text processing, which heavily leverages the spaCy library. We detail the performance of two packages of models released in scispaCy and demonstrate their robustness on several tasks and datasets. Models and code are available at https://allenai.github.io/scispacy/.

1 Introduction

The publication rate in the medical and biomedical sciences is growing at an exponential rate (Bornmann and Mutz, 2014). The information overload problem is widespread across academia, but is particularly apparent in the biomedical sciences, where individual papers may contain specific discoveries relating to a dizzying variety of genes, drugs, and proteins. In order to cope with the sheer volume of new scientific knowledge, there have been many attempts to automate the process of extracting entities, relations, protein interactions and other structured knowledge from scientific papers (Wei et al., 2016; Ammar et al., 2018; Poon et al., 2014).

Although there exists a wealth of tools for processing biomedical text, many focus primarily on named entity recognition and disambiguation. MetaMap and MetaMapLite (Aronson, 2001; Demner-Fushman et al., 2017), the two most widely used and supported tools for biomedical text processing, support entity linking with negation detection and acronym resolution. However, tools which cover more classical natural language processing (NLP) tasks such as the GENIA tagger (Tsuruoka et al., 2005; Tsuruoka and Tsujii, 2005), or phrase structure parsers such as those presented in McClosky and Charniak (2008) typically do not make use of new research innovations such as word representations or neural networks.

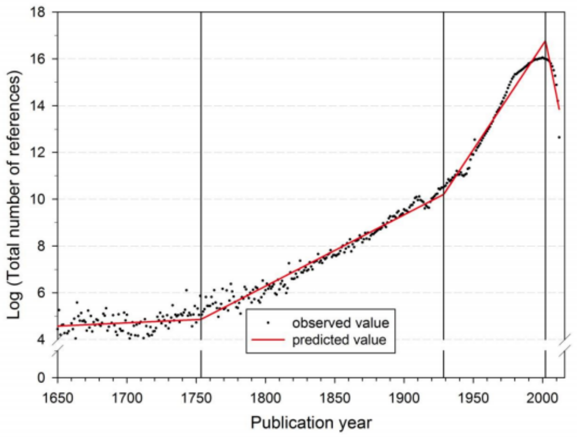

Figure 1: Growth of the annual number of cited references from 1650 to 2012 in the medical and health sciences (citing publications from 1980 to 2012). Figure from (Bornmann and Mutz, 2014).

In this paper, we introduce scispaCy, a specialized NLP library for processing biomedical texts which builds on the robust spaCy library,[1] and document its performance relative to state of the art models for part of speech (POS) tagging, dependency parsing, named entity recognition (NER) and sentence segmentation. Specifically, we:

- Release a reformatted version of the GENIA 1.0 (Kim et al., 2003) corpus converted into Universal Dependencies v1.0 and aligned with the original text from the PubMed abstracts.

[1] spacy.io

Proceedings of the BioNLP 2019 workshop, pages 319–327
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

Model	Vocab Size	Vector Count	Min Word Freq	Min Doc Freq
en_core_sci_sm	58,338	0	50	5
en_core_sci_md	101,678	98,131	20	5

Table 1: Vocabulary statistics for the two core packages in scispaCy.

- Benchmark 9 named entity recognition models for more specific entity extraction applications demonstrating competitive performance when compared to strong baselines.
- Release and evaluate two fast and convenient pipelines for biomedical text, which include tokenization, part of speech tagging, dependency parsing and named entity recognition.

2 Overview of (sci)spaCy

In this section, we briefly describe the models used in the spaCy library and describe how we build on them in scispaCy.

spaCy. The Python-based spaCy library (Honnibal and Montani, 2017)[2] provides a variety of practical tools for text processing in multiple languages. Their models have emerged as the defacto standard for practical NLP due to their speed, robustness and close to state of the art performance. As the spaCy models are popular and the spaCy API is widely known to many potential users, we choose to build upon the spaCy library for creating a biomedical text processing pipeline.

scispaCy. Our goal is to develop scispaCy as a robust, efficient and performant NLP library to satisfy the primary text processing needs in the biomedical domain. In this release of scispaCy, we retrain spaCy[3] models for POS tagging, dependency parsing, and NER using datasets relevant to biomedical text, and enhance the tokenization module with additional rules. scispaCy contains two core released packages: **en_core_sci_sm** and **en_core_sci_md**. Models in the **en_core_sci_md** package have a larger vocabulary and include word vectors, while those in **en_core_sci_sm** have a smaller vocabulary and do not include word vectors, as shown in Table 1.

