Multilingual Joint Parsing of Syntactic and Semantic Dependencies with a Latent Variable Model

James Henderson*
Xerox Research Centre Europe

Paola Merlo**
University of Geneva

Ivan Titov†
Saarland University

Gabriele Musillo‡
dMetrics

Current investigations in data-driven models of parsing have shifted from purely syntactic analysis to richer semantic representations, showing that the successful recovery of the meaning of text requires structured analyses of both its grammar and its semantics. In this article, we report on a joint generative history-based model to predict the most likely derivation of a dependency parser for both syntactic and semantic dependencies, in multiple languages. Because these two dependency structures are not isomorphic, we propose a weak synchronization at the level of meaningful subsequences of the two derivations. These synchronized subsequences encompass decisions about the left side of each individual word. We also propose novel derivations for semantic dependency structures, which are appropriate for the relatively unconstrained nature of these graphs. To train a joint model of these synchronized derivations, we make use of a latent variable model of parsing, the Incremental Sigmoid Belief Network (ISBN) architecture. This architecture induces latent feature representations of the derivations, which are used to discover correlations both within and between the two derivations, providing the first application of ISBNs to a multi-task learning problem. This joint model achieves competitive performance on both syntactic and semantic dependency parsing for several languages. Because of the general

* Most of the work in this paper was done while James Henderson was at the University of Geneva. He is currently at XRCE, 6 chemin de Maupertuis, 38240 Meylan, France.
E-mail: james.henderson@xrce.xerox.com.
** Department of Linguistics, University of Geneva, 5 rue de Candolle, Geneva, Switzerland.
E-mail: paola.merlo@unige.ch.
† MMCI Cluster of Excellence, Saarland University, Postfach 151150, 66041 Saarbrücken, Germany.
E-mail: titov@mmci.uni-saarland.de.
‡ dMetrics, 181 N 11th St, Brooklyn, NY 11211, USA. E-mail: gam@dmetrics.com.

Submission received: 31 August 2011; revised version received: 14 September 2012; accepted for publication: 1 November 2012.

doi:10.1162/COLI_a_00158

nature of the approach, this extension of the ISBN architecture to weakly synchronized syntactic-semantic derivations is also an exemplification of its applicability to other problems where two independent, but related, representations are being learned.

1. Introduction

Success in statistical syntactic parsing based on supervised techniques trained on a large corpus of syntactic trees—both constituency-based (Collins 1999; Charniak 2000; Henderson 2003) and dependency-based (McDonald 2006; Nivre 2006; Bohnet and Nivre 2012; Hatori et al. 2012)—has paved the way to applying statistical approaches to the more ambitious goals of recovering semantic representations, such as the logical form of a sentence (Ge and Mooney 2005; Wong and Mooney 2007; Zettlemoyer and Collins 2007; Ge and Mooney 2009; Kwiatkowski et al. 2011) or learning the propositional argument-structure of its main predicates (Miller et al. 2000; Gildea and Jurafsky 2002; Carreras and Màrquez 2005; Màrquez et al. 2008; Li, Zhou, and Ng 2010). Moving towards a semantic level of representation of language and text has many potential applications in question answering and information extraction (Surdeanu et al. 2003; Moschitti et al. 2007), and has recently been argued to be useful in machine translation and its evaluation (Wu and Fung 2009; Liu and Gildea 2010; Lo and Wu 2011; Wu et al. 2011), dialogue systems (Basili et al. 2009; Van der Plas, Henderson, and Merlo 2009), automatic data generation (Gao and Vogel 2011; Van der Plas, Merlo, and Henderson 2011) and authorship attribution (Hedegaard and Simonsen 2011), among others.

The recovery of the full meaning of text requires structured analyses of both its grammar and its semantics. These two forms of linguistic knowledge are usually thought to be at least partly independent, as demonstrated by speakers' ability to understand the meaning of ungrammatical text or speech and to assign grammatical categories and structures to unknown words and nonsense sentences.

These two levels of representation of language, however, are closely correlated. From a linguistic point of view, the assumption that syntactic distributions will be predictive of semantic role assignments is based on linking theory (Levin 1986). Linking theory assumes the existence of a ranking of semantic roles that are mapped by default on a ranking of grammatical functions and syntactic positions, and it attempts to predict the mapping of the underlying semantic component of a predicate's meaning onto the syntactic structure. For example, Agents are always mapped in syntactically higher positions than Themes. Linking theory has been confirmed statistically (Merlo and Stevenson 2001).

It is currently common to represent the syntactic and semantic role structures of a sentence in terms of dependencies, as illustrated in Figure 1. The complete graph of both the syntax and the semantics of the sentences is composed of two half graphs, which

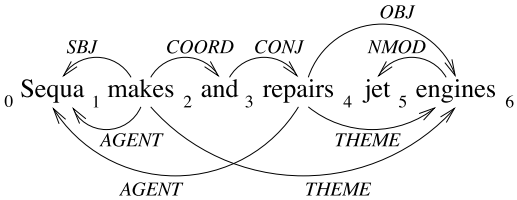

Figure 1
A semantic dependency graph labeled with semantic roles (lower half) paired with a syntactic dependency tree labeled with grammatical relations.

share all their vertices—namely, the words. Internally, these two half graphs exhibit different properties. The syntactic graph is a single connected tree. The semantic graph is just a set of one-level treelets, one for each proposition, which may be disconnected and may share children. In both graphs, it is not generally appropriate to assume independence across the different treelets in the structure. In the semantic graph, linguistic evidence that propositions are not independent of each other comes from constructions such as coordinations where some of the arguments are shared and semantically parallel. The semantic graph is also generally assumed not to be independent of the syntactic graph, as discussed earlier. As can be observed in Figure 1, however, arcs in the semantic graph do not correspond one-to-one to arcs in the syntactic graph, indicating that a rather flexible framework is needed to capture the correlations between graphs.

Developing models to learn these structured analyses of syntactic and shallow semantic representations raises, then, several interesting questions. We concentrate on the following two central questions.

- How do we design the interface between the syntactic and the semantic parsing representations?
- Are there any benefits to joint learning of syntax and semantics?

The answer to the second issue depends in part on the solution to the first issue, as indicated by the difficulty of achieving any benefit of joint learning with more traditional approaches (Surdeanu et al. 2008; Hajič et al. 2009; Li, Zhou, and Ng 2010). We begin by explaining how we address the first issue, using a semi-synchronized latent-variable approach. We then discuss how this approach benefits from the joint learning of syntax and semantics.

1.1 The Syntactic-Semantic Interface

The issue of the design of the interface between the syntactic and the semantic representations is central for any system that taps into the meaning of text. Standard approaches to automatic semantic role labeling use hand-crafted features of syntactic and semantic representations within linear models trained with supervised learning. For example, Gildea and Jurafsky (2002) formulate the shallow semantic task of semantic role labeling (SRL) as a classification problem, where the semantic role to be assigned to each constituent is inferred on the basis of its co-occurrence counts with syntactic features extracted from parse trees. More recent and accurate SRL methods (Johansson and Nugues 2008a; Punyakanok, Roth, and Yih 2008) use complex sets of lexico-syntactic features and declarative constraints to infer the semantic structure. Whereas supervised learning is more flexible, general, and adaptable than hand-crafted systems, linear models require complex features and the number of these features grows with the complexity of the task. To keep the number of features tractable, model designers impose hard constraints on the possible interactions within the semantic or syntactic structures, such as conditioning on grandparents but not great-great-grandparents. Likewise, hard constraints must be imposed on the possible interactions between syntax and semantics.

This need for complete specification of the allowable features is inappropriate for modeling syntactic–semantic structures because these interactions between syntax and semantics are complex, not currently well understood, and not identical from language to language. This issue is addressed in our work by developing a loosely coupled architecture and developing an approach that automatically discovers appropriate

features, thus better modeling both our lack of knowledge and the linguistic variability. We use latent variables to model the interaction between syntax and semantics. Latent variables serve as an interface between semantics and syntax, capturing properties of both structures relevant to the prediction of semantics given syntax and, conversely, syntax given semantics. Unlike hand-crafted features, latent variables are induced automatically from data, thereby avoiding a priori hard independence assumptions. Instead, the structure of the latent variable model is used to encode soft biases towards learning the types of features we expect to be useful.

