CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies

Daniel Zeman[1], Martin Popel[1], Milan Straka[1], Jan Hajič[1], Joakim Nivre[2], Filip Ginter[3], Juhani Luotolahti[3], Sampo Pyysalo[4], Slav Petrov[5], Martin Potthast[6], Francis Tyers[7], Elena Badmaeva[8], Memduh Gökırmak[9], Anna Nedoluzhko[1], Silvie Cinková[1], Jan Hajič jr.[1], Jaroslava Hlaváčová[1], Václava Kettnerová[1], Zdeňka Urešová[1], Jenna Kanerva[3], Stina Ojala[3], Anna Missilä[3], Christopher Manning[10], Sebastian Schuster[10], Siva Reddy[10], Dima Taji[11], Nizar Habash[11], Herman Leung[12], Marie-Catherine de Marneffe[13], Manuela Sanguinetti[14], Maria Simi[15], Hiroshi Kanayama[16], Valeria de Paiva[17], Kira Droganova[1], Héctor Martínez Alonso[18], Çağrı Çöltekin[19], Umut Sulubacak[9], Hans Uszkoreit[20], Vivien Macketanz[20], Aljoscha Burchardt[20], Kim Harris[21], Katrin Marheinecke[21], Georg Rehm[20], Tolga Kayadelen[5], Mohammed Attia[5], Ali Elkahky[5], Zhuoran Yu[5], Emily Pitler[5], Saran Lertpradit[5], Michael Mandl[5], Jesse Kirchner[5], Hector Fernandez Alcalde[5], Jana Strnadová[5], Esha Banerjee[5], Ruli Manurung[5], Antonio Stella[5], Atsuko Shimada[5], Sookyoung Kwak[5], Gustavo Mendonça[5], Tatiana Lando[5], Rattima Nitisaroj[5], and Josie Li[5]

[1]Charles University, Faculty of Mathematics and Physics
[2]Uppsala University, [3]University of Turku, [4]University of Cambridge
[5]Google, [6]Bauhaus-Universität Weimar, [7]UiT The Arctic University of Norway
[8]University of the Basque Country, [9]Istanbul Technical University
[10]Stanford University, [11]New York University Abu Dhabi, [12]City University of Hong Kong
[13]Ohio State University, [14]University of Turin, [15]University of Pisa, [16]IBM Research, [17]Nuance Communications, [18]INRIA – Paris 7, [19]University of Tübingen, [20]DFKI, [21]text & form

{zeman|popel|straka|hajic}@ufal.mff.cuni.cz
joakim.nivre@lingfil.uu.se, {figint|mjluot}@utu.fi
slav@google.com, martin.potthast@uni-weimar.de

Abstract

The Conference on Computational Natural Language Learning (CoNLL) features a shared task, in which participants train and test their learning systems on the same data sets. In 2017, one of two tasks was devoted to learning dependency parsers for a large number of languages, in a real-world setting without any gold-standard annotation on input. All test sets followed a unified annotation scheme, namely that of Universal Dependencies. In this paper, we define the task and evaluation methodology, describe data preparation, report and analyze the main results, and provide a brief categorization of the different approaches of the participating systems.

1 Introduction

Ten years ago, two CoNLL shared tasks were a major milestone for parsing research in general and dependency parsing in particular. For the first time dependency treebanks in more than ten languages were available for learning parsers. Many of them were used in follow-up work, evaluating parsers on multiple languages became standard, and multiple state-of-the-art, open-source parsers became available, facilitating production of dependency structures to be used in downstream applications. While the two tasks (Buchholz and Marsi, 2006; Nivre et al., 2007) were extremely important in setting the scene for the following years, there were also limitations that complicated application of their results: (1) gold-standard to-

kenization and part-of-speech tags in the test data moved the tasks away from real-world scenarios, and (2) incompatible annotation schemes made cross-linguistic comparison impossible. CoNLL 2017 has picked up the threads of those pioneering tasks and addressed these two issues.[1]

The focus of the 2017 task was learning syntactic dependency parsers that can work in a real-world setting, starting from raw text, and that can work over many typologically different languages, even surprise languages for which there is little or no training data, by exploiting a common syntactic annotation standard. This task has been made possible by the Universal Dependencies initiative (UD) (Nivre et al., 2016), which has developed treebanks for 50+ languages with cross-linguistically consistent annotation and recoverability of the original raw texts.

Participating systems had to find labeled syntactic dependencies between words, i.e., a syntactic head for each word, and a label classifying the type of the dependency relation. No gold-standard annotation (tokenization, sentence segmentation, lemmas, morphology) was available in the input text. However, teams wishing to concentrate just on parsing were able to use segmentation and morphology predicted by the baseline UDPipe system (Straka et al., 2016).

2 Data

In general, we wanted the participating systems to be able to use any data that is available free of charge for research and educational purposes (so that follow-up research is not obstructed). We deliberately did not place upper bounds on data sizes (in contrast to e.g. Nivre et al. (2007)), despite the fact that processing large amounts of data may be difficult for some teams. Our primary objective was to determine the capability of current parsers with the data that is currently available.

In practice, the task was formally closed, i.e., we listed the approved data resources so that all participants were aware of their options. However, the selection was rather broad, ranging from Wikipedia dumps over the OPUS parallel corpora (Tiedemann, 2012) to morphological transducers. Some of the resources were proposed by the participating teams.

[1]Outside CoNLL, there were several other parsing tasks in the meantime, which naturally also explored previously unadressed aspects—for example SANCL (Petrov and McDonald, 2012) or SPMRL (Seddah et al., 2013, 2014).

We provided dependency-annotated training and test data, and also large quantities of crawled raw texts. Other language resources are available from third-party servers and we only referred to the respective download sites.

2.1 Training Data: UD 2.0

Training and development data come from the Universal Dependencies (UD) 2.0 collection (Nivre et al., 2017b). Unlike previous UD releases, the test data was not included in UD 2.0. It was kept hidden until the evaluation phase of the shared task terminated. In some cases, the underlying texts had been known from previous UD releases but the annotation had not (UD 2.0 follows new annotation guidelines that are not backward-compatible).

64 UD treebanks in 45 languages were available for training. 15 languages had two or more training treebanks from different sources, often also from different domains.

56 treebanks contained designated development data. Participants were asked not to use it for training proper but only for evaluation, development, tuning hyperparameters, doing error analysis etc. The 8 remaining treebanks were small and had only training data (and even these were extremely small in some cases, especially for Kazakh and Uyghur). For those treebanks cross-validation had to be used during development, but the entire dataset could be used for training once hyperparameters were determined.

Participants received the training and development data with gold-standard tokenization, sentence segmentation, POS tags and dependency relations; and for some languages also lemmas and/or morphological features.

Cross-domain and cross-language training was allowed and encouraged. Participants were free to train models on any combination of the training treebanks and apply it to any test set. They were even allowed to use the training portions of the 6 UD 2.0 treebanks that were excluded from evaluation (see Section 2.3).

2.2 Supporting Data

To enable the induction of custom embeddings and the use of semi-supervised methods in general, the participants were provided with supporting resources primarily consisting of large text corpora for (nearly) all of the languages in the task, as well as embeddings pre-trained on these corpora.

Raw texts The supporting raw data was gathered from CommonCrawl, which is a publicly available web crawl created and maintained by the non-profit CommonCrawl foundation.[2] The data is publicly available in the Amazon cloud both as raw HTML and as plain text. It is collected from a number of independent crawls from 2008 to 2017, and totals petabytes in size.

We used cld2[3] as the language detection engine because of its speed, available Python bindings and large coverage of languages. Language detection was carried out on the first 1024 bytes of each plaintext document. Deduplication was carried out using hashed document URLs, a simple strategy found in our tests to be effective for coarse duplicate removal. The data for each language was capped at 100,000 tokens per a single input file.

Automatic tokenization, morphology and parsing The raw texts were further processed in order to generate automatic tokenization, segmentation, morphological annotations and dependency trees.

At first, basic cleaning was performed – paragraphs with erroneous encoding or less than 16 characters were dropped, remaining paragraphs converted to Normalization Form KC (NFKC)[4] and again deduplicated. Then the texts were segmented and tokenized, multi-word tokens split into words, and sentences with less than 5 words dropped. Because we wanted to publish the resulting corpus, we shuffled the sentences and also dropped sentences with more than 80 words at this point for licensing reasons. The segmentation and tokenization was obtained using the baseline UDPipe models described in Section 5. These models were also used to further generate automatic morphological annotations (lemmas, UPOS, XPOS and FEATS) and dependency trees.