	Processing Times Per	
Software Package	Abstract (ms)	Sentence (ms)
NLP4J (java)	19	2
Genia Tagger (c++)	73	3
Biaffine (TF)	272	29
Biaffine (TF + 12 CPUs)	72	7
jPTDP (Dynet)	905	97
Dexter v2.1.0	208	84
MetaMapLite v3.6.2	293	89
en_core_sci_sm	32	4
en_core_sci_md	33	4

Table 2: Wall clock comparison of different publicly available biomedical NLP pipelines. All experiments run on a single machine with 12 Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz and 62GB RAM. For the Biaffine Parser, a pre-compiled Tensorflow binary with support for AVX2 instructions was used in a good faith attempt to optimize the implementation. Dynet does support the Intel MKL, but requires compilation from scratch and as such, does not represent an "off the shelf" system. TF is short for Tensorflow.

Processing Speed. To emphasize the efficiency and practical utility of the end-to-end pipeline provided by scispaCy packages, we perform a speed comparison with several other publicly available processing pipelines for biomedical text using 10k randomly selected PubMed abstracts. We report results with and without segmenting the abstracts into sentences since some of the libraries (e.g., GENIA tagger) are designed to operate on sentences.

As shown in Table 2, both models released in scispaCy demonstrate competitive speed to pipelines written in C++ and Java, languages designed for production settings.

Whilst scispaCy is not as fast as pipelines designed for purely production use-cases (e.g., NLP4J), it has the benefit of straightforward integration with the large ecosystem of Python libraries for machine learning and text processing. Although the comparison in Table 2 is not an apples to apples comparison with other frameworks (different tasks, implementation languages etc), it is useful to understand scispaCy's runtime in the context of other pipeline components. Running scispaCy models *in addition to* standard Entity Linking software such as MetaMap would result in only a marginal increase in overall runtime.

In the following section, we describe the POS taggers and dependency parsers in scispaCy.

[2] Source code at `https://github.com/explosion/spaCy`

[3] scispaCy models are based on spaCy version 2.0.18

3 POS Tagging and Dependency Parsing

The joint POS tagging and dependency parsing model in spaCy is an arc-eager transition-based parser trained with a dynamic oracle, similar to Goldberg and Nivre (2012). Features are CNN representations of token features and shared across all pipeline models (Kiperwasser and Goldberg, 2016; Zhang and Weiss, 2016). Next, we describe the data we used to train it in scispaCy.

3.1 Datasets

GENIA 1.0 Dependencies. To train the dependency parser and part of speech tagger in both released models, we convert the treebank of McClosky and Charniak (2008),[4] which is based on the GENIA 1.0 corpus (Kim et al., 2003), to Universal Dependencies v1.0 using the Stanford Dependency Converter (Schuster and Manning, 2016). As this dataset has POS tags annotated, we use it to train the POS tagger jointly with the dependency parser in both released models.

As we believe the Universal Dependencies converted from the original GENIA 1.0 corpus are generally useful, we have released them as a separate contribution of this paper.[5] In this data release, we also align the converted dependency parses to their original text spans in the raw, untokenized abstracts from the original release,[6] and include the PubMed metadata for the abstracts which was discarded in the GENIA corpus released by McClosky and Charniak (2008). We hope that this raw format can emerge as a resource for practical evaluation in the biomedical domain of core NLP tasks such as tokenization, sentence segmentation and joint models of syntax.

Finally, we also retrieve from PubMed the original metadata associated with each abstract. This includes relevant named entities linked to their Medical Subject Headings (MeSH terms) as well as chemicals and drugs linked to a variety of ontologies, as well as author metadata, publication dates, citation statistics and journal metadata. We hope that the community can find interesting problems for which such natural supervision can be used.

Package/Model	GENIA
MarMoT	98.61
jPTDP-v1	98.66
NLP4J-POS	98.80
BiLSTM-CRF	98.44
BiLSTM-CRF- charcnn	98.89
BiLSTM-CRF - char lstm	98.85
en_core_sci_sm	98.38
en_core_sci_md	98.51

Table 3: Part of Speech tagging results on the GENIA Test set.

