Extracting Headless MWEs from Dependency Parse Trees: Parsing, Tagging, and Joint Modeling Approaches

Tianze Shi
Cornell University
tianze@cs.cornell.edu

Lillian Lee
Cornell University
llee@cs.cornell.edu

Abstract

An interesting and frequent type of multi-word expression (MWE) is the headless MWE, for which there are no true internal syntactic dominance relations; examples include many named entities ("Wells Fargo") and dates ("July 5, 2020") as well as certain productive constructions ("blow for blow", "day after day"). Despite their special status and prevalence, current dependency-annotation schemes require treating such flat structures as if they had internal syntactic heads, and most current parsers handle them in the same fashion as headed constructions. Meanwhile, outside the context of parsing, taggers are typically used for identifying MWEs, but taggers might benefit from structural information. We empirically compare these two common strategies—parsing and tagging—for predicting flat MWEs. Additionally, we propose an efficient joint decoding algorithm that combines scores from both strategies. Experimental results on the MWE-Aware English Dependency Corpus and on six non-English dependency treebanks with frequent flat structures show that: (1) tagging is more accurate than parsing for identifying flat-structure MWEs, (2) our joint decoder reconciles the two different views and, for non-BERT features, leads to higher accuracies, and (3) most of the gains result from feature sharing between the parsers and taggers.

1 Introduction

Headless multi-word expressions (MWEs), including many named entities and certain productive constructions, are frequent in natural language and are important to NLP applications. In the context of dependency-based syntactic parsing, however, they pose an interesting representational challenge. Dependency-graph formalisms for syntactic structure represent lexical items as nodes and head-dominates-modifier/argument relations between lexical items as directed arcs on the corresponding pair of nodes. Most words can be assigned clear linguistically-motivated syntactic heads, but several frequently occurring phenomena do not easily fit into this framework, including punctuation, coordinating conjunctions, and "flat", or headless MWEs. While the proper treatment of headless constructions in dependency formalisms remains debated (Kahane et al., 2017; Gerdes et al., 2018), many well-known dependency treebanks handle MWEs by giving their component words a "default head", which is not indicative of a true dominance relation, but rather as "a tree encoding of a flat structure without a syntactic head" (de Marneffe and Nivre, 2019, pg. 213). Fig. 1 shows an example: the headless MWE Mellon Capital has its first word, Mellon, marked as the "head" of Capital.

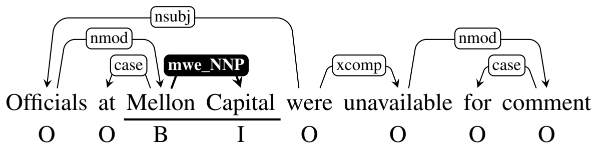

Figure 1: Dependency tree from the MWE-Aware English Dependency Corpus, imposing a "head" relationship between the words in the actually headless MWE Mellon Capital. Also shown are MWE BIO labels.

Despite the special status of flat structures in dependency tree annotations, most state-of-the-art dependency parsers treat all annotated relations equally, and thus do not distinguish between headed and headless constructions. When headless-span identification (e.g., as part of named-entity recognition (NER)) is the specific task at hand, begin-chunk/inside-chunk/outside-chunk (BIO) tagging (Ramshaw and Marcus, 1995) is generally adopted. It is therefore natural to ask whether *parsers* are as accurate as *taggers* in identifying these "flat branches" in dependency trees. Additionally, since parsing and tagging represent

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8780–8794
July 5 - 10, 2020.

two different views of the same underlying structures, can *joint decoding* that combines scores from the two modules and/or *joint training* under a multi-task learning (MTL) framework derive more accurate models than parsing or tagging alone?

To facilitate answering these questions, we introduce a joint decoder that finds the maximum sum of scores from both BIO tagging and parsing decisions. The joint decoder incorporates a special deduction item representing continuous headless spans, while retaining the cubic-time efficiency of projective dependency parsing. The outputs are consistent structures across the tagging view and the parsing view.

We perform evaluation of the different strategies on the MWE-Aware English Dependency Corpus and treebanks for five additional languages from the Universal Dependencies 2.2 corpus that have frequent multi-word headless constructions. On average, we find taggers to be more accurate than parsers at this task, providing 0.59% (1.42%) absolute higher F1 scores with(out) pre-trained contextualized word representations. Our joint decoder combining jointly-trained taggers and parsers further improves the tagging strategy by 0.69% (1.64%) absolute. This corroborates early evidence (Finkel and Manning, 2009) that joint modeling with parsing improves over NER. We also show that neural representation sharing through MTL is an effective strategy, as it accounts for a large portion of our observed improvements. Our code is publicly available at `https://github.com/tzshi/flat-mwe-parsing`.

2 Background on Headless Structures

A (multi-word) headless construction, or flat structure, is a span of lexical items that together reference a single concept and where no component is a syntactically more plausible candidate for the span's head than any other component. Examples are boldfaced in the following English sentences.

(1) *Within the scope of this paper:*
  a. ACL starts on **July 5, 2020**.
  b. My bank is **Wells Fargo**.
  c. The candidates matched each other **insult for insult**. (Jackendoff, 2008)

(1)a and (1)b show that dates and many named entities can be headless constructions, suggesting that they are frequent. Indeed, in the MWE-Aware English Dependency Corpus (Kato et al., 2017), nearly half of the sentences contain headless constructions, 75% of which are named entities. For comparison, (2) shows examples of non-flat MWEs, which are also interesting and important, but they are beyond the focus of our paper.

