# Weakly supervised parsing with rules

Christophe Cerisara, Alejandra Lorenzo, Pavel Kral

# Weakly supervised parsing with rules

*C. Cerisara*[1], *A. Lorenzo*[1] *and P. Kral*[2, 3]

[1]LORIA-UMR7503, Nancy, France
[2]Dept of Computer Science and Engineering, Univ. of West Bohemia, Plzeň, Czech Republic
[3]NTIS - New Technologies for the Information Society, Univ. of West Bohemia, Plzeň, Czech Rep.
cerisara@loria.fr, alelorenzo@gmail.com, pkral@kiv.zcu.cz

## Abstract

This work proposes a new research direction to address the lack of structures in traditional n-gram models. It is based on a weakly supervised dependency parser that can model speech syntax without relying on any annotated training corpus. Labeled data is replaced by a few hand-crafted rules that encode basic syntactic knowledge. Bayesian inference then samples the rules, disambiguating and combining them to create complex tree structures that maximize a discriminative model's posterior on a target unlabeled corpus. This posterior encodes sparse selectional preferences between a head word and its dependents. The model is evaluated on English and Czech newspaper texts, and is then validated on French broadcast news transcriptions.
**Index Terms**: speech parsing, unsupervised training, inference

## 1. Introduction

N-gram models have several well-known theoretical limitations, most notably regarding the fact that they reduce natural language syntax to a small linear context. Despite these limitations, most, if not all, current state-of-the-art automatic speech recognition systems are based on n-gram models. Several interesting alternative have been proposed that try to introduce structure back into such systems. Probably the most well-known is the Structured Language Model of Chelba and Jelinek [1]. However, these approaches have so far not been able to outperform the n-grams by a large-enough margin to make them become the new mainstream type of language models. The reason for this might come, partially, from both the lack of robustness of current parsers to speech recognition errors and the difficulty to accurately model speech syntax, which does not benefit from as large treebanks as written text in many languages. Thus, we believe that a solution to these issues would be to further investigate the weakly supervised machine learning area and how it could be used to replace corpus annotations by other types of knowledge that shall guide the unsupervised training of parsers designed for speech. Our long-term objective is thus to design accurate parsers for speech transcriptions that would not rely on large annotated treebanks and still would be good enough to be integrated into speech recognition systems thus compensating the lack of structures of current n-gram models.

This work describes the first part of this ambitious goal, that is the proposal of a new weakly supervised parser for speech transcripts that exploits a few hand-crafted rules as a substitute to corpus annotations. We first review the litterature about unsupervised training in Section 2, and then describe our model in Section 3. We then validate experimentally the proposed model in Section 5, first on written text corpora for English and Czech, in order to show that the approach can be used to train new models for different languages at a low cost, and finally on French speech transcripts. The integration of this model into a speech recognition system is another challenging task that is left for future work.

## 2. Related works

Unsupervised parsing aims at automatically producing syntactic trees on top of a raw, unlabeled text corpus. Many amongst the most successful approaches in the field [2, 3, 4, 5, 6] exploit some stochastic process to decompose parents-children dependencies. The parameters of these models are typically trained so as to maximize the sparsity of selectional preferences in the corpus. Although no manual annotations are given, one can argue that some "linguistic" knowledge is nevertheless introduced in the model's definition, for instance in the set of conditional independencies, priors and initial parameters. It has further been shown that adding some kind of knowledge might greatly improve the performances of unsupervised parsers and help to control their convergence and the resulting structures.

Hence, [7] exploit phylogenetic dependencies between human languages, [8] replace standard corpus annotations with a few syntactic prototypes and [9] make use of semantic cues. The posterior regularization framework [10] is often used to integrate constraints during inference, e.g., with a sparsity-inducing bias over unique dependency types [11] or with a few universal rules that are valid across languages [5].

Most of these works rely on a generative Bayesian model because of fundamental theoretical limitations concerning unsupervised training of discriminative models. Nevertheless, as discussed in Section 3.2, using knowledge to constrain inference makes the training of discriminative models possible even without any supervised annotation [12]. Several different approaches have thus been proposed to train discriminative parsers on unlabeled corpora, for instance by transferring dependency grammars from English to other languages within the posterior regularization framework and with discriminative models [13] or by defining preferred dependency constraints within the generalized expectation framework with tree CRFs [14]. Semi-supervised discriminative parsers can also make a very efficient use of even a few annotated sentences, such as in the SEARN paradigm [15, 16]. Our work has also been inspired by the "Constraint-Driven Learning" paradigm, as proposed in [17, 18] and generalized in [19], as well as with [20, 8].

