Valency-Augmented Dependency Parsing

Tianze Shi
Cornell University
tianze@cs.cornell.edu

Lillian Lee
Cornell University
llee@cs.cornell.edu

In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*

Abstract

We present a complete, automated, and efficient approach for utilizing valency analysis in making dependency parsing decisions. It includes extraction of valency patterns, a probabilistic model for tagging these patterns, and a joint decoding process that explicitly considers the number and types of each token's syntactic dependents. On 53 treebanks representing 41 languages in the Universal Dependencies data, we find that incorporating valency information yields higher precision and F1 scores on the core arguments (subjects and complements) and functional relations (e.g., auxiliaries) that we employ for valency analysis. Precision on core arguments improves from 80.87 to 85.43. We further show that our approach can be applied to an ostensibly different formalism and dataset, Tree Adjoining Grammar as extracted from the Penn Treebank; there, we outperform the previous state-of-the-art labeled attachment score by 0.7. Finally, we explore the potential of extending valency patterns beyond their traditional domain by confirming their helpfulness in improving PP attachment decisions.[1]

1 Introduction

Many dependency parsers treat attachment decisions and syntactic relation labeling as two independent tasks, despite the fact that relation labels carry important subcategorization information. For example, the number and types of the syntactic arguments that a predicate may take is rather restricted for natural languages — it is not common for an English verb to have more than one syntactic subject or more than two objects.

In this work, we present a parsing approach that explicitly models subcategorization of (some) syntactic dependents as *valency patterns* (see Fig. 1 for examples), and operationalize this notion as extracted supertags. An important distinction from prior work is that our definition of valency-pattern supertags is relativized to a user-specified subset of all possible syntactic relations (see §3). We train supertaggers that assign probabilities of potential valency patterns to each token, and leverages these probabilities during decoding to guide our parsers so that they favor more linguistically plausible output structures.

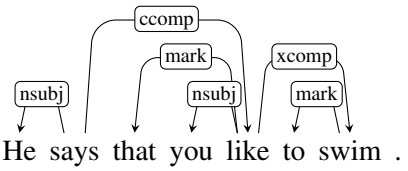


Figure 1: Sample annotation in UD, encoding the core valency pattern nsubj ⋄ ccomp for *says*, nsubj ⋄ xcomp for *like*, and so on (see §2-4)

We mainly focus on two subsets of relations in our analysis, those involving core arguments and those that represent functional relations, and perform experiments over a collection of 53 treebanks in 41 languages from the Universal Dependencies dataset (UD; Nivre et al., 2017). Our valency-aware parsers improve upon strong baseline systems in terms of output linguistic validity, measured as the accuracy of the assigned valency patterns. They also have higher precision and F1 scores on the subsets of relations under analysis, suggesting a potentially controlled way to balance precision-recall trade-offs.

We further show that our approach is not limited to a particular treebank annotation style. We apply our method to parsing another grammar formalism, Tree Adjoining Grammar, where dependency and valency also play an important role in both theory and parser evaluation. Our parser reaches a new state-of-the-art LAS score of 92.59, with more than 0.6 core-argument F1-score improve-

[1] Our implementation is available at https://github.com/tzshi/valency-parser-emnlp18

ment over our strong baseline parser.

Finally, we demonstrate the applicability of our valency analysis approach to other syntactic phenomena less associated with valency in its traditional linguistic sense. In a case study of PP attachment, we analyze the patterns of two syntactic relations commonly used in PP attachment, and include them in the joint decoding process. Precision of the parsers improves by an absolute 3.30% on these two relation types.

2 Syntactic Dependencies and Valencies

According to Nivre (2005), the modern dependency grammar can be traced back to Tesnière (1959), with its roots reaching back several centuries before the Common Era. The theory is centered on the notion of *dependency*, an asymmetrical relation between words of a sentence. Tesnière distinguishes three node types when analyzing simple predicates: verb equivalents that describe actions and events, noun equivalents as the arguments of the events, and adverb equivalents for detailing the (temporal, spatial, etc.) circumstances. There are two types of relations: (1) verbs dominate nouns and adverbs through a dependency relation; (2) verbs and nouns are linked through a *valency* relation. Tesnière compares a verb to an atom: a verb can attract a certain number of arguments, just as the valency of an atom determines the number of bonds it can engage in (Ágel and Fischer, 2015). In many descriptive lexicographic works (Helbig and Schenkel, 1959; Herbst et al., 2004), valency is not limited to verbs, but also includes nouns and adjectives. For more on the linguistic theory, see Ágel et al. (2003, 2006).

Strictly following the original notion of valency requires distinguishing between arguments and adjuncts, as well as obligatory and optional dependents. However, there is a lack of consensus as to how these categorizations may be distinguished (Tutunjian and Boland, 2008), and thus we adopt a more practical definition in this paper.

3 Computational Representation

Formally, we fix a set of syntactic relations $\mathcal{R}$, and define the *valency pattern* of a token w_i with respect to $\mathcal{R}$ as the linearly-ordered[2] sequence $a_{-j} \cdots a_{-1} \diamond a_1 \cdots a_k$: the $\diamond$ symbol denotes the center word w_i, and each a_l asserts the existence of a word w dominated by w_i via relation $a_l \in \mathcal{R}$, $w_i \xrightarrow{a_l} w$. For a_l and a_m, when $l < m$, the syntactic dependent for a_l linearly precedes the syntactic dependent for a_m. As an example, consider the UD-annotated sentence in Fig. 1. The token *says* has a core-relation[3] valency pattern nsubj $\diamond$ ccomp, and *like* has the pattern nsubj $\diamond$ xcomp. If we consider only functional relations, both *like* and *swim* have the pattern mark $\diamond$.[4] We sometimes employ the abbreviated notation $\alpha^L \diamond \alpha^R$, where α indicates a sequence and the letters L and R distinguish left dependencies from right dependencies.

Dataset	Subset	Syntactic Relations
UD	Core	nsubj, obj, iobj, csubj, ccomp, xcomp
	Func.	aux, cop, mark, det, clf, case
	PP (§8)	nmod, obl
TAG	Core	0 (subject), 1 (object), 2 (indirect object)
	Co-head	CO

Table 1: Sets of syntactic relations we used for valency analysis. UD subsets come from the official categorization in the annotation guidelines.

