The 2018 Shared Task on Extrinsic Parser Evaluation: On the Downstream Utility of English Universal Dependency Parsers

Murhaf Fares[♣] **Stephan Oepen**[♣], **Lilja Øvrelid**[♣], **Jari Björne**[♠], **Richard Johansson**[♡]

[♣] University of Oslo, Department of Informatics

[♠] University of Turku, Department of Future Technologies

[♡] Chalmers Technical University and University of Gothenburg, Department of Computer Science and Engineering

epe-organizers@nlpl.eu

Abstract

We summarize empirical results and tentative conclusions from the Second Extrinsic Parser Evaluation Initiative (EPE 2018). We review the basic task setup, downstream applications involved, and end-to-end results for seventeen participating parsers. Based on both quantitative and qualitative analysis, we correlate intrinsic evaluation results at different layers of morph-syntactic analysis with observed downstream behavior.

1 Background and Motivation

The Second Extrinsic Parser Evaluation Initiative (EPE 2018) was organized as an optional track of the 2018 Shared Task on Multilingual Parsing from Raw Text to Universal Dependencies (Zeman et al., 2018) at the Conference on Computational Natural Language Learning (CoNLL 2018). In the following, we distinguish the tracks as the EPE vs. the 'core' UD parsing tasks, respectively. One focus of the UD parsing task in 2018 was on different intrinsic evaluation metrics, such that the connection to the EPE framework provides new opportunities for correlating *intrinsic* metrics with *downstream* utility to three relevant applications, viz. biological event extraction, fine-grained opinion analysis, and negation resolution. Unlike the strongly multilingual core task, the EPE framework for the time being is limited to English.

A previous instance of the EPE initiative (see § 2 below) embraced diversity and accepted submissions of parser outputs that varied along several dimensions, including different types of syntactic or semantic dependency representations, variable parser training data in type and volume, and of course diverse approaches to input segmentation and parsing. In contrast, the association of EPE 2018 with the UD parsing task 'fixes' two of these dimensions: All submitted systems output basic Universal Dependency (UD; McDonald et al., 2013; Nivre et al., 2016) trees (following the conventions of UD version 2.x) and parser training data was limited to the English UD treebanks provided for the core task.

2 History: The EPE 2017 Infrastructure

What we somewhat interchangeably refer to as the EPE framework or the EPE infrastructure was originally assembled in mid-2017, to enable the First Shared Task on Extrinsic Parser Evaluation (EPE 2017; Oepen et al., 2017), which was organized as a joint event by the Fourth International Conference on Dependency Linguistics (DepLing 2017) and the 15th International Conference on Parsing Technologies (IWPT 2017). The framework is characterized by a collection of 'downstream' natural language 'understanding' applications that are assumed to depend on the analysis of grammatical structure. For each downstream application, there are commonly used reference data sets (often from past shared tasks) and evaluation metrics. In the EPE context, state-of-the-art systems for these applications have been generalized to accept as inputs a broad variety of syntactico-semantic dependency representations (i.e. parser outputs submitted for extrinsic evaluation) and to automatically retrain (and tune, to some degree) for each specific parser. The following paragraphs briefly summarize each of the downstream systems and main results from the EPE 2017 competition.

Dependency Representations For compatibility with different linguistic schools in syntactico-semantic analysis, the EPE framework assumes a comparatively broad definition of suitable interface representations to grammatical analysis (Oepen et al., 2017; p. 6):

> The term (bi-lexical) dependency representation in the context of EPE 2017 is interpreted as a graph whose nodes are anchored in surface lexical units, and whose edges represent labeled directed relations between two nodes. Each node corresponds to a sub-string of the underlying linguistic signal (input string), identified by character stand-off pointers. Node labels can comprise a non-recursive attribute–value matrix (or 'feature structure'), for example to encode lemma and part of speech information. Each graph can optionally designate one or more 'top' nodes, broadly interpreted as the root-level head or highest-scoping predicate (Kuhlmann and Oepen, 2016).

In principle, this notion of dependency representations is broad in that it allows nodes that do not correspond to (full) surface tokens, partial or full overlap of nodes, as well as graphs that transcend fully connected rooted trees. Participating teams in the original EPE 2017 initiative did in fact take advantage of all these degrees of freedom, whereas in connection to the 2018 UD parsing task such variation is excluded by design.

Biological Event Extraction The Turku Event Extraction System (TEES) (Björne, 2014) is a program developed for the automated extraction of *events*, complex relations used to define the semantic structure of a sentence. These events differ from pairwise binary relations in that they have a defined trigger node, usually a verb, they can have multiple arguments, and other events can be used as event arguments, forming complex nested relations. Events can be seen as graphs, where named entities and triggers are the *nodes* and the arguments linking these are the *edges*. In this graph model, an event is implicitly defined as a trigger node and its set of outgoing edges.

The TEES system approaches event extraction as a task of graph generation, modelling it as a pipeline of consecutive, atomic classification tasks. The first step is *entity detection* where each token in the sentence is predicted as an entity node or as negative. In the second step of *edge detection*, argument edges are predicted for all valid, directed pairs of nodes. In the third, *unmerging* step, overlapping events are 'pulled apart' by duplicating trigger nodes. In the optional fourth step of *modifier detection*, binary modifiers (such as speculation or negation) can be predicted for the detected events. All of the classification steps in the TEES system rely on rich feature representations generated to a large degree from syntactic dependency parses. All classification tasks are implemented using the SVM^{multiclass} classifier (Joachims, 1999).

TEES has been developed using corpora from the Biomedical Natural Language Processing (BioNLP) domain, in particular the event corpora from the BioNLP Shared Tasks. These tasks define their own annotation schemes and provide standardized evaluation services. In the context of the EPE challenge we use the BioNLP 2009 GENIA corpus and its associated evaluation program to measure the impact of different parses on event extraction performance (Kim et al., 2009). The metric used for comparing the EPE submissions is the primary 'approximate span and recursive mode' metric of the original Shared Task, a micro-averaged F_1 score for the nine event classes of the corpus.