We define a history-based model (Black et al. 1993) for joint parsing of semantic and syntactic structures. History-based models map structured representations to sequences of derivation steps, and model the probability of each step conditioned on the entire sequence of previous steps. There are standard shift-reduce algorithms (Nivre, Hall, and Nilsson 2004) for mapping a syntactic dependency graph to a derivation sequence, and similar algorithms can be defined for mapping a semantic dependency graph to a derivation sequence, as discussed subsequently. But defining a joint syntactic–semantic derivation presents a challenge. Namely, given the complex nature of correspondences between the structures, it is not obvious how to synchronize individual semantic–syntactic steps in the derivation. Previous joint statistical models of dependency syntax and SRL have either ignored semantic arcs not corresponding to single syntactic arcs (Thompson, Levy, and Manning 2003; Titov and Klementiev 2011) or resorted to pre-/post-processing strategies that modify semantic or syntactic structures (Lluís and Màrquez 2008; Lang and Lapata 2011; Titov and Klementiev 2012). In a constituency setting, Li, Zhou, and Ng (2010) explore different levels of coupling of syntax and semantics, and find that only explicit interleaving or explicit feature selection yield improvements in performance.

Instead of synchronizing individual steps, we (1) decompose both the syntactic derivation and the semantic derivation into subsequences, where each subsequence corresponds to a single word in the sentence, and then (2) synchronize syntactic and semantic subsequences corresponding to the same word with each other. To decide which steps correspond to a given word, we use a simple deterministic rule: A step of a derivation corresponds to the word appearing at the front of the queue prior to that step. For shift-reduce derivations, this definition breaks derivations into contiguous subsequences in the same order as the words of the sentence, both for syntax and for semantics. Each subsequence forms a linguistically meaningful chunk in that it includes all the decisions about the arcs on the left side of the associated word, both its parents and its children. Thus, synchronizing the syntactic and semantic subsequences according to their associated word places together subsequences that are likely to be correlated. Note that such pairs of syntactic and semantic subsequences will, in general, have different numbers of steps on each side and these numbers of steps are, in general, unbounded. Therefore, instead of defining atomic synchronized rules as in synchronous grammars (Wu 1997; Chiang 2005), we resort to parametrized models that exploit the internal structure of the paired subsequences.

This derivational, joint approach to handling these complex representations leads to a new proposal on how to learn them, which avoids extensive and complex feature engineering, as discussed in the following.

1.2 Joint Learning of Syntax and Semantics

Our probabilistic model is learned using Incremental Sigmoid Belief Networks (ISBNs) (Henderson and Titov 2010), a recent development of an early latent variable model

for syntactic structure prediction (Henderson 2003), which has shown very good performance for both constituency (Titov and Henderson 2007a) and dependency parsing (Titov and Henderson 2007d). Instead of hand-crafting features of the previous parsing decisions, as is standard in history-based models, ISBNs estimate the probability of the next parsing actions conditioned on a vector of latent-variable features of the parsing history. These features are induced automatically to maximize the likelihood of the syntactic–semantics graphs given in the training set, and therefore they encode important correlations between syntactic and semantic decisions. This makes joint learning of syntax and semantics a crucial component of our approach.

The joint learning of syntactic and semantic latent representations makes our approach very different from the vast majority of the successful SRL methods. Most of these approaches not only learn syntactic and semantic representations independently, but also use pipelines at testing time. Therefore, in these methods semantic information does not influence syntactic parsing (Punyakanok, Roth, and Yih 2008; Toutanova, Haghighi, and Manning 2008). Some of the recent successful methods learn their syntactic and semantic parsing components separately, optimizing two different functions, and then combine syntactic and semantic predictions either by simple juxtaposition or by checking their coherence in a final step (Chen, Shi, and Hu 2008; Johansson and Nugues 2008b).

A few other approaches do attempt joint learning of syntax and grammatical function or semantics (Lluís and Màrquez 2008; Hall and Nivre 2008; Morante, Van Asch, and van den Bosch 2009; Tsarfaty, Sima'an, and Scha 2009; Li, Zhou, and Ng 2010). Although these approaches recognize that joint learning requires treating the representations as correlated, they do not exploit the intuition that successful methods need, implicitly or explicitly, to tackle a number of sub-problems that are common across the goal problems. For instance, some way of modeling selectional preferences is arguably necessary both for semantic role labeling and for syntactic parse disambiguation, and therefore the corresponding component should probably be shared between the syntactic and semantic models.

In machine learning, the issue of joint learning of models for multiple, non-trivially related tasks is called **multi-task learning**. Though different multi-task learning methods have been developed, the underlying idea for most of them is very similar. Multi-task learning methods attempt to induce a new, less sparse representation of the initial features, and this representation is shared by the models for all the considered tasks. Intuitively, for any given set of primary tasks, if one were to expect that similar latent sub-problems needed to be solved to find a solution for these primary tasks, then one would expect an improvement from inducing shared representations.

Multi-task learning methods have been shown to be beneficial in many domains, including natural language processing (Ando and Zhang 2005a, 2005b; Argyriou, Evgeniou, and Pontil 2006; Collobert and Weston 2008). Their application in the context of syntactic-semantic parsing has been very limited, however. The only other such successful multi-task learning approach we are aware of targets a similar, but more restricted, task of function labeling (Musillo and Merlo 2005). Musillo and Merlo (2005) conclusively show that jointly learning functional and syntactic information can significantly improve syntax. Our joint learning approach is an example of a multi-task learning approach in that the induced representations in the vectors of latent variables can capture hidden sub-problems relevant to predicting both syntactic and semantic structures.

The rest of this article will first describe the data that are used in this work and their relevant properties. We then present our probabilistic model of joint syntactic parsing

and semantic role labeling. We introduce the latent variable architecture for structured prediction, before presenting our application of this architecture to modeling the distributions for the parsing model, and investigate a few variations. We then present the results on syntactic and semantic parsing of English, which we then extend to several languages. Finally, we discuss, compare to related work, and conclude.

2. Representations and Formulation of the Problem

The recovery of shallow meaning, and semantic role labels in particular, has a long history in linguistics (Fillmore 1968). Early attempts at systematically representing lexical semantics information in a precise way usable by computers, such as Levin's classification or WordNet, concentrated on defining semantic properties of words and classes of words in the lexicon (Miller et al. 1990; Levin 1993). But only recently has it become feasible to tackle these problems by using machine learning techniques, because of the development of large annotated databases, such as VerbNet (Kipper et al. 2008) and FrameNet (Baker, Fillmore, and Lowe 1998), and corpora, such as PropBank (Palmer, Gildea, and Kingsbury 2005). OntoNotes (Pradhan et al. 2007) is a current large-scale exercise in integrated annotation of several semantic layers.

Several corpus annotation efforts have been released, including FrameNet and PropBank. FrameNet is a large-scale, computational lexicography project (Baker, Fillmore, and Lowe 1998), which includes a set of labeled examples that have been used as a corpus. FrameNet researchers work at a level of representation called the frame, which is a schematic representation of situations involving various participants, or representations of objects involving their properties. The participants and properties in a frame are designated with a set of semantic roles called frame elements. One example is the MOTION DIRECTIONAL frame, and its associated frame elements include the THEME (the moving object), the GOAL (the ultimate destination), the SOURCE, and the PATH. The collection of sentences used to exemplify frames in the English FrameNet has been sampled to produce informative lexicographic examples, but no attempt has been made to produce representative distributions. The German SALSA corpus (Burchardt et al. 2006), however, has been annotated with FrameNet annotation. This extension to exhaustive corpus coverage and a new language has only required a few novel frames, demonstrating the cross-linguistic validity of this annotation scheme. FrameNets for other languages, Spanish and Japanese, are also under construction.

Another semantically annotated corpus—the one we use in this work for experiments on English—is called Proposition Bank (PropBank) (Palmer, Gildea, and Kingsbury 2005). PropBank is based on the assumption that the lexicon is not a list of irregularities, but that systematic correlations can be found between the meaning components of words and their syntactic realization. It does not incorporate the rich frame typology of FrameNet, because natural classes of predicates can be defined based on syntactic alternations, and it defines a limited role set. PropBank encodes propositional information by adding a layer of argument structure annotation to the syntactic structures of verbs in the Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993). Arguments of verbal predicates in the Penn Treebank (PTB) are annotated with abstract semantic role labels (A0 through A5 or AA) for those complements of the predicative verb that are considered arguments. Those complements of the verb labeled with a semantic functional label in the original PTB receive the composite semantic role label AM-*X*, where *X* stands for labels such as LOC, TMP, or ADV, for locative, temporal, and

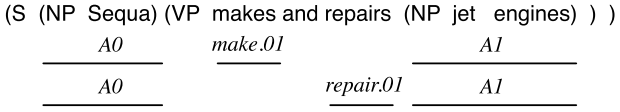

Figure 2
An example sentence from Penn Treebank annotated with constituent syntactic structure along with semantic role information provided in PropBank.

adverbial modifiers, respectively. A tree structure, represented as a labeled bracketing, with PropBank labels, is shown in Figure 2.