Package/Model	UAS	LAS
Stanford-NNdep	89.02	87.56
NLP4J-dep	90.25	88.87
jPTDP-v1	91.89	90.27
Stanford-Biaffine-v2	92.64	91.23
Stanford-Biaffine-v2(Gold POS)	92.84	91.92
en_core_sci_sm - SD	90.31	88.65
en_core_sci_md - SD	90.66	88.98
en_core_sci_sm	89.69	87.67
en_core_sci_md	90.60	88.79

Table 4: Dependency Parsing results on the GENIA 1.0 corpus converted to dependencies using the Stanford Universal Dependency Converter. We additionally provide evaluations using Stanford Dependencies(SD) in order for comparison relative to the results reported in (Nguyen and Verspoor, 2018).

OntoNotes 5.0. To increase the robustness of the dependency parser and POS tagger to generic text, we make use of the OntoNotes 5.0 corpus[7] when training the dependency parser and part of speech tagger (Weischedel et al., 2011; Hovy et al., 2006). The OntoNotes corpus consists of multiple genres of text, annotated with syntactic and semantic information, but we only use POS and dependency parsing annotations in this work.

3.2 Experiments

We compare our models to the recent survey study of dependency parsing and POS tagging for biomedical data (Nguyen and Verspoor, 2018) in Tables 3 and 4. POS tagging results show that both models released in scispaCy are competitive with state of the art systems, and can be considered of

[4] https://nlp.stanford.edu/~mcclosky/biomedical.html

[5] https://github.com/allenai/genia-dependency-trees

[6] Available at http://www.geniaproject.org/

[7] Instructions for download at http://cemantix.org/data/ontonotes.html

equivalent practical value. In the case of dependency parsing, we find that the Biaffine parser of Dozat and Manning (2016) outperforms the scispaCy models by a margin of 2-3%. However, as demonstrated in Table 2, the scispaCy models are approximately 9x faster due to the speed optimizations in spaCy. [8]

Robustness to Web Data. A core principle of the scispaCy models is that they are useful on a wide variety of types of text with a biomedical focus, such as clinical notes, academic papers, clinical trials reports and medical records. In order to make our models robust across a wider range of domains more generally, we experiment with incorporating training data from the OntoNotes 5.0 corpus when training the dependency parser and POS tagger. Figure 2 demonstrates the effectiveness of adding increasing percentages of web data, showing substantially improved performance on OntoNotes, at no reduction in performance on biomedical text. Note that mixing in web text during training has been applied to previous systems - the GENIA Tagger (Tsuruoka et al., 2005) also employs this technique.

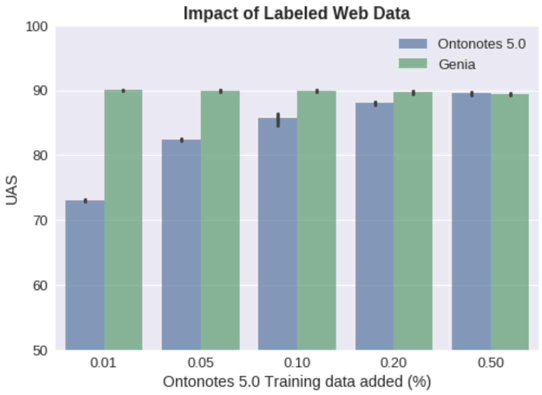

Figure 2: Unlabeled attachment score (UAS) performance for an **en_core_sci_md** model trained with increasing amounts of web data incorporated. Table shows mean of 3 random seeds.

4 Named Entity Recognition

The NER model in spaCy is a transition-based system based on the chunking model from Lample et al. (2016). Tokens are represented as hashed, embedded representations of the prefix, suffix, shape and lemmatized features of individual words. Next, we describe the data we used to train NER models in scispaCy.

4.1 Datasets

The main NER model in both released packages in scispaCy is trained on the mention spans in the MedMentions dataset (Murty et al., 2018). Since the MedMentions dataset was originally designed for entity linking, this model recognizes a wide variety of entity types, as well as non-standard syntactic phrases such as verbs and modifiers, but the model does not predict the entity type. In order to provide for users with more specific requirements around entity types, we release four additional packages **en_ner_{bc5cdr|craft|jnlpba|bionlp13cg}_md** with finer-grained NER models trained on BC5CDR (for chemicals and diseases; Li et al., 2016), CRAFT (for cell types, chemicals, proteins, genes; Bada et al., 2011), JNLPBA (for cell lines, cell types, DNAs, RNAs, proteins; Collier and Kim, 2004) and BioNLP13CG (for cancer genetics; Pyysalo et al., 2015), respectively.

