Neural Dependency Parsing of Biomedical Text: TurkuNLP entry in the CRAFT Structural Annotation Task

Thang Minh Ngo, Jenna Kanerva, Filip Ginter, and Sampo Pyysalo
Turku NLP Group, Department of Future Technologies,
University of Turku, Finland
{first.last}@utu.fi

Abstract

We present the approach taken by the TurkuNLP group in the CRAFT Structural Annotation task, a shared task on dependency parsing. Our approach builds primarily on the Turku neural parser, a native dependency parser that ranked among the best in the recent CoNLL tasks on parsing Universal Dependencies. To adapt the parser to the biomedical domain, we considered and evaluated a number of approaches, including the generation of custom word embeddings, combination with other in-domain resources, and the incorporation of information from named entity recognition. We achieved a labeled attachment score of 89.7%, the best result among task participants.

1 Introduction

Syntactic analysis (parsing) is a fundamental task in natural language processing (NLP) and a prerequisite for many related tasks. There is a long tradition of research in automatic parsing targeting both constituency (phrase structure) and dependency representations, with most work focusing on the analysis of English news texts (Marcus et al., 1994). Syntactic analyses are required also by many methods for the analysis of biomedical text; for example, information extraction methods commonly rely on the shortest path over syntactic dependencies to identify how entities mentioned in text are related (Airola et al., 2008; Björne et al., 2009; Liu et al., 2013; Luo et al., 2016). The performance of parsers is known to be domain-dependent: to create high-quality analyses of e.g. biomedical texts, the tools should be trained on annotated corpora reflecting the domain (Miwa et al., 2010). Syntactically annotated corpora of domain texts are thus required for much of biomedical NLP. These resources should also preferably follow the relevant standards in the representation of syntactic analyses to allow methods developed to these standards to be applied also for biomedical domain texts, thus allowing biomedical NLP to benefit from advances in parsing technology.

The CRAFT Structural Annotation (SA) task, organized in 2019 is a shared task on dependency parsing largely following the setting of the popular Conference on Computational Natural Language Learning (CoNLL) 2017 and 2018 shared tasks on dependency parsing (Zeman et al., 2017, 2018). These tasks emphasize real-world scenarios by casting the task as analyzing raw text (rather than e.g. pre-tokenized and tagged text) and applying universal, language-independent representations. The CRAFT SA task follows these tasks in providing only plain text as input, requiring participating systems to perform sentence segmentation, tokenization, part-of-speech tagging, lemmatization, and the identification of morphological features in addition to analyzing the syntactic structure of the input sentences. CRAFT SA also adopts the format and evaluation tools of the CoNLL tasks, and its representation matches the universal representation of these tasks in part. The CRAFT task is differentiated from the many corpus resources applied in the CoNLL tasks specifically in focusing on biomedical domain texts, and CRAFT is unique among syntactically annotated biomedical corpora in that its texts are drawn from full-text articles, rather than only article titles and abstracts.

We participated in the CRAFT SA task using an approach that builds primarily on the Turku neural parser (Kanerva et al., 2018), a native dependency parsing system that previously ranked among the best systems in the CoNLL 2018 task. As the parser is fully retrainable, designed to accept the format used for the CRAFT data, and agnostic to the details of the representation, it was possible to train it for the CRAFT task with little modification. Additionally, as the parser has not been de-

Proceedings of the 5th Workshop on BioNLP Open Shared Tasks, pages 206–215

veloped or previously applied to biomedical English, we consider a number of modifications and adaptions to improve on its performance, finding in particular that the strong baseline performance of the parser can be further improved through initialization with in-domain word vectors.