PropBank uses two levels of granularity in its annotation, at least conceptually. Arguments receiving labels A0–A5 or AA are specific to the verb, so these labels do not necessarily express consistent semantic roles across verbs, whereas arguments receiving an AM-*X* label are supposed to be adjuncts, and the roles they express are consistent across all verbs. A0 and A1 arguments are annotated based on the proto-role theory presented in Dowty (1991) and correspond to proto-agents and proto-patients, respectively. Although PropBank, unlike FrameNet, does not attempt to group different predicates evoking the same prototypical situation, it does distinguish between different senses of polysemous verbs, resulting in multiple *framesets* for such predicates.

NomBank annotation (Meyers et al. 2004) extends the PropBank framework to annotate arguments of nouns. Only the subset of nouns that take arguments are annotated in NomBank and only a subset of the non-argument siblings of nouns are marked as ARG-M. The most notable specificity of NomBank is the use of support chains, marked as SU. Support chains are needed because nominal long distance dependencies are not captured under the Penn Treebank's system of empty categories. They are used for all those cases in which the nominal argument is outside the noun phrase. For example, in a support verb construction, such as *Mary took dozens of walks*, the arcs linking *walks* to *of*, *of* to *dozens*, and *dozens* to *took* are all marked as support.

The data we use for English are the output of an automatic process of conversion of the original PTB, PropBank, and NomBank into dependency structures, performed by the algorithm described in Johansson and Nugues (2007). These are the data provided to participants to the CoNLL-2008 and CoNLL-2009 shared tasks (`http://ifarm.nl/signll/conll/`). An example is shown in Figure 3. This representation encodes both the grammatical functions and the semantic labels that describe the sentence.

Argument labels in PropBank and NomBank are assigned to constituents, as shown in Figure 2. After the conversion to dependency the PropBank and NomBank labels

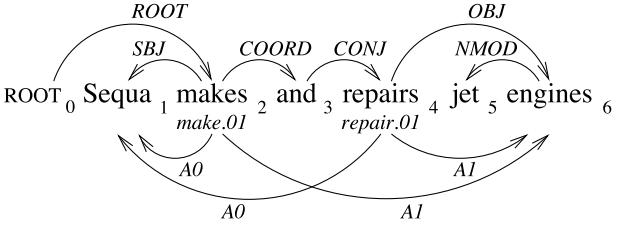

Figure 3
An example from the PropBank corpus of verbal predicates and their semantic roles (lower half) paired with syntactic dependencies derived from the Penn Treebank.

are assigned to individual words. Roughly, for every argument span, the preprocessing algorithm chooses a token that has the syntactic head outside of the span, though additional modifications are needed to handle special cases (Johansson and Nugues 2007; Surdeanu et al. 2008). This conversion implies that the span of words covered by the subtree headed by the word receiving the label can often be interpreted as receiving the semantic role label. Consequently, for the dependency-based representation, the syntactic and the semantic graphs jointly define the semantic role information. This is coherent with the original PropBank annotation, which is to be interpreted as a layer of annotation added to the Penn Treebank. Note, however, that the coherence of the syntactic annotation and the semantic role labels is not evaluated in the dependency-based SRL tasks (CoNLL-2008 and CoNLL-2009), so the two half-graphs are, in practice, considered independently.

Unfortunately, mapping from the dependency graphs to the argument spans is more complex than just choosing syntactic subtrees of headwords. This over-simplistic rule would result in only 88% of PropBank arguments correctly recovered (Choi and Palmer 2010). For example, it would introduce overlapping arguments or even cases where the predicate ends up in the argument span; both these situations are impossible under the PropBank and NomBank guidelines. These problems are caused by relative clauses, modals, negations, and verb chains, among others. A careful investigation (Choi and Palmer 2010), however, showed that a set of heuristics can be used to accurately retrieve the original phrase boundaries of the semantic arguments in PropBank from the dependency structures. This observation implies that both representations are nearly equivalent and can be used interchangeably.[1]

Several data sets in this format for six other languages were released for the CoNLL-2009 shared task. These resources were in some cases manually constructed in dependency format, and in some cases they were derived from existing resources, such as the data set for Czech, derived from the tectogrammatic Prague Dependency Treebank (Hajič et al. 2006), or a data set for German derived from the FrameNet-style SALSA corpus (Burchardt et al. 2006). Not only are these resources derived from different methodologies and linguistic theories, but they are also adapted to very different languages and different sizes of data sets. For the discussion of the conversion process, we refer the reader to the original shared task description (Surdeanu et al. 2008).

The two-layer graph representation, which was initially developed for English and then adapted to other languages, enables these very different encodings to be represented in the same form. The properties of these different data sets, though, are rather different, in some important respects. As can be clearly seen from Table 1 and as indicated in the Introduction, the properties of syntactic dependency graphs are very different from semantic dependency graphs: The former give rise to a tree, and the latter are a forest of treelets, each representing a proposition. The amount of crossing arcs are also different across the different data sets in the various languages.

The problem we need to solve consists of producing a syntactic–semantic graph given an input word string. Our formulation of this problem is very general: It does not assume that the two-half-graphs are coupled, nor that they form a single tree or a graph without crossing arcs. Rather, it considers that the syntactic and the semantic graphs are

1 Note though that the study in Choi and Palmer (2010) was conducted using gold-standard syntactic dependencies in the heuristics. Recovery of argument spans based on predicted syntactic analyses is likely to be a harder problem. Extending the heuristics in Choi and Palmer to recover the spans of the semantic arguments in NomBank also appears to be a challenging problem.

Table 1
For each language, percentages of training sentences with crossing arcs in syntax and semantics, and percentages of training sentences with semantic arcs forming a tree whose root immediately dominates the predicates.

	Syntactic crossings	Semantic crossings	Semantic tree
Catalan	0.0	0.0	61.4
Chinese	0.0	28.0	28.6
Czech	22.4	16.3	6.1
English	7.6	43.9	21.4
German	28.1	1.3	97.4
Japanese	0.9	38.3	11.2
Spanish	0.0	0.0	57.1

only loosely coupled, and share only the vertices (the words). The next section presents how we model these graph structures.

3. Modeling Synchronized Derivations

We propose a joint generative probabilistic model of the syntactic and semantic dependency graphs using two synchronized derivations. In this section, we describe how the probability of the two half-graphs can be broken down into the conditional probabilities of parser actions. The issue of how to estimate these conditional probabilities without making inappropriate independence assumptions will be addressed in Section 4, where we explain how we exploit induced latent variable representations to share information between action choices.

Our joint probability model of syntactic and semantic dependencies specifies the two dependency structures as synchronized sequences of actions for a parser that operates on two different data structures. The probabilities of the parser actions are further broken down to probabilities for primitive actions similar to those used in previous dependency parsing work. No independence assumptions are made in the probability decomposition itself. This allows the probability estimation technique (discussed in Section 4) to make maximal use of its latent variables to learn correlations between the different parser actions, both within and between structures.

3.1 Synchronized Derivations

We first specify the syntactic and semantic derivations separately, before specifying how they are synchronized in a joint generative model.

The derivations for syntactic dependency trees are based on a shift-reduce style parser (Nivre et al. 2006; Titov and Henderson 2007d). The derivations use a stack and an input queue. There are actions for creating a leftward or rightward arc between the top of the stack and the front of the queue, for popping a word from the stack, and for shifting a word from the queue to the stack.

A syntactic configuration of the parser is defined by the current stack, the queue of remaining input words, and the partial labeled dependency structure constructed by previous parser actions. The parser starts with an empty stack and terminates when it

reaches a configuration with an empty queue. The generative process uses four types of actions:

1. The action *Left-Arc*$_r$ adds a dependency arc from the next input word w_j to the word w_i on top of the stack, selects the label r for the relation between w_i and w_j, and finally pops the word w_i from the stack.
2. The action *Right-Arc*$_r$ adds an arc from the word w_i on top of the stack to the next input word w_j and selects the label r for the relation between w_i and w_j.
3. The action *Reduce* pops the word w_i from the stack.
4. The action *Shift*$_{w_{j+1}}$ shifts the word w_j from the input queue to the stack and predicts the next word in the queue w_{j+1}.[2]

The derivations for semantic dependencies also use a stack and an input queue, but there are three main differences between the derivations of the syntactic and semantic dependency graphs. The actions for semantic derivations include the actions used for syntactic derivations, but impose fewer constraints on their application because a word in a semantic dependency graph can have more than one parent. Namely, unlike the algorithm used for syntax, the *Left-Arc*$_r$ action does not pop a word from the stack. This modification allows a word to have multiple parents, as required for non-tree parsing. Also, the *Reduce* action does not require the word to have a parent, thereby allowing for disconnected structure. In addition, two new actions are introduced for semantic derivations:

5. The action *Predicate*$_s$ selects a frameset s for the predicate w_j at the front of the input queue.
6. The action *Swap* swaps the two words at the top of the stack.