The specialized domain language presents unique challenges for parsers not specifically optimized for this domain, so using this data set to evaluate open-domain parses may result in overall lower performance than with parsers specifically trained on e.g. the GENIA treebank (Tateisi et al., 2005). When using the EPE parse data, TEES features encompass the type and direction for the dependencies combined wit the text span and a single part of speech for the tokens; lemmas are not used.

Negation Resolution The EPE negation resolution system is called Sherlock (Lapponi et al., 2012, 2017) and implements the perspective on negation defined by Morante and Daelemans (2012) through the creation of the Conan Doyle Negation Corpus for the Shared Task of the 2012 Joint Conference on Lexical and Computational Semantics (*SEM 2012). Negation instances are annotated as tri-partite structures: Negation *cues* can be full tokens (e.g. *not*), multi-word expressions (*by no means*), or sub-tokens (*un* in *unfortunate*); for each cue, its *scope* is defined as the possibly discontinuous sequence of (sub-)tokens affected by the negation. Additionally, a subset of in-scope tokens can be marked as negated *events* or *states*, provided that the sentence is factual and the events in question did not take place. In the EPE context, gold-standard negation cues are provided, because this sub-task has been found relatively insensitive to grammatical structure (Velldal et al., 2012).

Sherlock approaches negation resolution as a sequence labeling problem, using a Conditional Random Field (CRF) classifier (Lavergne et al., 2010). The token-wise negation annotations contain multiple layers of information. Tokens may or may not be negation cues and they can be either in or out of scope for a specific cue; in-scope tokens may or may not be negated events. Moreover, multiple negation instances may be (partially or fully) overlapping. Before presenting the CRF with the annotations, Sherlock 'flattens' all negation instances in a sentence, assigning a six-valued extended 'begin–inside–outside' labeling scheme. After classification, hierarchical (overlapping) negation structures are reconstructed using a set of post-processing heuristics.

The features of the classifier include different combinations of token-level observations, such as surface forms, part-of-speech tags, lemmas, and dependency labels. In addition, we extract both token and dependency distance to the nearest cue, together with the full shortest dependency path. Standard evaluation measures from the original shared task include scope tokens (ST), scope match (SM), event tokens (ET), and full negation (FN) F_1 scores. ST and ET are token-level scores for in-scope and negated event tokens, respectively, where a true positive is a correctly retrieved token of the relevant class (Morante and Blanco, 2012). FN is the strictest of these measures and the primary negation metric used in the EPE context—counting as true positives only perfectly retrieved full scopes, including an exact match on negated events.

Opinion Analysis The system by Johansson and Moschitti (2013) marks up expressions of opinion and emotion in a pipeline comprised of three separate classification steps, combined with end-to-end reranking; it was previously generalized and adapted for the EPE framework by Johansson (2017). The system is based on the annotation model and the annotated corpus developed in the MPQA project (Wiebe et al., 2005). The main component in this annotation scheme is the *opinion expression*; examples include case such as *dislike*, *praise*, *horrible*, or *one of a kind*. Each expression is associated with an *opinion holder*: an entity that expresses the opinion or experiences the emotion. Furthermore, every non-objective opinion expression is assigned a *polarity*: positive, negative, or neutral.

The opinion expression and polarity classifiers rely near-exclusively on token-level information, viz. n-grams comprising surface forms, lemmas, and PoS tags. Conversely, the opinion holder extraction and reranking modules make central use of structural information, i.e. paths and topological properties in one or more syntactico-semantic dependency graph(s).

In the EPE context, we evaluated how well the participating systems extract the three types of structures mentioned above: expressions, holders, and polarities. In each case, soft-boundary precision and recall measures were computed (Johansson and Moschitti, 2013; Johansson, 2017). Furthermore, for the detailed analysis we evaluated the opinion holder extractor separately, using gold-standard opinion expressions. We refer to this task as *in-vitro holder extraction*, and this score is used for the overall ranking of submissions when averaging F_1 scores across the three EPE downstream applications. The reason for highlighting this score is that it is the one most strongly affected by the design of the dependency representation.

Participating Teams Nine teams participated in EPE 2017, in the order of overall rank: Stanford–Paris (Schuster et al., 2017), Szeged (Szántó and Farkas, 2017), Paris–Stanford (Schuster et al., 2017), Universitat Pompeu Fabra (Mille et al., 2017), East China Normal University (Ji et al., 2017), Peking (Chen et al., 2017), Prague (Straka et al., 2017), and the University of Washington (Peng et al., 2017). These teams submitted 49 distinct runs that encompassed many different families of dependency representations, various approaches to preprocessing and parsing, and variable types and volumes of training data. The dependency representations employed by the participants varied from more syntactically oriented schemes—e.g. Stanford Basic (de Marneffe et al., 2006), CoNLL 2008–style (Surdeanu et al., 2008), and UD—to more semantically oriented representations, such as the Deep Syntactic Structures of Ballesteros et al. (2015), DELPH-IN MRS Dependencies (DM; Ivanova et al., 2012), or Enju Predicate–Argument Structures (PAS; Miyao, 2006). The teams also employed wildly variable volumes of training data, ranging from around 200,000 tokens (the English UD treebanks) to 1,7 million tokens (combining the venerable Wall Street Journal, Brown, and GENIA treebanks).

Results The team with the overall best result was the Stanford–Paris system with an overall score of 60.51, followed by the Szeged (58.57) and Paris–

Stanford (56.81) teams. The Stanford–Paris system obtained the best results for event extraction (when using the Stanford Basic representation), as well as for negation resolution (with enhanced Universal Dependencies). The Szeged system was the top performer in the opinion analysis subtask and employed the 'classic' CoNLL 2008 representation. The results further showed that a larger training set had a positive impact on results for the Stanford–Paris and Prague teams, who systematically varied the amount of training data in their experimental runs. In general however, it proved difficult to compare results across different teams due to the fact that these varied along multiple dimensions: the parser (and its output quality), the representation, input preprocessing, and the volume and type of training data. In this respect, EPE 2018 controls for several of these factors (dependency representation and amount of training data) and thus enables a more straightforward comparison across teams and analysis of the relationship between intrinsic and extrinsic parser performance.