## 3. Proposed framework

We propose a weakly supervised approach that relies on two main components: a set of rules that generate dependency structures

over input sentences annotated with part-of-speech (POS) tags [1], and a model that evaluates the trees that are produced by the rules on some raw text corpus [2]. The rules shall describe, for every possible dependency type (such as subject, object...), the most standard situations in which this relation may occur.

### 3.1. Rules design process

The proposed framework was originally designed to be used with hand-crafted rules. However, validating the framework only with manually defined rules may leave some doubts about two potentially problematic aspects: first, the difficulty to reproduce our experimental results, because of the subjectivity in the rules design process. Hence, different users will most likely write different rules for the same task; second, excessive tuning to the task, as it is always possible to improve the results by writing more rules, or fine-tuning them.

To address both issues, we propose next to automatically train and extract the rules from a small labeled corpus. Thereafter, we further validate our approach with hand-crafted rules, to support our original motivation that is to develop a parser without any annotated corpus.

#### *3.1.1. Automatically trained rules*

In order to automatically extract our set of rules from a small labeled corpus $\mathcal{C}$, we first define the parametric form of every rule as $R(u, w, h, d, s_R, \mathcal{C})$, where $R$ is the rule that creates one and only one dependency arc with label $d$ from word $w$ of sentence $u$ to the head word $h$ of sentence $u$. The fifth parameter is the score $s_R$ that represents the level of confidence of this rule: the larger $s_R$ is the more chance has the rule to be correct. This score is computed with a Support Vector Machine (SVM) that is trained on $\mathcal{C}$. The SVM uses the same basic features as the ones used in the MATE parser [22]. An example of a feature is the tuple ($d$,form($h$),postag($h$),word_order($w$,$h$)). The full set of features used in our classifier includes all the first-order features [3] listed in Table 4 in [22].

At test time, all possible rules that link every pair of words with every possible dependency labels are first built for each input sentence. Then, the score $s_R$ of every rule is computed with the SVM, and all rules which score is below 0 are removed. The inference algorithm described in Section 3.3 is then applied with these rules, just as it is done with the manual rules, with the exception of two minor differences, which both come from the fact that there are much more automatic rules (up to 4000 rules per sentence) than manual rules (up to 40 rules instances per sentence), leading to a much larger search space. We have thus slightly adapted the proposed inference algorithm to accommodate this increased search space by choosing as initial configuration the trees produced by the MATE parser, which has also been trained on $\mathcal{C}$.

The performances of both the initial MATE model and the proposed model with automatically trained rules are shown in Table 1.

#### *3.1.2. Manually designed rules*

The proposed framework supports many types of rules, which shall only take as input a sequence of words, check that some preconditions are met in the input, such as the existence of a noun followed by a verb, and output new annotations on the sentence, such as a subject relation between the noun and the verb.

The user has thus a lot of freedom in the types of rules he may write, and even correlated, ambiguous, incomplete and logically inconsistent rules are allowed, thanks to Bayesian inference that filters-out irrelevant rules. For instance, on the one hand, the user could write a single rule that links any word to any other word with any dependency label. This situation reflects the purely unsupervised case [4]. On the other hand, the user could instead write a large set of so precise rules that the correct parse trees can be derived from them without ambiguity, leading to an ideal rule-based deterministic parser, in which case the proposed Bayesian model is useless. The current work rather aims at some intermediate stage, where the user should write one or a few rules per dependency type, which, when combined, lead to relatively ambiguous parses, and where Bayesian inference should take care of resolving ambiguity to find the correct tree. The potential of the proposed method to support such unconstrained and intuitive rules makes the proposed approach unique in the field.

Given these general guidelines, we have implemented a "rule definition language" that extends traditional regular expressions to manipulate tree-like structures. This allows the user to write a simple text file with regular expressions to match some tree or sequence patterns and produce new dependency arcs. When these regular expressions are not expressive enough for the user, he can directly implement the rule interface in Java code, and thus manipulate the dependency tree structure the way he wants. In the following French and English experiments, the rules are defined using both formats, while only regular expressions are used for Czech.

In order to decide which rules he shall write, the user may use an existing annotation guide or examples of annotated sentences. In the following experiments, as it is difficult to master the three English, French and Czech annotation guides, we have rather given the first 50 labeled sentences per language as examples to the rules designer. Although he actually used only a small fraction of these annotations, we have nevertheless compared the resulting system with other semi-supervised approaches that exploit twice as much annotated sentences.