The resulting corpus contains 5.9 M sentences and 90 G words in 45 languages and is available in CoNLL-U format (Ginter et al., 2017). The per-language sizes of the corpus are listed in Table 1

Precomputed word embeddings We also precomputed word embeddings using the segmented and tokenized plain texts. Because UD words can contain spaces, these in-word spaces were con-

[2] http://commoncrawl.org/ Except for Ancient Greek, which was gathered from the Perseus Digital Library.

[3] http://github.com/CLD2Owners/cld2

[4] http://unicode.org/reports/tr15/

Language	Words
English (en)	9,441 M
German (de)	6,003 M
Portuguese (pt)	5,900 M
Spanish (es)	5,721 M
French (fr)	5,242 M
Polish (pl)	5,208 M
Indonesian (id)	5,205 M
Japanese (ja)	5,179 M
Italian (it)	5,136 M
Vietnamese (vi)	4,066 M
Turkish (tr)	3,477 M
Russian (ru)	3,201 M
Swedish (sv)	2,932 M
Dutch (nl)	2,914 M
Romanian (ro)	2,776 M
Czech (cs)	2,005 M
Hungarian (hu)	1,624 M
Danish (da)	1,564 M
Chinese (zh)	1,530 M
Norwegian-Bokmål (no)	1,305 M
Persian (fa)	1,120 M
Finnish (fi)	1,008 M
Arabic (ar)	963 M
Catalan (ca)	860 M
Slovak (sk)	811 M
Greek (el)	731 M
Hebrew (he)	615 M
Croatian (hr)	583 M
Ukrainian (uk)	538 M
Korean (ko)	527 M
Slovenian (sl)	522 M
Bulgarian (bg)	370 M
Estonian (et)	328 M
Latvian (lv)	276 M
Galician (gl)	262 M
Latin (la)	244 M
Basque (eu)	155 M
Hindi (hi)	91 M
Norwegian-Nynorsk (no)	76 M
Kazakh (kk)	54 M
Urdu (ur)	46 M
Irish (ga)	24 M
Ancient Greek (grc)	7 M
Uyghur (ug)	3 M
Kurdish (kmr)	3 M
Upper Sorbian (hsb)	2 M
Buryat (bxr)	413 K
North Sámi (sme)	331 K
Old Church Slavonic (cu)	28 K
Total	90,669 M

Table 1: The supporting data overview: the number of words (M = million; K = thousand) for each language.

verted to Unicode character *NO-BREAK SPACE* (U+00A0).[5]

The dimensionality of the word embeddings was chosen to be 100 after thorough discussion – more dimensions may yield better results and are commonly used, but even with just 100, the uncompressed word embeddings for the 45 languages take 135 GiB. Also note that Andor et al. (2016) achieved state-of-the-art results with 64 dimensions.

The word embeddings were precomputed using `word2vec` (Mikolov et al., 2013) with the following options:

```
word2vec -min-count 10 -size 100
  -window 10 -negative 5 -iter 2
  -threads 16 -cbow 0 -binary 0.
```

The precomputed word embeddings are available on-line (Ginter et al., 2017).

2.3 Test Data: UD 2.0

The main part of test data comprises test sets corresponding to 63 of the 64 training treebanks.[6] Test sets from two different treebanks of one language were evaluated separately as if they were different languages. Every test set contained at least 10,000 words or punctuation marks. UD 2.0 treebanks that were smaller than 10,000 words were excluded from the evaluation. Among the treebanks that were able to provide the required amount of test data, there are 8 treebanks so small that the remaining data could not be split to training and development portions; for two of them, the data left for training is only a tiny sample (529 words in Kazakh, 1662 in Uyghur). There was no upper limit on the test data; the largest treebank had a test set comprising 170K words.

Although the 63 test sets correspond to UD 2.0 treebanks, they were not released with UD 2.0. They were kept hidden and only published after the evaluation phase of the shared task (Nivre et al., 2017a).

2.4 New Parallel Test Sets

In addition, there were test sets for which no corresponding training data sets exist: 4 "surprise" languages (described in Section 2.5) and 14 test sets of a new Parallel UD (PUD) treebank (described in this section). These test sets were created for this shared task, i.e., not included in any previous UD release.

The PUD treebank consists of 1000 sentences currently in 18 languages (15 K to 27 K words, depending on the language), which were randomly picked from on-line newswire and Wikipedia;[7] usually only a few sentences per source document. 750 sentences were originally English, the remaining 250 sentences come from German, French, Italian and Spanish texts. They were translated by professional translators to 14 languages (i.e., 15 languages with the original: Arabic, Chinese, English, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Portuguese, Russian, Spanish, Thai and Turkish; but four languages—Chinese, Indonesian, Korean and Thai—were excluded from the shared task due to consistency issues). Translators were instructed to prefer translations closer to original grammatical structure, provided it is still a fluent sentence in the target language. In some cases, picking a correct translation was difficult because the translators did not see the context of the original document. The translations were organized at DFKI and text & form, Germany; they were then tokenized, morphologically and syntactically annotated at Google following guidelines based on McDonald et al. (2013), and finally converted to proper UD v2 annotation style by volunteers from the UD community using the Udapi framework (Popel et al., 2017).[8] Three additional translations (Czech, Finnish and Swedish) were contributed and annotated natively in UD v2 by teams from Charles University, University of Turku and Uppsala University, respectively.

The Google dependency representation predates Universal Dependencies, deriving from the scheme used by McDonald et al. (2013), i.e., Stanford Dependencies 2.0 with the option to make copula verbs heads (de Marneffe and Manning, 2008, section 4.7) and Google Universal POS tags (Petrov et al., 2011). Various tree transformations were needed to convert it to UD.[9] For example, prepositions and copula verbs are phrasal heads in Google annotation but must be dependent function words in UD. Similarly, some POS tags differ in the two schemes; particularly hard were conjunc-

[5] Using `udpipe -output=horizontal`.

[6] We had to withdraw the test set from the Italian ParTUT treebank because it turned out to significantly overlap with the training data of the larger Italian treebank in UD 2.0.

[7] The two domains are encoded in sentence ids but this information is not visible to the systems participating in the shared task.

[8] `http://udapi.github.io/`

[9] using ud.Google2ud from the Udapi framework

tions, where the Google tag set does not distinguish coordinators (CCONJ in UD) from subordinators (SCONJ). Some bugs, for example where verbs had multiple subjects or objects, or where function words were not leaves, were detected automatically[10] and fixed manually.

Finally, the most prominent consistency issues lay in tokenization and word segmentation, especially in languages where it interacts with morphology or where the writing system does not clearly mark word boundaries. The tokenizers used before manual annotation were not necessarily compatible with existing UD treebanks, yet in the shared task it was essential to make the segmentation consistent with the training data. We were able to fix some problems, such as unmarked multi-word tokens in European languages,[11] and we were even able to re-segment Japanese (note that this often involved new dependency relations); on the other hand, we had to exclude Korean for not being able to fix it in time.

Many transformations were specific to individual languages. For example, in the original tokenization of Arabic, the definite article *al-* was separated from the modified word, which is comparable to the D3 tokenization scheme (Habash, 2010). This scheme was inconsistent with the tokenization of the Arabic training data, hence it had to be changed. Text-level normalization further involved removal of the *shadda* diacritical mark (marking consonant gemination), which is optional in Arabic orthography and does not occur in the training data. On the POS level, the active and passive participles and verbal nouns (masdars) were annotated as verbs. For Arabic, however, these should be mapped to NOUN. Once we changed the tags, we also had to modify the surrounding relations to those used with nominals.

Like some UD treebanks, the parallel data contains information on document boundaries. They are projected as empty lines to the raw text presented to parsers, and they can be exploited to improve sentence segmentation. Note that due to the way the sentences were collected, the paragraphs are rather short.[12]

[10] using ud.MarkBugs from the Udapi framework

[11] using Udapi's ud.de.AddMwt for German, and similarly for Spanish (es), French (fr) and Portuguese (pt). For all languages, we applied ud.ComplyWithText to make sure the concatenation of tokens matches exactly the original raw text.