3 Refinements: Towards EPE 2018

To integrate the EPE infrastructure with the 2018 UD parsing task, a number of extensions and revisions have been realized. These included provisioning the EPE data and a basic validation tool for parser outputs on the TIRA platform (Potthast et al., 2014) as well as technical improvements in two of the downstream systems (the opinion analysis system remains unchanged from EPE 2017). In the following paragraphs, we survey some of these adaptations for the EPE 2018 setup and comment on how these revisions limit comparability to end-to-end results from the 2017 campaign.

Document Collections The EPE parser inputs are comprised of training, development, and evaluation data for the three downstream applications, in total some 1900 documents, or around 850,000 tokens of running text. Reading and parsing thousands of small files (for the opinion analysis and event extraction tasks) proved to be a bottleneck for several systems in the EPE 2017 shared task, as parsers had to reload for each input file. For the convenience of 2018 participants, we have 'packed' the original large collections of small documents into three large files—one for each downstream application. The packing scheme inserts special 'delimiter paragraphs' at document boundaries, using the following general format:

```
Document 0020030 ends.
```

To not interfere with the grammatical analysis of immediate context, each delimiter is preceded and followed by three consecutive newlines—seeking to ensure that it is treated as a four-token utterance of its own in sentence splitting and tokenization.

When preparing submitted parser outputs for end-to-end evaluation, the delimiters allowed reconstructing the original document collections and data splits for each of the three EPE data sets. Overall, we did not observe unwanted side effects of the delimiters; there are, however, a few instances where the delimiter string itself can be tokenized (and sometimes sentence-split) in unexpected ways, including by the CoNLL 2018 baseline parser, such as splitting the numerical identifier into two tokens and breaking up the delimiter string as two sentences. The EPE 2018 unpacker robustly handles such cases, effectively ignoring sentence and token boundaries in scanning parser outputs for delimiter strings, and we have no reason to believe that the delimiters have negatively affected the parsing systems of participants.

Biological Event Extraction The TEES system used in the EPE 2018 task is largely unchanged from the 2017 version. However, the training and evaluation setup has been revised in order to achieve optimal performance when evaluating the submitted parses.

The BioNLP 2009 Shared Task, which serves as the EPE event extraction application, consists of three subtasks (Kim et al., 2009). Subtask 1 is the core task which defines a number of event types to extract. Subtask 2 extends the first with the addition of non-protein entities and secondary event arguments. Subtask 3 adds speculation and negation modifiers in the form of binary attributes to be predicted for each event. Thus, subtasks 1 and 2 define the event graph, and subtask 1 annotations can be seen as subgraphs of subtask 2.

In earlier versions of the TEES system, subtask evaluation was linked to subtask training, so that when the system was trained using subtask 1 annotations it was also evaluated for the same subtask. However, TEES generally achieves better performance on subtask 1 when trained on subtask 2 (or 3) annotations. We speculate this might be caused by the machine learning system trying to predict at least some edges for the 'gaps' left by not including subtask 2 annotations.

In the version of TEES updated for EPE 2018, evaluation has been decoupled from training data selection, so it is now possible to evaluate the system for the primary subtask 1 while still training on the full subtask 2 graphs. The end result is higher (and hopefully more stable) performance when evaluating the submitted parses, but unfortunately the EPE 2018 event extraction downstream task results are therefore not fully comparable with the 2017 ones.

Negation Resolution The Sherlock system used in the EPE 2018 task differs from the one used in EPE 2017 in two ways. First, we fixed a bug in the 2017 system related to a limited, but important, 'leak' of gold-standard annotations into system predictions. This leak was a side effect of the (legitimate) use of gold-standard information for negation cues, where the presence of multi-word cues (such as *neither ... nor* or *by no means*) could lead to the injection of gold-standard scope and event annotations in post-processing after classification, effectively overwriting actual system predictions under certain conditions.

The second difference between the 2017 and 2018 versions of Sherlock pertains to automated hyper-parameter tuning. The two main components in the Sherlock pipeline are two CRF classifiers, one for scope and one for event tokens. Sherlock in 2017 used the default hyper-parameters in the Wapiti implementation, i.e. unlike the other two EPE downstream systems it lacked the ability to automatically tune for each specific set of parser outputs. In EPE 2018, we introduced a comprehensive hyper-parameter grid search over the development set to identify the best-performing values for each system individually. Specifically, we optimized the L1 and L2 regularization hyper-parameters as well as the stopping threshold in Wapiti for both the scope and negated event classifiers. Briefly, the grid search starts with training Sherlock using all possible combinations of a broad range of candidate values along these six dimensions, leading to a total of some 6400 configurations trained using different hyper-parameter settings. These systems are then sorted in two consecutive steps that reflect the pipelined architecture of Sherlock: First, we rank the configurations based on their scope resolution scores on the development set and choose the best-performing hyper-parameters for the scope classifier among the n systems whose score falls within an experimentally defined range below the top-ranking system. Then, we re-rank this subset of n systems based on their full negation score on the development set and again select the best-performing hyper-parameters from among an experimentally defined range below the the best system. To mitigate the risk of overfitting, in both stages, the choice of the best-performing hyper-parameters is based a simple 'voting' scheme, picking hyper-parameter values that are most common in the top n configurations. This tuning process was applied separately to all parser outputs submitted to EPE 2018.

Overall, the corrected version of Sherlock combined with automated hyper-parameter tuning leads to a more robust and systematic evaluation on the downstream application of negation resolution. While this also means that the EPE 2018 results on negation resolution are not strictly compatible to the earlier 2017 campaign, it appears that the two Sherlock revisions offset each other at least when averaging over all submissions: the bug fix caused a drop in full negation scores of close to two F_1 points, but hyper-parameter tuning regained that performance loss to an accuracy of one decimal point (on average).