2 Background

Biomedical domain models have been available for a number of constituency parsers (e.g. Charniak and Johnson (2005), McClosky and Charniak (2008)) and have been widely applied in domain information extraction efforts, frequently in conjunction with heuristic conversions into dependency representations such as Stanford dependencies (De Marneffe and Manning, 2008). There have also been native dependency parsers available for the domain, such as Pro3Gres (Schneider and Rinaldi, 2004) and, later, GDep (Miyao et al., 2008), nevertheless the abovementioned McClosky-Charniak parser with Stanford dependencies conversion was the workhorse of biomedical dependency parsing for nearly a decade. Also the treebanks available for training the parsers in the biomedical domain have traditionally been constituency-based, for instance the Penn BioIE (Kulick et al., 2004) and especially the GENIA treebank (Tateisi et al., 2005). The BioInfer corpus (Pyysalo et al., 2007) was the first domain corpus to adopt Stanford Dependencies as the native annotation scheme, coinciding with a generally growing interest in dependency parsing and its applications.

The CoNLL 2006 and 2007 shared tasks (Buchholz and Marsi, 2006; Nivre et al., 2007) addressed multilingual dependency parsing, and while data was provided for different languages in the same format, the underlying representation (e.g. dependency types) was not standardized in these tasks. These tasks also included only prediction of syntactic trees, whereas tokenization and part-of-speech tags were given for the participants.

In recent years, there has been an increased interest in native dependency parsing, reflected in efforts such as Universal Dependencies (UD) (Nivre et al., 2016) and the CoNLL 2017 and 2018 shared tasks on multilingual parsing using UD data (Zeman et al., 2017, 2018). While these efforts have covered a wide range of languages, genres and text domains, and introduced end-to-end parsing from plain text as the objective, they have not specifically involved scientific articles or biomedical domain texts.

	Train	Test
Documents	67	30
Sentences	21 731	9 099
Tokens	561 032	232 619

Table 1: CRAFT Structural Annotation statistics

	Train	Devel	Eval
Documents	47	10	10
Sentences	15 007	3 421	3 303
Tokens	387 473	91 306	82 253

Table 2: CRAFT Train data split for development

3 Data

3.1 CRAFT data

The primary resource used for training systems for the task is the CRAFT corpus syntactic annotation provided by the task organizers. Table 1 summarizes the key statistics of the data.

The test annotations were only made available after participants had submitted their predictions, and no train/development split was defined for the provided data. For development purposes, we thus split the provided training dataset of 67 documents randomly into a set of 47 used for training, a devel set of 10 used for early stopping during training, and 10 used for evaluation during development. The statistics of this split are shown in Table 2

The original CRAFT corpus syntactic annotation uses a modified Penn Treebank (PTB) constituency formalism (Verspoor et al., 2012), and the dependency annotation provided for the task was automatically created by conversion from the constituency representation. The source data was first converted into the CoNLL-X format using the SD dependency representation and PTB POS tags using the approach of (Choi and Palmer, 2012), and this data was then further converted into the CoNLL-U format with custom scripts.

The resulting task dataset is in the UD *format* (CoNLL-U), but it only partially follows the UD standard in terms of its content. In particular, while the POS tags and morphological features conform to UD, the dependency representation – arguably the most important part of the data – does not, instead matching the SD representation of the CoNLL-X version of the data. Figure 1 shows SD

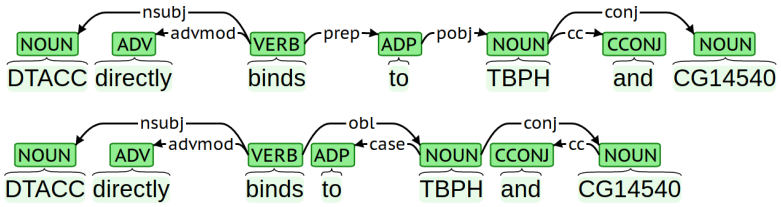

Figure 1: Illustration of Stanford Dependencies (top) and Universal Dependencies (bottom) analyses for an example sentence (from PMCID:15207008). The CRAFT dependency annotation follows the former representation.

and UD analyses for an example sentence from the CRAFT data. While a number of dependencies are identical between the two (e.g. `nsubj`), in UD dependencies primarily relate content words (e.g. verbs and nouns), with function words such as adpositions being dependents of content words rather than mediating their relations such as in SD (cf. *binds to TBPH* in Figure 1). There are also a number of minor differences such as the attachment of coordinating conjunctions to the first constituent in SD and the nearest to the right in UD.