[12] A special case is Arabic where we artificially marked every sentence as a separate paragraph, to make it more consistent with somewhat unusual segmentation of the existing UD Arabic treebank. This gave an advantage to systems that were able to take paragraph boundaries into account, including those that re-used the baseline segmentation.

The fact that the data is parallel was not exploited in this task. Participating systems were told the language code so they could select an appropriate model. All parallel test sets were in languages that have at least one training treebank in UD 2.0 (although the domain may differ).

After the evaluation phase these parallel test sets were published together with the main test data; in the future they will become part of regular UD releases.

2.5 Surprise Languages

The second type of additional test sets were surprise languages, which had not been previously released in UD. Names of surprise languages (Buryat, Kurmanji Kurdish, North Sámi and Upper Sorbian) and small samples of gold-standard data (about 20 sentences) were published one week before the beginning of the evaluation phase. Crawled raw texts were provided too, though in much smaller quantity than for the other languages. The point of having surprise languages was to encourage participants to pursue truly multilingual approaches to parsing, utilizing data from other languages.

As with all other test sets, the systems were able to use segmentation and part-of-speech tags predicted by the baseline UDPipe system (in this case UDPipe was trained and applied in a 10-fold cross-validation manner directly on the test data; hence this is the only annotation that the participants were given but could not produce with their own models).

Note that the smallest non-surprise languages (Kazakh, Uyghur) were asking for multilingual approaches as well, given that the amount of their own training data was close to zero. The difference was that participants at least knew in advance what these languages were and had more time to determine the most suitable training model. On the other hand, the segmentation and tagging models for these languages were only trained on the tiny training data, i.e., they were much worse than the models for the surprise languages. In this sense parsing of Kazakh and Uyghur was even harder than parsing the surprise languages.

When compared to the training data available in UD 2.0, the genetically closest language to Kazakh and Uyghur is Turkish; but it uses a dif-

ferent writing system, and the Turkish dataset itself is not particularly large. For Kurmanji Kurdish, the closest relative is Persian, again with different script and other reservations. Buryat is a Mongolic language written in Cyrillic script and does not have any close relative in UD. North Sámi is an Finno-Ugric language; Finnish and Estonian UD data could be expected to be somewhat similar. Finally, Upper Sorbian is a West Slavic language spoken in Germany; among the many Slavic languages in UD, Czech and Polish are its closest relatives.

In summary, the test data consisted of 81 files in 49 languages (55 test sets from "big" UD 2.0 treebanks, 8 "small" treebanks, 14 parallel test sets and 4 surprise-language test sets).

3 Evaluation Metrics

The standard evaluation metric of dependency parsing is the *labeled attachment score (LAS),* i.e., the percentage of nodes with correctly assigned reference to parent node, including the label (type) of the relation. When parsers are applied to raw text, the metric must be adjusted to the possibility that the number of nodes in gold-standard annotation and in the system output vary. Therefore, the evaluation starts with aligning system nodes and gold nodes. A dependency relation cannot be counted as correct if one of the nodes could not be aligned to a gold node. LAS is then re-defined as the harmonic mean (F_1) of precision P and recall R, where

$$P = \frac{\#\mathit{correctRelations}}{\#\mathit{systemNodes}} \quad (1)$$

$$R = \frac{\#\mathit{correctRelations}}{\#\mathit{goldNodes}} \quad (2)$$

$$LAS = \frac{2PR}{P+R} \quad (3)$$

Note that attachment of all nodes including punctuation is evaluated. LAS is computed separately for each of the 81 test files and a macro-average of all these scores serves as the main metric for system ranking in the task.

3.1 Token Alignment

UD defines two levels of token/word segmentation. The lower level corresponds to what is usually understood as tokenization. However, unlike some popular tokenization schemes, it does not include any normalization of the non-whitespace characters. We can safely assume that any two tokenizations of a text differ only in whitespace while the remaining characters are identical. There is thus a 1-1 mapping between gold and system non-whitespace characters, and two tokens are aligned if all their characters match.

3.2 Syntactic Word Alignment

The higher segmentation level is based on the notion of *syntactic word*. Some languages contain *multi-word tokens* (MWT) that are regarded as contractions of multiple syntactic words. For example, the German token *zum* is a contraction of the preposition *zu* "to" and the article *dem* "the".

Syntactic words constitute independent nodes in dependency trees. As shown by the example, it is not required that the MWT is a pure concatenation of the participating words; the simple token alignment thus does not work when MWTs are involved. Fortunately, the CoNLL-U file format used in UD clearly marks all MWTs so we can detect them both in system output and in gold data. Whenever one or more MWTs have overlapping spans of surface character offsets, the longest common subsequence algorithm is used to align syntactic words within these spans.

3.3 Sentence Segmentation

Words are aligned and dependencies are evaluated in the entire file without considering sentence segmentation. Still, the accuracy of sentence boundaries has an indirect impact on LAS: any missing or extra sentence boundary necessarily makes one or more dependency relations incorrect.

3.4 Invalid Output

If a system fails to produce one of the 81 files or if the file is not valid CoNLL-U format, the score of that file (counting towards the system's macro-average) is zero.

Formal validity is defined more leniently than for UD-released treebanks. For example, a non-existent dependency type does not render the whole file invalid, it only costs the system one incorrect relation. However, cycles and multi-root sentences are disallowed. A file is also invalid if there are character mismatches that could make the token alignment algorithm fail.

3.5 CLAS

Content-word Labeled Attachment Score (CLAS) has been proposed as an alternative parsing metric that is tailored to the UD annotation style and more suitable for cross-language comparison (Nivre and Fang, 2017). It differs from LAS in that it only considers relations between content words. Attachment of function words is disregarded because it corresponds to morphological features in other languages (and morphology is not evaluated in this shared task). Furthermore, languages with many function words (e.g., English) have longer sentences than morphologically rich languages (e.g., Finnish), hence a single error in Finnish costs the parser significantly more than an error in English. CLAS also disregards attachment of punctuation.

As CLAS is still experimental, we have designated full LAS as our main evaluation metric; nevertheless, a large evaluation campaign like this is a great opportunity to study the behavior of the new metric, and we present both scores in Section 6.

4 Evaluation Methodology

Key goals of any empirical evaluation are to ensure a blind evaluation, its replicability, and its reproducibility. To facilitate these goals, we employed the cloud-based evaluation platform TIRA (Potthast et al., 2014),[13] which implements the evaluation as a service paradigm (Hanbury et al., 2015). In doing so, we depart from the traditional submission of system output to shared tasks, which lacks in these regards, toward the submission of working software. Naturally, software submissions bring about additional overhead for both organizers and participants, whereas the goal of an evaluation platform like TIRA is to reduce this overhead to a bearable level. Still being an early prototype, though, TIRA fulfills this goal only with some reservations. Nevertheless, the scale of the CoNLL 2017 UD Shared Task also served as a test of scalability of the evaluation as a service paradigm in general as well as that of TIRA in particular.

4.1 Blind Evaluation

Traditionally, evaluations in shared tasks are half-blind (the test data are shared with participants while the ground truth is withheld), whereas outside shared tasks, say, during paper-writing, evaluations are typically pseudo-blind (the test data and ground truth are accessible, yet, ignored until the to-be-evaluated software is ready). In both cases, remaining blind to the test data is one of the cornerstones of evaluation, and has a significant impact on the validity of evaluation results. While outside shared tasks, one can only trust that paper authors do not spoil their evaluation by implicitly or explicitly exploiting their knowledge of the test data, within shared tasks, another factor comes into play, namely the fact that shared tasks are also competitions.

Dependent on its prestige, winning a shared task comes along with a lot of visibility, so that supplying participants with the test data up front bears risks of mistakes that spoil the ground truth, and of cheating. Here, TIRA implements a proper solution which ensures blind evaluation, an airlock for data. On demand, software deployed at TIRA is locked in the datalock together with the test data, where it can process the data and have its output recorded. Otherwise, all communication channels to the outside are closed or tightly moderated to prevent data leakage. However, closing down all communication channels also has its downsides, since participants cannot check up on their running software anymore, or have to ask organizers to do so, which increases the turnaround time to fix bugs. Participants were only able to learn whether they achieved a non-zero score on each of the 81 test files; a zero score signaled a bug, in which case the task moderator would make the diagnostic output visible to the participants. Such interaction was only possible when the system run completed; before that, even the task moderator would not see whether the system was really producing output and not just sitting in an endless loop. Especially given the scale of operations this year, this turned out to be a major obstacle for some participants; TIRA needs to be improved by offering more fine-grained process monitoring tools, both for organizers and participants.

