A Survey of Unsupervised Dependency Parsing

Wenjuan Han[1*] **Yong Jiang**[2*] **Hwee Tou Ng**[1]**, Kewei Tu**[3†]
[1]Department of Computer Science, National University of Singapore
[2]Alibaba DAMO Academy, Alibaba Group
[3]School of Information Science and Technology, ShanghaiTech University
dcshanw@nus.edu.sg
yongjiang.jy@alibaba-inc.com
nght@comp.nus.edu.sg
tukw@shanghaitech.edu.cn

Abstract

Syntactic dependency parsing is an important task in natural language processing. Unsupervised dependency parsing aims to learn a dependency parser from sentences that have no annotation of their correct parse trees. Despite its difficulty, unsupervised parsing is an interesting research direction because of its capability of utilizing almost unlimited unannotated text data. It also serves as the basis for other research in low-resource parsing. In this paper, we survey existing approaches to unsupervised dependency parsing, identify two major classes of approaches, and discuss recent trends. We hope that our survey can provide insights for researchers and facilitate future research on this topic.

1 Introduction

Dependency parsing is an important task in natural language processing that aims to capture syntactic information in sentences in the form of dependency relations between words. It finds applications in semantic parsing, machine translation, relation extraction, and many other tasks.

Supervised learning is the main technique used to automatically learn a dependency parser from data. It requires the training sentences to be manually annotated with their correct parse trees. Such a training dataset is called a *treebank*. A major challenge faced by supervised learning is that treebanks are not always available for new languages or new domains and building a high-quality treebank is very expensive and time-consuming.

There are multiple research directions that try to learn dependency parsers with few or even no syntactically annotated training sentences, including transfer learning, unsupervised learning, and semi-supervised learning. Among these directions, unsupervised learning of dependency parsers (a.k.a. *unsupervised dependency parsing* and dependency grammar induction) is the most challenging, which aims to obtain a dependency parser without using annotated sentences. Despite its difficulty, unsupervised parsing is an interesting research direction, not only because it would reveal ways to utilize almost unlimited text data without the need for human annotation, but also because it can serve as the basis for studies of transfer and semi-supervised learning of parsers. The techniques developed for unsupervised dependency parsing could also be utilized for other NLP tasks, such as unsupervised discourse parsing (Nishida and Nakayama, 2020). In addition, research in unsupervised parsing inspires and verifies cognitive research of human language acquisition.

In this paper, we conduct a survey of unsupervised dependency parsing research. We first introduce the definition and evaluation metrics of unsupervised dependency parsing, and discuss research areas related to it. Then we present in detail two major classes of approaches to unsupervised dependency parsing: generative approaches and discriminative approaches. Finally, we discuss important new techniques and setups of unsupervised dependency parsing that appear in recent years.

---

[*]Equal contributions.
[†]Corresponding author.

2 Background

2.1 Problem Definition

Dependency parsing aims at discovering the syntactic dependency tree $\mathbf{z}$ of an input sentence $\mathbf{x}$, where $\mathbf{x}$ is a sequence of words $x_1, \ldots, x_n$ with length n. A dummy root word x_0 is typically added at the beginning of the sentence. A dependency tree $\mathbf{z}$ is a set of directed edges between words that form a directed tree structure rooted at x_0. Each edge points from a parent word (also called a head word) to a child word.

In unsupervised dependency parsing, the goal is to obtain a dependency parser without using annotated sentences. Some work requires no training data and derives dependency trees from centrality or saliency information (Søgaard, 2012). We focus on learning a dependency parser from an unannotated dataset that consists of a set of sentences without any parse tree annotation. In many cases, part-of-speech (POS) tags of the words in the training sentences are assumed to be available during training.

Two evaluation metrics are widely used in previous work of unsupervised dependency parsing (Klein and Manning, 2004): directed dependency accuracy (DDA) and undirected dependency accuracy (UDA). DDA denotes the percentage of correctly predicted dependency edges, while UDA is similar to DDA but disregards the directions of edges when evaluating their correctness.

2.2 Related Areas

Supervised Dependency Parsing Supervised dependency parsing aims to train a dependency parser from training sentences that are manually annotated with their dependency parse trees. Generally, supervised dependency parsing approaches can be divided into graph-based approaches and transition-based approaches. A graph-based dependency parser searches for the best spanning tree of the graph that is formed by connecting all pairs of words in the input sentence. In the simplest form, a graph-based parser makes the first-order assumption that the score of a dependency tree is the summation of scores of its edges (McDonald et al., 2005). A transition-based dependency parser searches for a sequence of actions that incrementally constructs the parse tree, typically from left to right. While current start-of-the-art approaches have achieved strong results in supervised dependency parsing, their usefulness is limited to resource-rich languages and domains with many annotated datasets.