While this discrepancy does not prevent the use of tools that are agnostic to the details of the representation (including many UD parsers), it does mean that the data is incompatible with existing UD resources and greatly complicates combination with other corpora, none of which are available in this particular hybrid SD/UD CoNLL-U representation. We expand on this issue below in Section 6.2.

3.2 Word vectors

We considered a number of previously released word vectors for initializing the parser. As a baseline we use the English word embeddings by Ginter et al. (2017) trained on general English extracted from Wikipedia and Internet crawls. These embeddings are trained using the word2vec (Mikolov et al., 2013) tool with lowercased data, skip-gram algorithm, window size of 10 and 100 dimensions. The vectors were originally provided for the CoNLL 2017 and 2018 multilingual parsing shared task, and thus used by many of the participating systems in their English parsing models. We also considered a number of word vectors induced specifically on biomedical text for domain tasks, including those created by Pyysalo et al. (2013)[1] and Chiu et al. (2016)[2].

[1] http://bio.nlplab.org/

[2] https://github.com/cambridgeltl/BioNLP-2016

3.3 Unlabelled data

To induce new word vectors (Section 4.3) and conduct co-training experiments (Section 5.2), we used unlabelled texts from PubMed titles and abstracts and PubMed Central (PMC) full texts. The data was drawn from the PubMed 2017 baseline distribution and a 2017 download of the PMC Open Access subset.[3] The texts were segmented into sentences using the GENIA sentence splitter and then tokenized using the PTBTokenizer included in Stanford CoreNLP tools (Manning et al., 2014) and the tokenized sentences shuffled randomly. The resulting dataset consists of 12.5 billion tokens in 500 million sentences. As the text of the full-text articles of the CRAFT corpus contains characters outside of the basic ASCII character set, we created word vectors on the original extracted texts instead of first applying a mapping to ASCII characters as was done in a number of similar previous efforts (e.g. (Pyysalo et al., 2013)).

4 Methods

4.1 Turku Parser

Our primary parser used in all experiments is the Turku Neural Parser Pipeline[4] (Kanerva et al., 2018), a full parser pipeline meant for end-to-end analysis from raw text into UD. The pipeline includes sentence and word segmentation, part-of-speech and morphological tagging, syntactic parsing, and lemmatization.

The segmentation component in the Turku pipeline is built using UDPipe (Straka and Straková, 2017), where the token and sentence boundaries are jointly predicted using a single-layer bidirectional GRU network. Universal (`UPOS`) and language-specific (`XPOS`) part-of-speech tags, as well as morphological features

[3] We used 2017 data as we had a plain text version readily available from previous work.

[4] https://turkunlp.org/Turku-neural-parser-pipeline/

(FEATS) are predicted with a modified version of the one published by Dozat et al. (2017), a time-distributed classifier over tokens in a sentence embedded using bidirectional LSTM network. The tagger has two separate classification layers, one for universal part-of-speech and one originally used for language-specific part-of-speech tags. The bidirectional encoding is shared between both classifiers. In the modified version (Kanerva et al., 2018), the second classifier is used to jointly predict the language-specific POS tags together with morphological features by simply concatenating the two input columns into one. The syntactic analysis is based on a graph-based parser by Dozat et al. (2017), a biaffine classifier with MST decoder on top of a bidirectional LSTM network. The lemmatizer component by Kanerva et al. (2019) is a sequence-to-sequence model, where the lemma is generated one character at a time from the given input word form and morphological features.

In the Turku parser pipeline, all these components are wrapped into a single system. All components directly supports training with CoNNL-U formatted treebanks while being completely label agnostic, thus not requiring the treebank to actually follow the UD guidelines and label sets. Therefore, the parser can be trained on CRAFT corpus as is. The Turku Parser was ranked second on LAS and MLAS, and first on BLEX on the CoNLL-2018 Shared Task, making it highly competitive.