4.2 Replicability and Reproducibility

The replicability of an evaluation depends on whether the same results can be obtained from re-running an experiment using the same setup, whereas reproducibility refers to achieving results that are commensurate with a reference evaluation, for instance, when exchanging the test data with alternative test data. Both are important aspects of an evaluation, the former pertaining to

[13] http://www.tira.io/

its reliability, and the latter to its validity. Ensuring both requires that a to-be-evaluated software is preserved in working condition for as long as possible. Traditionally, shared tasks do not take charge of participant software preservation, mostly because the software remains with participants, and since open sourcing the software underlying a paper is still the exception rather than the rule. To ensure both, TIRA supplies participants with a virtual machine, offering a range of commonly used operating systems in order not to limit the choice of technology stacks and development environments. Once deployed and tested, the virtual machines are archived to preserve the software within.

Many participants agreed to share their code so that we decided to collect the respective projects in a kind of open source proceedings at GitHub.[14]

4.3 Resource Allocation

The allocation of an appropriate amount of computing resources (especially CPUs and RAM, whereas disk space is cheap enough) to each participant proved to be difficult, since minimal requirements were unknown. When asked, participants typically request liberal amounts of resources, just to be on the safe side, whereas assigning too much up front would not be economical nor scale well. We hence applied a least commitment strategy with an initial assignment of 1 CPU and 4 GB RAM. More resources were granted on request, the limit being the size of the underlying hardware. When it comes to exploiting available resources, a lot depends on programming prowess, whereas more resources do not necessarily translate into better performance. This is best exemplified by the fact that with 4 CPUs and 16 GB RAM, the winning team Stanford used only a quarter the amount of resources of the second and third winners, respectively. The team on fourth (sixth) place was even more frugal, getting by with 1 CPU and 8 GB RAM (4 GB RAM). All of the aforementioned teams' approaches exceed the LAS level of 70%.

5 Baseline System

5.1 UDPipe

We prepared a set of baseline models using UDPipe (Straka et al., 2016) version 1.1. A slightly improved version—UDPipe 1.2—was submitted by Straka and Straková (2017) as one of the competing systems. Straka and Straková (2017) describe both these versions in more detail.

The baseline models were released together with the UD 2.0 training data, one model for each treebank. Because only training and development data were available during baseline model training, we put aside a part of the training data for hyperparameter tuning, and evaluated the baseline model performance on development data. We called this data split *baseline model split*. The baseline models, the baseline model split, and also UD 2.0 training data with morphology predicted by 10-fold jack-knifing (cross-validation), are available on-line (Straka, 2017).

UDPipe baseline models are able to reconstruct nearly all annotation from CoNLL-U files – they can generate segmentation, tokenization, multiword token splitting, morphological annotation (lemmas, UPOS, XPOS and FEATS) and dependency trees. Participants were free to use any part of the model in their systems – for all test sets, we provided UDPipe processed variants in addition to raw text inputs. We provided the UDPipe processed variant even for surprise languages – however, only segmentation, tokenization and morphology, generated by 10-fold jack-knifing, as described in Section 2.5.

Baseline UDPipe Shared Task System We further used the baseline models as a baseline system in the shared task. We used the corresponding models for the UD 2.0 test data.

For the new parallel treebanks, we used UD 2.0 baseline models of the corresponding languages. If there were several treebanks for one language, we arbitrarily chose the one named after the language only (e.g., we chose `ru` and not `ru_syntagrus`). Unfortunately, we did not explicitly mention this choice to the participants and this arbitrary choice had a large impact on results – some contestant systems fell below UDPipe baseline just because of choosing different treebanks to train on for the parallel treebanks. (On the other hand, there was no guarantee that the models selected in the baseline system would be optimal.)

For each surprise language, we also chose one baseline model to apply. Even if most words are unknown to the baseline model, universal POS tags can be used to drive the parsing, making the baseline model act similar to a delexicalized parser. We chose a baseline model to maximize

[14] https://github.com/CoNLL-UD-2017

	Team	LAS
1.	Stanford (Dozat et al.)	76.30
2.	C2L2 (Shi et al.)	75.00
3.	IMS (Björkelund et al.)	74.42
4.	HIT-SCIR (Che et al.)	72.11
5.	LATTICE (Lim and Poibeau)	70.93
6.	NAIST SATO (Sato et al.)	70.14
7.	Koç University (Kırnap et al.)	69.76
8.	ÚFAL (Straka and Straková)	69.52
9.	UParse (Vania et al.)	68.87
10.	Orange (Heinecke and Asadullah)	68.61
11.	TurkuNLP (Kanerva et al.)	68.59
12.	darc (Yu et al.)	68.41
13.	BASELINE UDPipe 1.1	68.35
14.	MQuni (Nguyen et al.)	68.05
15.	fbaml (Qian and Liu)	67.87
16.	LyS (Vilares and Gómez-Rodríguez)	67.81
17.	LIMSI (Aufrant and Wisniewski)	67.72
18.	RACAI (Dumitrescu et al.)	67.71
19.	IIT Kharagpur (Das et al.)	67.61
20.	naistCL (no paper)	67.59
21.	Wanghao-ftd-SJTU (Wang et al.)	66.53
22.	UALING (Hornby et al.)	65.24
23.	Uppsala (de Lhoneux et al.)	65.11
24.	METU (Akkuş et al.)	61.98
25.	CLCL (Moor et al.)	61.82
26.	Mengest (Ji et al.)	61.33
27.	ParisNLP (De La Clergerie et al.)	60.02
28.	OpenU (More and Tsarfaty)	56.56
29.	TRL (Kanayama et al.)	43.07
30.	MetaRomance (Garcia and Gamallo)	34.05
31.	UT (no paper)	21.10
32.	ECNU (no paper)	3.18
33.	Wenba-NLU (no paper)	0.58

Table 2: Ranking of the participating systems by the main evaluation metric, the labeled attachment F_1-score, macro-averaged over 81 test sets. Pairs of systems with significantly ($p < 0.05$) different LAS are separated by a line. Names of several teams are abbreviated in the table: LyS-FASTPARSE, OpenU NLP Lab, Orange – Deskiñ and ÚFAL – UDPipe 1.2. Citations refer to the corresponding system-description papers in this volume.

the accuracy on the released sample for each surprise language, resulting in Finnish FTB, Polish, Finnish FTB and Slovak models for the surprise languages Buryat, Kurmanji, North Sámi and Upper Sorbian, respectively.

	Team	CLAS F_1
1.	Stanford (Stanford)	72.57
2.	C2L2 (Ithaca)	70.91
3.	IMS (Stuttgart)	70.18
4.	HIT-SCIR (Harbin)	67.63
5.	LATTICE (Paris)	66.16
6.	NAIST SATO (Nara)	65.15
7.	Koç University (İstanbul)	64.61
8.	ÚFAL – UDPipe 1.2 (Praha)	64.36
9.	Orange – Deskiñ (Lannion)	64.15
10.	TurkuNLP (Turku)	63.61
11.	UParse (Edinburgh)	63.55
12.	darc (Tübingen)	63.24
13.	BASELINE UDPipe 1.1	63.02

Table 3: Average CLAS F_1 score.

5.2 SyntaxNet

Another set of baseline models was prepared by Alberti et al. (2017) based on improved version of the SyntaxNet system (Andor et al., 2016). Pretrained models were provided for UD 2.0 data.

However, no SyntaxNet models were prepared for the surprise languages, therefore, the SyntaxNet baseline is not part of the official results.

6 Results

6.1 Official Parsing Results

Table 2 gives the main ranking of participating systems by the LAS F_1 score macro-averaged over all 81 test files. The table also shows the performance of the baseline UDPipe system; the baseline is relatively strong and only 12 of the 32 systems managed to outperform it.

We used bootstrap resampling to compute 95% confidence intervals: they are in the range ± 0.11 to ± 0.15 (% LAS) for all systems except the three lowest-scoring ones. We used paired bootstrap resampling to compute whether the difference in LAS is significant ($p < 0.05$) for each pair of systems.[15]

6.2 Secondary Metrics

In addition to the main LAS ranking, we evaluated the systems along multiple other axes, which may

[15] using Udapi's eval.Conll17, marked by the presence or absence of vertical lines in Table 2.