4 Task Overview

To minimize technical barriers to entry, the EPE parser inputs were installed on the TIRA platform alongside the data sets for the core UD parsing task, using the exact same general formats. The EPE document collections were provided as either 'raw', running text, or in pre-segmented form, with sentence and token boundaries predicted by the UDPipe baseline system of the core task. Parser outputs were collected in CoNLL-U format (again, for parallelism with the core task) and were then transferred from TIRA to the cluster that actually runs the EPE infrastructure. Here, all submissions were 'unpacked' (see § 3 above) and converted to the general EPE dependency graph format. Further details on the task schedule, technical infrastructure, submitted parser outputs, and end-to-end results are available from the task web site:

http://epe.nlpl.eu

Participating Teams Sixteen teams participated in the EPE 2018 campaign, in addition to the baseline parser provided by the core UD parsing task; we refer to the summary paper for the core task for a high-level characterization of participating

Team	Words		Sentences		Lemmas		UPOS		XPOS		LAS		MLAS		BLEX		Intrinsic
	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	⟨ ⟩	#	#
AntNLP	99.62	3	84.44	6	95.39	5	93.93	10	92.86	5	81.93	5	70.61	8	73.38	7	8
ArmParser	99.58	14	18.71	16	89.93	13	90.30	16	57.42	13	60.52	16	44.89	16	52.17	13	16
Baseline	99.62	3	84.44	6	95.39	5	93.93	10	92.86	5	76.33	13	65.86	13	67.43	11	11
IBM-NY	99.44	16	84.44	6	75.24	16	93.37	15	22.51	15	77.72	12	46.68	15	47.83	15	15
ICS-PAS	99.62	3	84.44	6	96.13	3	95.50	5	92.86	5	83.00	2	73.45	1	75.63	2	3
LATTICE-18	99.62	3	84.44	6	95.39	5	96.41	1	92.86	5	84.67	1	72.93	3	76.57	1	2
NLP-Cube	99.64	1	85.49	2	93.61	12	95.37	7	94.535	3	81.67	6	71.02	7	70.77	8	7
ONLP-lab	99.62	3	84.44	6	76.61	14	93.93	10	0.0525	16	65.92	15	57.01	14	46.32	16	13
ParisNLP-18	99.62	3	84.44	6	95.39	5	93.93	10	92.86	5	78.70	11	67.31	10	70.59	10	10
Phoenix	99.46	15	84.69	5	95.15	11	93.90	14	92.825	12	76.28	14	66.47	12	67.41	12	14
SLT-Interactions	99.62	3	84.44	6	95.39	5	95.86	2	92.86	5	81.51	8	69.91	9	73.44	6	5
SParse	0.00	17	0.00	17	0.00	17	0.00	17	0.00	17	0.00	17	0.00	17	0.00	17	17
Stanford-18	99.64	2	87.77	1	95.96	4	95.82	4	95.13	2	82.06	4	73.04	2	74.99	4	1
TurkuNLP-18	99.62	3	84.44	6	96.50	1	94.91	8	94.3	4	82.44	3	72.52	5	75.26	3	4
UDPipe-Future	99.59	13	84.81	4	96.43	2	95.86	3	95.195	1	81.64	7	72.56	4	74.81	5	6
Uppsala-18	99.62	12	85.44	3	75.40	15	95.41	6	22.52	14	81.37	9	71.34	6	50.46	14	12
UniMelb	99.62	3	84.44	6	95.39	5	94.68	9	92.86	5	79.19	10	66.95	11	70.77	9	9

Table 1: Summary of a selection of intrinsic evaluation scores from the core UD parsing task on English treebanks only. Columns labeled ⟨ ⟩ and # indicate the macro-averaged F_1 of each metric over the four English treebanks and the corresponding ranking of each team, respectively. The metrics are, from left to right: word and sentence segmentation; lemmatization; coarse and fine-grained parts of speech (UPOS and XPOS, respectively); labeled attachment score (LAS); morphology-aware labeled attachment score (MLAS); bi-lexical dependency score (BLEX); and finally an aggregate 'intrinsic' score, reflecting the average of ranks of each team. Teams shown in bold are included in the correlation analysis to intrinsic measures in § 5.

approaches and bibliographic references to individual system descriptions (Zeman et al., 2018). The names of all participants are shown in Table 1. Most teams submitted only one run with the exception of NLP-Cube (three runs) and SParse (four); in these cases, all runs have been scored, but only the most recent submission was considered for the final evaluation and comparison with intrinsic measures.

We conducted a post-submission survey among participants, to gauge the comparability of the parsing systems submitted to the core UD parsing task vs. those used for parsing the EPE data, e.g. software versions, training regimes, or other configuration options.[1] Twelve teams responded to the survey, and hence the following details only apply to those who responded. Almost all participants used (parts of) the English training data provided by the UD parsing shared task (which is the only training data allowed in EPE 2018), except for the UniMelb team who accidentally used their own UD conversions of the WSJ and GENIA treebanks. Therefore, UniMelb was excluded from the competition, but we report their scores as an additional point of comparison. Of all the systems that used 'legitimate' training data, only LATTICE used different training data for their EPE submission than in their core task system. Two of the survey respondents—NLP-Cube and SParse—indicated that they had made changes to their systems that render the EPE and core task results incomparable. The four teams that did not respond to the survey and the four teams for which the survey revealed limited comparability to core task results (i.e. UniMelb, LATTICE, NLP-Cube, and SParse; shown in italics in Tables 1 and 2) were not considered in our quantitative correlation analysis between intrinsic and extrinsic metrics (see § 5 below). Finally, only four of the survey respondents (NLP-Cube, Phoenix, UDPipe-Future, and Uppsala-18) indicated that their parsers had used raw texts as inputs, i.e. applied their own sentence and token segmentation. The other eight respondents, in contrast, had availed themselves of the pre-segmented inputs provided as an alternative form of the EPE parser inputs.