4.2 UDPipe

UDPipe[5] (Straka and Straková, 2017) is an easily trainable parsing pipeline including segmentation, morphological tagging, lemmatization and syntactic parsing. UDPipe has long been the "go-to" UD parser and has also served as the organizers' baseline in the 2017 and 2018 CoNLL Shared Tasks on Multilingual Parsing from Raw Text to Universal Dependencies. Tokenization and sentence segmentation is implemented jointly, using a single-layer GRU network, predicting for each character whether it is a sentence boundary, token boundary, or token-internal. The tagger is an averaged perceptron disambiguating from a set of candidate analyses generated based on the last four characters of the word. Lemmatization is carried out by generating a set of candidate lemma rules, each of which produce a lemma by removing and possibly substituting characters from the word prefix and suffix. As in tagging, an averaged perceptron then disambiguates among the candidates. The dependency parser is a transition-based parser with a feed-forward neural network serving as the classifier that decides on the next transition taken by the parser.

4.3 Word vectors

For inducing new sets of word vectors, we used the word2vec[6] (Mikolov et al., 2013) and FastText[7] (Joulin et al., 2016; Bojanowski et al., 2017) tools. In brief, these tools generate a vector representation for each token based on the similarity of the contexts in which they appear in a large corpus of unannotated text. Word vectors were induced on texts extracted from PubMed abstracts and PMC Open Access publications (Section 3.3) using both the skip-gram and continuous bag-of-words (CBOW) models implemented in both tools. Model parameters were primarily kept at their default values, but we performed a series of experiments with different values of the window parameter, which has been found to be particularly impactful in previous work (Chiu et al., 2016).

4.4 Evaluation

The CRAFT SA shared task adopted the evaluation metrics and evaluation implementation of the CoNLL'18 shared task. In particular, performance was evaluated in terms of the Labeled attachment score (LAS), Morphology-aware labeled attachment score (MLAS), and Bi-lexical dependency score (BLEX) metrics, defined as follows (Zeman et al., 2018):

LAS The percentage of nodes having correctly assigned parent token, as well as correct type of the dependency relation. All tokens are considered in the evaluation, including also punctuation.

MLAS Similar to LAS, but with an additional requirement of having also functional dependents and certain morphological features predicted correctly. In addition the metric is calculated only based on content bearing words discarding functional words and punctuation. Thus, MLAS measures the percentage of content words having correctly assigned parent token, relation type, func-

[5] http://ufal.mff.cuni.cz/udpipe

[6] https://github.com/tmikolov/word2vec

[7] https://fasttext.cc/

Parser	Word vectors	LAS
Turku	Bio, word2vec/CBOW (window 2)	89.86
Turku	Bio (CRAFT tokens), word2vec/CBOW (default parameters)	89.78
Turku	Bio (CRAFT tokens), word2vec/CBOW (win2)	89.69
Turku	Bio, word2vec/CBOW (default parameters)	89.55
Turku	Bio, word2vec/CBOW (window 20)	89.73
Turku	Bio, FastText/CBOW (default parameters)	89.50
Turku	Bio, word2vec/skipgram (default parameters)	89.63
Turku	CoNLL	89.27
UDPipe	Bio, word2vec/CBOW (window 2)	85.00
UDPipe	CoNLL	84.66
UDPipe	Bio, word2vec/CBOW (default parameters)	84.22

Table 3: Development set results with different word vectors. CoNLL = baseline CoNLL shared task word vectors, Bio = custom word vectors induced on PubMed and PMC articles, CRAFT tokens = input text tokenized with model trained on CRAFT data. (For details on the word2vec and FastText tools, their CBOW and skipgram models, and parameters, see Section 3.2)

tional dependents and certain morphological features.

BLEX The proportion of correct relations between two content bearing words with an additional requirement that the lemma of the dependent must be correct. Functional words and punctuation tokens are discarded.