	Team	Toks	Wrds	Sents
1.	IMS	98.92	98.81	89.10
2.	LIMSI	**98.95**	98.68	88.49
3.	ÚFAL – UDPipe 1.2	98.89	98.63	**88.68**
4.	HIT-SCIR	**98.95**	98.62	**88.91**
5.	ParisNLP	98.85	98.58	88.61
6.	Wanghao-ftd-SJTU	98.81	98.55	88.40
	darc	98.81	98.55	**88.66**
8.	BASELINE UDPipe	98.77	98.50	88.49
	C2L2	98.77	98.50	88.49
	CLCL	98.77	98.50	88.49
	IIT Kharagpur	98.77	98.50	88.49
	Koç University	98.77	98.50	88.49
	LATTICE	98.77	98.50	88.49
	LyS-FASTPARSE	98.77	98.50	88.49
	METU	98.77	98.50	88.49
	MQuni	98.77	98.50	88.49
	NAIST SATO	98.77	98.50	88.49
	Orange – Deskiñ	98.77	98.50	88.49
	Stanford	98.77	98.50	88.49
	TurkuNLP	98.77	98.50	88.49
	UALING	98.77	98.50	88.49
	UParse	98.77	98.50	88.49
	naistCL	98.77	98.50	88.49
24.	RACAI	98.58	98.39	87.52
25.	OpenU NLP Lab	**98.77**	98.38	**88.49**
26.	Uppsala	97.64	98.20	**89.03**

Table 4: Tokenization, word segmentation and sentence segmentation (ordered by word F_1 scores; out-of-order scores in the other two columns are bold).

shed more light on their strengths and weaknesses. This section provides an overview of selected secondary metrics for systems matching or surpassing the baseline; a large number of additional results is available at the shared task website.[16]

The website also features a LAS ranking of unofficial system runs, i.e. those that were not marked by their teams as primary runs, or were even run after the official evaluation phase closed and test data were unblinded. At least two differences from the official results are remarkable; both seem to be partially inflicted by the blind evaluation on TIRA and the inability of the participants to see the diagnostic messages from their software. In the first case, the Dynet library seems to produce suboptimal results when deployed on a machine different from the one where it was trained. Several teams used the library and may have been affected; for the Uppsala team (de Lhoneux et al., 2017) the issue led to official LAS = 65.11 (23rd place) instead of 69.66 (9th place). In the second case, the ParisNLP system (De La Clergerie et al., 2017) used a wrong method of recognizing the input language, which was not supported in the test data (but unfortunately it was possible to get along with it in development and trial data). Simply crashing could mean that the task moderator would show the team their diagnostic output and they would fix the bug; however, the parser was robust enough to switch to a language-agnostic mode and produced results that were not great, but also not so bad to alert the moderator and make him investigate. Thus the official LAS of the system is 60.02 (27th place) while without the bug it could have been 70.35 (6th place).

[16] http://universaldependencies.org/conll17/results.html

	Team	UPOS	Feats	Lemm
1.	Stanford	93.09	38.81	82.46
2.	IMS	91.98	82.99	62.83
3.	ParisNLP	91.91	38.89	75.32
4.	ÚFAL – UDPipe 1.2	91.22	82.50	71.17
5.	HIT-SCIR	91.13	81.90	83.74
6.	TurkuNLP	91.10	82.58	82.64
7.	LIMSI	91.05	82.49	82.64
8.	darc	91.00	82.48	82.60
9.	CLCL	90.88	82.31	82.46
10.	BASELINE UDPipe	90.88	82.31	82.45
	C2L2	90.88	82.31	82.46
	IIT Kharagpur	90.88	82.31	82.46
	Koç University	90.88	82.31	82.46
	LATTICE	90.88	82.31	82.46
	LyS-FASTPARSE	90.88	82.31	79.14
	NAIST SATO	90.88	82.31	82.46
	Orange – Deskiñ	90.88	38.81	15.38
	UALING	90.88	82.31	82.46
	UParse	90.88	82.31	82.46
	naistCL	90.88	82.31	82.46

Table 5: Universal POS tags, features and lemmas (ordered by UPOS F_1 scores).

Table 3 ranks the systems by CLAS instead of LAS (see Section 3.5). The scores are lower than LAS but differences in system ranking are minimal, possibly indicating that optimization towards

one of the metrics does not make the parser bad with respect to the other.

Table 4 evaluates detection of tokens, syntactic words and sentences. Half of the systems simply trusted the segmentation offered by the baseline system. 7 systems were able to improve baseline segmentation. For most languages and in aggregate, the ability to improve parsing scores through better segmentation was probably negligible, but for a few languages, such as Chinese and Vietnamese, the UDPipe baseline segmentation was not so strong and several teams, notably IMS, appear to have improved their LAS by several percent through use of improved segmentation.

The systems were not required to generate any morphological annotation (part-of-speech tags, features or lemmas). Some parsers do not even need morphology and learn to predict syntactic dependencies directly from text. Nevertheless, systems that did output POS tags, and had them at least as good as the baseline system, are evaluated in Table 5. Note that as with segmentation, morphology predicted by the baseline system was available and some systems simply copied it to the output.

6.3 Partial Results

Table 6 gives the LAS F_1 score averaged over the 55 "big" treebanks (training data larger than test data, development data available). Higher scores reflect the fact that models for these test sets are easier to learn: enough data is available, no cross-lingual or cross-domain learning is necessary (the parallel test sets are not included here). When compared to Table 2, four new teams now surpass the baseline, LyS-FASTPARSE being the best among them. The likely explanation is that the systems can learn good models but are not so good at picking the right model for unknown domains and languages.

Table 7 gives the LAS F_1 score on the four surprise languages only. The globally best system, Stanford, now falls back to the fourth rank while C2L2 (Cornell University) apparently employs the most successful strategy for underresourced languages. Another immediate observation is that our surprise languages are very hard to parse; accuracy under 50% is hardly useful for any downstream processing. However, there are significant language-by-language differences, the best score on Upper Sorbian surpassing 60%. This probably owes to the presence of many Slavic treebanks in training data, including some of the largest datasets in UD.

	Team	LAS F_1
1.	Stanford (Stanford)	81.77
2.	C2L2 (Ithaca)	79.85
3.	IMS (Stuttgart)	79.60
4.	HIT-SCIR (Harbin)	77.45
5.	LATTICE (Paris)	75.79
6.	NAIST SATO (Nara)	75.64
7.	LyS-FASTPARSE (A Coruña)	74.55
8.	Koç University (İstanbul)	74.39
9.	ÚFAL – UDPipe 1.2 (Praha)	74.38
10.	TurkuNLP (Turku)	74.19
11.	Orange – Deskiñ (Lannion)	74.13
12.	MQuni (Sydney)	74.03
13.	LIMSI (Paris)	73.64
14.	UParse (Edinburgh)	73.56
15.	darc (Tübingen)	73.31
16.	fbaml (Palo Alto)	73.11
17.	BASELINE UDPipe 1.1	73.04

Table 6: Average attachment score on the 55 "big" treebanks.

In contrast, the results on the 8 small non-surprise treebanks (Table 8) are higher on average, but again the variance is huge. Uyghur (best score 43.51) is worse than three surprise languages, and Kazakh (best score 29.22) is the least parsable test set of all (see Table 10). These two treebanks are outliers in the size of training data (529 words Kazakh and 1662 words Uyghur, while the other "small" treebanks have between 10K and 20K words). However, the only "training data" of the surprise languages are samples of 147 to 460 words, yet they seem to be easier for some systems. It would be interesting to know whether the more successful systems took a similar approach to Kazakh and Uyghur as to the surprise languages.

Table 9 gives the average LAS on the 14 new parallel test sets (PUD). Three of them (Turkish, Arabic and Hindi) proved difficult to parse for any model trained on the UD 2.0 training data; it seems likely that besides domain differences, inconsistent application of the UD annotation guidelines played a role, too.