Cross-Domain and Cross-Lingual Parsing One useful approach to handling the lack of treebank resources in the target domain or language is to adapt a learned parser from a resource-rich source domain or language (Yu et al., 2015; McDonald et al., 2011; Ma and Xia, 2014; Duong et al., 2015). This is very related to unsupervised parsing as both approaches do not rely on treebanks in the target domain or language. However, unsupervised parsing is more challenging because it does not have access to any source treebank either.

Unsupervised Constituency Parsing Constituency parsing aims to discover a constituency tree of the input sentence in which the leaf nodes are words and the non-leaf nodes (nonterminal nodes) represent phrases. Unsupervised constituency parsing is often considered more difficult than unsupervised dependency parsing because it has to induce not only edges but also nodes of a tree. Consequently, there have been far more papers in unsupervised dependency parsing than in unsupervised constituency parsing over the past decade. More recently, however, there is a surge in interest in unsupervised constituency parsing and several novel approaches were proposed in the past two years (Li et al., 2020). While we focus on unsupervised dependency parsing in this paper, most of our discussions on the classification of approaches and recent trends apply to unsupervised constituency parsing as well.

Latent Tree Models with Downstream Tasks Latent tree models treat the parse tree as a latent variable that is used in downstream tasks such as sentiment classification. While no treebank is used in training, these models rely on the performance of the downstream tasks to guide the learning of the latent parse trees. To enable end-to-end learning, the REINFORCE algorithm and the Gumbel-softmax trick (Jang et al., 2017) can be utilized (Yogatama et al., 2016; Choi et al., 2018). There also exists previous work on latent dependency tree models that utilizes structured attention mechanisms (Kim et al., 2017)

for applications. Latent tree models differ from unsupervised parsing in that they utilize training signals from downstream tasks and that they aim to improve performance of downstream tasks instead of syntactic parsing.

3 General Approaches

3.1 Generative Approaches

3.1.1 Models

A generative approach models the joint probability of the sentence and the corresponding parse tree. Traditional generative models are mostly based on probabilistic grammars. To enable efficient inference, they typically make one or more relatively strict conditional independence assumptions. The simplest assumption (a.k.a. the context-free assumption) states that the generation of a token is only dependent on its head token and is independent of anything else. Such assumptions make it possible to decompose the joint probability into a product of component probabilities or scores, leading to tractable inference. However, they also lead to unavailability of useful information (e.g., context and generation history) in generating each token.

Based on their respective independence assumptions, different generative models specify different generation processes of the sentence and parse tree. Paskin (2002) and Carroll and Charniak (1992) choose to first uniformly sample a dependency tree skeleton and then populate the tokens (words) conditioned on the dependency tree in a recursive root-to-leaf manner. The generation of a child token is conditioned on the head token and the dependency direction. In contrast, Klein and Manning (2004) propose the Dependency Model with Valence (DMV) that generates the sentence and the parse tree simultaneously. Without knowing the dependency tree structure, each head token has to sample a decision (conditioned on the head token and the dependency direction) of whether to generate a child token or not before actually generating the child token. Besides, the generation of a child token in DMV is additionally conditioned on the valence, defined as the number of the child tokens already generated from a head token. Headden III et al. (2009) propose to also introduce the valence into the condition of decision sampling. Spitkovsky et al. (2012) additionally condition decision and child token generation on sibling words, sentence completeness, and punctuation context. Yang et al. (2020) propose a second-order extension of DMV that incorporates grandparent-child or sibling information. In addition to these generative dependency models, other grammar formalisms have also been used for unsupervised dependency parsing, such as tree substitution grammars (Blunsom and Cohn, 2010) and combinatory categorial grammars (Bisk and Hockenmaier, 2012; Bisk and Hockenmaier, 2013).

Similar tokens may have similar syntactic behaviors in a grammar. For example, all the verbs are very likely to generate a noun to the left as the subject. One way to capture this prior knowledge is to compute generation probabilities from a set of features that conveys syntactic similarity. Berg-Kirkpatrick et al. (2010) use a log-linear model based on manually-designed local morpho-syntactic features (e.g., whether a word is a noun) and Jiang et al. (2016) employ a neural network to automatically learn such features. Both approaches are based on DMV.

3.1.2 Inference

Given a model parameterized by Θ and a sentence $\mathbf{x}$, the model predicts the parse $\mathbf{z}^*$ with the highest probability.

$$\mathbf{z}^* = \arg \max_{\mathbf{z} \in \mathcal{Z}(\mathbf{x})} P(\mathbf{x}, \mathbf{z}; \Theta) \tag{1}$$

where $\mathcal{Z}(\mathbf{x})$ is the set of all valid dependency trees of the sentence $\mathbf{x}$. Due to the independence assumptions made by generative models, the inference problem can be efficiently solved exactly in most cases. For example, chart parsing can be used for DMV.