Intrinsic Metrics In our view, one of the most intriguing opportunities of aligning EPE 2018 with the core UD parsing task lies in the comparison of intrinsic and extrinsic evaluation results. In other words, we seek to shed light on the degrees to which observations made in intrinsic evaluation allow one to predict downstream success for a specific application, as well as on which (intrinsically measurable) layers of grammatical analysis most directly impact end-to-end performance. For these

[1] To not interfere with the busy final weeks of the core task, the EPE submission deadline was two weeks later. Hence, we could not technically enforce that the exact same software configurations were used in both component tasks, and in fact at least two teams had to resort to revising their parsers in order to complete processing of the comparatively large EPE input files.

reasons, we extracted a comprehensive array of intrinsic evaluation results for parsers represented in EPE 2018 from the in-depth result summary for the core UD parsing task.[2]

Table 1 summarizes our selection of intrinsic observations, where the first six metrics seek to isolate performance at all relevant layers of grammatical analysis, viz. word and sentence segmentation, lexical analysis (lemmatization and tagging), and syntactic structure (labeled attachment scores, or LAS). The table further includes the other two official metrics of the core task, which by design blend together some of these layers, i.e. morphology-aware labeled attachment score and bi-lexical dependency score, which evaluate LAS plus tagging and morphological features[3] and LAS plus lemmatization, respectively.

In all cases, the results in Table 1 reflect (macro-averaged) performance over the English UD treebanks only. Several of the best-performing systems across all languages of the core task also submitted to EPE 2018, including ICS-PAS, LATTICE, Stanford, TurkuNLP-18, and UDPipe-Future. These systems also populate the top ranks in the aggregate English-only intrinsic evaluation, even though there is some 'jitter' in their relative ranks across individual metrics. In a few cases, the results in Table 1 actually reveal system idiosyncrasies: IBM-NY and Uppsala-18 do not predict XPOS values, whereas the XPOS field in the ONLP-lab parser outputs merely contains a copy of the coarse-grained UPOS predictions. The nine parsers that started from pre-segmented EPE documents all tie for third and sixth rank in sentence splitting and tokenization, respectively.

5 Official Results

End-to-end extrinsic evaluation results for the EPE 2018 campaign are summarized in Table 2.[4] For each of the three downstream applications, the table shows precision, recall, and F_1 scores on the corresponding EPE evaluation set. Additionally, we indicate for each application whether coarse- or fine-grained parts of speech were used (see below) and provide an aggregate ranking of participating teams based on macro-averaged F_1 scores.

The parser that gives rise to overall best downstream results across the three EPE applications is UDPipe-Future, even though it is not the top performer for any of the individual applications. Differences in average scores for the best-performing systems are small, however, with less than 0.4 F_1 points between the first and the fifth overall rank. Many of the best-performing systems when judged in terms of extrinsic results correspond to what one might have predicted from our summary of English-only intrinsic results (see § 4 above): in addition to UDPipe-Future, also SLT-Interactions, Stanford, and TurkuNLP-18 are in the intersection of the top-five intrinsic and extrinsic ranks. The system that ranks second in the extrinsic perspective (NLP-Cube), on the other hand, indicated in our participant survey that they had made changes to the parser inbetween their submissions to the core vs. the EPE tasks.

If one ranks systems individually for each downstream application and compares across each row, the majority of teams appear to obtain broadly comparable rankings on different applications. Nevertheless, there are a few notable exceptions. ArmParser achieves the best results on negation resolution but otherwise ranks in the bottom segment on event extraction and opinion analysis. Manual inspection of the parser outputs submitted reveals that ArmParser zealously over-segments (as is also evident in its low intrinsic score on sentence splitting in Table 1): it breaks the 1089 sentences of the gold-standard negation evaluation data into a little more than two thousand isolated token sequences. While the EPE infrastructure deals robustly with segmentation mismatches, this discrepancy uncovers a technical issue in our way of interfacing to the original *SEM 2012 scorer: the 'annotation projection' described by Lapponi et al. (2017) will present the scorer with shortened and, hence, simplified gold standards to compare to. In other words, the high negation scores for ArmParser indicate an unwarranted reward for its dealing in artificially short 'sentences'.

Another stark asymmetry in per-application ranks pertains to TurkuNLP-18, which shows top results on negation resolution and opinion analysis but ranks in the bottom quarter on the event extraction application (which happens to be developed at the same site). While the unexpectedly

[2] Intrinsic results were automatically scraped from the official http://universaldependencies.org/conll18/results.html page.

[3] None of the current EPE downstream systems actually considers morphological features, although the EPE interface format does in principle provide for their representation.

[4] A multitude of additional scores, including against the development sections for each downstream application, are available from the task web site at http://epe.nlpl.eu.