See Table 10 for a ranking of all test sets by the best LAS achieved on them by any parser. Note that this cannot be directly interpreted as a

Team	LAS F_1
1. C2L2 (Ithaca)	47.54
2. IMS (Stuttgart)	45.32
3. HIT-SCIR (Harbin)	42.64
4. Stanford (Stanford)	40.57
5. ParisNLP (Paris)	39.22
6. UParse (Edinburgh)	39.17
7. Koç University (İstanbul)	38.81
8. Orange – Deskiñ (Lannion)	38.72
9. LIMSI (Paris)	37.57
10. IIT Kharagpur (Kharagpur)	37.17
11. BASELINE UDPipe 1.1	37.07

Table 7: Average attachment score on the 4 surprise languages: Buryat (bxr), Kurmanji (kmr), North Sámi (sme) and Upper Sorbian (hsb).

Team	LAS F_1
1. C2L2 (Ithaca)	61.49
2. Stanford (Stanford)	61.02
3. IMS (Stuttgart)	58.76
4. LATTICE (Paris)	54.78
5. HIT-SCIR (Harbin)	54.77
6. fbaml (Palo Alto)	54.64
7. RACAI (Bucureşti)	54.26
8. TurkuNLP (Turku)	54.19
9. ÚFAL – UDPipe 1.2 (Praha)	53.76
10. NAIST SATO (Nara)	53.52
11. Koç University (İstanbul)	53.36
12. darc (Tübingen)	52.46
13. UALING (Tucson)	52.27
14. Wanghao-ftd-SJTU (Shanghai)	52.13
15. BASELINE UDPipe 1.1	51.80

Table 8: Average attachment score on the 8 small treebanks: French ParTUT, Galician TreeGal, Irish, Kazakh, Latin, Slovenian SST, Uyghur and Ukrainian.

ranking of languages by their parsing difficulty: many treebanks have high ranks simply because the corresponding training data is large. The table also gives a secondary ranking by CLAS and indicates the system that achieved the best LAS / CLAS (mostly the same system won by both metrics). Finally, the best score of word and sentence segmentation is given (without indicating the best-scoring system). Vietnamese proved to be the hardest language in terms of word segmentation; it is not surprising given that its writing system allows spaces inside words. Second hardest was Hebrew, probably due to a large number of multi-word tokens. In both cases the poor segmentation correlates with poor parsing accuracy. Sentence segmentation was particularly difficult for treebanks without punctuation, i.e., most of the classical languages and spoken data (the best score achieved on the Spoken Slovenian Treebank is only 21.41%). On the other hand, the paragraph boundaries available in some treebanks made sentence detection significantly easier (the extreme being Arabic PUD with one sentence per paragraph; some systems were able to exploit this anomaly and get 100% correct segmentation).

Team	LAS F_1
1. Stanford (Stanford)	73.73
2. C2L2 (Ithaca)	71.49
3. IMS (Stuttgart)	71.31
4. LATTICE (Paris)	70.77
5. NAIST SATO (Nara)	69.83
6. Koç University (İstanbul)	69.76
7. HIT-SCIR (Harbin)	69.51
8. MQuni (Sydney)	69.28
9. ÚFAL – UDPipe 1.2 (Praha)	69.00
10. UParse (Edinburgh)	68.91
11. Orange – Deskiñ (Lannion)	68.64
12. TurkuNLP (Turku)	68.56
13. BASELINE UDPipe 1.1	68.33

Table 9: Average attachment score on the 14 parallel test sets (PUD).

7 Analysis of Submitted Systems

Table 11 gives an overview of 29 of the systems evaluated in the shared task. The overview is based on a post-evaluation questionnaire to which 29 of 32 teams responded. The abbreviations used in Table 11 are explained in Table 12.

As we can see from Table 11, the typical system uses the baseline models for segmentation and morphological analysis (including part-of-speech tagging), employs a single parsing model with pretrained word embeddings provided by the organizers, and does not make use of any additional data. For readability, all the cells corresponding to use of baseline models (and lack of additional data) have been shaded gray.

Only 7 teams have developed their own word and sentence segmenters, while an additional 5

Treebank	LAS F_1	CLAS F_1	Best system	Word	Sent
1. ru_syntagrus	92.60	1. 90.11	Stanford	99.69	98.64
2. hi	91.59	6. 87.92	Stanford	100.00	99.29
3. sl	91.51	**2. 88.98**	Stanford	99.96	99.24
4. pt_br	91.36	8. 87.48	Stanford	99.86	96.84
5. ja	91.13	26. 83.18	TRL	98.59	95.11
6. ca	90.70	**10. 86.70**	Stanford	99.97	99.43
7. it	90.68	13. 86.18	Stanford	99.85	99.07
8. cs_cac	90.43	**4. 88.31**	Stanford	99.99	100.00
9. pl	90.32	5. 87.94	Stanford	99.90	99.59
10. cs	90.17	**3. 88.44**	Stanford	99.99	95.10
11. es_ancora	89.99	14. 86.15	Stanford	99.95	98.67
12. no_bokmaal	89.88	**7. 87.67**	Stanford	99.88	96.44
13. bg	89.81	11. 86.53	Stanford	99.92	93.36
14. no_nynorsk	88.81	12. 86.41	Stanford	99.93	94.56
15. fi_pud	88.47	**9. 86.82**	Stanford	99.63	93.67
16. it_pud	88.14	17. 84.49	Stanford	99.27	97.81
17. fr_partut	88.13	24. 83.58	C2L2	99.56	99.13
18. nl_lassysmall	87.71	**15. 85.22**	Stanford	99.99	85.33
19. pt	87.65	25. 83.27	Stanford	99.54	91.67
20. el	87.38	**23. 83.59**	Stanford	99.94	92.68
21. fr_sequoia	87.31	**20. 84.09**	C2L2	99.49	84.60
22. es	87.29	32. 82.08	Stanford	99.81	95.37
23. la_ittb	87.02	**16. 84.94**	Stanford	99.99	94.34
24. fi_ftb	86.81	19. 84.12	Stanford	99.99	86.98
25. fa	86.31	28. 82.93	Stanford	99.65	99.25
26. sk	86.04	**21. 83.86**	Stanford	100.00	85.32
27. ro	85.92	33. 81.87	Stanford	99.77	96.57
28. sv	85.87	**22. 83.71**	Stanford	99.87	97.26
29. cs_cltt	85.82	27. 83.05	C2L2	99.82	95.69
30. fi	85.64	**18. 84.25**	Stanford	99.69	90.88
31. en_pud	85.51	29. 82.63	Stanford	99.74	98.06
32. fr	85.51	31. 82.14	Stanford	99.50	94.58
33. hr	85.25	**30. 82.36**	Stanford	99.93	97.75
34. en_partut	84.46	39. 79.80	C2L2	99.61	98.40
35. cs_pud	84.42	**35. 81.60**	Stanford	99.29	96.43
36. ja_pud	83.75	50. 75.63	HIT-SCIR	94.93	97.52
37. ru	83.65	**34. 81.80**	Stanford	99.94	97.16
38. gl	83.23	43. 78.05	Stanford	99.98	96.36
39. da	82.97	**37. 80.03**	Stanford	100.00	82.59
40. sv_lines	82.89	38. 79.92	Stanford	99.98	87.89
41. ko	82.49	**36. 80.85**	Stanford	99.73	93.05
42. ur	82.28	49. 75.88	Stanford	100.00	98.60
43. en	82.23	**41. 78.99**	Stanford	99.03	78.01
44. en_lines	82.09	42. 78.71	Stanford	99.96	87.55
45. eu	81.44	**40. 79.71**	Stanford	99.99	99.83
46. es_pud	81.05	53. 74.60	Stanford	99.48	98.19
47. de	80.71	**46. 76.97**	Stanford	99.67	80.47
48. nl	80.48	52. 75.19	Stanford	99.88	77.14
49. id	79.19	**45. 77.15**	Stanford	100.00	92.66
50. fr_pud	78.81	**44. 77.37**	Stanford	98.87	96.55
51. sv_pud	78.49	47. 76.48	Stanford	98.56	95.52
52. pt_pud	78.48	56. 72.80	C2L2	99.45	97.32
53. hu	77.56	**48. 76.08**	Stanford	99.85	96.56
54. cu	76.84	51. 75.59	IMS	100.00	50.44
55. ru_pud	75.71	55. 73.13	Stanford	98.29	98.95
56. uk	75.33	57. 71.72	Stanford	99.92	95.75
57. grc_proiel	75.28	60. 69.73	IMS	100.00	51.38
58. de_pud	74.86	**54. 73.96**	Stanford	98.00	91.40
59. gl_treegal	74.34	65. 67.59	C2L2	98.76	86.74
60. lv	74.01	**58. 70.22**	Stanford	99.45	98.80
61. grc	73.19	64. 67.59	Stanford	100.00	98.96
62. ar	72.90	**61. 69.15**	IMS	95.53	85.69
63. et	71.65	**59. 69.85**	Stanford	99.89	93.66
64. la_proiel	71.55	63. 68.93	IMS	100.00	40.63
65. got	71.36	**62. 69.02**	IMS	100.00	41.65
66. ga	70.06	67. 61.38	Stanford	99.73	96.92
67. zh	68.56	**66. 64.23**	IMS	94.57	98.80
68. he	68.16	68. 61.10	IMS	91.37	100.00
69. la	63.37	70. 58.96	Stanford	100.00	99.20
70. tr	62.79	**69. 60.01**	Stanford	97.95	97.04
71. hsb	61.70	71. 56.32	C2L2 / Stanford	99.84	91.65
72. sl_sst	59.07	72. 54.30	C2L2	100.00	21.41
73. hi_pud	54.49	73. 48.87	Stanford	99.65	94.85
74. ar_pud	49.94	75. 46.32	IMS	96.05	100.00
75. sme	48.96	**74. 48.42**	C2L2	99.88	99.13
76. kmr	47.53	76. 44.54	C2L2	98.85	98.64
77. vi	47.51	77. 44.12	IMS	87.30	92.95
78. ug	43.51	78. 34.07	IMS	99.94	70.47
79. tr_pud	38.22	79. 32.32	IMS	96.93	93.91
80. bxr	32.24	80. 26.32	IMS / ParisNLP	99.35	93.69
81. kk	29.22	81. 25.14	RACAI	96.56	89.35