3.1.3 Learning Objective

Log marginal likelihood is typically employed as the objective function for learning generative models. It is defined on N training sentences $X = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(N)}\}$:

$$L(\Theta) = \sum_{i=1}^{N} \log \mathrm{P}(\mathbf{x}^{(i)}; \Theta) \tag{2}$$

where the model parameters are denoted by Θ. The likelihood of each sentence $\mathbf{x}$ is as follows:

$$P(\mathbf{x}; \Theta) = \sum_{\mathbf{z} \in \mathcal{Z}(\mathbf{x})} P(\mathbf{x}, \mathbf{z}; \Theta) \tag{3}$$

where $\mathcal{Z}(\mathbf{x})$ is the set of all valid dependency trees of sentence $\mathbf{x}$. As we mentioned earlier, the joint probability of a sentence and its dependency tree can be decomposed into the product of the probabilities of the components in the dependency tree.

Apart from the vanilla marginal likelihood, priors and regularization terms are often added into the objective function to incorporate various inductive biases. Smith and Eisner (2006) insert penalty terms into the objective to control dependency lengths and the root number of the parse tree. Cohen and Smith (2008; 2009) leverage logistic-normal prior distributions to encourage correlations between POS tags in DMV. Naseem et al. (2010) design a posterior constraint based on a set of manually-specified universal dependency rules. Gillenwater et al. (2011) add a posterior regularization term to encourage rule sparsity. The approaches of Spitkovsky et al. (2011b) can be seen as adding posterior constraints over parse trees based on punctuation. Tu and Honavar (2012) introduce an entropy term to prevent the model from becoming too ambiguous. Mareček and Žabokrtský (2012) insert a term that prefers reducible subtrees (i.e., their removal does not break the grammaticality of the sentence) in the parse tree. The same reducibility principle is used by Mareček and Straka (2013) to bias the decision probabilities in DMV. Noji et al. (2016) place a hard constraint in the objective that limits the degree of center-embedding of the parse tree.

3.1.4 Learning Algorithm

The Expectation-Maximization (EM) algorithm is typically used to optimize log marginal likelihood. For each sentence, the EM algorithm aims to maximize the following lower-bound of the objective function and alternates between the E-step and M-step.

$$\log P(\mathbf{x}; \Theta) - KL(Q(\mathbf{z}) \| P(\mathbf{z}|\mathbf{x}, \Theta)) \tag{4}$$

where $Q(\mathbf{z})$ is an auxiliary distribution with regard to $\mathbf{z}$. In the E-step, Θ is fixed and $Q(\mathbf{z})$ is set to $P(\mathbf{z}|\mathbf{x}, \Theta)$. A set of so-called expected counts can be derived from $Q(\mathbf{z})$ to facilitate the subsequent M-step and they are typically calculated using the inside-outside algorithm. In the M-step, Θ is optimized based on the expected counts with $Q(\mathbf{z})$ fixed.

There are a few variants of the EM algorithm. If $Q(\mathbf{z})$ represents a point-estimation (i.e., the best dependency tree has a probability of 1), the algorithm becomes hard-EM or Viterbi EM, which is found to outperform standard EM in unsupervised dependency parsing (Spitkovsky et al., 2010b). Softmax-EM (Tu and Honavar, 2012) falls between EM (considering all possible dependency trees) and hard-EM (only considering the best dependency tree), applying a softmax-like transformation to $Q(\mathbf{z})$. During the EM iterations, an annealing schedule (Tu and Honavar, 2012) can be used to gradually shift from hard-EM to softmax-EM and finally to the EM algorithm, which leads to better performance than sticking to a single algorithm. Lateen EM (Spitkovsky et al., 2011c) repeatedly alternates between EM and hard-EM, which is also found to produce better results than both EM and hard-EM.

Approaches with more complicated objectives often require more advanced learning algorithms, but many of the algorithms can still be seen as extensions of the EM algorithm that revise either the E-step (e.g., to update $Q(\mathbf{z})$ based on posterior regularization terms) or the M-step (e.g., to optimize the posterior probability that incorporates parameter priors).