(2) *Outside the scope of this paper:*
  a. **congressman at large** (Sag et al., 2002) [head = "congressman"]
  b. I have **moved on**. [verb-particle construction, head = "moved"]
  c. I **take** your argument **into account**. (Constant et al., 2017) [light-verb construction, head = "take"]

Returning to headless MWEs, the choice of representation for headless spans depends on the task. In *named-entity recognition*, such spans are often treated as BIO tag sequences:[1] for example, in Fig. 1, "Mellon" is tagged as "B" and "Capital" is tagged as "I". In *dependency parsing*, where labeled dependency arcs are the only way to express a syntactic analysis (short of treating MWEs as atomic lexical items, which would result in a chicken-and-egg problem) is to impose arcs within the MWE's span. Different corpora adopt different annotation conventions. The MWE-Aware English Dependency Corpus uses the arc label mwe_NNP, as shown in Fig. 1. The Universal Dependencies (UD; Nivre et al., 2018) annotation guidelines have all following tokens in such constructions attached to the first one via arcs labeled flat, a choice that is admittedly "in principle arbitrary".[2]

The frequency of flat structures across different treebanks varies according to language, genre, and even tokenization guidelines, among other factors. Table 1 lists the UD 2.2 treebanks with the highest and lowest percentage of flat relations. While the Korean treebank ko_gsd (with the highest percentage) splits up most names into multiple tokens and connects them through flat, the Japanese treebank ja_gsd (no flats at all) treats all names as compound nouns, and thus represents them as having internal structure without any indication that a special case has occurred.[3] Fig. 2 shows examples from the UD parallel treebanks, illustrating

[1] In this paper, we adopt the original BIO tagset, which cannot properly represent discontinuous MWEs. See Schneider et al. (2014) for modified tagsets providing such support.

[2] `universaldependencies.org/u/dep/flat.html`

[3] Some flat structures can end up using other dependency labels such as compound, as a result of the fact that many UD treebanks, including ja_gsd, are automatically converted from non-UD style annotations. The UD annotations depend

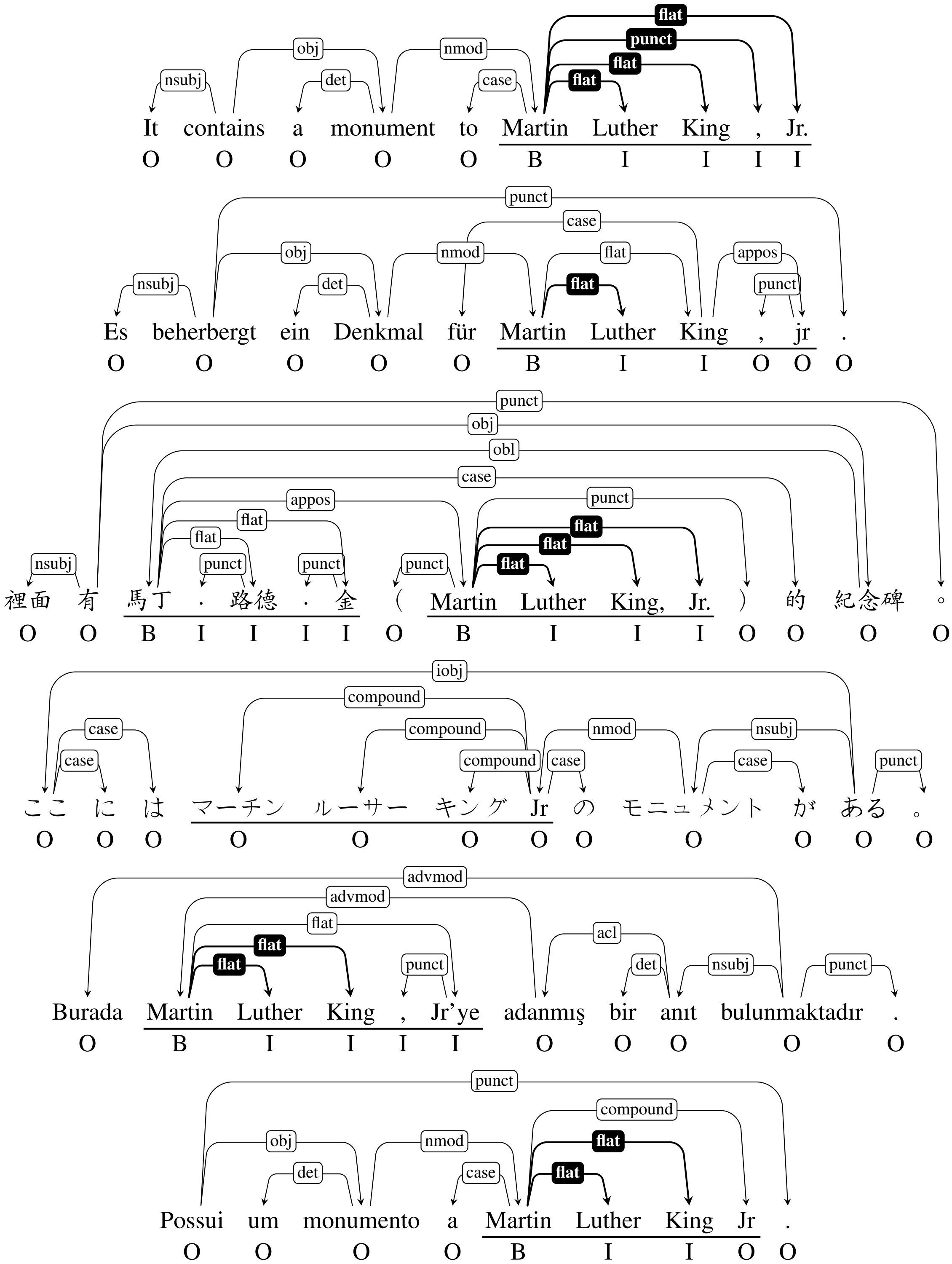

Figure 2: An illustration of flat-structure annotation variation across treebanks: a set of parallel sentences, all containing the conceptually headless MWE "Martin Luther King, Jr." (underlined), from UD 2.2 (treebank code _pud) in English, German, Chinese, Japanese, Turkish, and Portuguese (top to bottom). The intent of this figure is not to critique particular annotation decisions, but to demonstrate the notation, concepts, and data extraction methods used in our paper. To wit: Highlights/black-background indicate well-formed flat-MWE tree fragments according to the principles listed in §4. BIO sequences are induced by the longest-spanning flat arcs. When there is a mismatch between the highlighted tree fragments and the BI spans—here, in the German, Chinese and Turkish examples—it is because the dependency trees do not fully conform to the UD annotation guidelines on headless structures.