We make our definition of valency patterns dependent on choice of $\mathcal{R}$ not only because some dependency relations are more often obligatory and closer to the original theoretical definition of valency, but also because the utility of different types of syntactic relations can depend on the downstream task. For example, purely functional dependency labels are semantically vacuous, so they are often omitted in the semantic representations extracted from dependency trees for question answering (Reddy et al., 2016, 2017). There are also recent proposals for parser evaluation that downplay the importance of functional syntactic relations (Nivre and Fang, 2017).

[2] Our approach, whose full description is in §5, can be adapted to cases where linear ordering is de-emphasized. The algorithm merely requires a distinction between left and right dependents. We choose to encode linearity since it appears that most languages empirically exhibit word order *preferences* even if they allow for relatively free word order.

[3] UD core and functional relations are listed in Table 1.

[4] The (possibly counterintuitive) direction for *that* and *to* is a result of UD's choice of a content-word-oriented design.

4 Pilot Study: Sanity Checks

We consider two questions that need to be addressed at the outset:[5]

1. How well do the extracted patterns generalize to unseen data?

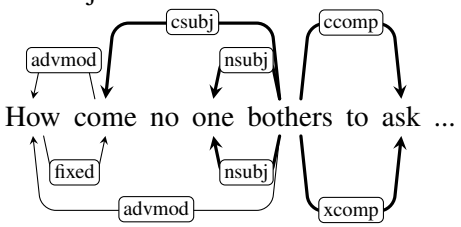
2. Do state-of-the-art parsers already capture the notion of valency implicitly, though they are not explicitly optimized for it?

The first question checks the feasibility of learning valency patterns from a limited amount of data; the second probes the potential for any valency-informed parsing approach to improve over current state-of-the-art systems.

To answer these questions, we use the UD 2.0 dataset for the CoNLL 2017 shared task (Zeman et al., 2017) and the system outputs[6] of the top five performing submissions (Dozat et al., 2017; Shi et al., 2017b; Björkelund et al., 2017; Che et al., 2017; Lim and Poibeau, 2017). Selection of treebanks is the same as in §6. We extract valency patterns relative to the set of 6 UD core arguments given in Table 1 because they are close to the original notion of valency and we hypothesize that these patterns should exhibit few variations. This is indeed the case: the average number of valency patterns we extract is 110.4 per training treebank, with Turkish (`tr`) having the fewest at 34, and Galician (`gl`) having the most at 298 patterns. We observe that in general, languages with higher degree of flexibility in word order tend to generate more patterns in the data, as our patterns encode linear word order information.

Next, we extract valency patterns from the test set and compare them against those from the training set. On average, out of the 55.4 patterns observed in the gold-standard test sets, only 5.5, or 9.98%, are new and unseen with respect to training. In comparison, 36.2% of the word types appearing in the test sets are not seen during training. This suggests that the valency pattern space is relatively restricted, and the patterns extracted from training sets do generalize well to test sets.

Finally, we consider the average number of valency patterns extracted from the top-performing system outputs and the number of those not observed in training.[7] All 5 systems are remarkably "hallucinatory" in inventing valency relations, introducing 16.8 to 35.5 new valency patterns, significantly larger than the actual number of unseen patterns. Below we show an error committed by the state-of-the-art Dozat et al. (2017) parser (upper half) as compared to the gold-standard annotation (lower half), and we highlight the core argument valency relations of the verb *bothers* in bold. The system incorrectly predicts *how come* to be a clausal subject.



Each such non-existent new pattern implies at least some (potentially small) parsing error that can contribute to the degradation of downstream task performance.

5 Valency-Aware Dependency Parsing

5.1 Overview

Our model is based on the following probability factorization for a given sentence $x = w_1, \ldots, w_n$ and parse tree y for x:

$$P(y|x) = \frac{1}{Z_x} \prod_{i=1}^{n} P(v_i|w_i) P(h_i|w_i) P(r_i|w_i, h_i),$$

where Z_x is the normalization factor, v_i is the valency pattern extracted for w_i from y, h_i is the index of the syntactic governor of w_i, and r_i is the syntactic relation label of the dependency relation between w_{h_i} and w_i. We first assume that we have a feature extractor that associates each token in the sentence w_i with a contextualized feature vector $\mathbf{w}_i$, and explain how to calculate the factored probabilities (§5.2). Then we discuss decoding (§5.3) and training (§5.4). Our decoder can be viewed as a special-case implementation of head-automaton grammars (Alshawi, 1996; Eisner and Satta, 1999). Finally, we return to the issue of feature extraction (§5.5).

[5] We actually performed these sanity checks after implementation and experiments of our approach, because we missed this idea and because it requires access to test sets that we abstained from looking at during model development.

[6] Retrieved from `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2424`.

[7] The CoNLL 2017 shared task is an end-to-end parsing task, so the participating systems do not have access to gold-standard tokenization, which is a potential explanation for the presented analysis. On the other hand, the conclusion still holds even if we restrict to system outputs with perfect or nearly perfect segmentations.

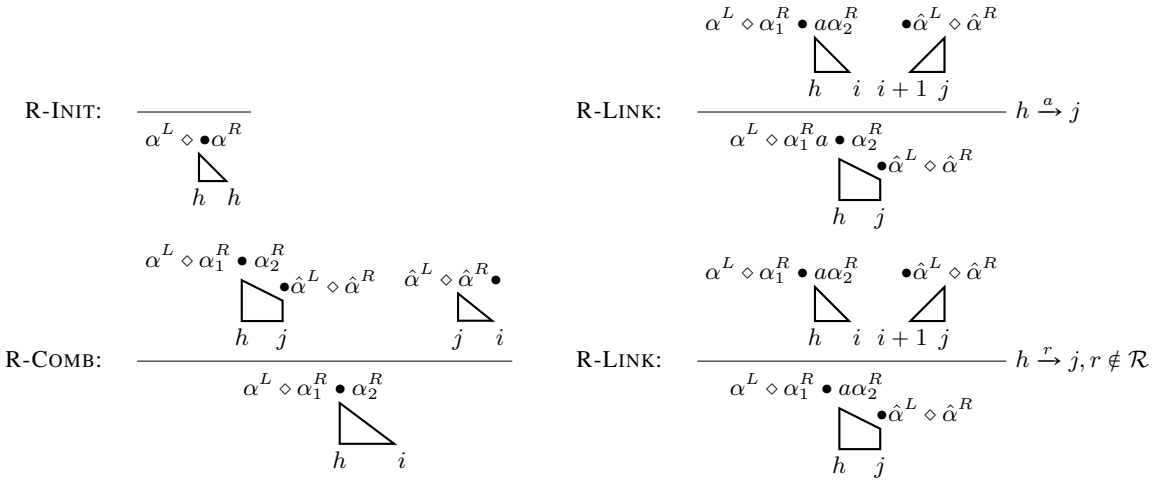


Figure 2: Eisner's (1996)/Eisner and Satta's (1999) algorithm, with valency-pattern annotations, incorporated as state information, shown explicitly. We show only the R-rules; the L-rules are symmetric.