4.2 Experiments

As NER is a key task for other biomedical text processing tasks, we conduct a through evaluation of the suitability of scispaCy to provide baseline performance across a wide variety of datasets. In particular, we retrain the spaCy NER model on each of the four datasets mentioned earlier (BC5CDR, CRAFT, JNLPBA, BioNLP13CG) as well as five more datasets in Crichton et al. (2017): AnatEM, BC2GM, BC4CHEMD, Linnaeus, NCBI-Disease. These datasets cover a wide variety of entity types required by different biomedical domains, including cancer genetics, disease-drug interactions, pathway analysis and trial population extraction. Additionally, they vary considerably in size and number of entities. For example, BC4CHEMD (Krallinger et al., 2015) has 84,310 annotations while Linnaeus (Gerner et al., 2009) only has 4,263. BioNLP13CG (Pyysalo et al., 2015) annotates 16 entity types while five of the datasets only annotate a single entity type.[9]

Table 5 provides a thorough comparison of the scispaCy NER models compared to a variety of models. In particular, we compare the models to

[8] We refer the interested reader to Nguyen and Verspoor (2018) for a comprehensive description of model architectures considered in this evaluation.

[9] For a detailed discussion of the datasets and their creation, we refer the reader to `https://github.com/cambridgeltl/MTL-Bioinformatics-2016/blob/master/Additional%20file%201.pdf`

strong baselines which do not consider the use of 1) multi-task learning across multiple datasets and 2) semi-supervised learning via large pretrained language models. Overall, we find that the scispaCy models are competitive baselines for 5 of the 9 datasets.

Additionally, in Table 6 we evaluate the recall of the pipeline mention detector available in both scispaCy models (trained on the MedMentions dataset) against all 9 specialised NER datasets. Overall, we observe a modest drop in average recall when compared directly to the MedMentions results in Table 7, but considering the diverse domains of the 9 specialised NER datasets, achieving this level of recall across datasets is already non-trivial.

Dataset	**sci_sm**	**sci_md**
BC5CDR	75.62	78.79
CRAFT	58.28	58.03
JNLPBA	67.33	70.36
BioNLP13CG	58.93	60.25
AnatEM	56.55	57.94
BC2GM	54.87	56.89
BC4CHEMD	60.60	60.75
Linnaeus	67.48	68.61
NCBI-Disease	65.76	65.65
Average	62.81	64.14

Table 6: Recall on the test sets of 9 specialist NER datasets, when the base mention detector is trained on MedMentions. The base mention detector is available in both **en_core_sci_sm** and **en_core_sci_md** models.

Model	Precision	Recall	F1
en_core_sci_sm	69.22	67.19	68.19
en_core_sci_md	70.44	67.56	68.97

Table 7: Performance of the base mention detector on the MedMentions Corpus.

5 Candidate Generation for Entity Linking

In addition to Named Entity Recognition, scispaCy contains some initial groundwork needed to build an Entity Linking model designed to link to a subset of the Unified Medical Language System (UMLS; Bodenreider, 2004). This reduced subset is comprised of sections 0, 1, 2 and 9 (SNOMED) of the UMLS 2017 AA release, which are publicly distributable. It contains 2.78M unique concepts and covers 99% of the mention concepts present in the MedMentions dataset (Murty et al., 2018).

5.1 Candidate Generation

To generate candidate entities for linking a given mention, we use an approximate nearest neighbours search over our subset of UMLS concepts and concept aliases and output the entities associated with the nearest K. Concepts and aliases are encoded using the vector of TF-IDF scores of character 3-grams which appears in 10 or more entity names or aliases (i.e., document frequency $\geq$ 10). In total, all data associated with the candidate generator including cached vectors for 2.78M concepts occupies 1.1GB of space on disk.