Treebank (Language)	% of flat graphs ↓	arcs
19 treebanks with highest percentages:		
`ko_gsd` (Korean)	67.84	15.35
`id_gsd` (Indonesian)	61.63	9.39
`ca_ancora` (Catalan)	41.11	3.32
`nl_lassysmall` (Dutch)	38.90	5.87
`ar_nyuad` (Arabic)	37.63	2.19
`es_ancora` (Spanish), `sr_set` (Serbian), `it_postwita` (Italian), `pt_bosque` (Portuguese), `pt_gsd` (Portuguese), `fa_seraji` (Persian), `de_gsd` (German), `hu_szeged` (Hungarian), `fr_gsd` (French), `es_gsd` (Spanish), `he_htb` (Hebrew), `kk_ktb` (Kazakh), `be_hse` (Belarusian), `nl_alpino` (Dutch)	> 20.00	
...	...	
12 treebanks without flat arcs:		
`cs_cltt` (Czech), `grc_perseus` (Ancient Greek), `hi_hdtb` (Hindi), `ja_gsd` (Japanese), `ja_bccwj` (Japanese), `la_ittb` (Latin), `la_perseus` (Latin), `no_nynorsklia` (Norwegian), `swl_sslc` (Swedish Sign Language), `ta_ttb` (Tamil), `ur_udtb` (Urdu), `vi_vtb` (Vietnamese)	0.00	0.00

Table 1: The UD 2.2 training treebanks with highest and lowest percentage of flat arcs, out of 90 treebanks.

the diversity of annotation for the same sentence rendered in different languages.

Overall, more than 20% of the treebanks in the UD 2.2 collection have flat structures in more than 20% of their training-set sentences.[4] Therefore, a parsing approach taking into account the special status of headless structural representations can potentially benefit models for a large number of languages and treebanks.

2.1 Notation and Definitions

Formally, given an n-word sentence $w = w_1, w_2, \ldots, w_n$, we define its dependency structure to be a graph $G = (V, E)$. Each node in V corresponds to a word in the sentence. Each (labeled) edge $(h, m, r) \in E$ denotes a syntactic relation labeled r between the head word w_h and modifier word w_m, where $h, m \in \{0, 1, \ldots, n\}$ and 0 denotes the dummy root of the sentence. Since we work with dependency treebanks, we require that the edges in E form a tree. To represent a multi-word headless span $w_i, \ldots, w_j$, all subsequent words in the span are attached to the beginning word w_i, i.e., $\forall k \in \{i+1, \ldots, j\}, (i, k, f) \in E$, where f is the special syntactic relation label denoting headless structures (flat in UD annotation). Alternatively, one can also use a BIO tag sequence $T = (t_1, t_2, \ldots, t_n) \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}^n$ to indicate the location of any headless spans within w. The headless MWE span $w_i, \ldots, w_j$ has the corresponding tags $t_i = \mathrm{B}$ and $\forall k \in \{i+1, \ldots, j\}, t_k = \mathrm{I}$; tokens outside any spans are assigned the tag O. We call G and T *consistent* if they indicate the same set of headless spans for w.

3 Three Approaches

We first present the standard approaches of edge-factored parsing (§3.2) and tagging (§3.3) for extracting headless spans in dependency trees, and then introduce a joint decoder (§3.4) that finds the global maximum among consistent (tree structure, tag sequence) pairs.

3.1 Preliminaries

Given a length-n sentence w—which we henceforth denote with the variable x for consistency with machine-learning conventions—we first extract contextualized representations from the input to associate each word with a vector $\mathbf{x}_0$ (for the dummy word "root"), $\mathbf{x}_1, \ldots, \mathbf{x}_n$. We consider two common choices of feature extractors: (1) bi-directional long short-term memory networks (bi-LSTMs; Graves and Schmidhuber, 2005) which

on how detailed the original syntactic analyses are and the accuracies of the conversion algorithms.

[4]Measured on the 90 treebanks with training splits.

have been widely adopted in dependency parsing (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017) and sequence tagging (Ma and Hovy, 2016); and (2) the Transformer-based (Vaswani et al., 2017) BERT feature extractor (Devlin et al., 2019), pre-trained on large corpora and known to provide superior accuracies on both tasks (Kitaev et al., 2019; Kondratyuk and Straka, 2019). For BERT models, we fine-tune the representations from the final layer for our parsing and tagging tasks. When the BERT tokenizer renders multiple tokens from a single pre-tokenized word, we follow Kitaev et al. (2019) and use the BERT features from the last token as its representation.

3.2 (Edge-Factored) Parsing

Since we consider headless structures that are embedded inside parse trees, it is natural to identify them through a rule-based post-processing step after full parsing. Our parsing component replicates that of the state-of-the-art Che et al. (2018) parser, which has the same parsing model as Dozat and Manning (2017). We treat unlabelled parsing as a head selection problem (Zhang et al., 2017) with deep biaffine attention scoring:

$$\begin{aligned} \mathbf{h}_i^{\text{attach}} &= \text{MLP}^{\text{attach-head}}(\mathbf{x}_i) \\ \mathbf{m}_j^{\text{attach}} &= \text{MLP}^{\text{attach-mod}}(\mathbf{x}_j) \\ \mathbf{s}_{i,j} &= [\mathbf{h}_i^{\text{attach}}; 1]^\top U^{\text{attach}} [\mathbf{m}_j^{\text{attach}}; 1] \\ P(h_j = i \mid x) &= \text{softmax}_i(\mathbf{s}_{:,j}), \end{aligned}$$

where $\text{MLP}^{\text{attach-head}}$ and $\text{MLP}^{\text{attach-mod}}$ are multilayer perceptrons (MLPs) that project contextualized representations into a d-dimensional space; $[\cdot; 1]$ indicates appending an extra entry of 1 to the vector; $U^{\text{att}} \in \mathbb{R}^{(d+1)\times(d+1)}$ generates a score $s_{i,j}$ for w_j attaching to w_i (which we can then refer to as the head of w_j, h_j); a softmax function defines a probability distribution over all syntactic head candidates in the argument vector (we use the range operator ":" to evoke a vector); and, recall, we represent potential heads as integers, so that we may write $h_j = i \in \{0, \ldots, n\}$.