5.2 Parameterization

We parameterize $P(v_i|w_i)$ as a softmax distribution over all candidate valency patterns:

$$P(v_i|w_i) \propto \exp(\text{score}^{\text{VAL}}_{v_i}(\mathbf{w}_i)),$$

where $\text{score}^{\text{VAL}}$ is a multi-layer perceptron (MLP).

For each word w_i, we generate a probability distribution over all potential syntactic heads in the sentence (Zhang et al., 2017). After we have selected the head of w_i to be w_{h_i}, we decide on the syntactic relation label based on another probability distribution. We use two softmax functions:

$$P(h_i|w_i) \propto \exp(\text{score}^{\text{HEAD}}(\mathbf{w}_{h_i}, \mathbf{w}_i)),$$
$$P(r_i|w_i, h_i) \propto \exp(\text{score}^{\text{LABEL}}_{r_i}(\mathbf{w}_{h_i}, \mathbf{w}_i)),$$

where both $\text{score}^{\text{HEAD}}$ and $\text{score}^{\text{LABEL}}$ are parameterized by deep biaffine scoring functions (Dozat and Manning, 2017).

5.3 Decoding

For joint decoding, we adopt the Eisner's (1996) algorithm annotated with valency patterns as the state information in Eisner and Satta (1999). The algorithm is depicted in Fig. 2. For each complete and incomplete span, visualized as triangles and trapezoids respectively, we annotate the head with its valency pattern. We adopt Earley's (1970) notation of $\bullet$ to outward-delimit the portion of a valency pattern, starting from the center word $\diamond$, that has already been collected within the span. INIT generates a minimal complete span with hypothesized valency pattern; the $\bullet$ is put adjacent to $\diamond$. COMB matches an incomplete span to a complete span with compatible valency pattern, yielding a complete analysis on the relevant side of $\diamond$. LINK either advances the $\bullet$ by attaching a syntactic dependent with the corresponding relation label, or attaches a dependent with a relation label irrelevant to the current valency analysis. This algorithm can be easily extended to cases where we analyze multiple subsets of valency relations simultaneously: we just need to annotate each head with multiple layers of valency patterns, one for each subset.[8]

The time complexity of a naïve dynamic programming implementation is $O(|V|^2|\alpha|n^3)$, where $|V|$ is the number of valency patterns and $|\alpha|$ is the maximum length of a valency pattern. In practice, $|V|$ is usually larger than n, making the algorithm prohibitively slow. We thus turn to A* parsing for a more practical solution.

A* parsing We take inspiration from A* CCG parsing (Lewis and Steedman, 2014; Lewis et al., 2016; Yoshikawa et al., 2017). The idea (see Alg. 1) is to estimate the best compatible full parse for every chart item (in our case, complete and incomplete spans), and expand the chart based on the estimated priority scores. Our factorization of probability scores allows the following admissible heuristic: for each span, we can optimistically estimate its best full parse score by assigning to

[8] To allow our model to account for unseen patterns in new data, we create a special wildcard valency pattern that allows dependents with arbitrary relations in the decoding process, and during training, treat valency patterns occurring fewer than 5 times as examples of the wildcard pattern.

Algorithm 1 Agenda-based best-first parsing algorithm, adapted from Lewis et al. (2016), Alg. 1.

Helper Functions: INIT(s) returns the set of spans generated by INIT. C.RULES(p) returns the set of spans that can be derived by combining p with existing entries in C through COMB or LINK.

procedure PARSE(s)
  // Empty priority queue A
  $A \leftarrow \varnothing$
  // Initialize A with minimal complete spans
  for $p \in$ INIT(s) **do**
    A.INSERT(p);
  // Empty chart C
  $C \leftarrow \varnothing$
  while $A \neq \varnothing$ **do**
    $p \leftarrow A$.POPMAX()
    // Found the global optimal solution
    if p is a full parse **then return** p
    else if $p \notin C$ **then**
      C.ADD(p)
      // Extend the chart
      for $p' \in C$.RULES(p) **do**
        A.INSERT(p')

every token outside the span the best possible valency pattern, best possible attachment and best relation label.

5.4 Training

We train all components jointly and optimize for the cross entropy between our model prediction and the gold standard, or, equivalently, the sum of the log-probabilities for the three distributions comprising our factorization from §5.1. This can be thought of as an instance of multi-task learning (MTL; Caruana, 1997), which has been shown to be useful in parsing (Kasai et al., 2018). To further reduce error propagation, instead of using part-of-speech tags as features, we train a tagger jointly with our main parser components (Zhang and Weiss, 2016).

5.5 Feature Extraction

We adopt bi-directional long short-term memory networks (bi-LSTMs; Hochreiter and Schmidhuber, 1997) as our feature extractors, since they have proven successful in a variety of syntactic parsing tasks (Kiperwasser and Goldberg, 2016; Cross and Huang, 2016; Stern et al., 2017; Shi et al., 2017a). As inputs to the bi-LSTMs, we concatenate one pre-trained word embedding, one randomly-initialized word embedding, and the output of character-level LSTMs for capturing sub-token level information (Ballesteros et al., 2015). The bi-LSTM output vectors at each timestep are then assigned to each token as its contextualized representation $\mathbf{w}_i$.