Aliases. Canonical concepts in UMLS have *aliases* - common names of drugs, alternative spellings, and otherwise words or phrases that are often linked to a given concept. Importantly, aliases may be shared across concepts, such as "cancer" for the canonical concepts of both "Lung Cancer" and "Breast Cancer". Since the nearest neighbor search is based on the surface forms, it returns K string values. However, because a given string may be an alias for multiple concepts, the list of K nearest neighbor strings may not translate to a list of K candidate entities. This is the correct implementation in practice, because given a possibly ambiguous alias, it is beneficial to score all plausible concepts, but it does mean that we cannot determine the exact number of candidate entities that will be generated for a given value of K. In practice, the number of retrieved candidates for a given K is much lower than K itself, with the exception of a few long tail aliases, which are aliases for a large number of concepts. For example, for K=100, we retrieve 54.26 ± 12.45 candidates, with the max number of candidates for a single mention being 164.

Abbreviations. During development of the candidate generator, we noticed that abbreviated mentions account for a substantial proportion of the failure cases where none of the generated candidates match the correct entity. To partially remedy this, we implement the unsupervised abbreviation detection algorithm of Schwartz and Hearst (2002), substituting mention candidates marked as abbreviations for their long form definitions before searching for their nearest neighbours. Figure 3 demonstrates the improved recall of gold concepts

Dataset	Baseline	SOTA	+ Resources	**sci_sm**	**sci_md**
BC5CDR (Li et al., 2016)	83.87	86.92b	89.69^{bb}	78.83	83.92
CRAFT (Bada et al., 2011)	79.55	-	-	72.31	76.17
JNLPBA (Collier and Kim, 2004)	68.95	73.48^{b}	75.50^{bb}	71.78	73.21
BioNLP13CG (Pyysalo et al., 2015)	76.74	-	-	72.98	77.60
AnatEM (Pyysalo and Ananiadou, 2014)	88.55	91.61^{**}	-	80.13	84.14
BC2GM (Smith et al., 2008)	84.41	80.51^{b}	81.69^{bb}	75.77	78.30
BC4CHEMD (Krallinger et al., 2015)	82.32	88.75^{a}	89.37^{aa}	82.24	84.55
Linnaeus (Gerner et al., 2009)	79.33	95.68^{**}	-	79.20	81.74
NCBI-Disease (Dogan et al., 2014)	77.82	85.80^{b}	87.34^{bb}	79.50	81.65

bb: LM model from Sachan et al. (2017) **b**: LSTM model from Sachan et al. (2017)
a: Single Task model from Wang et al. (2018) **aa**: Multi-task model from Wang et al. (2018)
** Evaluations use dictionaries developed without a clear train/test split.

Table 5: Test F1 Measure on NER for the small and medium scispaCy models compared to a variety of strong baselines and state of the art models. The **Baseline** and **SOTA** (State of the Art) columns include only single models which do not use additional resources, such as language models, or additional sources of supervision, such as multi-task learning. **+ Resources** allows any type of supervision or pretraining. All scispaCy results are the mean of 5 random seeds.

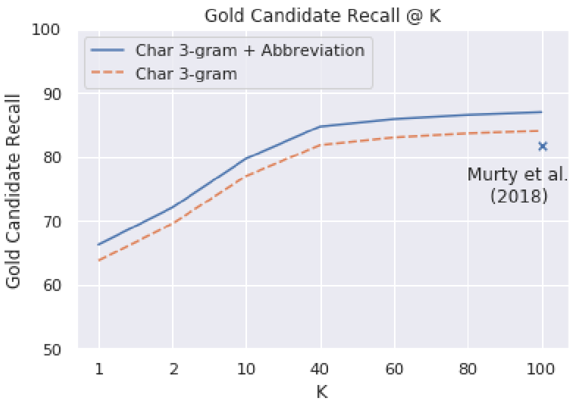

Figure 3: Gold Candidate Generation Recall for different values of K. Note that K refers to the number of nearest neighbour queries, and not the number of considered candidates. Murty et al. (2018) do not report this distinction, but for a given K the same amount of work is done (retrieving K neighbours from the index), so results are comparable. For all K, the actual number of candidates is considerably lower on average.

for various values of K nearest neighbours. Our candidate generator provides a 5% absolute improvement over Murty et al. (2018) despite generating 46% fewer candidates per mention on average.

6 Sentence Segmentation and Citation Handling

Accurate sentence segmentation is required for many practical applications of natural language processing. Biomedical data presents many difficulties for standard sentence segmentation algorithms: abbreviated names and noun compounds containing punctuation are more common, whilst the wide range of citation styles can easily be misidentified as sentence boundaries.

We evaluate sentence segmentation using both sentence and full-abstract accuracy when segmenting PubMed abstracts from the raw, untokenized GENIA development set (the **Sent/Abstract** columns in Table 8).