		Intermediate Representation	Encoder	Decoder
Autoencoder	CRFAE (Cai et al., 2017)	Z	$P(\mathbf{z}|\mathbf{x})$	$P(\hat{\mathbf{x}}|\mathbf{z})$
	D-NDMV (Han et al., 2019a) Deterministic Variant	S	$P(\mathbf{s}|\mathbf{x})$	$P(\mathbf{z}, \hat{\mathbf{x}}|\mathbf{s})$
Variational Autoencoder	(Li et al., 2019)	Z	$P(\mathbf{z}|\mathbf{x})$	$P(\mathbf{z}, \mathbf{x})$
	D-NDMV (Han et al., 2019a) Variational Variant	S	$P(\mathbf{s}|\mathbf{x})$	$P(\mathbf{z}, \mathbf{x}|\mathbf{s})$
	(Corro and Titov, 2018)	Z	$P(\mathbf{z}|\mathbf{x})$	$P(\mathbf{x}|\mathbf{z})$

Table 1: Major approaches based on autoencoders and variational autoencoders for unsupervised dependency parsing. Z: dependency tree. S: continuous sentence representation. $\hat{\mathbf{x}}$ is a copy of $\mathbf{x}$ representing the reconstructed sentence. $\mathbf{z}$ is the dependency tree. $\mathbf{s}$ is the continuous representation of sentence $\mathbf{x}$.

In addition to the EM algorithm, the learning objective can also be optimized with gradient descent. Yang et al. (2020) recently observe that gradient descent can sometimes significantly outperform EM when learning neural DMV.

Better learning results can also be achieved by manipulating the training data. Spitkovsky et al. (2010a) apply curriculum learning to DMV training, which starts with only the shortest sentences and then progresses to increasingly longer sentences. Tu and Honavar (2011) provide a theoretical analysis on the utility of curriculum learning in unsupervised dependency parsing.

Spitkovsky et al. (2013) propose to treat different learning algorithms and configurations as modules and connect them to form a network. Some approaches discussed above, such as Lateen EM and curriculum learning, can be seen as special cases of this approach.

3.1.5 Pros and Cons

It is often straightforward to incorporate various inductive biases and manually-designed local features into generative approaches. Moreover, generative models can be easily trained via the EM algorithm and its extensions. On the other hand, generative models often have limited expressive power because of the independence assumptions they make.

3.2 Discriminative Approaches

Because of the limitation of generative approaches, more recently, researchers have paid more attention to discriminative approaches. Discriminative approaches model the conditional probability or score of the dependency tree given the sentence. By conditioning on the whole sentence, discriminative approaches are capable of utilizing not only local features (i.e., features related to the current dependency) but also global features (i.e., contextual features from the whole sentence) in scoring a dependency tree.

3.2.1 Autoencoder-Based Approaches

Autoencoder-based approaches aim to map a sentence to an intermediate representation (encoding) and then reconstruct the observed sentence from the intermediate representation (decoding). In the two existing autoencoder approaches (summarized in Table 1), the intermediate representation is the dependency tree and a continuous sentence vector respectively.

The reconstruction loss is typically employed as the learning objective function for autoencoder models. For a training dataset including N sentences $X = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$, the objective function is as follows:

$$L(\Theta) = \sum_{i=1}^{N} \log \mathrm{P}(\hat{\mathbf{x}}^{(i)}|\mathbf{x}^{(i)}; \Theta) \tag{5}$$

where Θ is the model parameter and $\hat{\mathbf{x}}^{(i)}$ is a copy of $\mathbf{x}^{(i)}$ representing the reconstructed sentence[1]. In some cases, there is an additional regularization term (e.g., L1) of Θ.

---

[1] In Han et al. (2019a), $\mathbf{x}$ is the word sequence, while $\hat{\mathbf{x}}$ is the POS tag sequence of the same sentence.

The first autoencoder model for unsupervised dependency parsing, proposed by Cai et al. (2017), is based on the conditional random field autoencoder framework (CRFAE). The encoder is a first-order graph-based discriminative dependency parser mapping an input sentence to the space of dependency trees. The decoder independently generates each token of the reconstructed sentence conditioned on the head of the token specified by the dependency tree. Both the encoder and the decoder are arc-factored, meaning that the encoding and decoding probabilities can be factorized by dependency arcs. Coordinate descent is applied to minimize the reconstruction loss and alternately updates the encoder parameters and the decoder parameters.

D-NDMV (Han et al., 2019a) (the deterministic variant) is the second autoencoder model proposed for unsupervised dependency parsing, in which the intermediate representation is a continuous vector representing the input sentence. The encoder is an LSTM summarizing the sentence with a continuous vector $\mathbf{s}$, while the decoder models the joint probability of the sentence and the dependency tree. More specifically, the decoder is a generative neural DMV that generates the sentence and its parse simultaneously, and its parameters are computed based on the continuous vector $\mathbf{s}$. The reconstruction loss is optimized using the EM algorithm. In the E-step, Θ is fixed and $Q(\mathbf{z})$ is set to $P(\mathbf{z}|\mathbf{x},\mathbf{s};\Theta)$. After we compute all the grammar rule probabilities given Θ, the inside-outside algorithm can be used to calculate the expected counts. In the M-step, Θ is optimized based on the expected counts with $Q(\mathbf{z})$ fixed.