6 Experiments

Data and Evaluation Our main experiments are based on UD version 2.0, which was prepared for the CoNLL 2017 shared task (Zeman et al., 2017). We used 53 of the treebanks[9] across 41 languages that have train and development splits given for the shared task. In contrast to the shared-task setting, where word and sentence segmentation are to be performed by the system, we directly use the test-set gold segmentations in order to focus directly on parsing; but this does mean that the performance of our models cannot be directly compared to the officially-reported shared-task results. For evaluation, we report unlabeled and labeled attachment scores (UAS and LAS respectively). Further, we explicitly evaluate precision, recall and F1 scores (P/R/F) for the syntactic relations from Table 1, as well as valency pattern accuracies (VPA) involving those relations.

Implementation Details We use three-layer bi-LSTMs with 500 hidden units (250 in each direction) for feature extraction. The valency analyzer uses a one-hidden-layer MLP with ReLU activation function (Nair and Hinton, 2010), while the head selector and labeler use 512- and 128-dimensional biaffine scoring functions respectively. Our models are randomly initialized (Glorot and Bengio, 2010) and optimized with AMSgrad (Reddi et al., 2018) with initial learning rate 0.002. We apply dropout (Srivastava et al., 2014) to our MLPs and variational dropout (Gal and Ghahramani, 2016) to our LSTMs with a keep rate of 0.67 during training.

Efficiency Our A* parsers are generally reasonably efficient; for the rare ($< 1\%$) cases where the A* search does not finish within 500,000 chart expansion steps, we back off to a model without valency analysis. When analyzing three or more relation subsets, the initialization steps become pro-

[9] We exclude the two large treebanks `cs` and `ru_syntagrus` due to experiment resource constraints. There are other Czech and Russian treebanks in our selected collection.

Subsets	UAS	LAS	Core #	Core VPA	Core P / R / F	Func. #	Func. VPA	Func. P / R / F
Baseline	87.59	83.64	2.75	95.83	80.87 / 81.31 / 81.08	4.85	97.51	91.99 / 92.43 / 92.20
Core MTL	87.71	83.80	2.73	96.02	81.96 / **81.98** / 81.96	4.85	97.51	91.96 / 92.50 / 92.23
+ Joint Decoding	87.80	83.93	2.60	**96.68**	**85.43** / 81.75 / **83.53**	4.86	97.50	91.81 / 92.65 / 92.22
Func. MTL	87.67	83.71	2.75	95.80	80.69 / 81.21 / 80.94	4.84	97.58	92.30 / 92.57 / 92.44
+ Joint Decoding	87.72	83.75	2.75	95.80	80.64 / 81.32 / 80.96	4.80	**97.74**	**93.16** / 92.42 / 92.79
Core + Func. MTL	87.67	83.79	2.73	95.99	81.72 / 81.81 / 81.75	4.84	97.59	92.27 / 92.62 / 92.44
+ Joint Decoding	**87.81**	**83.99**	2.63	96.60	84.70 / 81.90 / 83.26	4.82	**97.74**	92.98 / **92.69** / **92.83**

Table 2: Macro-averaged results on UD 2.0 across 53 treebanks. Detailed results in the Suppl. Material. VPA=valency pattern accuracy; MTL=multi-task learning; #=average number of predicted attachments per sentence. Best results for each metrics are highlighted in bold.

Treebank	Baseline	Joint	ER	Treebank	Baseline	Joint	ER	Treebank	Baseline	Joint	ER
Dutch$_{MAX}$	84.91	89.83	32.63	Portuguese$_{MAX}$	92.24	93.46	15.66	Norwegian$_{MIN}$	90.38	91.41	10.76
Greek	85.82	89.50	25.95	Portuguese$_{MIN}$	86.90	88.88	15.10	Indonesian	81.53	83.51	10.75
Swedish$_{MAX}$	86.97	90.21	24.92	Spanish$_{MAX}$	85.04	87.29	15.06	Latvian	71.33	74.41	10.74
Finnish$_{MAX}$	88.14	90.87	22.96	Hungarian	79.11	82.13	14.45	French$_{MIN}$	90.56	91.51	10.01
Italian	87.04	90.00	22.85	Arabic	74.33	77.97	14.16	Basque	76.79	78.97	9.39
Latin$_{MAX}$	82.52	86.37	22.03	Urdu	70.47	74.63	14.07	Hindi	80.38	82.16	9.04
Danish	85.85	88.93	21.75	Dutch$_{MIN}$	76.03	79.18	13.13	German	81.05	82.73	8.85
Finnish$_{MIN}$	87.95	90.44	20.68	Swedish$_{MIN}$	85.51	87.36	12.76	Czech$_{MIN}$	76.68	78.64	8.42
Slovenian	86.03	88.59	18.31	Croatian	84.14	86.12	12.46	Polish	86.67	87.72	7.84
Old Slavonic	76.79	81.02	18.25	Gothic	72.15	75.60	12.38	A. Greek$_{MIN}$	59.00	62.17	7.74
French$_{MAX}$	89.86	91.71	18.21	A. Greek$_{MAX}$	74.31	77.48	12.34	Turkish	58.43	61.59	7.61
Estonian	72.02	77.09	18.09	Hebrew	80.27	82.60	11.80	Korean	83.33	84.55	7.31
Slovak	80.39	83.78	17.32	Persian	80.83	83.08	11.72	Chinese	73.81	75.66	7.07
Czech$_{MAX}$	85.58	88.02	16.90	Norwegian$_{MAX}$	91.20	92.21	11.50	English$_{MIN}$	84.69	85.60	5.96
Latin$_{MIN}$	76.60	80.55	16.89	Catalan	88.19	89.54	11.49	Vietnamese	48.45	51.49	5.91
Romanian	82.60	85.51	16.74	English$_{MID}$	84.26	86.05	11.37	Galician	72.17	73.70	5.49
English$_{MAX}$	90.96	92.43	16.20	Spanish$_{MIN}$	88.72	89.98	11.17	Japanese	91.87	91.88	0.14
Russian	82.17	85.05	16.13	Bulgarian	83.98	85.75	11.02	Average	81.08	83.53	13.80

Table 3: Treebank-specific F1 scores on core argument relations, comparing the baseline models to our Core MTL + joint decoding models, sorted by the error reduction (ER, %) rate. When comparing a model with performance s_2 against baseline score s_1, ER is defined as $(s_2 - s_1)/(1 - s_1)$. For languages with two or three treebanks, we include multiple entries differentiated by the subscripts MAX/MID/MIN, corresponding to the treebanks with the highest/median/lowest ER, respectively. A. Greek = Ancient Greek.

hibitively slow due to the large number of valency pattern combinations. Thus, we limit the number of combinations for each token to the highest-scoring 500.