### 3.2. Scoring model

The second component of the framework is the model, which scores the full set of trees produced by the rules on the corpus. This score reflects two linguistic criteria: the sparsity of lexical preferences and lists of dependents for a given head word [5]. The search algorithm, described in Section 3.3, then looks for the best sequences of rules, one per sentence of the corpus, that maximize the global model's score. Note that the size of the search space depends on the level of ambiguity of the rules set.

In the following, the scoring model is implemented as a discriminative directed graphical model, shown in Figure 1. Classically, generative models are used in unsupervised systems, because discriminative models cannot learn from unlabeled data [6]. However, our model is not purely unsupervised, because of the rules that constrain the values that the latent variables can take, leading to a *weakly supervised* training algorithm.

In Figure 1, $R_u$ is the main latent variable; it is the se-

[1] In the following English and Czech experiments, the gold POS-tags are considered, while in the French experiments, they are automatically computed with the Treetagger [21]

[2] Initially, the corpus is unlabeled, and the only supervision considered in this work comes from the rules.

[3] i.e., excluding grandparent and siblings features

[4] This case requires to use a generative model instead of the discriminative model proposed in Section 3.2

[5] The DMV model [2] uses similar criteria. Another choice may be to maximize sparsity of recurrent elementary trees, such as in [4]

[6] At least in a theoretically purely unsupervised setting without constraints. See [12] for a discussion and some solutions.

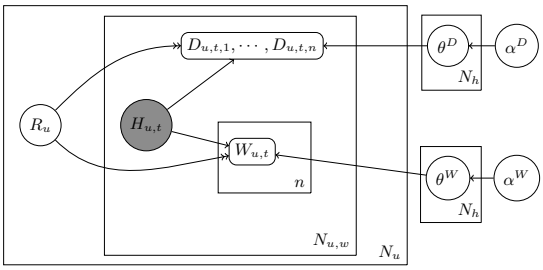



Figure 1: Plate diagram of the scoring model. Shaded nodes are observed and double arrows represent deterministic relations. $N_u$ is the number of utterances in the corpus, $N_{u,w}$ is the number of words in utterance $u$, $N_h$ is the size of the vocabulary and $n$ is the constant (per head word) maximum number of dependents pre-computed over all possible rules sequences.

quence of rules currently applied onto utterance $u$. $H_{u,t}$ is the only observed variable; its value is the $t^{th}$ word of utterance $u$. $D_{u,t} = (D_{u,t,1}, \cdots, D_{u,t,n})$ represents the resulting syntactic frame of $H_u$, i.e., the ordered list of dependency types governed by $H_u$ and produced by the rules sequence $R_u$. $W_{u,t}$ encodes lexical preferences, i.e., all words governed by $H_{u,t}$ produced by $R_u$. The prior of $H_{u,t}$ is assumed uniform, and the only parameters are thus the multinomial parameters $\theta^D$ and $\theta^W$, which have respective symmetric Dirichlet priors $\alpha^D$ and $\alpha^W$. Their concentration parameter is arbitrarily set to 0.001 for both in all experiments.

### 3.3. Inference

Following standard practice, we perform inference using a Collapsed Gibbs sampler, where the model parameters, $\theta^D$ and $\theta^W$, are marginalized out. In each iteration, we want to sample a sequence of rules $R_u$ for each sentence $u$ in turn. In this sampling process, the $R_u$ variable is not decomposed into each of the individual rules that form this sequence, because this may lead to very slow mixing chains, for instance in cases where several rules in the sequence have to be permuted to jump from one mode of the posterior to another. Therefore, besides Gibbs, we still face the challenge of sampling a full sequence of rules per sentence. This may be achieved in several ways. The solution used in the following experiments explores the full search tree of all possible rules permutations in a depth-first manner, which is made possible thanks to the limited length of sentences and to an aggressive pruning based on a topological score that favors projective trees. For longer sentences, $R_u$ sampling may be approximated with an inner loop of Metropolis sampling, which would reject samples that make $P(R_u|R_{-u}, H)$ decrease. Some local hill-climbing search may also be considered to speed-up convergence towards a local optimum, eventually with multiple random restarts or simulated annealing.