As LAS is the best established and most frequently applied of these metrics, we focused on optimizing this metric during development and report results for experiments conducted during development in terms of LAS only. For the final three test set submissions, we provide results for the full set of metrics implemented in the CoNLL evaluation script. In addition to the three metrics above, this includes measures of token, sentence and word segmentation agreement with gold (**Tokens**, **Sentences** and **Words** metrics), agreement of the universal (**UPOS**) and language-specific (**XPOS**) part-of-speech tags and morphological features (**UFeats**), the three previous together (**AllTags**), and agreement on lemmas (**Lemmas**). We refer to Zeman et al. (2018) for further details on these additional metrics.

We note that in the CRAFT test set evaluation, performance for each metric was calculated as an average of the results for the 30 test set documents, rather than over the catenation of the documents as in the CoNLL evaluation.

5 Results

During the development of our system, we considered a number of approaches in an iterative and incremental process. In this section, we first present the strategies we found effective, namely the use of custom in-domain word vectors and data augmentation. We then present the results from our three test set submissions and an analysis of these results using various additional metrics.

5.1 Word vectors

A simple but highly effective way to adapt machine learning systems that operate on vector representations of words to new domains is to initialize them with word embeddings induced on domain texts. We evaluated a variety of previously introduced and newly induced word embeddings in this way (see above) using both the Turku and UDPipe parsers, and summarize results for notable baseline vectors and selected in-domain word vectors in Table 3.

We find that using the general out-of-domain CoNLL word vectors, the parsers already achieve high baseline LAS scores, 84.66% for UDPipe and 89.27% for our primary, Turku system. In our limited experiments with UDPipe we found somewhat mixed results from the use of custom biomedical domain word vectors. For the Turku parser, a number of the in-domain word embeddings did prove effective, with the best-performing combination of data preprocessing, model and parameters achieving a LAS of 89.86%, a 5% relative reduction in LAS error from the CoNLL word vector baseline. Regarding the alternative settings for inducing word vectors, we broadly found CBOW to be more effective than the skipgram model and small windows to be more effective than either default parameters or large windows. We did not see an advantage of FastText over word2vec vectors and conducted the majority of our experiments with the latter tool.

Two of the runs submitted for the final evaluation used settings from these experiments, namely

Parser	Word vectors	Extra data (size, source)	LAS
Turku	Bio, word2vec/CBOW (window 2)	4k sentences, PMC	89.92
Turku	Bio, word2vec/CBOW (window 2)	10k sentences, PMC and PubMed	89.87
Turku	Bio, word2vec/CBOW (window 2)	6k sentences, PubMed	89.78
Turku	Bio, word2vec/CBOW (window 2)	10k sentences, PMC	89.84
Turku	Bio, word2vec/CBOW (window 2)	20k sentences, PMC	89.41

Table 4: Development set results with extra training data

Bio, word2vec/CBOW (window 2) and CRAFT tokenized, word2vec/CBOW (default parameters).

5.2 Training data augmentation

After identifying the word vectors that achieved the highest LAS score for this data, we implemented and evaluated multiple techniques to increase the number of training examples beyond the given training data. Most of the approaches we considered failed to improve on performance, largely due to incompatibilities in annotation (see Section 6.1), but we found limited success with a co-training approach (Blum and Mitchell, 1998).

Specifically, we first used the best Turku and UDPipe parser models introduced in our previous experiments to analyze a large sample of unannotated text from PubMed abstracts and PMC full text articles. We then compared the results to identify sentences that are identically segmented and tokenized and given identical syntactic analyses (heads and dependency relations) by the two systems. We then created random samples of varying sources and sizes from this data, generating comparatively high-quality automatically annotated additional training data. This data was combined with the original CRAFT training data to create an extended training set that was then used to create a new model with the Turku parser. We present a selection of development results from this setting in Table 4.