3.2.2 Variational Autoencoder-Based Approaches

As mentioned in Section 3.1, the training objective of a generative model is typically the probability of the training sentence and the dependency tree is marginalized as a hidden variable. However, the marginalized probability cannot usually be calculated accurately for more complex models that do not make strict independence assumption. Instead, a variational autoencoder maximizes the Evidence Lower Bound (ELBO), a lower bound of the marginalized probability. Since the intermediate representation follows a distribution, different sampling approaches are used to optimize the objective function (i.e., likelihood) according to different model schema.

Three unsupervised dependency parsing models were proposed in recent years based on variational autoencoders (shown in Table 1). There are three probabilities involved in ELBO: the prior probability of the syntactic structure, the probability of generating the sentence from the syntactic structure (the decoder), and the variational posterior (the encoder) from the sentence to the syntactic structure.

Recurrent Neural Network Grammars (RNNG) (Dyer et al., 2016) is a transition-based constituent parser, with a discriminative and a generative variant. Discriminative RNNG incrementally constructs the constituency tree of the input sentence through three kinds of operations: generating a non-terminal token, shifting, and reducing. Generative RNNG replaces the shifting operation with a word generation operation and incrementally generates a constituency tree and its corresponding sentence. The probability of each operation is calculated by a neural network. Li et al. (2019) modify RNNG for dependency parsing and use discriminative RNNG and generative RNNG as the encoder and decoder of a variational autoencoder respectively. However, because RNNG has a strong expressive power, it is prone to overfitting in the unsupervised setting. Li et al. (2019) propose to use posterior regularization to introduce linguistic knowledge as a constraint in learning, thereby mitigating this problem to a certain extent.

The model proposed by Corro and Titov (2018) is also based on a variational autoencoder. It is designed for semi-supervised dependency parsing, but in principle it can also be applied for unsupervised dependency parsing. The encoder of this model is a conditional random field model while the decoder generates a sentence based on a graph convolutional neural network whose structure is specified by the dependency tree. Since the variational autoencoder needs Monte Carlo sampling to approximate the gradient and the complexity of sampling a dependency tree is very high, Corro and Titov (2018) use Gumbel random perturbation. Jang et al. (2017) use differentiable dynamic programming to design an efficient approximate sampling algorithm.

The variational variant of D-NDMV (Han et al., 2019a) has the same structure as the deterministic variant described in Section 3.2.1, except that the variational variant probabilistically models the intermediate continuous vector conditioned on the input sentence using a Gaussian distribution. It also specifies a Gaussian prior over the intermediate continuous vector.

3.2.3 Other Discriminative Approaches

Apart from the approaches based on autoencoder and variational autoencoder, there are also a few other discriminative approaches based on discriminative clustering (Grave and Elhadad, 2015), self-training (Le and Zuidema, 2015), or searching (Daumé III, 2009). Because of space limit, below we only introduce the approach based on discriminative clustering called Convex MST (Grave and Elhadad, 2015).

Convex MST employs a first-order graph-based discriminative parser. It searches for the parses of all the training sentences and learns the parser simultaneously, with a learning objective that the searched parses are close to the predicted parses by the parser. In other words, the parses should be easily predictable by the parser. The objective function can be relaxed to become convex and then can be optimized exactly.

3.2.4 Pros and Cons

Discriminative models are capable of accessing global features from the whole input sentence and are typically more expressive than generative models. On the other hand, discriminative approaches are often more complicated and do not admit tractable exact inference.

4 Recent Trends

4.1 Combined Approaches

Generative approaches and discriminative approaches have different pros and cons. Therefore, a natural idea is to combine the strengths of the two types of approaches to achieve better performance. Jiang et al. (2017) propose to jointly train two state-of-the-art models of unsupervised dependency parsing, the generative LC-DMV (Noji et al., 2016) and the discriminative Convex MST, with the dual decomposition technique that encourages the two models to gradually influence each other during training.

4.2 Neural Parameterization

Traditional generative approaches either directly learn or use manually-designed features to compute dependency rule probabilities. Following the recent rise of deep learning in the field of NLP, Jiang et al. (2016) propose to predict dependency rule probabilities using a neural network that takes as input the vector representations of the rule components such as the head and child tokens. The neural network can automatically learn features that capture correlations between tokens and rules. Han et al. (2019a) extend this generative approach to a discriminative approach by further introducing sentence information into the neural network in order to compute sentence-specific rule probabilities. Compared with generative approaches, it is more natural for discriminative approaches to use neural networks to score dependencies or parsing actions, so recent discriminative approaches all make use of neural networks (Li et al., 2019; Corro and Titov, 2018).