Team	Event Extraction				Negation Resolution				Opinion Analysis					
	PoS	P	R	F_1	PoS	P	R	F_1	PoS	P	R	F_1	⟨ ⟩	#
AntNLP	U	54.00	44.97	49.07	X	100	39.54	56.67	X	64.37	55.76	59.76	55.17	10
ArmParser	U	53.76	39.28	45.39	X	99.12	42.75	**59.74**	X	60.13	51.65	55.57	53.57	15
Baseline	U	39.63	53.43	45.51	X	100	40.30	57.45	X	61.67	55.95	58.67	53.88	13
IBM-NY	U	53.08	43.81	48.00	U	100	39.16	56.28	U	62.03	56.03	58.88	54.39	12
ICS-PAS	U	56.41	43.97	49.42	X	100	39.54	56.67	X	63.73	57.67	60.55	55.55	6
LATTICE-18	X	58.93	43.12	**49.80**	X	100	39.16	56.28	X	63.91	56.88	60.19	55.42	9
NLP-Cube	U	56.54	42.65	48.62	X	100	40.15	57.30	X	64.95	59.24	61.96	55.96	3
ONLP-lab	U	54.08	41.67	47.07	U	100	36.88	53.89	U	62.94	56.37	59.47	53.48	16
ParisNLP-18	X	55.66	43.56	48.87	X	100	40.68	57.83	X	63.01	56.78	59.73	55.48	8
Phoenix	U	47.23	40.98	43.88	X	100	41.06	58.22	X	63.16	55.87	59.29	53.80	14
SLT-Interactions	X	56.32	43.97	49.38	X	100	41.06	58.22	U	65.47	56.56	60.69	56.10	2
SParse	X	50.62	41.04	45.33	X	100	40.30	57.45	X	63.44	57.94	60.57	54.45	11
Stanford-18	U	59.26	41.14	48.56	X	100	41.29	58.45	X	63.33	57.68	60.37	55.80	5
TurkuNLP-18	U	52.64	42.05	46.75	X	100	42.59	**59.74**	X	64.23	58.26	61.10	55.86	4
UDPipe-Future	U	53.97	45.98	49.66	X	100	41.29	58.45	X	63.47	57.72	60.46	56.19	1
Uppsala-18	U	58.04	43.43	49.68	U	100	36.74	53.74	U	64.67	61.68	**63.14**	55.52	7
UniMelb	X	58.52	49.43	53.59	X	100	41.83	58.99	X	66.67	62.88	64.72	59.10	

Table 2: Summary of EPE 2018 results. The columns show, from left to right: team name, PoS tags used (UPOS or XPOS), precision, recall, and F_1 across the three downstream applications, average F_1 across applications, and finally the overall rank of each team. The best F_1 score for each downstream task is indicated in bold. The UniMelb submission is considered outside the competition due to the use of additional training data; teams shown in bold are included in the correlation analysis to intrinsic measures in § 5.

low performance in the combination of the Turku parser with the Turku event extraction system reassuringly indicates that there was no collusion in Finland, we have so far been unable to form a hypothesis about what might be the cause for this performance discrepancy. Conversely, Uppsala-18 is among the top performers for event extraction and opinion analysis but obtains the lowest F_1 results on negation resolution in the EPE 2018 field. The Uppsala parser is one of the few that does not predict fine-grained parts of speech, which the Sherlock negation system appears to strongly prefer over the far more coarse-grained UPOS tags (see below). We conjecture that the lack of XPOS predictions in the Uppsala-18 parser outputs is at least an important factor in the uncharacteristically poor negation results for this system.

UPOS vs. XPOS Recall that the EPE 2018 infrastructure automatically retrains and tunes each downstream system for each system submission. An additional aspect in which the downstream systems could be optimized towards a particular parser is, of course, feature engineering and selection. For full generality and applicability across different types of syntactico-semantic dependency representations, the current EPE applications restrict themselves to a range of broad token-level and structural features that do not invoke individual linguistic configurations (e.g. indicators of passive voice)—including conjunctions of individual features that have been clearly observed to be beneficial (see § 2 above and references there). All three downstream systems employ 'vintage' classifiers (CRFs and SVMs) for which regularization techniques and best practices are well established, such that one can hope for a certain degree of feature selection during training.

Reflecting availability of two distinct assignments of parts of speech in all but a few of the EPE 2018 submissions, we conducted one round of feature adaptation in the downstream systems, viz. determining whether to use the coarse-grained, universal UPOS or the finer-grained, English-specific XPOS values for each combination of parser outputs and downstream system. This selection was based on optimizing the primary metric for each application on the development data, and the results are indicated in the three PoS columns in Table 2.

XPOS appears to work better in general, possibly reflecting that it makes available additional distinctions, including some inflectional morphology.[5] There are a few notable exceptions to this generalization, however, and they appear application-dependent to some degree. In particular the event extraction system often obtains better results when using UPOS, whereas for negation resolution

[5] Reflecting the above design constraints and desire for cross-framework applicability, the EPE downstream systems do not currently consider the morphological features that are increasingly an integral part of the Universal Dependencies framework.

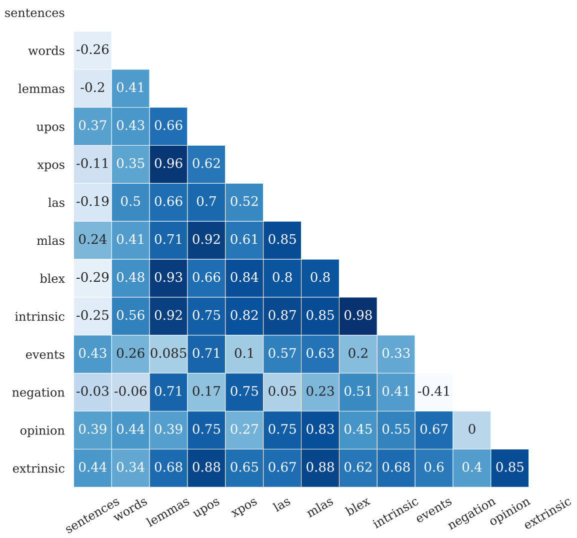

Figure 1: Correlation matrix of intrinsic and extrinsic metrics.

XPOS (where available) universally yields higher end-to-end scores, and UPOS is only used with the three systems that do not predict fine-grained tags. Almost the same holds for the opinion analysis application, with the one exception of the SLT-Interactions submission, whose UPOS predictions actually yield better results (though the actual differences are small). Based on these observations, one might expect Uppsala-18 (which only predicts UPOS) to be at a disadvantage for opinion analysis too, but other factors in this combination appear more important (as Uppsala-18 actually obtains the best overall opinion results).