Additionally, we examine the ability of the segmentation learned by our model to generalise to the body text of PubMed articles. Body text is typically more complex than abstract text, but in particular, it contains citations, which are considerably less frequent in abstract text. In order to examine the effectiveness of our models in this scenario, we design the following synthetic experiment. Given sentences from Cohan et al. (2019) which were originally designed for citation intent prediction, we run these sentences individually through our models. As we know that these sentences should be single sentences, we can simply count the frequency with which our models segment the individual sentences containing citations into multiple sentences (the **Citation** column in Table 8).

As demonstrated by Table 8, training the dependency parser on in-domain data (both the scispaCy models) completely obviates the need for rule-based sentence segmentation. This is a positive result - rule based sentence segmentation is a brittle, time consuming process, which we have replaced with a domain specific version of an existing pipeline component.

Both scispaCy models are released with the custom tokeniser, but without a custom sentence segmenter by default.

Model	Sent	Abstract	Citation
web-small	88.2%	67.5%	74.4%
web-small + ct	86.6%	62.1%	88.6%
web-small + cs	91.9%	77.0%	87.5%
web-small + cs + ct	92.1%	78.3%	94.7%
sci-small + ct	97.2%	81.7%	97.9%
sci-small + cs + ct	97.2%	81.7%	98.0%
sci-med + ct	97.3%	81.7%	98.0%
sci-med + cs + ct	97.4%	81.7%	98.0%

Table 8: Sentence segmentation performance for the core spaCy and scispaCy models. **cs** = custom rule based sentence segmenter and **ct** = custom rule based tokenizer, both designed explicitly to handle citations and common patterns in biomedical text.

7 Related Work

Apache cTakes (Savova et al., 2010) was designed specifically for clinical notes rather than the broader biomedical domain. MetaMap and MetaMapLite (Aronson, 2001; Demner-Fushman et al., 2017) from the National Library of Medicine focus specifically on entity linking using the Unified Medical Language System (UMLS) (Bodenreider, 2004) as a knowledge base. Buyko et al. adapt Apache OpenNLP using the GENIA corpus, but their system is not openly available and is less suitable for modern, Python-based workflows. The GENIA Tagger (Tsuruoka et al., 2005) provides the closest comparison to scispaCy due to it's multi-stage pipeline, integrated research contributions and production quality runtime. We improve on the GENIA Tagger by adding a full dependency parser rather than just noun chunking, as well as improved results for NER without compromising significantly on speed.

In more fundamental NLP research, the GENIA corpus (Kim et al., 2003) has been widely used to evaluate transfer learning and domain adaptation. McClosky et al. (2006) demonstrate the effectiveness of self-training and parse re-ranking for domain adaptation. Rimell and Clark (2008) adapt a CCG parser using only POS and lexical categories, while Joshi et al. (2018) extend a neural phrase structure parser trained on web text to the biomedical domain with a small number of partially annotated examples. These papers focus mainly of the problem of domain adaptation itself, rather than the objective of obtaining a robust, high-performance parser using existing resources.

NLP techniques, and in particular, *distant supervision* have been employed to assist the curation of large, structured biomedical resources. Poon et al. (2015) extract 1.5 million cancer pathway interactions from PubMed abstracts, leading to the development of Literome (Poon et al., 2014), a search engine for genic pathway interactions and genotype-phenotype interactions. A fundamental aspect of Valenzuela-Escarcega et al. (2018) and Poon et al. (2014) is the use of handwritten rules and triggers for events based on dependency tree paths; the connection to the application of scispaCy is quite apparent.

8 Conclusion

In this paper we presented several robust model pipelines for a variety of natural language processing tasks focused on biomedical text. The scispaCy models are fast, easy to use, scalable, and achieve close to state of the art performance. We hope that the release of these models enables new applications in biomedical information extraction whilst making it easy to leverage high quality syntactic annotation for downstream tasks. Additionally, we released a reformatted GENIA 1.0 corpus augmented with automatically produced Universal Dependency annotations and recovered and aligned original abstract metadata. Future work on scispaCy will include a more fully featured entity linker built from the current candidate generation work, as well as other pipeline components such as negation detection commonly used in the clinical and biomedical natural language processing communities.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Chris Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *NAACL-HLT*.