While we achieved some minor improvements in some of the experiments, the co-training approach did not improve the performance as system as much as could be hoped based on e.g. the effectiveness of self-training for parsing (McClosky et al., 2006). There may be a number of reasons for the limited effectiveness of our approach, potentially including sub-domain mismatch between our unlimited samples of PubMed and PMC documents and the comparatively narrow and focused domain of CRAFT texts. We nevertheless chose to include the model with the best result in these experiments with **4k sentences, PMC** as extra training data to include in our final submissions.

5.3 Test set results

The properties of the three runs we submitted to the task are summarized in Table 5 together with their development and test set LAS scores. We find that test set performance closely follows the results of development experiments, producing the same ranking of the three runs as well as results within 0.3% points of the development results in all three cases.

As expected on the basis of the development experiments, the two runs without extra training data are highly competitive, and augmenting the training data via co-training while keeping the word vectors constant provides only a modest benefit. Nevertheless, the run that combined custom in-domain word vectors and co-training to adapt the Turku parser to biomedical text achieved the highest performance not only among our runs but also out of all six runs submitted to the task.

5.4 Analysis of final results

Table 6 provides a detailed look at the performance of our three final submissions using all metrics implemented in the CoNLL 2018 shared task evaluation script (see Section 4.4). All of the metrics are averaged F1 scores across all 30 test files.

We find very similar results across all three runs. Segmentation performance is acceptable for sentence splitting (over 97.5%) and very high for tokenization (over 99.5%), indicating limited remaining benefit from further focus on identifying sentence and token boundaries. Part-of-speech tags (UPOS and XPOS) as well as morphological features are each assigned at a high level of consistency (approx98% each), and lemmas are correctly identified in approx. 99% of cases, indicating that the parser is well adapted to the challenges of specialized biomedical domain terminology. The only metrics showing notable remaining room for improvement are dependency-based (last five rows in Table 6). The relatively close results for the unlabeled and labeled attachment score metrics (UAS and LAS) indicate that the identification of the correct dependency relation is not a key factor

Parser	Word vectors	Extra data	LAS(dev)	LAS(test)
Turku	Bio, word2vec/CBOW (window 2)	4k sentences, pmcoa articles	89.92	89.695
Turku	Bio, word2vec/CBOW (window 2)	No	89.86	89.650
Turku	Bio (CRAFT tokens), word2vec/CBOW (defaults)	No	89.78	89.536

Table 5: Final submission results on test data

Metrics	Run 1	Run 2	Run 3
Tokens	99.593	99.555	99.593
Sentences	97.590	97.621	97.590
Words	99.593	99.555	99.593
UPOS	98.221	98.179	98.184
XPOS	97.806	97.758	97.789
UFeats	98.282	98.233	98.265
AllTags	97.752	97.718	97.729
Lemmas	98.999	98.981	99.048
UAS	90.942	90.882	90.794
LAS	89.695	89.650	89.536
CLAS	87.373	87.294	87.201
MLAS	85.549	85.441	85.318
BLEX	86.630	86.595	86.544

Table 6: Final submission test results for all metrics

limiting the performance of the parser, and that the remaining challenges for substantially advancing the performance of the system lie specifically in more accurately recovering the dependency structure of the sentences.

6 Discussion

In the following, we briefly discuss a number of ideas we considered that failed to improve on the performance of the parser and address the relationship between the CRAFT SA task data and Universal Dependencies.

6.1 What did not work

During the relatively brief development period for participating in the shared task, we considered a number of variants and potential extensions of our approach that failed to improve on the performance of the system. Although these were not developed and evaluated with the rigor required to report full experimental results, we summarize some of these ideas here in the hope that they may help others in their work.

Corpus combinations As the CRAFT dependency annotations were created by automatic conversion from PTB source, we considered the possibility of combining the task training data with additional similarly converted annotations. We performed several preliminary experiments converting the PTB Wall Street Journal section (Marcus et al., 1994) and the original GENIA treebank data (Tateisi et al., 2005) as well as a version of the GENIA treebank that as previously converted using the Stanford Dependency Converter.[8] The results of these experiments were disappointing; initial single-corpus experiments using the converted data failed to reach the expected level of performance, and all combinations of this data with CRAFT data resulted in decreased performance. We also initially considered attempting combinations with English corpora from the Universal Dependencies collection, but abandoned this idea due to incompatibilities in the representations (see below).