The model for arc labeling employs an analogous deep biaffine scoring function:

$$\begin{aligned} \mathbf{h}_i^{\text{rel}} &= \text{MLP}^{\text{rel-head}}(\mathbf{x}_i) \\ \mathbf{m}_j^{\text{rel}} &= \text{MLP}^{\text{rel-mod}}(\mathbf{x}_j) \\ \mathbf{v}_{i,j,r} &= [\mathbf{h}_i^{\text{rel}}; 1]^\top U_r^{\text{rel}} [\mathbf{m}_j^{\text{rel}}; 1] \\ P(r_j = r \mid x, h_j = i) &= \text{softmax}_r(\mathbf{v}_{i,j,:}), \end{aligned}$$

where r_j is the arc label between w_{h_j} and w_j.

The objective for training the parser is to minimize the cumulative negative log-likelihood

$$\begin{aligned} L^{\text{parse}} = \sum_{(i^*, j^*, r^*) \in E} & [-\log P(h_{j^*} = i^* \mid x) \\ & - \log P(r_i = r^* \mid x, h_{j^*} = i^*)]. \end{aligned}$$

After the model predicts a full parse, we extract headless structures as the tokens "covered" by the longest-spanning f-arcs ($f =$ flat in UD).

3.3 Tagging

For extracting spans in texts, if one chooses to ignore the existence of parse trees, BIO tagging is a natural choice. We treat the decision for the label of each token as an individual multi-class classification problem. We let

$$P(t_i = t \mid x) = \text{softmax}_t(\text{MLP}^{\text{tag}}(\mathbf{x}_i)),$$

where MLP^{tag} has 3 output units corresponding to the scores for tags B, I and O respectively.[5]

We train the tagger to minimize

$$L^{\text{tag}} = \sum_i - \log P(t_i = t_i^* \mid x),$$

where t^* corresponds to the gold BIO sequence. During inference, we predict the BIO tags independently at each token position and interpret the tag sequence as a set of MWE spans. As a post-processing step, we discard all single-token spans, since the task is to predict multi-word spans.

3.4 A Joint Decoder

A parser and a tagger take two different views of the same underlying data. It is thus reasonable to hypothesize that a joint decoding process that combines the scores from the two models might yield more accurate predictions. In this section, we propose such a joint decoder to find the parser+tagger-consistent structure with the highest product of probabilities. Formally, if $\mathcal{Y}$ is the output space for all consistent parse tree structures and BIO tag sequences, for $y \in \mathcal{Y}$ with components consisting

[5] Sequence tagging is traditionally handled by conditional random fields (Lafferty et al., 2001, CRFs). However, in recent experiments using contextualized representations on tagging (Clark et al., 2018; Devlin et al., 2019), CRF-style loss functions provide little, if any, performance gains compared with simple multi-class classification solutions, at slower training speeds, to boot. Our preliminary experiments with both biLSTM and BERT-based encoders corroborate these findings, and thus we report results trained without CRFs.

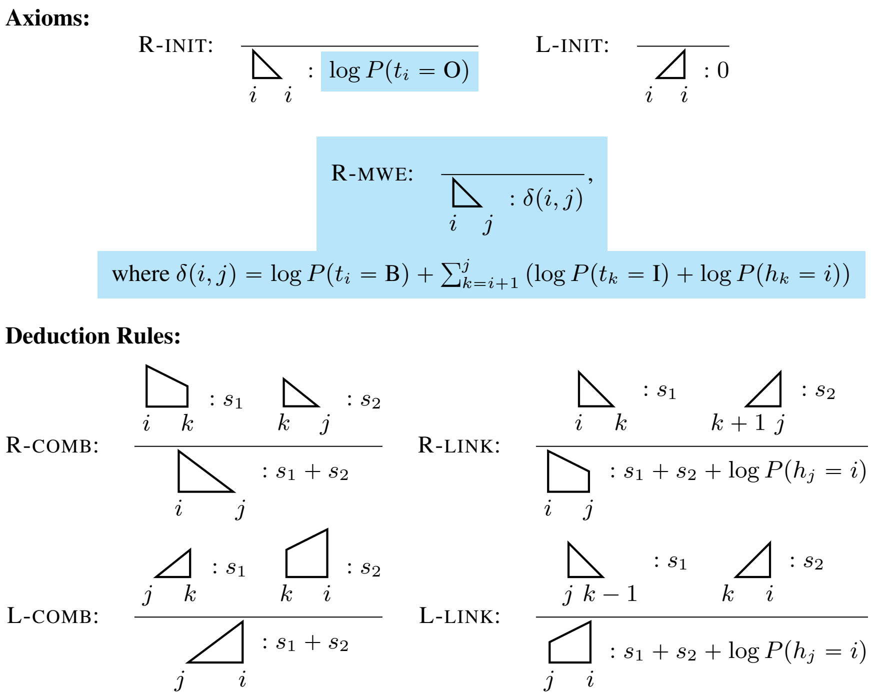

Figure 3: Eisner's (1996) algorithm adapted to parsing headless structures (unlabeled case), our modifications highlighted in blue. All deduction items are annotated with their scores. R-MWE combines BIO tagging scores and head selection parsing scores. We need no L-MWE because of the rightward headless-structure-arc convention.

of tags t_i, head assignments h_i, and relation labels r_i, our decoder aims to find $\hat{y}$ satisfying

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P(y \mid x),$$

where

$$P(y \mid x) = \prod_i P(t_i \mid x) P(h_i \mid x) P(r_i \mid x, h_i).$$

Fig. 3 illustrates our joint decoder in the unlabeled case.[6] It builds on Eisner's (1996) decoder for projective dependency parsing. In addition to having single-word spans as axioms in the deduction system, we further allow multi-word spans to enter the decoding procedures through the axiom R-MWE. Any initial single-word spans receive an O-tag score for that word, while the newly introduced MWE spans receive B-tag, I-tag, attachment and relation scores that correspond to the two consistent views of the same structure. The time complexity for this decoding algorithm remains the same $O(n^3)$ as the original Eisner algorithm.

During training, we let the parser and the tagger share the same contextualized representation $\mathbf{x}$ and optimize a linearly interpolated joint objective

$$L^{\text{joint}} = \lambda L^{\text{parse}} + (1 - \lambda) L^{\text{tag}},$$

where λ is a hyper-parameter adjusting the relative weight of each module.[7] This is an instance of multi-task learning (MTL; Caruana, 1993, 1997). MTL has proven to be a successful technique (Collobert and Weston, 2008) on its own; thus, in our experiments, we compare the joint decoder and using the MTL strategy alone.