The *Swap* action, introduced to handle non-planar structures, will be discussed in more detail in Section 3.2.

One of the crucial intuitions behind our approach is that the parsing mechanism must correlate the two half-graphs, but allow them to be constructed separately as they have very different properties. Let T_d be a syntactic dependency tree with derivation $D_d^1, \ldots, D_d^{m_d}$, and T_s be a semantic dependency graph with derivation $D_s^1, \ldots, D_s^{m_s}$. To define derivations for the joint structure T_d, T_s, we need to specify how the two derivations are synchronized, and in particular make the important choice of the granularity of the synchronization step. Linguistic intuition would perhaps suggest that syntax and semantics are connected at the clause level—a big step size—whereas a fully integrated system would synchronize at each parsing decision, thereby providing the most communication between these two levels. We choose to synchronize the construction of the two structures at every word—an intermediate step size. This choice is simpler, as it is based on the natural total order of the input, and it avoids the problems of the more linguistically motivated choice, where chunks corresponding to different semantic propositions would be overlapping.

2 For clarity, we will sometimes write *Shift*$_j$ instead of *Shift*$_{w_{j+1}}$.

We divide the two derivations into the sequence of actions, which we call **chunks**, between shifting each word onto the stack, $c_d^t = D_d^{b^t}, \ldots, D_d^{e^t}$ and $c_s^t = D_s^{b^t}, \ldots, D_s^{e^t}$, where $D_d^{b^t-1} = D_s^{b^t-1} = Shift_{t-1}$ and $D_d^{e^t+1} = D_s^{e^t+1} = Shift_t$. Then the actions of the synchronized derivations consist of quadruples $C^t = (c_d^t, Switch, c_s^t, Shift_t)$, where *Switch* means switching from syntactic to semantic mode. A word-by-word illustration of this synchronized process is provided in Figure 4. This gives us the following joint probability model, where n is the number of words in the input.

$$\begin{aligned} P(T_d, T_s) &= P(C^1, \ldots, C^n) \\ &= \prod_t P(C^t | C^1, \ldots, C^{t-1}) \end{aligned} \tag{1}$$

Chunk probabilities are then decomposed into smaller steps. The probability of each synchronized derivation chunk C^t is the product of four factors, related to the syntactic level, the semantic level, and the two synchronizing steps. An illustration of the individual derivation steps is provided in Figure 5.

$$\begin{aligned} P(C^t | C^1, \ldots, C^{t-1}) = \; & P(c_d^t | C^1, \ldots, C^{t-1}) \times \\ & P(Switch | c_d^t, C^1, \ldots, C^{t-1}) \times \\ & P(c_s^t | Switch, c_d^t, C^1, \ldots, C^{t-1}) \times \\ & P(Shift_t | c_d^t, c_s^t, C^1, \ldots, C^{t-1}) \end{aligned} \tag{2}$$

These synchronized derivations $C^1, \ldots, C^n$ only require a single input queue, since the *Shift* operations are synchronized, but they require two separate stacks, one for the syntactic derivation and one for the semantic derivation.

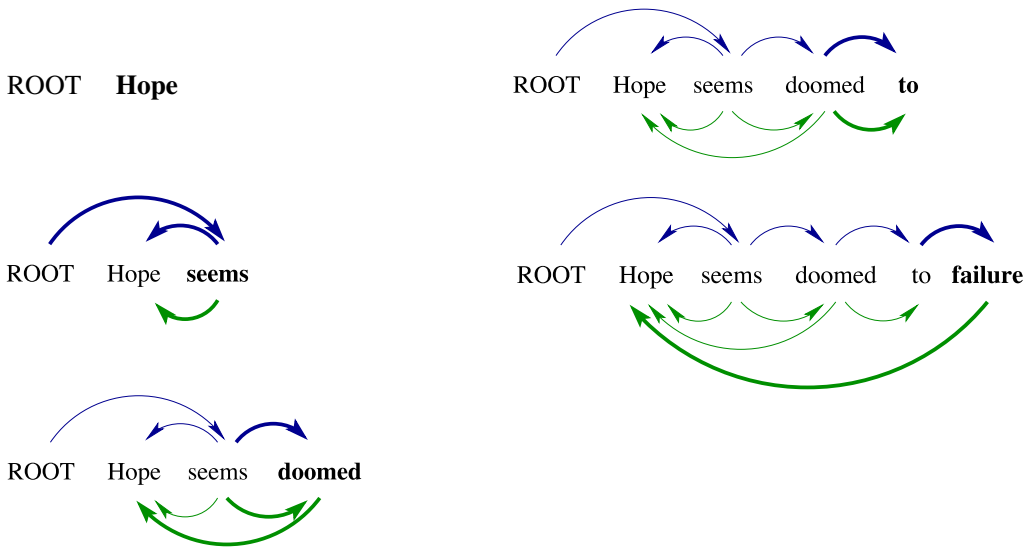

Figure 4
A word-by-word illustration of a joint synchronized derivation, where the blue top half is the syntactic tree and the green bottom half is the semantic graph. The word at the front of the queue and the arcs corresponding to the current chunk are shown in **bold**.

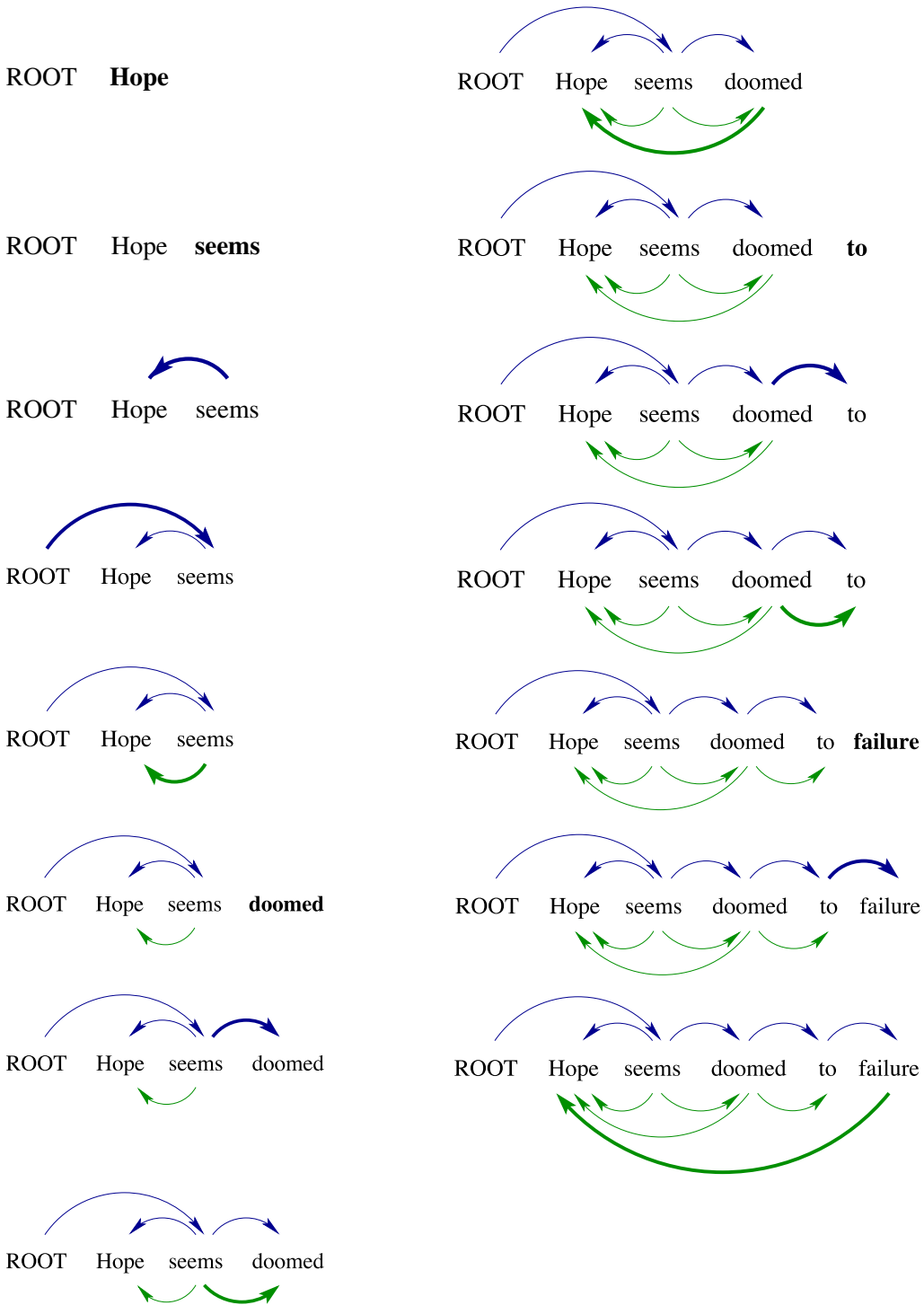

Figure 5
A joint, synchronized derivation, illustrating individual syntactic and semantic steps. The results of each derivation step are shown in **bold**, with the blue upper arcs for syntax and the green lower arcs for semantics. *Switch* and *Reduce* actions are not shown explicitly.