Results on UD We present our main experimental results on UD in Table 2. The baseline system does not leverage any valency information (we only train the head selectors and labelers, and use the original Eisner decoder). We compare the baseline to settings where we train the parsers jointly with our proposed valency analyzers, distinguishing the effect of using this information only at training (multi-task learning; MTL) vs. both at training and decoding.

Including valency analysis into the training objective already provides a slight improvement in parsing performance, in line with the findings of Kasai et al. (2018). With our proposed joint decoding, there is a mild improvement to the overall UAS and LAS, and a higher boost to VPA. The output parse trees are now more precise in the analyzed valency relations: on core arguments, precision increases by as much as 4.56. As shown by Table 3, the performance gain of joint decoding varies across treebanks, ranging from an error reduction rate of over 30% (Dutch Lassy Small Treebank) on core argument relations to nearly 0% (Japanese). Overall, our approach exhibits a clearly positive impact on most of the treebanks in UD. We do not see performance correlating to language typology, although we do observe smaller error-reduction rates on treebanks with lower baseline performances, that is, on "harder" languages.

7 Parsing Tree Adjoining Grammar

Dependency and valency relations also play an important role in formalisms other than dependency grammar. In this section, we apply our proposed valency analysis to Tree Adjoining Grammar (TAG; Joshi and Schabes, 1997), because TAG derivation trees, representing the process of inserting obligatory arguments and adjoining modifiers, can be treated as a dependency representation (Rambow and Joshi, 1997). We follow prior art and use Chen's (2001) automatic conversion of the Penn Treebank (Marcus et al., 1993) into TAG derivation trees. The dataset annotation has labels 0, 1 and 2, corresponding to subject, direct object, and indirect object; we treat these as our core argument subset in valency analysis.[10] Additionally, we also analyze CO (co-head for phrasal verbs) as a separate singleton subset. We leave out adj (adjuncts) in defining our valency patterns. We strictly follow the experiment protocol of previous work (Bangalore et al., 2009; Chung et al., 2016; Friedman et al., 2017; Kasai et al., 2017, 2018), and report the results in Table 4. The findings are consistent with our main experiments: MTL helps parsing performance, and joint decoding further improves on core argument F1 scores, reaching a new state-of-the-art result of 92.59 LAS. The precision recall trade-off is pronounced for the CO relation subset.

8 Case Study on PP Attachment

Although valency information has traditionally been used to analyze complements or core arguments,[11] in this section, we show the utility of our approach in analyzing other types of syntactic relations. We choose the long-standing problem of prepositional phrase (PP) attachment (Hindle and Rooth, 1993; Brill and Resnik, 1994; Collins and Brooks, 1995; de Kok et al., 2017), which is known to be a major source of parsing mistakes (Kummerfeld et al., 2012; Ng and Curran, 2015). In UD analysis, PPs usually have the labels obl or nmod with respect to their syntactic parents, whereas adpositions are attached via a case relation, which is included in the functional relation subset. Thus, we add another relation subset, obl and nmod, to our valency analysis.

Table 5 presents the results for different combinations of valency relation subsets. We find that PP-attachment decisions are generally harder to make, compared with core and functional relations. Including them during training distracts other parsing objectives (compare Core + PP with only analyzing Core in §6). However, they do permit improvements on precision for PP attachment by 3.30, especially with our proposed joint decoding. This demonstrates the usage of our algorithm outside the traditional notions of valency — it can be a general method for training parsers to focus on specific subsets of syntactic relations.

[10] We choose not to use the sparse labels 3 and 4, which encode additional complements.

[11] There are also recent proposals to analyze valency without distinguishing complements and adjuncts (Čech et al., 2010).

	UAS	LAS	Core VPA	Core P / R / F	CO VPA	CO P / R / F
Friedman et al. (2017)	90.31	88.96	–	–	–	–
Kasai et al. (2017)	90.97	89.68	–	–	–	–
Kasai et al. (2018)	93.26	91.89	–	–	–	–
Baseline	93.66	92.44	97.06	92.45 / 92.76 / 92.60	99.22	73.11 / **87.20** / 79.54
Core + CO MTL	93.71	92.53	97.19	92.74 / 93.20 / 92.97	**99.24**	75.43 / 84.44 / 79.68
+ Joint Decoding	**93.75**	**92.59**	**97.47**	**93.27 / 93.22 / 93.24**	**99.24**	**76.06** / 83.70 / **79.70**

Table 4: Experimental results on parsing TAGs.

	UAS	LAS	Core P / R / F	Func. P / R / F	PP P / R / F
Baseline	87.59	83.64	80.87 / 81.31 / 81.08	91.99 / 92.43 / 92.20	77.29 / 77.99 / 77.62
PP MTL	87.67	83.70	80.61 / 81.23 / 80.91	92.03 / 92.50 / 92.26	78.30 / **78.38** / 78.32
+ Joint Decoding	87.68	83.69	79.93 / 81.50 / 80.69	91.92 / 92.51 / 92.21	**80.59** / 77.68 / **79.04**
Core + PP MTL	87.70	83.77	81.62 / 81.81 / 81.71	91.93 / 92.52 / 92.22	77.93 / 78.25 / 78.08
+ Joint Decoding	87.80	83.91	**84.18 / 81.97 / 83.05**	91.68 / **92.65** / 92.16	79.71 / 78.03 / 78.83
Core + Func. + PP MTL	87.67	83.75	81.35 / 81.68 / 81.50	92.18 / 92.61 / 92.39	77.99 / 78.22 / 78.08
+ Joint Decoding	**87.81**	**83.94**	83.88 / **81.97** / 82.90	**92.78** / 92.63 / **92.70**	79.54 / 78.11 / 78.78

Table 5: Experimental results involving analyzing PPs as valency patterns.