4 Experiments

Data We perform experiments on the MWE-Aware English Dependency Corpus (Kato et al., 2017) and treebanks selected from Universal Dependencies 2.2 (UD; Nivre et al., 2018) for having frequent occurrences of headless MWE structures. The MWE-Aware English Dependency Corpus provides automatically unified named-entity annotations based on OntoNotes 5.0 (Weischedel et al., 2013) and Stanford-style dependency trees (de Marneffe and Manning, 2008). We extract MWE spans according to mwe_NNP dependency relations. We choose the UD treebanks based on two basic properties that hold for flat structures

[6]In the labeled case, the parser further adds the arc-labeling scores to the R-MWE and LINK rules.

[7]The joint decoder combines tagging and parsing scores regardless of whether the two modules are jointly trained. However, since feature extraction is the most time-consuming step in our neural models, especially with BERT-based feature extractors, it is most practical to save memory and time by sharing common feature representations across modules.

Treebank		# tokens	# headless arcs	%	# headless spans	Average span length	Compliance ratio
English		731,677	32,065	4.38%	16,997	2.89	100.00%
UD 2.2	de_gsd	263,804	6,786	2.57%	5,663	2.59	93.00%
	it_postwita	99,441	2,733	2.75%	2,277	2.26	94.89%
	nl_alpino	186,046	4,734	2.54%	3,269	2.45	100.00%
	nl_lassysmall	75,134	4,408	5.87%	3,018	2.46	99.82%
	no_nynorsk	245,330	5,578	2.27%	3,670	2.54	99.78%
	pt_bosque	206,739	5,375	2.60%	4,310	2.25	97.38%

Table 2: Dataset statistics. Language codes: de=German; it=Italian; nl=Dutch; no=Norwegian; pt=Portuguese.

conforming to the UD annotation guidelines: (1) all words that are attached via flat relations must be leaf nodes and (2) all words within a flat span should be attached to a common "head" word, and each arc label should be either flat or punct.[8] For each treebank, we compute its *compliance ratio*, defined as the percentage of its trees containing flat arc labels that satisfy both properties above; and we filter out those with compliance ratios below 90%.[9] We rank the remaining treebanks by their ratios of flat relations among all dependency arcs, and pick those with ratios higher than 2%. Six treebanks representing 5 languages, German (McDonald et al., 2013), Italian (Sanguinetti et al., 2018), Dutch (Bouma and van Noord, 2017), Norwegian (Solberg et al., 2014) and Portuguese (Rademaker et al., 2017), are selected for our experiments.[10] Data statistics are given in Table 2. To construct gold-standard BIO labels, we extract MWE spans according to the longest-spanning arcs that correspond to headless structures.

Implementation Details We use 3-layer bi-LSTMs where each layer has 400 dimensions in both directions and the inputs are concatenations of 100-dimensional randomly-initialized word embeddings with the final hidden vectors of 256-dimensional single-layer character-based bi-LSTMs; for BERT, we use pre-trained cased multi-lingual BERT models[11] and fine-tune the weights. We adopt the parameter settings of Dozat and Manning (2017) and use 500 and 100 dimensions for U^{att} and U_r^{rel}, respectively. The MLP in the taggers have 500 hidden dimensions. We use a dropout (Srivastava et al., 2014) rate of 0.33, a single hidden layer, and a ReLU activation function (Nair and Hinton, 2010) for all MLPs. The models are trained with the Adam optimizer (Kingma and Ba, 2015) using a batch size of 16 sentences. The learning rates are set to $1e^{-3}$ for bi-LSTMs and $1e^{-5}$ for BERT initially and then multiplied by a factor of 0.1 if the performance on the development set stops improving within 3200 training iterations. For the parsing models, we use the projective Eisner (1996) decoder algorithm. For the joint training and joint decoding models, we tune $\lambda \in \{0.02, 0.05, 0.1, 0.3, 0.5, 0.9\}$ for each treebank independently and fix the settings based on the best dev-set scores. We run each model with 5 different random seeds and report the mean and standard deviation for each setting.

Results We report F1 scores based on multi-word headless-structure extraction. Table 3 compares different strategies for identifying headless MWEs in parse trees. Tagging is consistently better than parsing except for two treebanks with BERT feature extractor. Tagging beats parsing in all but two combinations of treebank and feature extractor. As hypothesized, our joint decoder improves over both strategies by 0.69% (1.64%) absolute through combined decisions from parsing and tagging with(out)

[8] punct inside a headless span is often used for hyphens and other internal punctuation in named entities. See the English sentence in Fig. 2 for an example.

[9] The two properties defined in the UD guidelines for headless structures provide us with a common basis for uniform treatment across languages and treebanks. Unfortunately, the two properties can be violated quite often, due to issues in annotation and automatic treebank conversion into UD style. In 6 out of the top 10 treebanks containing the most flat relations, (at least one of) these properties are violated in more than 35% of the sentences with flat relations and have to be excluded from our experiments. We hope that ongoing community effort in data curation will facilitate evaluation on more diverse languages.

[10] It is a coincidence that all the selected languages are Indo-European (IE). Although there are some non-IE treebanks with high flat ratio, such as Korean (see Table 1), the annotated structures frequently break one or both of the basic properties. See Fig. 2 for violation examples.