The probability of c_d^t is decomposed into the probabilities of the derivation actions D_d^i

$$P(c_d^t|C^1,\ldots,C^{t-1}) = \prod_{b_d^t \le i \le e_d^t} P(D_d^i|D_d^{b_d^t},\ldots,D_d^{i-1},C^1,\ldots,C^{t-1}) \quad (3)$$

and then the probability of c_s^t is decomposed into the probabilities of the derivation actions D_s^i

$$P(c_s^t|Switch,c_d^t,C^1,\ldots,C^{t-1}) = \prod_{b_s^t \le i \le e_s^t} P(D_s^i|D_s^{b_s^t},\ldots,D_s^{i-1},Switch,c_d^t,C^1,\ldots,C^{t-1}) \quad (4)$$

Note that in all these equations we have simply applied the chain rule, so all equalities are exact. The order in which the chain rule has been applied gives us a complete ordering over all decisions in $C^1,\ldots,C^n$, including all the decisions in $D_d^1,\ldots,D_d^{m_d}$ and $D_s^1,\ldots,D_s^{m_s}$. For notational convenience, we refer to this complete sequence of decisions as $D^1,\ldots,D^m$, allowing us to state

$$P(T_d,T_s) = \prod_i P(D^i|D^1,\ldots,D^{i-1}) \quad (5)$$

Instead of treating each D^i as an atomic decision, it will be convenient in the subsequent discussion to sometimes split it into a sequence of elementary decisions $D^i = d_1^i,\ldots,d_m^i$:

$$P(D^i|D^1,\ldots,D^{i-1}) = \prod_k P(d_k^i|hist(i,k)) \quad (6)$$

where $hist(i,k)$ denotes the parsing history $D^1,\ldots,D^{i-1},d_1^i,\ldots,d_{k-1}^i$. Each conditional distribution is estimated using the latent variable model, ISBN, which we will describe in Section 4.1.

This way of synchronizing the syntactic and semantic derivations is not formally equivalent to a synchronous grammar. A synchronous grammar would generate the sequence of synchronized steps $C^1,\ldots,C^n$, which would require a finite vocabulary of possible synchronized steps C^i. But these synchronized steps C^i are themselves specified by a generative process which is capable of generating arbitrarily long sequences of actions. For example, there may be an unbounded number of *Reduce* actions in between two *Shift* actions. Thus there are an infinite number of possible synchronized steps C^i, and the synchronous grammar would itself have to be infinite.

Instead, we refer to this model as "semi-synchronized." The two derivations are synchronized on the right-hand side of each dependency (the front of the queue), but not on the left-hand side (the top of the stack). This approach groups similar dependencies together, in that they all involve the same right-hand side. But the lack of restrictions on the left-hand side means that this approach does not constrain the possible structures or the relationship of syntax to semantics.

3.2 Planarization of Dependencies

Without including the *Swap* action, the derivations described above could only specify planar syntactic or semantic dependency graphs. Planarity requires that the graph can be drawn in the semi-plane above the sentence without any two arcs crossing, and without changing the order of words.[3]

Exploratory data analysis indicates that many instances of non-planarity in the complete graph are due to crossings of the syntactic and semantic graphs. For instance, in the English training set, there are approximately 7.5% non-planar arcs in the joint syntactic–semantic graphs, whereas summing the non-planarity within each graph gives us only roughly 3% non-planar arcs in the two separate graphs. Because our synchronized derivations use two different stacks for the syntactic and semantic dependencies, respectively, we only require each individual graph to be planar.

The most common approach to deal with non-planarity transforms crossing arcs into non-crossing arcs with augmented labels (Nivre and Nilsson 2005). This is called the pseudo-projective parsing with HEAD encoding method (HEAD for short, see Section 6). We use this method to projectivize the syntactic dependencies. Despite the shortcomings that will be discussed later, we adopt this method because the amount of non-planarity in syntactic structures is often small: only 0.39% of syntactic dependency arcs in the English training set are non-planar. Therefore, choice of the planarization strategy for syntactic dependencies is not likely to seriously affect the performance of our method for English.

One drawback of this approach is theoretical. Augmented structures that do not have any interpretation in terms of the original non-planar trees receive non-zero probabilities. When parsing with such a model, the only computationally feasible search consists of finding the most likely augmented structure and removing inconsistent components of the dependency graph (Nivre et al. 2006; Titov and Henderson 2007d). But this practically motivated method is not equivalent to a statistically motivated—but computationally infeasible—search for the most probable consistent structure. Moreover, learning these graphs is hard because of the sparseness of the augmented labels. Empirically, it can be observed that a parser that uses this planarization method tends to output only a small number of augmented labels, leading to a further drop of recall on non-planar dependencies.

Applying the same planarization approach to semantic dependency structures is not trivial and would require a novel planarization algorithm, because semantic dependency graphs are highly disconnected structures, and direct application of any planarization algorithm, such as the one proposed in Nivre and Nilsson (2005), is unlikely to be appropriate. For instance, a method that extends the planarization method to semantic predicate-argument structures by exploiting the connectedness of the corresponding syntactic dependency trees has been tried in Henderson et al. (2008). Experimental results reported in Section 6 indicate that the method that we will illustrate in the following paragraphs yields better performance.

A different way to tackle non-planarity is to extend the set of parsing actions to a more complex set that can parse any type of non-planarity (Attardi 2006). This approach is discussed in more detail in Section 7. We adopt a conservative version of this approach

3 Note that this planarity definition is stricter than the definition normally used in graph theory where the entire plane is used. Some parsing algorithms require *projectivity*: this is a stronger requirement than planarity and the notion of projectivity is only applicable to trees (Nivre and Nilsson 2005).

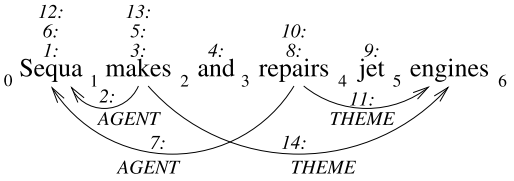

Figure 6
A non-planar semantic dependency graph whose derivation is the sequence of operations *1:Shift(1), 2:LeftArc(1,2), 3:Shift(2), 4:Shift(3), 5:Reduce(3), 6:Swap(1,2), 7:LeftArc(1,4), 8:Shift(4), 9:Shift(5), 10:Reduce(5), 11:RightArc(4,6), 12:Reduce(4), 13:Reduce(1), 14:RightArc(2,6).* In the figure, these steps are associated with either the created arc or the resulting top of the stack.

as described in Titov et al. (2009). Specifically, we add a single action that is able to handle most crossing arcs occurring in the training data. The decision *Swap* swaps the two words at the top of the stack.

The *Swap* action is inspired by the planarization algorithm described in Hajičová et al. (2004), where non-planar trees are transformed into planar ones by recursively rearranging their sub-trees to find a linear order of the words for which the tree is planar (also see the discussion of Nivre [2008], Nivre, Kuhlmann, and Hall [2009] in Section 7). Important differences exist, however, because changing the order of adjacent nodes in the stack is not equivalent to changing the order of adjacent phrases in the word sequences. In our method, nodes can appear in different orders at different steps of the derivation, so some arcs can be specified using one ordering, then other arcs can be specified with another ordering.[4] This makes our algorithm more powerful than just a single adjacent transposition of sub-trees.

In our experiments on the CoNLL-2008 shared task data set (Surdeanu et al. 2008), reported subsequently, introducing this action was sufficient to parse the semantic dependency structures of 37,768 out of 39,279 training sentences (96%).

Moreover, among the many linguistic structures which this parsing algorithm can handle, one of the frequent ones is coordination. The algorithm can process non-planarity introduced by coordination of two conjuncts sharing a common argument or being arguments of a common predicate (e.g., *Sequa makes and repairs jet engines*), as well as similar structures with three verb conjuncts and two arguments (e.g., *Sequa makes, repairs, and sells jet engines*). The derivation of a typical non-planar semantic graph involving coordination is illustrated in Figure 6. Inspection of example derivations also indicates that swaps occur frequently after verbs like *expect to, thought to,* and *helped,* which take a VP complement in a dependency representation. This is a coherent set of predicates, suggesting that swapping enables the processing of constructions such as *John expects Bill to come* that establish a relation between the higher verb and the lower infinitival head word (*to*), but with an intervening expressed subject (*Bill*). This is indeed a case in which two predicate-argument structures cross in the CoNLL shared task representation. More details and discussion on this action can be found in Titov et al. (2009).