Table 10: Treebank ranking by best parser LAS. Bold CLAS is higher than the preceding one. Best F_1 of word and sentence segmentation is also shown. ISO 639 language codes are optionally followed by a treebank code.

teams have retrained or improved the baseline models, or combined them with other techniques. When it comes to part-of-speech tags and morphology, 7 teams use their own systems and 4 use modified versions of the baseline, while 2 teams predict tags jointly with parsing and 3 teams do not predict morphology at all.

For parsing, most teams use a single parsing model – transition-based, graph-based or even rule-based – but 4 teams build ensemble systems in one way or the other. It is worth noting that, whereas the C2L2 and IMS systems are ensembles, the winning Stanford system is not, which makes its performance even more impressive.

The majority of parsers incorporate pre-trained word embeddings. Only 3 parsers use word embeddings without pre-training, and only 4 parsers do not incorporate word embeddings at all. Except for training word embeddings, the additional data provided (or permitted) appears to have been used very sparingly.

When it comes to the surprise languages (and some of the other low-resource languages), the dominant approach is to use a cross-lingual parser, single- or multi-source, and often delexicalized. Finally, for the parallel test sets, most teams have picked a model trained on a single treebank from the same language, but at least 4 teams have trained models on multiple treebanks.

8 Conclusion

The CoNLL 2017 Shared Task on UD parsing was novel in several respects. Besides using cross-linguistically consistent linguistic representations and emphasizing end-to-end processing of text, as discussed in the introduction, it was unusual also in featuring a very large number of languages, in integrating cross-lingual learning for resource-poor languages, and in using a multiply parallel test set.

It was the first large-scale evaluation on data annotated in the Universal Dependencies style. For most UD languages the results represent a new state of the art for dependency parsing. The numbers are not directly comparable to some older work for various reasons (different annotation schemes, gold-standard POS tags, tokenization etc.) but the way the task was organized should ensure their reproducibility and comparability in the future. Furthermore, parsing results are now more comparable across languages than ever before.

System	R	Segment	Morph/POS	Parsing	Embed	AddData	Surp	Para
C2L2	2	Base	Aux	Ensemble-GT	Random	None	Cross-MD	Single
CLCL	25	Base	Base	Single	Random	None	Cross-M	?
darc	12	UDP	Own	Single-T	Base	None	Cross	Single
fbaml	15	Own	Own	Single	Base	None	Mono	?
HIT-SCIR	4	Own	None	Single/Ensemble	Base	OPUS	Cross	Single
IIT Kharagpur	19	Base	Base	Single-T	Base	None	Cross-MD	All/Single
IMS	3	B/O	Own	Ensemble-GT	Base	None	Cross	All
Koç University	7	Base	Base	Single	Crawl	None	Cross	?
LATTICE	5	Base	Base	Single	B/O/FB	Wiki/OPUS	Cross	All
LIMSI	17	B/UDP	Base	B/Single-T	Base	OPUS	Cross	Single
LyS-FASTPARSE	16	Base	Base	Single-T	Base	None	Cross	Single
Mengest	26	Base	Base	Single-T	Crawl	None	Canon	Single
MetaRomance	30	Base	Base	Single-R	None	None	Canon	?
METU	24	Base	Base	Single-T	Base	PTB/CCG	Cross	Single
MQuni	14	Base	Joint	Single-G	Random	None	Mono	Single
NAIST SATO	6	Base	Base	Single	Base	None	Canon	Single
OpenU NLP Lab	28	B/UDP	B/O	Single-T	None	None	Cross	Single
Orange-Deskiñ	10	Base	None	Single	Crawl	None	Cross	Single
ParisNLP	27	B/UDP/O	B/UDP/O/AG	Single	B/C	None	Cross	Single
RACAI	18	Own	Own	Single-G	Crawl	None	Cross	?
Stanford	1	Base	Own	Single-G	B/FB	None	Cross-MD	Single
TRL	29	Own	Own	Single-R	None	None	Cross-SD	Single
TurkuNLP	11	Base	Base/UDP	Single-T	Crawl	None	Cross-S	Single
UALING	22	Base	Base	Base	Base	None	Cross	?
ÚFAL – UDPipe 1.2	8	Own	UDP	Single-T	Treebank	None	Cross	All/Single
UParse	9	Base	Base	B/Single-G	O/FB	OPUS	Cross	Single
Uppsala	23	Own	None	Single-T	Treebank	None	Cross-M	Single
UT	31	Base	Own/AG	Ensemble	FB	None	Cross-S	?
Wanghao-ftd-SJTU	21	Own	Base	Single	None	None	Cross-D	?

Table 11: Classification of participating systems. The second column repeats the main system ranking.

Two new language resources were produced whose usefulness reaches far beyond the task itself: A UD-style parallel treebank in 18 languages, and a large, web-crawled parsebank in 48 languages, over 90 billion words in total.

The analysis of the shared task results has so far only scratched the surface, and we refer to the system description papers for more in-depth analysis of individual systems and their performance. For many previous CoNLL shared tasks, the task itself has only been the starting point of a long and fruitful research strand, enabled by the resources created for the task. We hope and believe that the 2017 UD parsing task will join this tradition.

Acknowledgments

We are grateful to all the contributors to Universal Dependencies; without their effort a task like this simply wouldn't be possible.

The work described herein, including data preparation for the *CoNLL 2017 UD Shared Task*, has been supported by the following grants and projects: "CRACKER," H2020 Project No. 645357 of the European Commission; "MANYLA," Grant No. GA15-10472S of the Grant Agency of the Czech Republic; FIN-CLARIN.

The data for the *CoNLL 2017 UD Shared Task* are available via the LINDAT/CLARIN repository, which is part of a research infrastructure project funded by the Ministry of Education, Youth and Sports of the Czech Republic, Project. No. LM2015071.

The parallel evaluation set was made possible by contributions from DFKI (sentence selection and translation), Google (initial treebanking) and UD volunteers (translation to additional languages, annotation and conversion to UD v2).

References

Burak Kerim Akkuş, Heval Azizoğlu, and Ruket Çakıcı. 2017. Initial explorations of CCG supertagging for Universal Dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Chris Alberti, Daniel Andor, Ivan Bogatyy, Michael Collins, Dan Gillick, Lingpeng Kong, Terry Koo, Ji Ma, Mark Omernick, Slav Petrov, Chayut Thanapirom, Zora Tung, and David Weiss. 2017.