[11] https://github.com/huggingface/transformers

w/ bi-LSTM Treebank		Compl. Ratio ↓	Parsing	Tagging	MTL Parsing	MTL Tagging	Joint Decoding
	English	100.00	91.24$_{\pm 0.60}$	91.81$_{\pm 0.45}$	93.00$_{\pm 0.83}$	**93.24**$_{\pm 0.76}$	**93.49**$_{\pm 0.43}$
UD 2.2	nl_alpino	100.00	72.66$_{\pm 1.73}$	74.94$_{\pm 1.00}$	77.29$_{\pm 0.80}$	75.58$_{\pm 1.18}$	**79.65**$_{\pm 1.05}$
UD 2.2	nl_lassysmall	99.82	76.44$_{\pm 1.56}$	**77.98**$_{\pm 1.56}$	**78.13**$_{\pm 0.98}$	77.58$_{\pm 1.17}$	**78.92**$_{\pm 1.00}$
UD 2.2	no_nynorsk	99.78	85.34$_{\pm 0.81}$	87.67$_{\pm 0.90}$	86.72$_{\pm 0.76}$	87.44$_{\pm 0.76}$	**88.40**$_{\pm 0.39}$
UD 2.2	pt_bosque	97.38	89.55$_{\pm 1.10}$	90.97$_{\pm 0.46}$	**91.30**$_{\pm 0.75}$	**92.07**$_{\pm 1.04}$	90.63$_{\pm 1.56}$
UD 2.2	it_postwita	94.89	75.35$_{\pm 1.05}$	76.37$_{\pm 1.72}$	**78.46**$_{\pm 1.08}$	**77.87**$_{\pm 0.57}$	**78.38**$_{\pm 1.04}$
UD 2.2	de_gsd	93.00	63.32$_{\pm 1.36}$	64.10$_{\pm 1.31}$	**64.81**$_{\pm 2.05}$	**65.07**$_{\pm 1.35}$	**65.86**$_{\pm 1.34}$
Average			79.13	80.55	81.39	81.26	82.19

w/ BERT Treebank		Compl. Ratio ↓	Parsing	Tagging	MTL Parsing	MTL Tagging	Joint Decoding
	English	100.00	94.98$_{\pm 0.26}$	95.45$_{\pm 0.23}$	95.01$_{\pm 0.20}$	**95.86**$_{\pm 0.19}$	95.51$_{\pm 0.58}$
UD 2.2	nl_alpino	100.00	83.87$_{\pm 1.61}$	83.32$_{\pm 1.01}$	84.65$_{\pm 1.48}$	**85.90**$_{\pm 1.51}$	**86.61**$_{\pm 1.52}$
UD 2.2	nl_lassysmall	99.82	87.16$_{\pm 1.20}$	87.52$_{\pm 0.59}$	**88.10**$_{\pm 0.80}$	87.68$_{\pm 0.78}$	**88.35**$_{\pm 0.49}$
UD 2.2	no_nynorsk	99.78	92.16$_{\pm 0.93}$	**93.48**$_{\pm 0.48}$	92.45$_{\pm 0.34}$	**93.11**$_{\pm 0.21}$	**93.08**$_{\pm 0.62}$
UD 2.2	pt_bosque	97.38	92.98$_{\pm 0.82}$	93.47$_{\pm 0.55}$	93.42$_{\pm 0.65}$	**93.85**$_{\pm 0.57}$	**94.01**$_{\pm 0.19}$
UD 2.2	it_postwita	94.89	80.80$_{\pm 1.51}$	80.80$_{\pm 1.52}$	**80.90**$_{\pm 1.78}$	**81.33**$_{\pm 0.43}$	80.83$_{\pm 1.20}$
UD 2.2	de_gsd	93.00	68.21$_{\pm 1.43}$	**70.28**$_{\pm 0.70}$	**70.04**$_{\pm 1.14}$	**71.05**$_{\pm 1.12}$	**70.72**$_{\pm 0.90}$
Average			85.74	86.33	86.37	86.97	87.02

Table 3: Flat-structure identification test-set F1 scores (%) with bi-LSTM (top) and BERT (bottom). The cell with the best result for each treebank has blue shading; results within one standard deviation of the best are bolded.

BERT. We also compare the joint decoding setting with MTL training strategy alone. While joint decoding yields superior F1 scores, MTL is responsible for a large portion of the gains: it accounts for over half of the average gains with bi-LSTMs, and when we use pre-trained BERT feature extractors, the accuracies of jointly-trained taggers are essentially as good as joint decoding models.

Interestingly, the choice of feature extractors also has an effect on the performance gap between parsers and taggers. With bi-LSTMs, tagging is 1.42% absolute F1 higher than parsing, and the gap is mitigated through MTL. While pre-trained BERT reduces the performance difference dramatically down to 0.59% absolute, MTL no longer helps parsers overcome this gap. Additionally, we observe that MTL helps both parsing and tagging models, demonstrating that the two views of the same underlying structures are complementary to each other and that learning both can be beneficial to model training. By resolving such representational discrepancies, joint decoding exhibits further accuracy improvement.

In terms of dependency parsing accuracies, we confirm that our parsing-only models achieve state-of-the-art performance on the UD treebanks, but there are no significant differences in parsing results among parsing-only, MTL and jointly-decoded models. See Appendix for detailed results.

5 Related Work

Syntactic analysis in conjunction with MWE identification is an important line of research (Wehrli, 2000). The span-based representations that form the basis of phrase-structure trees (as opposed to dependency trees) are arguably directly compatible with headless spans. This motivates approaches using joint constituency-tree representations based on context-free grammars (Arun and Keller, 2005; Constant et al., 2013) and tree substitution grammars (Green et al., 2011, 2013). Finkel and Manning (2009) add new phrasal nodes to denote named entities, enabling statistical parsers trained on this modified representation to produce both parse trees and named entity spans simultaneously. Le Roux et al. (2014) use dual decomposition to develop a joint system that combines phrase-structure parsers and taggers for compound recognition. These approaches do not directly transfer to dependency-based representations since dependency trees do not explicitly represent phrases.

In the context of dependency parsing, Eryiğit et al. (2011) report that MWE annotations have a large impact on parsing. They find that the dependency parsers are more accurate when MWE spans are not unified into single lexical items. Similar to the phrase-structure case, Candito and Constant (2014) consider MWE identification as a side product of dependency parsing into joint representations. This parse-then-extract strategy is widely adopted (Vincze et al., 2013; Nasr et al., 2015; Simkó et al., 2017). Waszczuk et al. (2019) introduce additional parameterized scoring functions for the arc labelers and use global decoding to produce consistent structures during arc-labeling steps once unlabeled dependency parse trees are predicted. Our work additionally proposes a joint decoder that combines the scores from both parsers and taggers. Alternative approaches to graph-based joint parsing and MWE identification include transition-based (Constant and Nivre, 2016) and easy-first (Constant et al., 2016) dependency parsing. These approaches typically rely on greedy decoding, whereas our joint decoder finds the globally optimal solution through dynamic programming.