4.3 Lexicalization

In the most common setting of unsupervised dependency parsing, the parser is unlexicalized with POS tags being the tokens in the sentences. The POS tags are either human annotated or induced from the training corpus (Spitkovsky et al., 2011a; He et al., 2018). However, words with the same POS tag may have very different syntactic behavior and hence it should be beneficial to introduce lexical information into unsupervised parsers. Headden III et al. (2009), Blunsom and Cohn (2010), and Han et al. (2017) use partial lexicalization in which infrequent words are replaced by special symbols or their POS tags. Yuret (1998), Seginer (2007), Pate and Johnson (2016), and Spitkovsky et al. (2013) experiment with full lexicalization. However, because the number of words is huge, a major problem with full lexicalization is that the grammar becomes much larger and thus learning requires more data. To mitigate the negative impact of data scarcity, smoothing techniques can be used. For instance, Han et al. (2017) use neural networks to predict dependency probabilities that are automatically smoothed.

In principle, lexicalized approaches could also benefit from pretrained word embeddings, which capture syntactic and semantic similarities between words. Recently proposed contextual word embeddings

METHODS	≤ 10	ALL			
Generative Approaches			Generative Approaches (cont'd)		
Klein and Manning (2004)	46.2	34.9	Spitkovsky et al. (2011a)	-	59.1
Cohen et al. (2008)	59.4	40.5	Gimpel and Smith (2012)	64.3	53.1
Cohen and Smith (2009)	61.3	41.4	Tu and Honavar (2012)	71.4	57.0
Headden III et al. (2009)	68.8	-	Bisk and Hockenmaier (2012)	71.5	53.3
Spitkovsky et al. (2010a)	56.2	44.1	Spitkovsky et al. (2013)	72.0	64.4
Berg-Kirkpatrick et al. (2010)	63.0	-	Jiang et al. (2016)	72.5	57.6
Gillenwater et al. (2010)	64.3	53.3	Han et al. (2017)	75.1	59.5
Spitkovsky et al. (2010b)	65.3	47.9	He et al. (2018)*	60.2	47.9
Blunsom and Cohn (2010)	65.9	53.1	Discriminative Approaches		
Naseem et al. (2010)	71.9	-	Daumé III (2009)	-	45.4
Blunsom and Cohn (2010)	67.7	55.7	Le and Zuidema (2015) †	73.2	65.8
Spitkovsky et al. (2011c)	-	55.6	Cai et al. (2017)	71.7	55.7
Spitkovsky et al. (2011b)	69.5	58.4	Li et al. (2019)	54.7	37.8
			Han et al. (2019a)	75.6	61.4

Table 2: Reported directed dependency accuracies on section 23 of the WSJ corpus, evaluated on sentences of length ≤ 10 and all lengths. *: without gold POS tags. †: with more training data in addition to WSJ.

(Devlin et al., 2019) are even more informative, capturing contextual information. However, word embeddings have not been widely used in unsupervised dependency parsing. One concern is that word embeddings are too informative and may make unsupervised models more prone to overfitting. One exception is He et al. (2018), who propose to use invertible neural projections to map word embeddings into a latent space that is more amenable to unsupervised parsing.

4.4 Big Data

Although unsupervised parsing does not require syntactically annotated training corpora and can theoretically use almost unlimited raw texts for training, most of the previous work conducts experiments on the WSJ10 corpus (the Wall Street Journal corpus with sentences no longer than 10 words) containing no more than 6,000 training sentences. There are a few papers that try to go beyond such a small training corpus. Pate and Johnson (2016) use two large corpora containing more than 700k sentences. Mareček and Straka (2013) utilize a very large corpus based on Wikipedia in learning an unlexicalized dependency grammar. Han et al. (2017) use a subset of the BLLIP corpus that contains around 180k sentences. With the advancement of computing power and deep neural models, we expect to see more future work on training with big data.

4.5 Unsupervised Multilingual Parsing

To tackle the lack of supervision in unsupervised dependency parsing, some previous work considers learning models of multiple languages simultaneously (Berg-Kirkpatrick and Klein, 2010; Liu et al., 2013; Jiang et al., 2019; Han et al., 2019b). Ideally, these models can learn from each other by identifying shared syntactic behaviors of different languages, especially those in the same language family. For example, Berg-Kirkpatrick and Klein (2010) propose to utilize the similarity of different languages defined by a phylogenetic tree and learn several dependency parsers jointly. Han et al. (2019b) propose to learn a unified multilingual parser with language embeddings as input. Jiang et al. (2019) propose to guide the learning process of unsupervised dependency parser from the knowledge of another language by using three types of regularization to encourage similarity between model parameters, dependency edge scores, and parse trees respectively.

5 Benchmarking on the WSJ Corpus

Most papers of unsupervised dependency parsing report the accuracy of their approaches on the test set of the Wall Street Journal (WSJ) corpus. We list the reported accuracy on WSJ in Table 2. It must be emphasized that the approaches listed in this table may use different training sets and different external

knowledge in their experiments, and one should check the corresponding papers to understand such differences before comparing these accuracies.