9 Further Related Work

Supertagging Supertagging (Bangalore and Joshi, 2010) has been proposed for and used in parsing TAG (Bangalore and Joshi, 1999; Nasr and Rambow, 2004), CCG (Curran and Clark, 2003; Curran et al., 2006), and HPSG (Ninomiya et al., 2006; Blunsom and Baldwin, 2006). Within dependency parsing, supertags have also been explored in the literature, but prior work mostly treats them as additional features. Ambati et al. (2013, 2014) use CCG supertags to improve dependency parsing results, while Ouchi et al. (2014, 2016) leverage dependency-based supertags as features. Faleńska et al. (2015) compare supertagging to parser stacking, where they extract supertags from base parsers to provide additional features for stacked parsers, instead of having a supertagger as a separate component.

Constrained Dependency Grammar Another line of research (Wang and Harper, 2004; Foth et al., 2006; Foth and Menzel, 2006; Bharati et al., 2002, 2009; Husain et al., 2011) utilizes supertags in dependency parsing within the framework of constraint dependency grammar (CDG; Maruyama, 1990; Heinecke et al., 1998). Constraints in CDG may be expressed in very general terms (and are usually hand-crafted for specific languages), so prior work in CDG involves a constraint solver that iteratively or greedily update hypotheses without optimality guarantees. In contrast, our work focuses on a special form of constraints — the valency patterns of syntactic dependents within a subset of relations — and we provide an efficient A*-based exact decoding algorithm.

Valency in Parsing To the best of our knowledge, there have been few attempts to utilize lexical valency information or to improve specifically on core arguments in syntactic parsing apart from CDG. Øvrelid and Nivre (2007) target parsing core relations in Swedish with specifically-designed features such as animacy and definiteness that are useful in argument realization. Jakubıček and Kovář (2013) leverage external lexicons of verb valency frames for reranking. Mirroshandel et al. (2012, 2013) and Mirroshandel and Nasr (2016) extract selectional constraints and subcategorization frames from large unannotated corpora, and enforce them through forest reranking. Our approach does not rely on external resources or lexicons, but directly extracts valency patterns from labeled dependency parse trees. Earlier works in this spirit include Collins (1997).

Semantic Dependency Parsing and Semantic Role Labeling The notion of valency is also used to describe predicate-argument structures that are adopted in semantic dependency parsing and semantic role labeling (Surdeanu et al.,

2008; Hajič et al., 2009; Oepen et al., 2014, 2015). While semantic frames clearly have patterns, previous work (Punyakanok et al., 2008; Flanigan et al., 2014; Täckström et al., 2015; Peng et al., 2017; He et al., 2017) incorporates several types of constraints, including uniqueness and determinism constraints that require that certain labels appear as arguments for a particular predicate only once. They perform inference through integer linear programming, which is usually solved approximately, and cannot easily encode linear ordering constraints for the arguments.

A* parsing Best-first search uses a heuristic to expand the parsing chart instead of doing so exhaustively. It was first applied to PCFGs (Ratnaparkhi, 1997; Caraballo and Charniak, 1998; Sagae and Lavie, 2006), and then to dependency parsing (Sagae and Tsujii, 2007; Zhao et al., 2013; Vaswani and Sagae, 2016). Our probability factorization permits a simple yet effective A* heuristic. A* parsing was introduced for parsing PCFGs (Klein and Manning, 2003; Pauls and Klein, 2009), and has been widely used for grammar formalisms and parsers with large search spaces, for example CCG (Auli and Lopez, 2011) and TAG (Waszczuk et al., 2016, 2017). Our decoder is similar to the supertag and dependency factored A* CCG parser (Yoshikawa et al., 2017), which in turn builds upon the work of Lewis and Steedman (2014) and Lewis et al. (2016). Our model additionally adds syntactic relations into the probability factorizations.

10 Conclusions

We have presented a probability factorization and decoding process that integrates valency patterns into the parsing process. The joint decoder favors syntactic analyses with higher valency-pattern supertagging probabilities. Experiments on a large set of languages from UD show that our parsers are more precise in the subset of syntactic relations chosen for valency analysis, in addition to enjoying the benefits gained from jointly training the parsers and supertaggers in a multi-task learning setting.

Our method is not limited to a particular type of treebank annotation or a fixed subset of relations. We draw similar conclusions when we parse TAG derivation trees. Most interestingly, in a case study on PP attachment, we confirm the utility of our parsers in handling syntactic relations beyond the traditional domain of valency.

A key insight of this paper that departs from prior work on automatic extraction of supertags from dependency annotations is that our definition of valency patterns is relativized to a subset of syntactic relations. This definition is closer to the linguistic notion of valency and alleviates the data sparsity problems in that the number of extracted valency patterns is small. At the same time, the patterns generalize well, and empirically, they are effective in our proposed joint decoding process.

Our findings point to a number of directions for future work. First, the choice of subsets of syntactic relations for valency analysis impacts the parsing performance in those categories. This may suggest a controllable way to address precision-recall trade-offs targeting specific relation types. Second, we experimented with a few obvious subsets of relations; characterizing what subsets can be most improved with valency augmentation is an open question. Finally, our decoder builds upon projective dependency-tree decoding algorithms. In the future, we will explore the possibility of removing the projective constraint and the tree requirement, extending the applicability of valency patterns to other tasks such as semantic role labeling.

Acknowledgments

We thank the three anonymous reviewers for their insightful comments, Jungo Kasai for assistance in setting up the TAG parsing experiments, and Xilun Chen, Jason Eisner, Jungo Kasai and Ana Smith for discussion and comments. We also thank CoNLL'17 shared task organizers and participants for publicizing system outputs. TS and LL were supported in part by a Google Focused Research Grant to Cornell University. LL was also supported in part by NSF grant SES-1741441. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other sponsors.

References

Vilmos Ágel, Ludwig M. Eichinger, Hans Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, and Henning Lobin, editors. 2003. *Dependency and valency: An international handbook of contemporary research*, volume 1. De Gruyter Mouton.

Vilmos Ágel, Ludwig M. Eichinger, Hans Werner Eroms, Peter Hellwig, Hans Jürgen Heringer, and Henning Lobin, editors. 2006. *Dependency and valency: An international handbook of contemporary research*, volume 2. De Gruyter Mouton.

Vilmos Ágel and Klaus Fischer. 2015. Dependency grammar and valency theory. In Bernd Heine and

Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, 2nd edition. Oxford University Press, Oxford.