Our work only focuses on a subset of MWEs that do not have internal structures. There is substantial research interest in the broad area of MWEs (Sag et al., 2002; Constant et al., 2017) including recent releases of datasets (Schneider and Smith, 2015), editions of shared tasks (Savary et al., 2017; Ramisch et al., 2018) and workshops (Savary et al., 2018, 2019). We leave it to future work to extend the comparison and combination of taggers and dependency parsers to other MWE constructions.

6 Conclusion and Further Directions

Our paper provides an empirical comparison of different strategies for extracting headless MWEs from dependency parse trees: parsing, tagging, and joint modeling. Experiments on the MWE-Aware English Dependency Corpus and UD 2.2 across five languages show that tagging, a widely-used methodology for extracting spans from texts, is more accurate than parsing for this task. When using bi-LSTM (but not BERT) representations, our proposed joint decoder reaches higher F1 scores than either of the two other strategies, by combining scores of the two different and complementary representations of the same structures. We also show that most of the gains stem from a multi-task learning strategy that shares common neural representations between the parsers and the taggers.

An interesting additional use-case for our joint decoder is when a downstream task, e.g., relation extraction, requires output structures from both a parser and a tagger. Our joint decoder can find the highest-scoring consistent structures among all candidates, and thus has the potential to provide simpler model designs in downstream applications.

Our study has been limited to a few treebanks in UD partially due to large variations and inconsistencies across different treebanks. Future community efforts on a unified representation of flat structures for all languages would facilitate further research on linguistically-motivated treatments of headless structures in "headful" dependency treebanks.

Another limitation of our current work is that our joint decoder only produces projective dependency parse trees. To handle non-projectivity, one possible solution is pseudo-projective parsing (Nivre and Nilsson, 2005). We leave it to future work to design a non-projective decoder for joint parsing and headless structure extraction.

Acknowledgments

We thank the three anonymous reviewers for their comments, and Igor Malioutov, Ana Smith and the Cornell NLP group for discussion and comments. TS was supported by a Bloomberg Data Science Ph.D. Fellowship.

References

Abhishek Arun and Frank Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 306–313, Ann Arbor, Michigan. Association for Computational Linguistics.

Gosse Bouma and Gertjan van Noord. 2017. Increasing return on annotation investment: The automatic construction of a Universal Dependency treebank for Dutch. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*, pages 19–26, Gothenburg, Sweden. Association for Computational Linguistics.

Marie Candito and Matthieu Constant. 2014. Strategies for contiguous multiword expression analysis and dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Baltimore, Maryland. Association for Computational Linguistics.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on International Conference on Machine Learning*, ICML'93, pages 41–48, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. 2017. Multiword expression processing: A survey. *Computational Linguistics*, 43(4):837–892.

Matthieu Constant, Joseph Le Roux, and Anthony Sigogne. 2013. Combining compound recognition and PCFG-LA parsing with word lattices and conditional random fields. *ACM Transactions on Speech and Language Processing*, 10(3):8:1–8:24.

Matthieu Constant, Joseph Le Roux, and Nadi Tomeh. 2016. Deep lexical segmentation and syntactic parsing in the easy-first dependency framework. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1095–1101, San Diego, California. Association for Computational Linguistics.

Matthieu Constant and Joakim Nivre. 2016. A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 161–171, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Jason Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 340–345.

Gülşen Eryiğit, Tugay İlbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 45–55, Dublin, Ireland. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 326–334, Boulder, Colorado. Association for Computational Linguistics.

Kim Gerdes, Joakim Nivre, Agata Savary, and Nathan Schneider. 2018. Working group on multiword expressions. `https://universaldependencies.org/workgroups/mwe.html`. Webpage accessed May 5, 2020.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610.

Spence Green, Marie-Catherine de Marneffe, John Bauer, and Christopher D. Manning. 2011. Multiword expression identification with tree substitution grammars: A parsing *tour de force* with French. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 725–735, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Spence Green, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. Parsing models for identifying multiword expressions. *Computational Linguistics*, 39(1):195–227.

Ray Jackendoff. 2008. *Construction after construction* and its theoretical challenges. *Language*, 84(1):8–28.

Sylvain Kahane, Marine Courtin, and Kim Gerdes. 2017. Multi-word annotation in syntactic treebanks – Propositions for Universal Dependencies. In *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, pages 181–189, Prague, Czech Republic.

Akihiko Kato, Hiroyuki Shindo, and Yuji Matsumoto. 2017. English multiword expression-aware dependency parsing including named entities. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 427–432, Vancouver, Canada. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California.

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

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Joseph Le Roux, Antoine Rozenknop, and Matthieu Constant. 2014. Syntactic parsing and compound recognition via dual decomposition: Application to French. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1875–1885. Dublin City University and Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report, Stanford University.

Marie-Catherine de Marneffe and Joakim Nivre. 2019. Dependency grammar. *Annual Review of Linguistics*, 5(1):197–218.

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 (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, Haifa, Israel. Omnipress.

Alexis Nasr, Carlos Ramisch, José Deulofeu, and André Valli. 2015. Joint dependency parsing and multiword expression tokenization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1116–1126, Beijing, China. Association for Computational Linguistics.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, John Bauer, Sandra Bellato, Kepa Bengoetxea, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Tolga Kayadelen, Václava Kettnerová, Jesse Kirchner, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, 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, Niko Miekka, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Shoval Sadde, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal Dependencies 2.2. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria de Paiva. 2017. Universal Dependencies for Portuguese. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 197–206, Pisa, Italy. Linköping University Electronic Press.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Cambridge, Massachusetts.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15, Mexico City, Mexico. Springer.

Manuela Sanguinetti, Cristina Bosco, Alberto Lavelli, Alessandro Mazzei, Oronzo Antonelli, and Fabio Tamburini. 2018. PoSTWITA-UD: An Italian Twitter treebank in Universal Dependencies. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 1768–1775, Miyazaki, Japan. European Language Resources Association (ELRA).

Agata Savary, Carla Parra Escartín, Francis Bond, Jelena Mitrović, and Verginica Barbu Mititelu, editors. 2019. *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*. Association for Computational Linguistics, Florence, Italy.

Agata Savary, Carlos Ramisch, Silvio Cordeiro, Federico Sangati, Veronika Vincze, Behrang QasemiZadeh, Marie Candito, Fabienne Cap, Voula Giouli, Ivelina Stoyanova, and Antoine Doucet. 2017. The PARSEME shared task on automatic identification of verbal multiword expressions. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 31–47, Valencia, Spain. Association for Computational Linguistics.