## 4. Experimental validation

All experiments exploit the very same model, with the same $\alpha$ parameters, but of course with different rules sets. The validation procedure consists of first removing the punctuation [7] from all sentences of the corpus, as it is common in the other works we compare to; second, filtering out all sentences that are strictly longer than 10 (for English and Czech) or 15 words (for French); third, initializing the dependency trees by applying all applicable rules in a random order [8]; fourth, running 5000 iterations of Gibbs sampling. The corpus trees that give the highest log-posterior probability are chosen, and the "test" subset of these trees is compared against the gold corpus to compute the standard CoNLL Labeled (LAS) and Unlabeled (UAS) Attachment Scores metrics. The LAS is the ratio of correct head attachments with correct dependency label, while the UAS is the accuracy of head attachments, independently of their labels.

In order to limit the issue of data sparsity during inference, $W_{u,t}$ takes as value the inflected form for words that occur more than 50 times in the corpus and their POS-tag otherwise; the domain of $D_{u,t}$ also only considers a few amongst all possible dependency types: (OBJ, NMOD, PRD, SBJ, VC) for English, (Sb, Obj, Pred, AuxV and AuxT) for Czech, and (AUX, DET, OBJ, SUJ and POBJ) for French.

### 4.1. English evaluations

Our validation corpus for English is derived from the Penn Treebank [23], after removing all punctuation marks and filtering out all sentences that are strictly longer than 10 words, leading to the standard WSJ10 corpus. The LAS and UAS are reported on Section 23 of this corpus. The first 100 sentences of Sections 2 to 21 are extracted and used with their gold dependency tree to train the MATE parser and to extract our set of automatic rules (see 3.1.1). The rest of Sections 2 to 21 are merged with Section 23 to perform inference, after all dependency trees have been deleted from this corpus.

The last four rows in Table 1 report performances respectively for (i) a baseline that applies the manual rules in a random order; (ii) the proposed system with manual rules; (iii) a baseline formed by the supervised MATE parser trained on the first 100 sentences; (iv) the proposed system with automatic rules trained on the same first 100 sentences and combined with Bayesian inference.

| | UAS [%] | LAS [%] |
|---|---|---|
| DMV (no rules) | 47.1 | - |
| Improved DMV (Headden) | 68.8 | - |
| TSG-DMV (Cohn) | 66.4 | - |
| Phylogenetic (Berg-Kirkpatrick) | 62.3 | - |
| Posterior regularization (Druck) | 61.3 | - |
| Post. reg. Universal rules (Naseem) | 71.9 | - |
| Post. reg. Collins rules (Naseem) | 73.8 | - |
| SEARN 10 sentences (Daumé) | $\sim 72$ | - |
| SEARN 100 sentences (Daumé) | $\sim 75$ | - |
| SEARN 1000 sentences (Daumé) | $\sim 78$ | - |
| **Random rules order (10 runs avg)** | 71.1 | 50.7 |
| **Bayesian inference (5000 iters)** | **75.6** | 57.0 |
| **MATE parser trained on 100 sent.** | 73.4 | 64.9 |
| **Bayes. inf. with automatic rules** | 75.0 | **65.9** |

Table 1: Dep. parsers on the WSJ10 corpus. The confidence interval is $\pm 0.4\%$.

The state-of-the-art results presented in Table 1 come from

[7] Arcs below a punctuation mark are recursively moved up to rather attach to the first head word that is not a punctuation.

[8] Or, for the automatically trained rules, with the trees produced by the MATE parser that has been trained on the first 100 labeled sentences of the training corpus

DMV [2], Improved DMV [3], TSG-DMV [4], Phylogenetic [7], Posterior regularization [14], Post. reg. Universal and Collins rules [5] and SEARN [15].

22 rules have been used in these experiments[9]. This experiment validates the proposed approach on a standard English corpus, and shows that it obtains good results as compared to the state-of-the-art. Our weakly supervised model obtains the best UAS scores with 22 manual rules only, while the best LAS scores are obtained by our model with rules automatically extracted from 100 annotated sentences. The large difference in LAS scores between manual and automatic rules is due to the limited number of manual rules, which cover only 19 out of the 44 dependency types in the WSJ10.