While the accuracy of unsupervised dependency parsing has increased by over thirty points in the last fifteen years, it is still well below that of supervised models, which leaves much room for improvement and challenges for future research.

6 Future Directions

6.1 Utilization of Syntactic Information in Pretrained Language Modeling

Pretrained language modeling (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019), as a new NLP paradigm, has been utilized in various areas including question answering, machine translation, grammatical error correction, and so on. Pretrained language models leverage a large-scale corpus for pretraining and then small data sets of specific tasks for finetuning, reducing the difficulty of downstream tasks and boosting their performance. Current state-of-the-art approaches on supervised dependency parsing, such as Zhou and Zhao (2019), adopt the new paradigm and benefit from pretrained language modeling. However, pretrained language models have not been widely used in unsupervised dependency parsing. One major concern is that pretrained language models are too informative and may make unsupervised models more prone to overfitting. Besides, massive syntactic and semantic information is encoded in pretrained language models and how to extract the syntactic part from them is a challenging task.

6.2 Inspiration for Other Tasks

Unsupervised dependency parsing is a classic unsupervised learning task. Many techniques developed for unsupervised dependency parsing can serve as the inspiration for studies of other unsupervised tasks, especially unsupervised structured prediction tasks. A recent example is Nishida and Nakayama (2020), who study unsupervised discourse parsing (inducing discourse structures for a given text) by borrowing techniques from unsupervised parsing such as Viterbi EM and heuristically designed initialization.

Unsupervised dependency parsing techniques can also be used as building blocks for transfer learning of parsers. Some of the approaches discussed in this paper have already been applied to cross-lingual parsing (He et al., 2019; Li and Tu, 2020), and more such endeavors are expected in the future.

6.3 Interpretability

One prominent problem of deep neural networks is that they act as black boxes and are generally not interpretable. How to improve the interpretability of neural networks is a research topic that gains much attention recently. For natural language texts, their linguistic structures reveal important information of the texts and at the same time can be easily understood by human. It is therefore an interesting direction to integrate techniques of unsupervised parsing into various neural models of NLP tasks, such that the neural models can build their task-specific predictions on intermediate linguistic structures of the input text, which improves the interpretability of the predictions.

7 Conclusion

In this paper, we present a survey on the current advances of unsupervised dependency parsing. We first motivate the importance of the unsupervised dependency parsing task and discuss several related research areas. We split existing approaches into two main categories, and explain each category in detail. Besides, we discuss several recent trends in this research area. While there is a growing body of work that improves unsupervised dependency parsing, its performance is still below that of supervised dependency parsing by a large margin. This suggests that more investigation and research are needed to make unsupervised parsers useful for real applications. We hope that our survey can promote further development in this research direction.

Acknowledgments

References

Taylor Berg-Kirkpatrick and Dan Klein. 2010. Phylogenetic grammar induction. In *ACL*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *NAACL*.

Yonatan Bisk and Julia Hockenmaier. 2012. Simple robust grammar induction with combinatory categorial grammars. In *AAAI*.

Yonatan Bisk and Julia Hockenmaier. 2013. An HDP model for inducing combinatory categorial grammars. *TACL*.

Phil Blunsom and Trevor Cohn. 2010. Unsupervised induction of tree substitution grammars for dependency parsing. In *EMNLP*.

Jiong Cai, Yong Jiang, and Kewei Tu. 2017. CRF autoencoder for unsupervised dependency parsing. In *EMNLP*.

Glenn Carroll and Eugene Charniak. 1992. Two experiments on learning probabilistic dependency grammars from corpora. Technical report, Department of Computer Science, Brown University.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Unsupervised learning of task-specific tree structures with tree-lstms. In *AAAI*.

Shay B Cohen and Noah A Smith. 2009. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *NAACL*.

Shay B Cohen, Kevin Gimpel, and Noah A Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. In *NIPS*.

Caio Corro and Ivan Titov. 2018. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. In *ICLR*.

Hal Daumé III. 2009. Unsupervised search-based structured prediction. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Cross-lingual transfer for unsupervised dependency parsing without parallel data. In *CoNLL*.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *NAACL*.

Jennifer Gillenwater, Kuzman Ganchev, Joao Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *ACL*.

Jennifer Gillenwater, Kuzman Ganchev, Fernando Pereira, Ben Taskar, et al. 2011. Posterior sparsity in unsupervised dependency parsing. *Journal of Machine Learning Research*.

Kevin Gimpel and Noah A Smith. 2012. Concavity and initialization for unsupervised dependency parsing. In *NAACL*.

Edouard Grave and Noémie Elhadad. 2015. A convex and feature-rich discriminative approach to dependency grammar induction. In *ACL-IJCNLP*.