The addition of the *Swap* action completes the specification of our semi-synchronized derivations for joint syntactic–semantic parsing. We now present the

4 Note that we do not allow two *Swap* actions in a row, which would return to an equivalent parser configuration. All other actions make an irreversible change to the parser configuration, so by requiring at least one other action between any two *Swap* actions, we prevent infinite loops.

latent variable method that allows us to accurately estimate the conditional probabilities of these parser actions.

4. The Estimation Method

The approach of modeling joint syntactic–semantic dependency parsing as a semi-synchronized parsing problem relies crucially on an estimation architecture that is flexible enough to capture the correlations between the two separate structures. For problems where multiple structured representations are learned jointly, and syntactic and semantic parsing in particular, it is often very difficult to precisely characterize the complex interactions between the two tasks. Under these circumstances, trying to design by hand features that capture these interactions will inevitably leave out some relevant features, resulting in independence assumptions that are too strong. We address this problem by using a learning architecture that is able to induce appropriate features automatically using latent variables.

Latent variables are used to induce features that capture the correlations between the two structures. Alternatively, these latent variables can be regarded as capturing correlations between the parsing tasks, as needed for effective multi-task learning. Roughly, we can assume that there exist some sub-problems that are shared between the two tasks, and then think of the latent variables as the outputs of classifiers for these sub-problems. For example, latent variables may implicitly encode if a word on top of the stack belongs to a specific cluster of semantically similar expressions.[5] This information is likely to be useful for both parsing tasks.

We use the Incremental Sigmoid Belief Network (ISBN) architecture (Henderson and Titov 2010) to learn latent variable models of our synchronized derivations of syntactic–semantic parsing. ISBNs postulate a vector of latent binary features associated with each state in each derivation. These features represent properties of the derivation history at that state which are relevant to future decisions. ISBNs learn these features as part of training the model, rather than a designer specifying them by hand. Instead, the designer specifies which previous states are the most relevant to a given state, based on locality in the structures being built by the derivation, as discussed later in this section. By conditioning each state's latent features on the latent features of these locally relevant states, ISBNs tend to learn correlations that are local in the structures. But by passing information repeatedly between latent features, the learned correlations are able to extend within and between structures in ways that are not constrained by independence assumptions.

In this section we will introduce ISBNs and specify how they are used to model the semi-synchronized derivations presented in the previous section. ISBNs are Bayesian networks based on sigmoid belief networks (Neal 1992) and dynamic Bayesian networks (Ghahramani 1998). They extend these architectures by allowing their model structure to be incrementally specified based on the partial structure being built by a derivation. They have previously been applied to constituency and dependency parsing (Titov and Henderson 2007a, 2007b). We successfully apply ISBNs to a more complex, multi-task parsing problem without changing the machine learning methods.

5 Development of methods for making explicit the regularities encoded in distributed latent representations remains largely an open problem, primarily due to statistical dependencies between individual latent variables. Therefore, we can only speculate about the range of modeled phenomena and cannot reliably validate our hypotheses.

4.1 Incremental Sigmoid Belief Networks

Like all Bayesian networks, ISBNs provide a framework for specifying a joint probability model over many variables. The conditional probability distribution of each variable is specified as a function of the other variables that have edges directed to it in the Bayesian network. Given such a joint model, we can then infer specific probabilities, such as computing the conditional probability of one variable given values for other variables.

This section provides technical details about the ISBN architecture. It begins with background on Sigmoid Belief Networks (SBNs) and Dynamic SBNs, a version of SBNs developed for modeling sequences. Then it introduces the ISBN architecture and the way we apply it to joint syntactic–semantic dependency parsing. Throughout this article we will use **edge** to refer to a link between variables in a Bayesian network, as opposed to **arc** for a link in a dependency structure. The pattern of edges in a Bayesian network is called the **model structure**, which expresses the types of correlations we expect to find in the domain.

4.1.1 Sigmoid Belief Networks. ISBNs are based on SBNs (Neal 1992), which have binary variables $s_i \in \{0, 1\}$ whose conditional probability distributions are of the form

$$P(s_i = 1 | Par(s_i)) = \sigma(\sum_{s_j \in Par(s_i)} J_{ij} s_j) \tag{7}$$

where $Par(s_i)$ denotes the variables with edges directed to s_i, σ denotes the logistic sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, and J_{ij} is the weight for the edge from variable s_j to variable s_i.[6] Each such conditional probability distribution is essentially a logistic regression (also called maximum-entropy) model, but unlike standard logistic regression models where the feature values are deterministically computable (i.e., observable), here the features may be latent. SBNs are also similar to feed-forward neural networks, but, unlike neural networks, SBNs have a precise probabilistic semantics for their hidden variables.

In ISBNs we consider a generalized version of SBNs where we allow variables with any range of discrete values. The normalized exponential function is used to define the conditional probability distributions at these variables:

$$P(s_i = v | Par(s_i)) = \frac{\exp(\sum_{s_j \in Par(s_i)} W^i_{vj} s_j)}{\sum_{v'} \exp(\sum_{s_j \in Par(s_i)} W^i_{v'j} s_j)} \tag{8}$$

where W^i is the weight matrix for the variable s_i.

4.1.2 Dynamic Sigmoid Belief Networks. SBNs can be easily extended for processing arbitrarily long sequences, for example, to tackle the language modeling problem or other sequential modeling tasks.

Such problems are often addressed with dynamic Bayesian networks (DBN) (Ghahramani 1998). A typical example of DBNs is the first-order hidden Markov model

6 For convenience, where possible, we will not explicitly include bias terms in expressions, assuming that every latent variable in the model has an auxiliary parent variable set to 1.

(HMM) which models two types of distributions, transition probabilities corresponding to the state transitions and emission probabilities corresponding to the emission of words for each state. In a standard HMM these distributions are represented as multinomial distributions over states and words for transition and emission distributions, respectively, and the parameters of these distributions are set to maximize the likelihood of the data. The Dynamic SBNs (Sallans 2002) instead represent the states as vectors of binary latent variables $S^i = (s^i_1, \ldots, s^i_n)$, and model the transitions and the emission distributions in the log-linear form, as in Equations (7) and (8). Formally, the distribution of words x given the state is given by

$$P(x^i = x|S^i) \propto \exp(\sum_j W_{xj} s^i_j) \tag{9}$$

The distributions of the current state vector S^i given the previous vector S^{i-1} is defined as a product of distributions for individual components s^i_j, and the distributions of these components is defined as in Equation (7):

$$P(s^i_j = 1|S^{i-1}) = \sigma(\sum_{j'} J_{jj'} s^{i-1}_{j'}) \tag{10}$$

Note that the same weight matrices are reused across all the positions due to the stationarity assumption. These weight matrices can be regarded as a template applied to every position of the sequence. A schematic representation of such a dynamic SBN is given in Figure 7.

As with HMMs, all the standard DBNs only allow edges between adjacent (or a bounded window of) positions in the sequence. This limitation on the model structure imposes a Markov assumption on statistical dependencies in the Bayesian network, which would only be appropriate if the derivation decision sequences were Markovian. But derivations for the syntactic and semantic structures of natural language are clearly not Markovian in nature, so such models are not appropriate. ISBNs are not limited to Markovian models because their model structure is specified incrementally as a function of the derivation.

4.1.3 Incrementally Specifying Model Structure. Like DBNs, ISBNs model unboundedly long derivations by connecting together unboundedly many Bayesian network templates, as illustrated in the final graph of Figure 8. But unlike DBNs, the way these templates are connected depends on the structure specified by the derivation. For

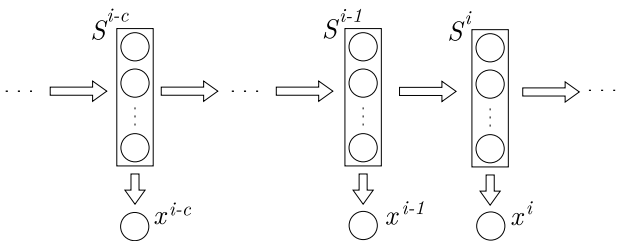

Figure 7
An example of a Dynamic Sigmoid Belief Network.