Correlation Analysis To obtain a better understanding of the relationships between intrinsic and extrinsic perspectives on parser performance, we perform a quantitative correlation analysis over pairs of evaluation metrics. We compute a rank correlation matrix of intrinsic and extrinsic measures, limited to the sub-set of nine systems which are known to be fully comparable across intrinsic and extrinsic evaluation, i.e. where there were no substantive changes to the parsers following the completion of the core UD parsing task. We further limit our analysis to the intrinsic evaluation metrics pertaining to English (see Table 1), combined with the downstream per-application F_1 scores and an average rank score called *extrinsic* in the following, which aggregates the average rank of each system across the three downstream applications. Figure 1 shows a heatmap of Spearman's rank correlation coefficients (ρ) for all pairs of intrinsic and extrinsic metrics.

In general, we observe high degrees of correlation among intrinsic measures, albeit less so for the segmentation metrics, in particular sentence segmentation.[6] We find the strongest correlations between the intrinsic average and the BLEX measure (0.98), XPOS and lemmas (0.96), BLEX and lemmas (0.93), and UPOS and MLAS (0.92). Further, BLEX correlates stronger with the average intrinsic metric than LAS and MLAS, so if one were to search for a single, indicative intrinsic measure, BLEX might offer a combined indicator across analysis layers. We note that the correlation scores pertaining to XPOS must be interpreted with some care, given that two of the systems involved (IBM-NY and Uppsala-18) do not predict XPOS, so that their ranks according to this metric will not correspond to their performance on other metrics.

If we examine the correlation between intrinsic and extrinsic metrics, we also observe some strong correlations—which is of course a very welcome observation. In particular, we find a strong correlation between the average extrinsic metric and the intrinsic UPOS and MLAS metrics (0.88). The correlation with UPOS is perhaps somewhat surprising as UPOS is not used by the majority of systems. Still, it appears that the ability to correctly predict universal PoS tags provides a useful indicator of downstream parser performance. We further observe strong to moderate correlations between the individual intrinsic metrics and the overall extrinsic average.

When examining per-application correlations to intrinsic performance we find that each of the individual downstream metrics shows a correlation with the intrinsic average, but for all three less so than the extrinsic average. While seemingly counter-intuitive, maybe, we interpret this as indicative of a certain degree of complementarity among the three downstream applications. Taken together, they lead to better correspondences with intrinsic metrics, an observation which holds also true for several of the individual intrinsic metrics, viz. UPOS, MLAS, and BLEX. This is in accordance with the observation in the results overview above: there is no parser to suit all needs, such that in principle at least it would make sense to pick a different parser for each of the three downstream

[6] Only one third of the systems considered in the correlation matrix actually apply their own sentence splitting and tokenization (see § 4 above). Accordingly, the corresponding metrics are bound to exhibit far less interesting variation in the correlation analysis.

applications.

Downstream results obtained by the different parsers for the event extraction application, correlate most strongly with the UPOS metric (0.71), followed by LAS (0.63) and MLAS (0.57). This fits well with the observation that most of the top-scoring systems in the event task actually make use of UPOS (see above). The event extraction application does not use lemmas among its features, hence it shows no observable correlation to this particular intrinsic metric. For the negation application, on the other hand, the strongest correlation is with the XPOS metric (0.75), followed by lemmas (0.71) and BLEX (0.51). XPOS seems to be the favoured PoS choice for this task (see Table 2), so this again is in line with the most effective type of PoS for the majority of systems.

When it comes to the opinion analysis application, its rankings correlate most strongly with the intrinsic ranking of parsers by MLAS (0.83), followed by LAS and UPOS (both 0.75). It thus seems that this application depends more strongly on a syntactic or structural metric such as MLAS, in comparison to the other downstream applications. We also find that the opinion scores somewhat surprisingly correlate more with UPOS (0.75) than XPOS (0.27), which does not obviously follow from the best-performing choice of tag set. We leave further investigation of the relative importance of PoS tagging to the EPE opinion analysis system to future work (see § 6 below).

Comparison to 2017 Owing to the updates in downstream systems summarized in § 3 above, the end-to-end scores in Table 2 are not strictly comparable to results from the EPE 2017 campaign (Oepen et al., 2017). Nevertheless, we believe that a 'ballpark' comparison can be informative.[7] The best-performing parser in 2017 enabled end-to-end scores of 50.23, 66.16, and 65.14 F_1 points on event extraction, negation resolution, and opinion analysis, respectively. This was the Stanford–Paris submission (run #06), outputting enhanced UD graphs and trained on about 1.7 million tokens of annotated text from the Brown, WSJ, and GENIA corpora (Schuster et al., 2017). In contrast, the overall best parser in the EPE 2018 field delivers F_1 results of 49.66, 58.45, and 56.19 (UDPipe-Future). Taking into account that event scores in 2017 may have been slightly under-estimated, negation scores moderately inflated, and opinion scores fully comparable—it seems fair to say that the 'pure' English UD parsers from the EPE 2018 campaign do not facilitate the same high levels of downstream performance. In the 2017 campaign, end-to-end results for the event extraction application were very competitive, and those for negation resolution advanced the state of the art. This is not the case in the 2018 field, which we tentatively attribute to the limited volume of English training data, the strict 'treeness' assumptions in most current dependency parsers, and quite possibly the inability of the EPE downstream applications to take advantage of the UD morphological features.

6 Reflections and Outlook

In our view, the considerable effort for both participants and organizers of running an additional track at the 2018 CoNLL Shared Task on Universal Dependency Parsing is rewarded through (a) a valuable, complementary perspective on the contrastive evaluation of different parsing systems, as well as through (b) a window of comparison to the state of the art in three representative language 'understanding' applications. From a sufficiently high level of abstraction, we see many reassuring correspondences between intrinsic parser evaluation and actual downstream utility. At the same time, we find that not even a comprehensive 'battery' of layered intrinsic metrics can fully inform the relative comparison of different parsers with regard to their contributions to downstream performance.