Entity mentions As the CRAFT corpus annotation integrates not only syntactic but also entity mention (or concept) annotation, there is an opportunity to integrate information on named entities and related concepts into the parsing process.[9] Briefly, the intuition is that a model that has information on which tokens are e.g. part of chemical or species names could better parse their mentions and associated text. To explore this idea, we converted the CRAFT concept annotation into a token-level begin-in-out (BIO) representation using custom tools, and appended these annotations into the XPOS column of the CoNLL-U data, creating merged POS and entity tags. We then trained on this data, creating joint models that integrate dependency parsing and entity mention information. However, the performance of these models was mixed, with minimal improvements in few cases and a reduction in LAS in others, and we chose not to pursue the idea further.

Previously introduced in-domain word embeddings Throughout development, we evaluated many word vectors, including both previously introduced and newly induced as well as biomedical domain and out-of-domain embeddings. The

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

[9] This idea was also advanced by the organizers in the CRAFT SA task description.

general pattern we found was the vectors introduced for the CoNLL shared task represented a very strong baseline, and many in-domain word vectors previously made available by the biomedical NLP community (including ones previously introduced by some of the authors) failed to improve on the results achieved with these vectors. We were only consistently able to improve over the CoNLL word vector baseline by newly inducing custom in-domain word vectors for the parser. We attribute some of this effect to the differences in the dimensionality of previously introduced vectors: although the parser can be configured to accept vectors of any size, some part of its development may have specifically optimized for the 100-dimensional CoNLL word vectors. It is also likely that part of the effect is explained by the presence of non-ASCII characters in the CRAFT data, as many in-domain word vectors were created on texts specifically mapped to ASCII as a pre-processing step.

6.2 CRAFT and Universal Dependencies

Universal Dependencies have become the *de facto* standard representation for computational dependency parsing, and the UD repository[10], containing over 100 UD treebanks covering more than 70 languages as of this writing, is a key interface connecting corpus creators and researchers working on parsing technology. There are several potential benefits to a biomedical domain UD corpus, especially the potential for combining existing English resources and domain transfer techniques. However, the CRAFT Structural Annotation shared task dataset differs from UD standards and conventions on a number of points, hindering its adoption as a UD resource.

Most obviously, despite being provided in the CoNLL-U format, the CRAFT data does not fully adopt UD types and annotation conventions. As noted above, the dependency relation types are drawn from a predecessor of UD, Stanford dependencies (SD), and the dependency annotation similarly follows SD rather than UD conventions. While the SD and UD representations are quite similar in many ways, they differ systematically in particular in that UD emphasizes content words over function words (see also Figure 1) and diverge in many details of the representation.

We also noted that the lemmas in the CRAFT data don't always correspond to the canonical (or base) forms of the words. In addition to numbers expressed as digits all having the lemma value "0", spelled-out cardinal numbers (e.g. "one") have the value "#crd#" in place of a lemma, ordinal numbers (e.g. "first") have "#ord#", and hyperlinks (e.g. "http://www.ncbi.nlm.nih.gov/") have "#hlink#". These exceptions are not part of UD and contrary to the representation of lemmas in existing English UD resources.

Based on our experience with the SD and UD representations and in creating UD corpora by conversion from other formats, we believe it should be possible to automatically convert the present CRAFT corpus annotations into a full UD representation using a combination of existing tools and some deterministic mappings addressing issues specific to this data. Such conversion would allow the inclusion of the corpus in the UD repository, increasing the availability of biomedical English training data to the parsing community.