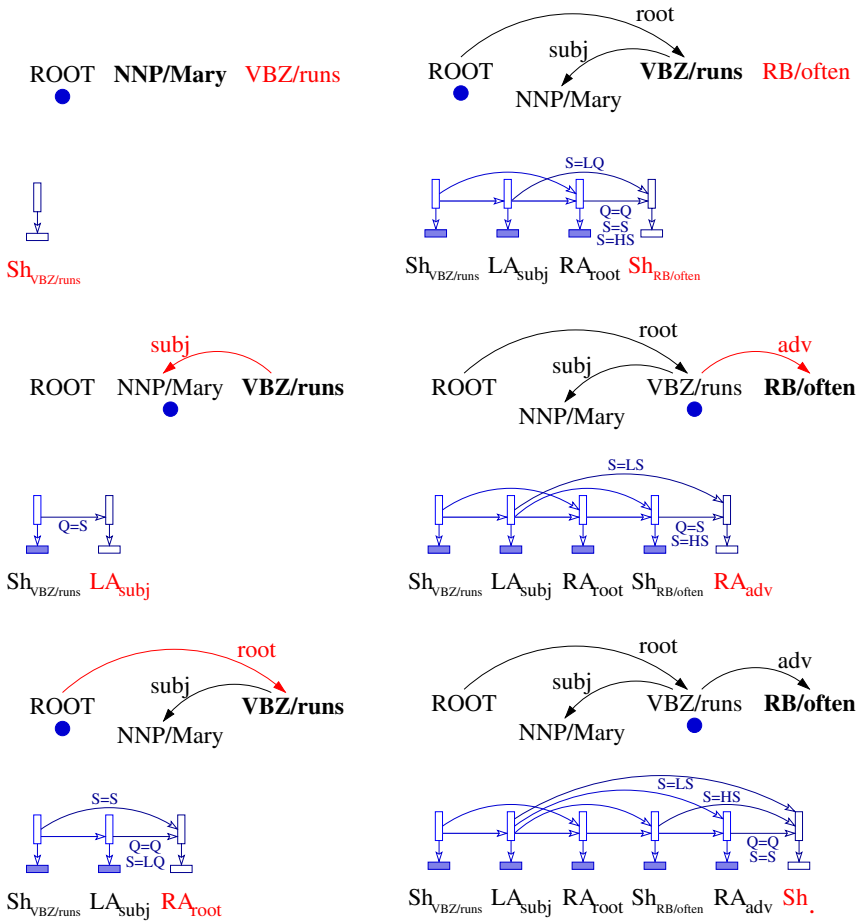

Figure 8
Illustration of the derivation of a syntactic output structure and its associated incremental specification of an ISBN model structure (ordered top-to-bottom, left-to-right). The blue dot indicates the top of the syntactic derivation's stack and the **bold** word indicates the front of the input queue. New model structure edges are labeled with the relationship between their source state and the current state, respectively, with *Q* for queue front, *S* for stack top, *HS* for head of stack top, *LQ* for leftmost child of queue front, and *LS* for leftmost child of stack top.

parsing problems, this means that the structure of the model depends on the structure of the output of parsing. This allows us to build models which reflect the fact that correlations in natural language parsing tend to be local in the syntactic and semantic structures.

In order to have edges in the Bayesian network that reflect locality in the output structure, we need to specify edges based on the actual outputs of the decision sequence $D^1, \ldots, D^m$, not just based on adjacency in this sequence. In ISBNs, the incoming edges for a given position are a discrete function of the sequence of decisions that precede that position, or, equivalently, a discrete function of the partial parse constructed by the previous actions of the parsers. This is why ISBNs are called "incremental" models, not just dynamic models; the structure of the model is determined incrementally as the decision sequence proceeds.

Intuitively, defining this discrete function is very similar to defining a set of history features in a traditional history-based model. In such methods, a model designer decides which previous decisions are relevant to the current one, whereas for ISBNs one needs to define which previous latent parsing states are relevant to the current decision. The crucial difference is that when making this choice in a traditional history-based model, the model designer inevitably makes strong independence assumptions because features that are not included are deemed totally irrelevant. In contrast, ISBNs can avoid such a priori independence assumptions because information can be passed repeatedly from latent variables to latent variables along the edges of the graphical model.[7] Nonetheless, the learning process is biased towards learning correlations with latent states that are close in the chain of edges, so the information that is passed tends to be information which was also useful for the decision made at the previous state. This inductive bias allows the model designer to encode knowledge about the domain in soft biases instead of hard constraints. In the final trained model, the information that is passed to a decision is determined in part on the basis of the data, not entirely on the basis of the model design. The flexibility of this latent variable approach also helps when building new models, such as for new languages or treebanks. The same model can be applied successfully to the new data, as demonstrated in the multilingual experiments that follow, whereas porting the traditional methods across languages would often require substantial feature-engineering effort.

This notion of incremental specification of the model structure is illustrated for syntactic parsing in Figure 8 (the blue directed graphs at the bottom of each panel), along with the partial output structures incrementally specified by the derivation (the black dependency trees in the upper portion of each panel). In Figure 8, the partial output structure also indicates the state of the parser, with the top of the parser's stack indicated by the blue dot and the front of the input queue indicated by the bold word. Red arcs indicate the changes to the structure that result from the parser action chosen in that step. The associated model is used to estimate the probability of this chosen parser action, also shown in red. The edges to the state that is used to make this decision are specified by identifying the most recent previous state that shares some property with this state. In Figure 8, these edges are labeled with the property, such as having the same word on the top of the stack (S=S) or the top of the stack being the same as the current leftmost child of the top of the stack (S=LS).

The argument for the incremental specification of model structure can be applied to any Bayesian network architecture, not just SBNs (e.g., Garg and Henderson 2011). We focus on ISBNs because, as shown in Section 4.1.5, they are closely related to the empirically successful neural network models of Henderson (2003), and they have achieved very good results on the sub-problem of parsing syntactic dependencies (Titov and Henderson 2007d).

4.1.4 ISBNs for Derivations of Structures. The general form of ISBN models that have been proposed for modeling derivations of structures is illustrated in Figure 9. Figure 9 illustrates a situation where we are given a derivation history preceding the elementary decision d_k^i in decision D^i, and we wish to compute a probability distribution for the decision d_k^i, $P(d_k^i|hist(i,k))$. Variables whose values are given are shaded, and latent

7 In particular, our ISBN model for syntactic and semantic derivations makes no hard independence assumptions, because every previous latent state is connected, possibly via intermediate latent variable vectors, to every future state.

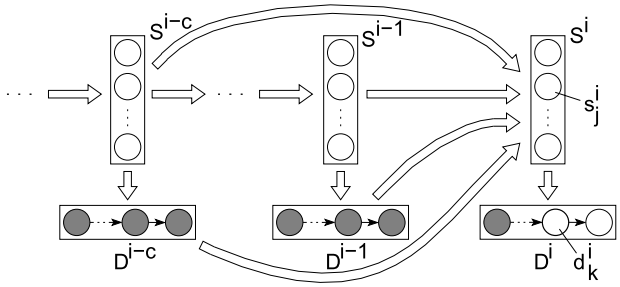

Figure 9
An ISBN for estimating $P(d_k^i|hist(i,k))$—one of the elementary decisions. Variables whose values are given in $hist(i,k)$ are shaded, and latent and current decision variables are unshaded.

and current decision variables are left unshaded. Arrows show how the conditional probability distributions of variables depend on other variables. As discussed earlier, the model includes vectors S^i of latent variables $s^i_{j'}$, which represent features of the parsing history relevant to the current and future decisions.

As illustrated by the arrows in Figure 9, the probability of each latent variable s^i_j depends on all the variables in a finite set of relevant previous latent and decision vectors, but there are no direct dependencies between the different variables in a single latent vector S^i. As discussed in Section 4.1.3, this set of previous latent and decision vectors is specified as a function of the partial parse and parser configuration resulting from the derivation history $D^1, \ldots, D^{i-1}$. This function returns a labeled list of positions in the history that are connected to the current position i. The label of each position $i-c$ in the list represents a relation between the current position i and the positions $i-c$ in the history. We denote this labeled list of positions as $\{R_1(i), \ldots, R_m(i)\}$, where $R_r(i)$ is the position for relation label r. For example, r could be the most recent state where the same word was on the top of the parser's stack, and a decision variable representing that word's part-of-speech tag. Each such selected relation has its own distinct weight matrix for the resulting edges in the graph, but the same weight matrix is used at each position where the relation is relevant (see Section 4.2 for examples of relation types we use in our experiments).