Wenjuan Han, Yong Jiang, and Kewei Tu. 2017. Dependency grammar induction with neural lexicalization and big training data. In *EMNLP*.

Wenjuan Han, Yong Jiang, and Kewei Tu. 2019a. Enhancing unsupervised generative dependency parser with contextual information. In *ACL*.

Wenjuan Han, Ge Wang, Yong Jiang, and Kewei Tu. 2019b. Multilingual grammar induction with continuous language identification. In *EMNLP*.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2018. Unsupervised learning of syntactic structure with invertible neural projections. In *EMNLP*.

Junxian He, Zhisong Zhang, Taylor Berg-Kirkpatrick, and Graham Neubig. 2019. Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *ACL*.

William P Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *NAACL*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumble-softmax. In *ICLR*.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2016. Unsupervised neural dependency parsing. In *EMNLP*.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2017. Combining generative and discriminative approaches to unsupervised dependency parsing via dual decomposition. In *EMNLP*.

Yong Jiang, Wenjuan Han, and Kewei Tu. 2019. A regularization-based framework for bilingual grammar induction. In *EMNLP-IJCNLP*.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. Structured attention networks. In *ICLR*.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *ACL*.

Phong Le and Willem Zuidema. 2015. Unsupervised dependency parsing: Let's use supervised parsers. In *NAACL*.

Zhao Li and Kewei Tu. 2020. Unsupervised cross-lingual adaptation of dependency parsers using CRF autoencoders. In *In Findings of EMNLP*.

Bowen Li, Jianpeng Cheng, Yang Liu, and Frank Keller. 2019. Dependency grammar induction with a neural variational transition-based parser. In *AAAI*.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020. An empirical comparison of unsupervised constituency parsing methods. In *ACL*.

Kai Liu, Yajuan Lü, Wenbin Jiang, and Qun Liu. 2013. Bilingually-guided monolingual dependency grammar induction. In *ACL*.

Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *ACL*.

David Mareček and Milan Straka. 2013. Stop-probability estimates computed on a large corpus improve unsupervised dependency parsing. In *ACL*.

David Mareček and Zdeněk Žabokrtský. 2012. Exploiting reducibility in unsupervised dependency parsing. In *EMNLP-IJCNLP*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *EMNLP*.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *EMNLP*.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *EMNLP*.

Noriki Nishida and Hideki Nakayama. 2020. Unsupervised discourse constituency parsing using Viterbi EM. *TACL*, 8:215–230.

Hiroshi Noji, Yusuke Miyao, and Mark Johnson. 2016. Using left-corner parsing to encode universal structural constraints in grammar induction. In *EMNLP*.

Mark A Paskin. 2002. Grammatical bigrams. In *NIPS*.

John K Pate and Mark Johnson. 2016. Grammar induction from (lots of) words alone. In *COLING*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391.

Noah A Smith and Jason Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *ACL*.

Anders Søgaard. 2012. Unsupervised dependency parsing without training. *Natural Language Engineering*, 18(2):187–203.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2010a. From baby steps to leapfrog: How "less is more" in unsupervised dependency parsing. In *NAACL*.

Valentin I Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D Manning. 2010b. Viterbi training improves unsupervised dependency parsing. In *CoNLL*.

Valentin I Spitkovsky, Hiyan Alshawi, Angel X Chang, and Daniel Jurafsky. 2011a. Unsupervised dependency parsing without gold part-of-speech tags. In *EMNLP*.

Valentin I Spitkovsky, Hiyan Alshawi, and Dan Jurafsky. 2011b. Punctuation: Making a point in unsupervised dependency parsing. In *CoNLL*.

Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011c. Lateen EM: unsupervised training with multiple objectives, applied to dependency grammar induction. In *EMNLP*.

Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2012. Three dependency-and-boundary models for grammar induction. In *EMNLP-CoNLL*.

Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *EMNLP*.

Kewei Tu and Vasant Honavar. 2011. On the utility of curricula in unsupervised learning of probabilistic grammars. In *IJCAI*.

Kewei Tu and Vasant Honavar. 2012. Unambiguity regularization for unsupervised learning of probabilistic grammars. In *EMNLP-CoNLL*.

Songlin Yang, Yong Jiang, Wenjuan Han, and Kewei Tu. 2020. Second-order unsupervised neural dependency parsing. In *COLING*.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2016. Learning to compose words into sentences with reinforcement learning. In *ICLR*.

Juntao Yu, Mohab El-karef, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *the 14th International Conference on Parsing Technologies*.

Deniz Yuret. 1998. Discovery of linguistic relations using lexical attraction. *arXiv preprint cmp-lg/9805009*.

Junru Zhou and Hai Zhao. 2019. Head-driven phrase structure grammar parsing on Penn Treebank. In *ACL*.