7 Conclusions

In this paper, we have presented the approach of the TurkuNLP team to the CRAFT SA dependency parsing shared task. Building on the basis of the Turku neural parser and UDPipe, we considered a number of modifications and adaptations to better address the full-text biomedical domain articles of the task, including the induction of custom word vectors and the extension of the training data with additional automatically parsed data. Experiments showed the Turku parser to clearly outperform the UDPipe baseline at the task and demonstrated that initializing the parser with custom in-domain word vectors could further improve on its strong off-the-shelf performance. Our adapted version of the Turku parser achieved the highest result on the test set of the shared task with a labeled attachment score of 89.7%.

All of the tools and resources applied in this work, as well as the newly trained parsing models, are made available under open licenses.

Acknowledgments

We thank the task organizers for their help and responsiveness to feedback during the development period. We are grateful to CSC – IT Center for Science for computational resources used to train our models.

[10] https://universaldependencies.org/

References

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, 9(11):S2.

Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2009. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of BioNLP Shared Task*, pages 10–18.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning*, pages 149–164. Association for Computational Linguistics.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 173–180. Association for Computational Linguistics.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174.

Jinho D Choi and Martha Palmer. 2012. Guidelines for the clear style constituent to dependency conversion. *Technical Report 01–12*.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. The stanford typed dependencies representation. In *Coling 2008: proceedings of the workshop on cross-framework and cross-domain parser evaluation*, pages 1–8. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanfords graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. 2017. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku Neural Parser Pipeline: An end-to-end system for the CoNLL 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2019. Universal lemmatizer: A sequence to sequence model for lemmatizing universal dependencies treebanks. *arXiv preprint arXiv:1902.00972*.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, Lyle Ungar, Scott Winters, and Pete White. 2004. Integrated annotation for biomedical information extraction. In *HLT-NAACL 2004 Workshop: Linking Biological Literature, Ontologies and Databases*, pages 61–68.

Haibin Liu, Lawrence Hunter, Vlado Kešelj, and Karin Verspoor. 2013. Approximate subgraph matching-based literature mining for biomedical events and relations. *PloS one*, 8(4):e60954.

Yuan Luo, Özlem Uzuner, and Peter Szolovits. 2016. Bridging semantics and syntax with graph algorithmsstate-of-the-art of extracting biomedical relations. *Briefings in bioinformatics*, 18(1):160–178.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 101–104. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of NAACL*, pages 152–159.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Makoto Miwa, Sampo Pyysalo, Tadayoshi Hara, and Jun'ichi Tsujii. 2010. A comparative study of syntactic parsers for event extraction. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 37–45. Association for Computational Linguistics.

Yusuke Miyao, Rune Sætre, Kenji Sagae, Takuya Matsuzaki, and Junichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proceedings of ACL*, pages 46–54.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Sampo Pyysalo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen, and Tapio Salakoski. 2007. On the unification of syntactic annotations under the stanford dependency scheme: A case study on BioInfer and GENIA. In *ACL'07 workshop on Biological, translational, and clinical language processing (BioNLP'07)*, pages 25–32. Association for Computational Linguistics.

Sampo Pyysalo, Filip Ginter, Hand Moen, Sophia Ananiadou, and Tapio Salakoski. 2013. Distributional semantics resources for biomedical text processing. *Proceedings of LBM*, pages 39–44.

Gerold Schneider and James Rinaldi, Fabio; Dowdall. 2004. Fast, deep-linguistic statistical minimalist dependency parsing. In *COLING-2004 Recent Advances in Dependency Grammars*, pages 33–40.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Junichi Tsujii. 2005. Syntax annotation for the genia corpus. In *Proceedings of IJCNLP'05*.

Karin Verspoor, Kevin Bretonnel Cohen, Arrick Lanfranchi, Colin Warner, Helen L Johnson, Christophe Roeder, Jinho D Choi, Christopher Funk, Yuriy Malenkiy, Miriam Eckert, et al. 2012. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC bioinformatics*, 13(1):207.

Daniel Zeman, Jan Hajič, 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.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. 2017. Conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task Multilingual Parsing from Raw Text to Universal Dependencies*.