Hiyan Alshawi. 1996. Head automata and bilingual tiling: Translation with minimal representations (invited talk). In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 167–176, Santa Cruz, California, USA. Association for Computational Linguistics.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2013. Using CCG categories to improve Hindi dependency parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 604–609, Sofia, Bulgaria. Association for Computational Linguistics.

Bharat Ram Ambati, Tejaswini Deoskar, and Mark Steedman. 2014. Improving dependency parsers using Combinatory Categorial Grammar. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 159–163, Gothenburg, Sweden. Association for Computational Linguistics.

Michael Auli and Adam Lopez. 2011. Efficient CCG parsing: A* versus adaptive supertagging. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1577–1585, Portland, Oregon, USA. Association for Computational Linguistics.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal. Association for Computational Linguistics.

Srinivas Bangalore, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA: A probabilistic dependency parser based on tree insertion grammars (application note). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 185–188, Boulder, Colorado. Association for Computational Linguistics.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics.*, 25(2):237–265.

Srinivas Bangalore and Aravind K. Joshi. 2010. *Supertagging: Using Complex Lexical Descriptions in Natural Language Processing*. The MIT Press.

Akshar Bharati, Samar Husain, Dipti Misra, and Rajeev Sangal. 2009. Two stage constraint based hybrid approach to free word order language dependency parsing. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 77–80. Association for Computational Linguistics.

Akshar Bharati, Rajeev Sangal, and T. Papi Reddy. 2002. A constraint based parser using integer programming. *Proceedings of the International Conference on Natural Language Processing*.

Anders Björkelund, Agnieszka Falenska, 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*, pages 40–51, Vancouver, Canada. Association for Computational Linguistics.

Phil Blunsom and Timothy Baldwin. 2006. Multilingual deep lexical acquisition for HPSGs via supertagging. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 164–171, Sydney, Australia. Association for Computational Linguistics.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of the 15th Conference on Computational Linguistics – Volume 2*, pages 1198–1204, Kyoto, Japan. Association for Computational Linguistics.

Sharon A. Caraballo and Eugene Charniak. 1998. New figures of merit for best-first probabilistic chart parsing. *Computational Linguistics*, 24(2):275–298.

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

Radek Čech, Petr Pajas, and Ján Majčutek. 2010. Full valency. Verb valency without distinguishing complements and adjuncts. *Journal of Quantitative Linguistics*, 17(4):291–302.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, 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*, pages 52–62, Vancouver, Canada. Association for Computational Linguistics.

John Chen. 2001. *Towards efficient statistical parsing using lexicalized grammatical information*. Ph.D. thesis, University of Delaware.

Wonchang Chung, Suhas Siddhesh Mhatre, Alexis Nasr, Owen Rambow, and Srinivas Bangalore. 2016. Revisiting supertagging and parsing: How to use supertags in transition-based parsing. In *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 85–92, Düsseldorf, Germany.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Michael Collins and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the 3rd Workshop on Very Large Corpora*, Cambridge, MA.

James Cross and Liang Huang. 2016. Incremental parsing with minimal features using bi-directional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 32–37.

James R. Curran and Stephen Clark. 2003. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics – Volume 1*, pages 91–98, Budapest, Hungary. Association for Computational Linguistics.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-tagging for lexicalized-grammar parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 697–704, Sydney, Australia. 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*.

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*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Jay Earley. 1970. An efficient context-free parsing algorithm. *Communations of the ACM*, 13(2):94–102.

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.

Jason Eisner and Giorgio Satta. 1999. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 457–464, College Park, Maryland, USA. Association for Computational Linguistics.

Agnieszka Faleńska, Anders Björkelund, Özlem Çetinoğlu, and Wolfgang Seeker. 2015. Stacking or supertagging for dependency parsing – what's the difference? In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 118–129, Bilbao, Spain. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Baltimore, Maryland. Association for Computational Linguistics.

Kilian A. Foth, Tomas By, and Wolfgang Menzel. 2006. Guiding a constraint dependency parser with supertags. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 289–296, Sydney, Australia. Association for Computational Linguistics.

Kilian A. Foth and Wolfgang Menzel. 2006. Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 321–328, Sydney, Australia. Association for Computational Linguistics.

Dan Friedman, Jungo Kasai, R. Thomas McCoy, Robert Frank, Forrest Davis, and Owen Rambow. 2017. Linguistically rich vector representations of supertags for TAG parsing. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 122–131, Umeå, Sweden. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What

works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483, Vancouver, Canada. Association for Computational Linguistics.

Johannes Heinecke, Jürgen Kunze, Wolfgang Menzel, and Ingo Schröder. 1998. Eliminative parsing with graded constraints. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics – Volume 1*, pages 526–530, Montreal, Quebec, Canada. Association for Computational Linguistics.

Gerhard Helbig and Wolfgang Schenkel. 1959. *Wörterbuch zur Valenz und Distribution deutscher Verben*. Bibliographisches Institut, Leipzig.

Thomas Herbst, David Heath, Ian F. Roe, and Dieter Götz. 2004. *A Valency Dictionary of English: A Corpus-Based Analysis of the Complementation Patterns of English Verbs, Nouns and Adjectives*, volume 40 of *Topics in English Linguistics*. De Gruyter Mouton, Berlin, Boston.

Donald Hindle and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Samar Husain, Raghu Pujitha Gade, and Rajeev Sangal. 2011. Linguistically rich graph based data driven parsing for Hindi. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 56–61, Dublin, Ireland. Association for Computational Linguistics.

Miloš Jakubıček and Vojtěch Kovář. 2013. Enhancing Czech parsing with verb valency frames. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 282–293, Samos, Greece. Springer.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. In *Handbook of formal languages, Volume 3: Beyond Words*, pages 69–124. Springer, New York.

Jungo Kasai, Robert Frank, R. Thomas McCoy, Owen Rambow, and Alexis Nasr. 2017. TAG parsing with neural networks and vector representations of supertags. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1712–1722, Copenhagen, Denmark. Association for Computational Linguistics.

Jungo Kasai, Robert Frank, Pauli Xu, William Merrill, and Owen Rambow. 2018. End-to-end graph-based TAG parsing with neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1181–1194. Association for Computational Linguistics.

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.

Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology – Volume 1*, pages 40–47, Edmonton, Canada. Association for Computational Linguistics.