In hindsight, we would have liked to obtain an even tigher experimental setup, without any remaining uncertainty about comparability of participating systems across the two tracks. If we were to run another EPE campaign (unlikely as that may feel just now), the EPE data bundles should also include relevant test data for intrinsic evaluation. In more immediate follow-up work, we plan to re-compute and publish end-to-end results for the submissions from the EPE 2017 campaign, for full comparability, as well as further investigate the relative contributions of individual analysis layers to the various downstream applications through additional control experiments and ablation studies.

[7] In addition to the parameters suggested for such comparison in § 3 above, we find this belief supported by alignment of results for the one system that participated in both EPE campaigns in very similar configurations: the Prague submission (run #00) in 2017 (Straka et al., 2017) corresponds closely to the 2018 UDPipe baseline. F_1 results for the three downstream applications in 2017 were 43.58, 58.83, and 59.79—compared to 2018 scores of 45.51, 57.45, and 58.67.

Acknowledgments

This work was in large parts conducted while the second and third authors were fellows in residence of the Center for Advanced Studies (CAS) at the Norwegian Academy of Science and Letters. The EPE 2018 campaign was in part funded by the Nordic e-Infrastructure Collaboration (NeIC) through their support to the Nordic Language Processing Laboratory (NLPL; http://www.nlpl.eu). We are grateful to our NLPL and CAS colleagues and to the Nordic tax payers. Dan Zeman, Martin Potthast, Jan Hajič, and Milan Straka have been exceptionally helpful on behalf of the organizing team for the core UD parsing task. We gratefully acknowledge the support by Sophia Ananiadou and Matthew Shardlow with our using the BioNLP 2009 evaluation data. We thank Joakim Nivre and Erik Velldal for their contributions to the overall task design and result analysis.

References

Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. 2015. Data-driven deep-syntactic dependency parsing. *Natural Language Engineering* 22:1 – 36.

Jari Björne. 2014. *Biomedical Event Extraction with Machine Learning*. Ph.D. thesis, University of Turku, Turku, Finland.

Yufei Chen, Junjie Cao, Weiwei Sun, and Xiaojun Wan. 2017. Peking at EPE 2017: A comparison of tree approximation, transition-based, and maximum subgraph models for semantic dependency analysis. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 60 – 64.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*. Genoa, Italy, page 449 – 454.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop*. Jeju, Republic of Korea, page 2 – 11.

Tao Ji, Yuekun Yao, Qi Zheng, Yuanbin Wu, and Man Lan. 2017. ECNU at EPE 2017: Universal dependencies representations parser. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 40 – 46.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods. Support Vector Learning*, MIT Press, Cambridge, MA, USA, page 41 – 56.

Richard Johansson. 2017. EPE 2017: The Trento–Gothenburg opinion extraction system. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 31 – 39.

Richard Johansson and Alessandro Moschitti. 2013. Relational features in fine-grained opinion analysis. *Computational Linguistics* 39(3):473 – 509.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing. Shared Task*. Boulder, CO, USA, page 1 – 9.

Marco Kuhlmann and Stephan Oepen. 2016. Towards a catalogue of linguistic graph banks. *Computational Linguistics* 42(4):819 – 827.

Emanuele Lapponi, Stephan Oepen, and Lilja Øvrelid. 2017. EPE 2017: The Sherlock negation resolution downstream application. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 25 – 30.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO2. Sequence-labeling negation using dependency features. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*. Montréal, Canada, page 319 – 327.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, page 504 – 513.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, and Oscar Täckström. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51th Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria, page 92 – 97.

Simon Mille, Roberto Carlini, Ivan Latorre, and Leo Wanner. 2017. UPF at EPE 2017: Transduction-based deep analysis. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency*

Linguistics and the 15th International Conference on Parsing Technologies. Pisa, Italy, page 76 – 84.

Yusuke Miyao. 2006. *From Linguistic Theory to Syntactic Analysis. Corpus-Oriented Grammar Development and Feature Forest Model*. Doctoral dissertation, University of Tokyo, Tokyo, Japan.

Roser Morante and Eduardo Blanco. 2012. *SEM 2012 Shared Task. Resolving the scope and focus of negation. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*. Montréal, Canada, page 265 – 274.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg. Annotation of negation in Conan Doyle stories. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*. Istanbul, Turkey, page 1563 – 1568.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1. A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*. Portorož, Slovenia, page 1659 – 1666.

Stephan Oepen, Lilja Øvrelid, Jari Björne, Richard Johansson, Emanuele Lapponi, Filip Ginter, and Erik Velldal. 2017. The 2017 Shared Task on Extrinsic Parser Evaluation. Towards a reusable community infrastructure. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 1 – 16.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Meeting of the Association for Computational Linguistics*. Vancouver, Canada, page 2037 – 2048.

Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. 2014. Improving the reproducibility of PAN's shared tasks. Plagiarism detection, author identification, and author profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative*. Springer, Berlin, Germany, page 268 – 299.

Sebastian Schuster, Eric De La Clergerie, Marie Candito, Benoît Sagot, Christopher D. Manning, and Djamé Seddah. 2017. Paris and Stanford at EPE 2017: Downstream evaluation of graph-based dependency representations. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 47 – 59.

Milan Straka, Jana Straková, and Jan Hajič. 2017. Prague at EPE 2017: The UDPipe system. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 65 – 74.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the 12th Conference on Natural Language Learning*. Manchester, UK, page 159 – 177.

Zsolt Szántó and Richárd Farkas. 2017. Szeged at EPE 2017: First experiments in a generalized syntactic parsing framework. In *Proceedings of the 2017 Shared Task on Extrinsic Parser Evaluation at the Fourth International Conference on Dependency Linguistics and the 15th International Conference on Parsing Technologies*. Pisa, Italy, page 75 – 79.

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing. Companion Volume*. Jeju, Korea.

Erik Velldal, Lilja Øvrelid, Jonathon Read, and Stephan Oepen. 2012. Speculation and negation: Rules, rankers, and the role of syntax. *Computational Linguistics* 38(2):369 – 410.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation* 39(2-3):165 – 210.

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*. Brussels, Belgium, page 1 – 20.