Agata Savary, Carlos Ramisch, Jena D. Hwang, Nathan Schneider, Melanie Andresen, Sameer Pradhan, and Miriam R. L. Petruck, editors. 2018. *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah A. Smith. 2014. Discriminative lexical semantic segmentation with gaps: Running the MWE gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, Denver, Colorado. Association for Computational Linguistics.

Katalin Ilona Simkó, Viktória Kovács, and Veronika Vincze. 2017. USzeged: Identifying verbal multiword expressions with POS tagging and parsing techniques. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 48–53, Valencia, Spain. Association for Computational Linguistics.

Per Erik Solberg, Arne Skjærholt, Lilja Øvrelid, Kristin Hagen, and Janne Bondi Johannessen. 2014. The Norwegian dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 789–795, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Veronika Vincze, János Zsibrita, and István Nagy T. 2013. Dependency parsing for identifying Hungarian light verb constructions. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 207–215, Nagoya, Japan. Asian Federation of Natural Language Processing.

Jakub Waszczuk, Rafael Ehren, Regina Stodden, and Laura Kallmeyer. 2019. A neural graph-based approach to verbal MWE identification. In *Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019)*, pages 114–124, Florence, Italy. Association for Computational Linguistics.

Eric Wehrli. 2000. Parsing and collocations. In *Proceedings of the Second International Conference on Natural Language Processing*, NLP '00, pages 272–282, London, UK. Springer-Verlag.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0 LDC2013T19.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Xingxing Zhang, Jianpeng Cheng, and Mirella Lapata. 2017. Dependency parsing as head selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 665–676, Valencia, Spain. Association for Computational Linguistics.

Appendix A Evaluation of the Strengths of Our Parsing Models

To confirm that we work with reasonable parsing models, we compare our parsers with those in the CoNLL 2018 shared task (Zeman et al., 2018). The shared task featured an end-to-end parsing task, requiring all levels of text processing including tokenization, POS tagging, morphological analysis, etc. We focus on the parsing task only, and predict syntactic trees based on sentences tokenized by the Qi et al. (2018) submission.[12] Table A1 shows that our parsing models are highly competitive with the current state-of-the-art. Indeed, on four out of the six treebanks we selected for their density of flat structures, our baseline models actually achieve higher labeled attachment scores (LAS) than the the top scorer did in the official shared task.

Treebank	Our Parsers	CoNLL 2018 Best
de_gsd	**80.65**	80.36
it_ostwita	79.33	**79.39**
nl_alpino	**89.78**	89.56
nl_lassysmall	**87.96**	86.84
no_nynorsk	90.44	**90.99**
pt_bosque	**89.25**	87.81

Table A1: Comparison of our (non-MTL) parsing models with the best-performing systems (Che et al., 2018; Qi et al., 2018) from the CoNLL 2018 shared task, measured by labeled attachment scores (LAS, %).

Appendix B Do MTL and Joint Decoding Help Parsing Performance?

In Table A2 (next page), we investigate whether MTL and combining scores from both representations of flat-structure MWEs can improve parsing performance. We observe very little difference among the various strategies. This fact can be explained by the relatively low ratios of flat relations and the already-high base performance: the room for improvement on the standard LAS metrics is quite small.

[12] We thank the shared task participants and the organizers for making system predictions available at https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2885.

w/ bi-LSTM Treebank		Compl. Ratio ↓	Parsing	MTL Parsing	Joint Decoding
English		100.00	**89.30**$_{\pm 0.41}$	**89.39**$_{\pm 0.67}$	**89.77**$_{\pm 0.52}$
UD 2.2	nl_alpino	100.00	81.97$_{\pm 1.27}$	**82.57**$_{\pm 0.99}$	**82.79**$_{\pm 0.77}$
	nl_lassysmall	99.82	82.06$_{\pm 1.30}$	**82.90**$_{\pm 0.64}$	81.55$_{\pm 1.26}$
	no_nynorsk	99.78	**86.54**$_{\pm 0.50}$	**86.35**$_{\pm 0.37}$	**86.65**$_{\pm 0.64}$
	pt_bosque	97.38	84.29$_{\pm 2.15}$	84.48$_{\pm 1.61}$	**85.28**$_{\pm 0.25}$
	it_postwita	94.89	**77.39**$_{\pm 0.69}$	**76.75**$_{\pm 1.29}$	76.59$_{\pm 1.46}$
	de_gsd	93.00	**76.66**$_{\pm 0.64}$	**76.35**$_{\pm 0.83}$	75.22$_{\pm 1.98}$
Average			82.60	82.69	82.55

w/ BERT Treebank		Compl. Ratio ↓	Parsing	MTL Parsing	Joint Decoding
English		100.00	**93.73**$_{\pm 0.24}$	**93.52**$_{\pm 0.17}$	93.38$_{\pm 0.39}$
UD 2.2	nl_alpino	100.00	**89.82**$_{\pm 0.55}$	**89.95**$_{\pm 0.41}$	**89.86**$_{\pm 0.59}$
	nl_lassysmall	99.82	**89.78**$_{\pm 0.46}$	**89.76**$_{\pm 0.17}$	**89.67**$_{\pm 0.16}$
	no_nynorsk	99.78	**90.77**$_{\pm 0.20}$	**90.98**$_{\pm 0.38}$	**90.85**$_{\pm 0.32}$
	pt_bosque	97.38	**89.78**$_{\pm 0.32}$	**89.51**$_{\pm 0.39}$	**89.79**$_{\pm 0.39}$
	it_postwita	94.89	**81.61**$_{\pm 0.32}$	**81.70**$_{\pm 0.14}$	81.53$_{\pm 0.63}$
	de_gsd	93.00	81.51$_{\pm 0.23}$	**81.74**$_{\pm 0.23}$	**81.52**$_{\pm 0.17}$
Average			88.14	88.17	88.09

Table A2: Dependency-parsing labeled attachment scores (LAS, %) on the test sets with bi-LSTM (top) and BERT (bottom) feature extractors. The cell containing the best result for each treebank has blue shading; results within one standard deviation of the best are in boldface.