### 4.2. Czech evaluations

Our Czech evaluation corpus is extracted from the training part of the Prague Dependency Treebank(PDT) [24], with punctuations removed and sentences longer than 10 words filtered out. The remaining corpus, composed of 140 Kwords in 24.5 Ksentences, has further been split into a training (80%) and a test part (20%) in order to match the experimental conditions in [11]. Bayesian inference is realized on the joint train and test corpus, and performances are computed on the test part only. The PDT contains about 2% of non-projective arcs. Although our pruning strategy favors projective trees (see Section 3.3), it does allow crossing dependencies and the rules provide enough constraints to output non-projective trees. Hence, we have observed 2.5% of non-projective arcs in the trees produced by our model. Table 2 reports the obtained results and compare them with the system described in [11], which is an unsupervised DMV-based approach that is trained with additional constraints for dependency type sparsity. The proposed system gives very competitive results with only 14 simple rules.

| System | UAS [%] | LAS [%] |
|---|---|---|
| DMV (EM algorithm, no rules) | 29.6 | - |
| E-DMV (EM-(3,3)) | 48.9 | - |
| Posterior regularization (Gillenwater) | 55.5 | - |
| **Random rules order (10 runs avg)** | 57.3 | 48.1 |
| **Bayesian inference (5000 iters)** | 58.8 | 49.4 |

Table 2: Dep. parsers on the Czech PDT corpus. The confidence interval is $\pm 0.57\%$.

### 4.3. French evaluations

Although previous semi-supervised parsers have been proposed for French written texts [25], there is no semi-supervised state-of-the-art results to compare with for French broadcast news. We thus compare in Table 3 the proposed model with the supervised MATE parser trained on the 50,000 words that form the training corpus of the Ester Treebank [26]. The Ester Treebank is the only corpus available annotated with dependencies and composed of broadcast news manual speech transcriptions in French. After filtering out all utterances longer than 15 words, the total corpus size on which Bayesian inference is applied is 16,000 words, while the gold contains 1309 words, which leads to a much larger statistical confidence interval than in English.

Although our model's performances are still well below those of a fully trained supervised parser, they give encouraging results without relying on any annotated corpus. We further expect better results by dedicating more time to writing new rules.

[9]We only list next the French rules, because of paper size. English and Czech rules will be made available with the software

| System | UAS [%] | LAS [%] |
|---|---|---|
| **Random rules order (10 runs avg)** | 61.2 | 57.0 |
| **Bayesian inference (5000 iters)** | 67.2 | 62.6 |
| Supervised **MATE** | 83.3 | 78.2 |

Table 3: Experimental results on the French broadcast news corpus. The confidence interval is $\pm 2.48\%$.

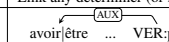
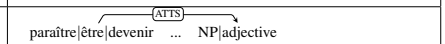
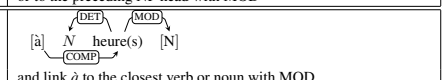
| | Rule |
|---|---|
| Root | Any verb or NP head can be the root of the utterance. |
| DET | Link any determiner (or number) to the NP head with DET. |
| AUX |  |
| ATTS |  |
| OBJ | Link with OBJ any NP head or VER:infi or relative pronoun to the preceding verb, or any personal pronoun to the following verb. |
| SUJ | Link any pronoun or NP head to the next verb with SUJ |
| COMP | Link any NP head to the preceding preposition, or any verb to the preceding conjunction with COMP |
| MOD | Link any adverb to the closest adverb, verb, or adjective with MOD |
| REF | Link any *se* or *s'* to the following verb with REF |
| DUMMY | Link any *y* to the following verb with DUMMY |
| *rel.* |  |
| *PP* | Link any preposition to the preceding verb with POBJ or MOD, or to the preceding NP head with MOD |
| **time** |  and link *à* to the closest verb or noun with MOD |
| **proper names** | Link any NAM to the immediately preceding NAM with MOD |

Table 4: Rules set for French broadcast news

## 5. Conclusions

We have proposed a weakly supervised parser that may be used, in a future work, to leverage traditional n-grams with structured dependencies. The integration of this model into a speech recognizer is not described here, but we plan to use it as an additional nbest rescoring pass to start with. This work focuses on the definition and training of the model, which is realized with Bayesian inference on a raw unlabeled corpus. Hand-crafted rules act as constraints to guide inference towards the most plausible solutions from the point of view of the target domain, and especially speech transcripts. To the best of our knowledge, the proposed approach is the first one that includes the rules as latent variables in a discriminative model for parsing, which allows to precisely define their influence on the other meaningful model's variables. Furthermore, the rules are sampled just like any other latent variable, hence giving the model the possibility to ignore some badly defined constraints and increasing its robustness to user mistakes. Another advantage is the high degree of freedom that the user has to write the rules, and the fact that our framework supports both generative and discriminative models. The proposed model is evaluated on three languages, English, French and Czech, on two domains, newspaper texts and broadcast news transcriptions, and with and without gold part-of-speech tags. The model's performances are encouraging across all conditions, and match those of related state-of-the-art weakly and semi-supervised systems.