Abbreviation	Explanation
Segment	**Word and sentence segmentation**
B(ase)	Baseline UDPipe
UDP	Adapted/retrained UDPipe
O(wn)	Own system for word and sentence segmentation
Morph	**Morphological analysis (including part-of-speech tags)**
None	No prediction of tags or morphological features
B(ase)	Baseline UDPipe
UDP	Adapted/retrained UDPipe
Own	Own system for predicting tags and/or morphological features
AG	Apertium/Giellatekno morphological analyzers
Aux	Tags predicted as an auxiliary task during parser training
Joint	Tags predicted jointly with parsing
Parsing	**Parsing technique**
B(ase)	Baseline UDPipe
Single	Single parser
Ensemble	Ensemble of several parsers
Suffixes	G = Graph-based
	T = Transition-based
	R = Rule-based (with statistics or unsupervised learning)
Embed	**Word embeddings**
B(ase)	Pre-trained word embeddings provided by organizers
C(rawl)	Word embeddings trained on the crawled data provided by organizers
Treebank	Word embeddings trained (only) on the UD treebanks
O	Word embeddings trained on data from OPUS
FB	Pre-trained embeddings released by Facebook
Random	Randomly initialized (trained only with parser)
AddData	**Additional data used (over and above treebanks + raw text)**
None	No additional data
OPUS	Parallel data from OPUS
Wiki	Wikipedia dumps
PTB	Penn Treebank (sic)
CCG	CCGBank (sic)
Surp	**Approach to surprise languages**
Mono	Monolingual parser trained on dev data
Canon	Single canonical model for all surprise languages
Cross	Cross-lingual parsing
Suffixes	S = Single-source
	M = Multi-source
	D = Delexicalized
Para	**Approach to parallel test sets**
Single	Model trained on a single treebank from each language
All	Model trained on all treebanks from each language

Table 12: Abbreviations used in Table 11.

Syntaxnet models for the conll 2017 shared task. *CoRR* abs/1703.04929. http://arxiv.org/abs/1703.04929.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1231.pdf.

Lauriane Aufrant and Guillaume Wisniewski. 2017. LIMSI@CoNLL'17: UD shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Anders Björkelund, Agnieszka Faleńska, Xiang Yu, and Jonas Kuhn. 2017. IMS at the CoNLL 2017 UD shared task: CRFs and perceptrons meet neural networks. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*. Association for Computational Linguistics, pages 149–164. http://anthology.aclweb.org/W/W06/W06-29.pdf#page=165.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, and Ting Liu. 2017. The HIT-SCIR system for end-to-end parsing of Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Ayan Das, Zaffar Affan, and Sudeshna Sarkar. 2017. Delexicalized transfer parsing for low-resource languages using transformed and combined treebanks. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Eric De La Clergerie, Benoît Sagot, and Djamé Seddah. 2017. The ParisNLP entry at the CoNLL UD shared task 2017: A tale of a #parsingtragedy. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to Universal Dependencies - look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report, Stanford University. http://nlp.stanford.edu/software/dependencies_manual.pdf.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's 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*. Association for Computational Linguistics.

Stefan Daniel Dumitrescu, Tiberiu Boroş, and Dan Tufiş. 2017. RACAI's natural language processing pipeline for Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Marcos Garcia and Pablo Gamallo. 2017. A rule-based system for cross-lingual parsing of Romance languages with universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

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, Charles University. http://hdl.handle.net/11234/1-1989.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Allan Hanbury, Henning Müller, Krisztian Balog, Torben Brodt, Gordon V. Cormack, Ivan Eggel, Tim Gollub, Frank Hopfgartner, Jayashree Kalpathy-Cramer, Noriko Kando, Anastasia Krithara, Jimmy Lin, Simon Mercer, and Martin Potthast. 2015. Evaluation-as-a-Service: Overview and Outlook. *ArXiv e-prints* http://arxiv.org/abs/1512.07454.

Johannes Heinecke and Munshi Asadullah. 2017. Multi-model and crosslingual dependency analysis. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Ryan Hornby, Clark Taylor, and Jungyeul Park. 2017. Corpus selection approaches for multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Tao Ji, Yuanbin Wu, and Man Lan. 2017. A fast and lightweight system for multilingual dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Hiroshi Kanayama, Masayasu Muraoka, and Katsumasa Yoshikawa. 2017. A semi-universal pipelined approach to the CoNLL 2017 UD shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Jenna Kanerva, Juhani Luotolahti, and Filip Ginter. 2017. TurkuNLP: Delexicalized pre-training of word embeddings for dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Ömer Kırnap, Berkay Furkan Önder, and Deniz Yuret. 2017. Parsing with context embeddings. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Kyungtae Lim and Thierry Poibeau. 2017. A system for multilingual dependency parsing based on bidirectional LSTM feature representations. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*. pages 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

Christophe Moor, Paola Merlo, James Henderson, and Haozhou Wang. 2017. Geneva DINN parser: a neural network dependency parser ten years later. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Amir More and Reut Tsarfaty. 2017. Universal joint morph-syntactic processing: The Open University of Israel's submission to the CoNLL 2017. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Dat Quoc Nguyen, Mark Dras, and Mark Johnson. 2017. A novel neural network model for joint POS tagging and graph-based dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Aljoscha Burchardt, Marie Candito, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Silvie Cinková, Çağrı Çöltekin, Miriam Connor, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Marhaba Eli, Ali Elkahky, Tomaž Erjavec, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonça, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Pinkey Nainwani, Anna Nedoluzhko, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Livy Real, Siva Reddy, Georg Rehm, Larissa Rinaldi, Laura Rituma, Rudolf Rosa, Davide Rovati, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Antonio Stella, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zhuoran Yu, Zdeněk Žabokrtský, Daniel Zeman, and Hanzhi Zhu. 2017a. Universal dependencies 2.0 – CoNLL 2017 shared task development and test data. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-2184.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phương Lê Hồng,

Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017b. Universal Dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague. http://hdl.handle.net/11234/1-1983.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher 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 (LREC 2016)*. European Language Resources Association, Portorož, Slovenia, pages 1659–1666.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*. pages 86–95.

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 CoNLL Shared Task Session of EMNLP-CoNLL 2007*. Association for Computational Linguistics, pages 915–932. http://www.aclweb.org/anthology/D/D07/D07-1.pdf#page=949.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2011. A universal part-of-speech tagset. *CoRR* abs/1104.2086. http://arxiv.org/abs/1104.2086.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Proceedings of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*. Montréal, Canada. http://www.petrovi.de/data/sancl12.pdf.

Martin Popel, Zdeněk Žabokrtský, and Martin Vojtek. 2017. Udapi: Universal API for universal dependencies. In *NoDaLiDa 2017 Workshop on Universal Dependencies*. Göteborgs universitet, Göteborg, Sweden, pages 96–101. http://aclweb.org/anthology/W/W17/W17-0412.pdf.

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 (CLEF 14)*. Springer, Berlin Heidelberg New York, pages 268–299. https://doi.org/10.1007/978-3-319-11382-1_22.

Xian Qian and Yang Liu. 2017. A non-DNN feature engineering approach to dependency parsing – FBAML at CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Djamé Seddah, Sandra Kübler, and Reut Tsarfaty. 2014. Introducing the SPMRL 2014 shared task on parsing morphologically-rich languages. In *First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. Dublin, Ireland, pages 103–109. http://www.aclweb.org/anthology/W14-6111.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2013. Overview of the SPMRL 2013 shared task: Cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*. Association for Computational Linguistics, Seattle, Washington, USA, pages 146–182. http://www.aclweb.org/anthology/W13-4917.

Tianze Shi, Felix G. Wu, Xilun Chen, and Yao Cheng. 2017. Combining global models for parsing Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Milan Straka. 2017. CoNLL 2017 shared task - UDPipe baseline models and supplementary materials. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. http://hdl.handle.net/11234/1-1990.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association, Portorož, Slovenia.

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*. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA), Istanbul, Turkey.

Clara Vania, Xingxing Zhang, and Adam Lopez. 2017. UParse: the Edinburgh system for the CoNLL 2017 UD shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

David Vilares and Carlos Gómez-Rodríguez. 2017. A non-projective greedy dependency parser with bidirectional LSTMs. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Hao Wang, Hai Zhao, and Zhisong Zhang. 2017. A transition-based system for universal dependency parsing. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Kuan Yu, Pavel Sofroniev, and Erik Schill. 2017. The parse is darc and full of errors: Universal dependency parsing with transition-based and graph-based algorithms. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.