Daniël de Kok, Jianqiang Ma, Corina Dima, and Erhard Hinrichs. 2017. PP attachment: Where do we stand? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 311–317, Valencia, Spain. Association for Computational Linguistics.

Jonathan K. Kummerfeld, David Hall, James R. Curran, and Dan Klein. 2012. Parser showdown at the Wall Street Corral: An empirical investigation of error types in parser output. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1048–1059, Jeju Island, Korea. Association for Computational Linguistics.

Mike Lewis, Kenton Lee, and Luke Zettlemoyer. 2016. LSTM CCG parsing. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 221–231, San Diego, California. Association for Computational Linguistics.

Mike Lewis and Mark Steedman. 2014. A* CCG parsing with a supertag-factored model. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 990–1000, Doha, Qatar. 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*, pages 63–70, Vancouver, Canada. Association for Computational Linguistics.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proceedings of the 28th Annual Meeting on Association for Computational Linguistics*, pages 31–38, Pittsburgh, Pennsylvania. Association for Computational Linguistics.

Seyed Abolghasem Mirroshandel and Alexis Nasr. 2016. Integrating selectional constraints and subcategorization frames in a dependency parser. *Computational Linguistics*, 42(1):55–90.

Seyed Abolghasem Mirroshandel, Alexis Nasr, and Joseph Le Roux. 2012. Semi-supervised dependency parsing using lexical affinities. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 777–785. Association for Computational Linguistics.

Seyed Abolghasem Mirroshandel, Alexis Nasr, and Benoît Sagot. 2013. Enforcing subcategorization constraints in a parser using sub-parses recombining. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 239–247. 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*, pages 807–814, Haifa, Israel.

Alexis Nasr and Owen Rambow. 2004. Supertagging and full parsing. In *Proceedings of the 7th International Workshop on Tree Adjoining Grammar and Related Formalisms*, pages 56–63, Vancouver, Canada.

Dominick Ng and James R. Curran. 2015. Identifying cascading errors using constraints in dependency parsing. 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 1148–1158, Beijing, China. Association for Computational Linguistics.

Takashi Ninomiya, Takuya Matsuzaki, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Extremely lexicalized models for accurate and fast HPSG parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 155–163, Sydney, Australia. Association for Computational Linguistics.

Joakim Nivre. 2005. Dependency grammar and dependency parsing. Technical Report MSI 05133, Växjö University, School of Mathematics and Systems Engineering.

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. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre and Chiao-Ting Fang. 2017. Universal dependency evaluation. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies*, pages 86–95, Gothenburg, Sweden. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Work-*

shop on Semantic Evaluation (SemEval 2015), pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Hiroki Ouchi, Kevin Duh, and Yuji Matsumoto. 2014. Improving dependency parsers with supertags. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 154–158, Gothenburg, Sweden. Association for Computational Linguistics.

Hiroki Ouchi, Kevin Duh, Hiroyuki Shindo, and Yuji Matsumoto. 2016. Transition-based dependency parsing exploiting supertags. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2059–2068.

Lilja Øvrelid and Joakim Nivre. 2007. When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 447–451, Borovets, Bulgaria.

Adam Pauls and Dan Klein. 2009. K-best A* parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 958–966, Suntec, Singapore. Association for Computational Linguistics.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2048, Vancouver, Canada. Association for Computational Linguistics.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Owen Rambow and Aravind Joshi. 1997. A formal look at dependency grammars and phrase-structure grammars, with special consideration of word-order phenomena. volume 39, pages 167–190. John Benjamins, Amsterdam and Philadelphia.

Adwait Ratnaparkhi. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 1–10, Providence, Rhode Island, USA.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *Proceedings of the 6th International Conference on Learning Representations*.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *Transactions of the Association for Computational Linguistics*, 4:127–140.

Siva Reddy, Oscar Täckström, Slav Petrov, Mark Steedman, and Mirella Lapata. 2017. Universal semantic parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 89–101, Copenhagen, Denmark. Association for Computational Linguistics.

Kenji Sagae and Alon Lavie. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, pages 691–698, Sydney, Australia. Association for Computational Linguistics.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1044–1050, Prague, Czech Republic. Association for Computational Linguistics.

Tianze Shi, Liang Huang, and Lillian Lee. 2017a. Fast(er) exact decoding and global training for transition-based dependency parsing via a minimal feature set. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 12–23, Copenhagen, Denmark. Association for Computational Linguistics.

Tianze Shi, Felix G. Wu, Xilun Chen, and Yao Cheng. 2017b. Combining global models for parsing Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 31–39, Vancouver, Canada. Association for Computational Linguistics.

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.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 818–827, Vancouver, Canada. Association for Computational Linguistics.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008:*

Proceedings of the Twelfth Conference on Computational Natural Language Learning, pages 159–177, Manchester, England. Coling 2008 Organizing Committee.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Lucien Tesnière. 1959. *Éléments de Syntaxe Structurale*. Librairie C. Klincksieck, Paris.

Damon Tutunjian and Julie E. Boland. 2008. Do we need a distinction between arguments and adjuncts? Evidence from psycholinguistic studies of comprehension. *Language and Linguistics Compass*, 2(4):631–646.

Ashish Vaswani and Kenji Sagae. 2016. Efficient structured inference for transition-based parsing with neural networks and error states. *Transactions of the Association for Computational Linguistics*, 4:183–196.

Wen Wang and Mary P. Harper. 2004. A statistical constraint dependency grammar (CDG) parser. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 42–49, Barcelona, Spain. Association for Computational Linguistics.

Jakub Waszczuk, Agata Savary, and Yannick Parmentier. 2016. Promoting multiword expressions in A* TAG parsing. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pages 429–439, Osaka, Japan. The COLING 2016 Organizing Committee.

Jakub Waszczuk, Agata Savary, and Yannick Parmentier. 2017. Multiword expression-aware A* TAG parsing revisited. In *Proceedings of the 13th International Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 84–93, Umeå, Sweden. Association for Computational Linguistics.

Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* CCG parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287, Vancouver, Canada. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. 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.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1566, Berlin, Germany. Association for Computational Linguistics.

Kai Zhao, James Cross, and Liang Huang. 2013. Optimal incremental parsing via best-first dynamic programming. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 758–768, Seattle, Washington, USA. Association for Computational Linguistics.