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

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A. Baumgartner, K. Bretonnel Cohen, Karin M. Verspoor, Judith A. Blake, and Lawrence Hunter. 2011. Concept annotation in the CRAFT corpus. In *BMC Bioinformatics*.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32 Database issue:D267–70.

Lutz Bornmann and Rüdiger Mutz. 2014. Growth rates of modern science: A bibliometric analysis. *CoRR*, abs/1402.4578.

Ekaterina Buyko, Joachim Wermter, Michael Poprat, and Udo Hahn. Automatically adapting an NLP core engine to the biology domain. In *Proceedings of the ISMB 2006 Joint Linking Literature, Information and Knowledge for Biology and the 9th Bio-Ontologies Meeting*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. *CoRR*, abs/1904.01608.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *NLP-BA/BioNLP*.

Gamal K. O. Crichton, Sampo Pyysalo, Billy Chiu, and Anna Korhonen. 2017. A neural network multi-task learning approach to biomedical named entity recognition. In *BMC Bioinformatics*.

Dina Demner-Fushman, Willie J. Rogers, and Alan R. Aronson. 2017. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *Journal of the American Medical Informatics Association : JAMIA*, 24 4:841–844.

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

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *CoRR*, abs/1611.01734.

Martin Gerner, Goran Nenadic, and Casey M. Bergman. 2009. LINNAEUS: A species name identification system for biomedical literature. In *BMC Bioinformatics*.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Coling 2012*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Eduard H. Hovy, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, and Ralph M. Weischedel. 2006. OntoNotes: The 90% solution. In *HLT-NAACL*.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *ACL*.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:i180–2.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional lstm feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzábal, and Alfonso Valencia. 2015. CHEMDNER: The drugs and chemical names extraction challenge. In *J. Cheminformatics*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *HLT-NAACL*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database : the journal of biological databases and curation*, 2016.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *ACL*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *ACL*.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *ACL*.

Dat Quoc Nguyen and Karin Verspoor. 2018. From POS tagging to dependency parsing for biomedical event extraction. *arXiv preprint arXiv:1808.03731*.

Hoifung Poon, Chris Quirk, Charlie DeZiel, and David Heckerman. 2014. Literome: PubMed-scale genomic knowledge base in the cloud. *Bioinformatics*, 30 19:2840–2.

Hoifung Poon, Kristina Toutanova, and Chris Quirk. 2015. Distant supervision for cancer pathway extraction from text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 120–31.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. In *Bioinformatics*.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Jun'ichi Tsujii, and Sophia Ananiadou. 2015. Overview of the cancer genetics and pathway curation tasks of BioNLP shared task 2013. In *BMC Bioinformatics*.

Laura Rimell and Stephen Clark. 2008. Adapting a lexicalized-grammar parser to contrasting domains. In *EMNLP*.

Devendra Singh Sachan, Pengtao Xie, Mrinmaya Sachan, and Eric P. Xing. 2017. Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition. In *MLHC*.

Guergana K. Savova, James J. Masanz, Philip V. Ogren, Jiaping Zheng, Sunghwan Sohn, Karin 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 : JAMIA*, 17 5:507–13.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*.

Ariel S. Schwartz and Marti A. Hearst. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 451–62.

Larry Smith, Lorraine K. Tanabe, Rie Johnson nee Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Y Lin, Roman Klinger, Christoph M. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence E. Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter Adriaans, Christian Blaschke, Rafael Torres, Mariana Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology*, 9:S2 – S2.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT/EMNLP*.

Marco Antonio Valenzuela-Escarcega, Ozgun Babur, Gus Hahn-Powell, Dane Bell, Thomas Hicks, Enrique Noriega-Atala, Xia Wang, Mihai Surdeanu, Emek Demir, and Clayton T. Morrison. 2018. Large-scale automated machine reading discovers new cancer-driving mechanisms. In *Database*.

Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis P. Langlotz, and Jiawei Han. 2018. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu. 2016. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. *Database : the journal of biological databases and curation*, 2016.

Ralph Weischedel, Eduard Hovy, Martha Palmer, Mitch Marcus, Robert Belvin adn Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In Joseph Olive, Caitlin Christianson, and John McCary, editors, *Handbook of Natural Language Processing and Machine Translation*. Springer.

Yuan Zhang and David I Weiss. 2016. Stack-propagation: Improved representation learning for syntax. *CoRR*, abs/1603.06598.