## 6. Acknowledgment

This work has been partly supported by the European Regional Development Fund (ERDF), project "NTIS - New Technologies for Information Society", European Centre of Excellence, CZ.1.05/1.1.00/02.0090.

# 7. References


[1] C. Chelba and F. Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, pp. 283–332, 2000.

[2] D. Klein, "The unsupervised learning of natural language structure," Ph.D. dissertation, Stanford University, 2005.

[3] W. P. Headden III, M. Johnson, and D. McClosky, "Improving unsupervised dependency parsing with richer contexts and smoothing," in *Proc. NAACL*, 2009.

[4] P. Blunsom and T. Cohn, "Unsupervised induction of tree substitution grammars for dependency parsing," in *Proc. EMNLP*, 2010.

[5] T. Naseem, H. Chen, R. Barzilay, and M. Johnson, "Using universal linguistic knowledge to guide grammar induction," in *Proc. EMNLP*. ACL, 2010, pp. 1234–1244.

[6] V. I. Spitkovsky, H. Alshawi, A. X. Chang, and J. D., "Unsupervised dependency parsing without gold part-of-speech tags," in *Proc. EMNLP*, 2011.

[7] T. Berg-Kirkpatrick and D. Klein, "Phylogenetic grammar induction," in *Proc. ACL*, Uppsala, Sweden, Jul. 2010, pp. 1288–1297.

[8] P. Boonkwan and M. Steedman, "Grammar induction from text using small syntactic prototypes," in *Proc. IJCNLP*, Chiang Mai, Thailand, Nov. 2011, pp. 438–446.

[9] T. Naseem and R. Barzilay, "Using semantic cues to learn syntax," in *AAAI*, W. Burgard and D. Roth, Eds. AAAI Press, 2011.

[10] J. Graça, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," in *Proc. NIPS*, 2007.

[11] J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar, "Posterior sparsity in unsupervised dependency parsing," *Journal of Machine Learning Research*, vol. 12, pp. 455–490, Feb. 2011.

[12] H. Daumé III, "Semi-supervised or semi-unsupervised?" in *Proc. NAACL Wokshop on Semi-supervised Learning for NLP*, 2009.

[13] K. Ganchev, J. Gillenwater, and B. Taskar, "Dependency grammar induction via bitext projection constraints," in *Proc. ACL*, 2009.

[14] G. Druck, G. Mann, and A. McCallum, "Semi-supervised learning of dependency parsers using Generalized Expectation criteria," in *Proc. ACL*, Suntec, Singapore, Aug. 2009, pp. 360–368.

[15] H. Daumé III, "Unsupervised search-based structured prediction," in *Proc. ICML*, Montreal, Canada, 2009.

[16] M. S. Rasooli and H. Faili, "Fast unsupervised dependency parsing with arc-standard transitions," in *Proc. EACL*, 2012.

[17] M. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," in *ACL*. Prague, Czech Republic: Association for Computational Linguistics, 6 2007, pp. 280–287. [Online]. Available: http://cogcomp.cs.illinois.edu/papers/ChangRaRo07.pdf

[18] S. Singh, L. Yao, S. Riedel, and A. McCallum, "Constraint-driven rank-based learning for information extraction," in *Proc. NAACL*, 2010, pp. 729–732.

[19] P. Liang, M. I. Jordan, and D. Klein, "Learning from measurements in exponential families," in *Proc. ICML*, 2009.

[20] A. Haghighi and D. Klein, "Prototype-driven grammar induction," in *Proc. ACL*, Sydney, Jul. 2006, pp. 881–888.

[21] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," in *Proc. International Conference on New Methods in Language Processing*, 1994, pp. 44–49.

[22] B. Bohnet, "Top accuracy and fast dependency parsing is not a contradiction," in *Proc. International Conference on Computational Linguistics*, Beijing, China, 2010.

[23] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of English: the Penn Treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[24] J. Hajič, *Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová*. Charles University Press, 1999, ch. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank, pp. 106–132.

[25] M. Candito, B. Crabbé, and D. Seddah, "On statistical parsing of french with supervised and semi-supervised strategies," in *Proc. EACL*, 2009.

[26] C. Cerisara, C. Gardent, and C. Anderson, "Building and exploiting a dependency treebank for french radio broadcasts," in *Proc. Intl Workshop on Treebanks and Linguistic Theories (TLT)*, Tartu, Estonia, Dec. 2010.