We can write the dependency of a latent variable component s^i_j on previous latent variable vectors and the decision history:

$$P(s^i_j = 1|S^1, \ldots, S^{i-1}, hist(i,1)) = \sigma\left(\sum_{r:\exists R_r(i)} \sum_{j'} J^r_{jj'} s^{R_r(i)}_{j'} + \sum_k B^{rk}_{id^{R_r(i)}_k}\right) \quad (11)$$

where $J^r_{jj'}$ is the latent-to-latent weight matrix and $B^{rk}_{id^{R_r(i)}_k}$ is the decision-to-latent weight matrix for relation r. If there is no previous step that is in relation r to the time step i, then the corresponding index is skipped in the summation, as denoted by the predicate $\exists R_r(i)$. For each relation r, the weight $J^r_{jj'}$ determines the influence of j'th variable $s^{R_r(i)}_{j'}$ in the related previous latent vector $S^{R_r(i)}$ on the distribution of the jth variable s^i_j of the considered latent vector S^i. Similarly, $B^{rk}_{id^{R_r(i)}_k}$ defines the influence of the past decision $d^{R_r(i)}_k$ on the distribution of the considered latent vector component s^i_j.

As indicated in Figure 9, the probability of each elementary decision d_k^i depends both on the current latent vector S^i and on the previously chosen elementary action d_{k-1}^i from D^i. This probability distribution has the normalized exponential form:

$$P(d_k^i = d|S^i, d_{k-1}^i) = \frac{\Phi_{hist(i,k)}(d)\exp(\sum_j W_{dj}s_j^i)}{\sum_{d'}\Phi_{hist(i,k)}(d')\exp(\sum_j W_{d'j}s_j^i)} \quad (12)$$

where $\Phi_{hist(i,k)}$ is the indicator function of the set of elementary decisions that can possibly follow the last decision in the history $hist(i,k)$, and the W_{dj} are the weights of the edges from the latent variables. Φ is essentially switching the output space of the elementary inference problems $P(d_k^i = d|S^i, d_{k-1}^i)$ on the basis of the previous decision. For example, in our generative history-based model of parsing, if decision d_1^i was to create a new node in the tree, then the next possible set of decisions defined by $\Phi_{hist(i,2)}$ will correspond to choosing a node label, whereas if decision d_1^i was to generate a new word then $\Phi_{hist(i,2)}$ will select decisions corresponding to choosing this word.

4.1.5 Approximating Inference in ISBNs. Computing the probability of a derivation, as needed in learning, is straightforward with ISBNs, but not tractable. Inference involves marginalizing out the latent variables, that is, a summation over all possible variable values for all the latent variable vectors. The presence of fully connected latent variable vectors does not allow us to use efficient belief propagation methods (MacKay 2003). Even in the case of dynamic SBNs (i.e., Markovian models), the large size of each individual latent vector would not allow us to perform the marginalization exactly. This makes it clear that we need methods for approximating the inference problems required for parsing.

Previous work on approximate inference in ISBNs has used mean field approximations (Saul, Jaakkola, and Jordan 1996; Titov and Henderson 2007c). In mean field approximations, the joint distribution over all latent variables conditioned on observable variables is approximated using independent distributions for each variable. The parameters that define these individual distributions (the variable's mean values) are set to make the approximate joint distribution as similar as possible to the true joint distribution in terms of the Kullback-Leibler divergence. Unfortunately, there is no closed form solution to finding these means and an iterative estimation procedure involving all the means would be required.

Work on approximate inference in ISBNs has developed two mean field approximations for estimating the decision probabilities $P(d_k^i|hist(i,k))$ (Titov and Henderson 2007c), one more accurate and one more efficient. Titov and Henderson (2007c) show that their more accurate approximation leads to more accurate parsers, but the improvement is small and the computational cost is high. Because we need to build larger more complex models than those considered by Titov and Henderson (2007c), in this article we only make use of the more efficient approximation.

The more efficient approximation assumes that each variable's mean can be effectively tuned by only considering the means of its parent variables (i.e., the variables with edges directed to the variable in question). This assumption leads to a closed form solution to minimizing the Kullback-Leibler divergence between the approximate and true distributions. This closed form solution replicates exactly the computation of the feed-forward neural network model of Henderson (2003), where the neural

network hidden unit activations are the means of the individual variable's distributions. So, instead of Equations (11) and (12), the computations of the approximate model are

$$\mu_j^i = \sigma\left(\sum_{r:\exists R_r(i)} \sum_{j'} J^r_{jj'} \mu_{j'}^{R_r(i)} + \sum_k B^{rk}_{id_k^{R_r(i)}}\right) \tag{13}$$

$$P(d_k^i = d | S^i, d_{k-1}^i) = \frac{\Phi_{hist_{(i,k)}}(d) \exp(\sum_j W_{dj} \mu_j^i)}{\sum_{d'} \Phi_{hist_{(i,k)}}(d') \exp(\sum_j W_{d'j} \mu_j^i)} \tag{14}$$

where μ_j is the mean parameter of the latent variables s_j. Consequently, the neural network probability model can be regarded as a fast approximation to the ISBN graphical model.

This feed-forward approximation does not update the latent vector means for positions $i' \leq i$ after observing a decision d_k^i, so information about decision d_k^i does not propagate back to its associated latent vector S^i. In the model design, edges from decision variables directly to subsequent latent variables (see Figure 9) are used to mitigate this limitation. We refer the interested reader to Garg and Henderson (2011) for a discussion of this limitation and an alternative architecture that avoids it.

4.2 ISBNs for Syntactic–Semantic Parsing

In this section we describe how we use the ISBN architecture to design a joint model of syntactic–semantic dependency parsing. In traditional fully supervised parsing models, designing a joint syntactic–semantic parsing model would require extensive feature engineering. These features pick out parts of the corpus annotation that are relevant to predicting other parts of the corpus annotation. If features are missing then predicting the annotation cannot be done accurately, and if there are too many features then the model cannot be learned accurately. Latent variable models, such as ISBNs and Latent PCFGs (Matsuzaki, Miyao, and Tsujii 2005; Petrov et al. 2006), have the advantage that the model can induce new, more predictive, features by composing elementary features, or propagate information to include predictive but non-local features. These latent annotations are induced during learning, allowing the model to both predict them from other parts of the annotation and use them to predict the desired corpus annotation. In ISBNs, we use latent variables to induce features of the parse history $D^1, \ldots, D^{i-1}$ that are used to predict future parser decisions $D^i, \ldots, D^m$.

The main difference between ISBNs and Latent PCFGs is that ISBNs have vectors of latent features instead of latent atomic categories. To train a Latent PCFG, the learning method must search the space of possible latent atomic categories and find good configurations of these categories in the different PCFG rules. This has proved to be difficult, with good performance only being achieved using sophisticated induction methods, such as split-merge (Petrov et al. 2006). In contrast, comparable accuracies have been achieved with ISBNs using simple gradient descent learning to induce their latent feature spaces, even with large numbers of binary features (e.g., 80 or 100) (Henderson and Titov 2010). This ability to effectively search a large informative

space of latent variables is important for our model because we are relying on the latent variables to capture complex interactions between and within the syntactic and semantic structures.

The ability of ISBNs to induce features of the parse history that are relevant to the future decisions avoids reliance on the system designer coming up with hand-crafted features. ISBNs still allow the model designer to influence the types of features that are learned through the design of the ISBN model structure, however—illustrated as arrows in Figure 9 and as the blue arrows between states in Figure 8. An arrow indicates which properties of the derivation history $D^1, \ldots, D^{i-1}$ are directly input to the conditional probability distribution of a vector of latent variables S^i. There are two types of properties: predefined features extracted from the previous decisions $D^1, \ldots, D^{i-1}$, and latent feature vectors computed at a previous position $i-c$ of the derivation. In either case, there are a fixed number of these relevant properties.

Choosing the set of relevant previous latent vectors is one of the main design decisions in building an ISBN model. By connecting to a previous latent vector, we allow the model to directly exploit features that have been induced for making that latent vector's decision. Therefore, we need to choose the set of connected latent vectors in accordance with our prior knowledge about which previous decisions are likely to induce latent features that are particularly relevant to the current decision. This design choice is illustrated for dependency parsing in Figure 8, where the model designer has chosen to condition each latent vector on previous latent vectors whose associated partial parse and parser configuration share some property with the current partial parse and parser configuration.

For syntactic-semantic dependency parsing, each of the two individual derivations is mapped to a set of edges in the ISBN in a similar way to that for syntactic dependency parsing. In addition, there are edges that condition each of the two derivations on latent representations and decisions from the other derivation. Both these types of connections are shown in Figure 10. Conditioning on latent representations from the other task allows the correlations between derivations to be captured automatically. In addition, by training the two derivations jointly, the model is able to share induced representations of auxiliary subproblems between the two tasks. For example, many selectional preferences for the syntactic arguments of verbs are semantic in nature, and inducing these semantic distinctions may be easier by combining evidence from both syntax and semantic roles. The presence of these edges between semantic and syntactic states enables our systems to learn these common representations, as needed for multi-task learning.

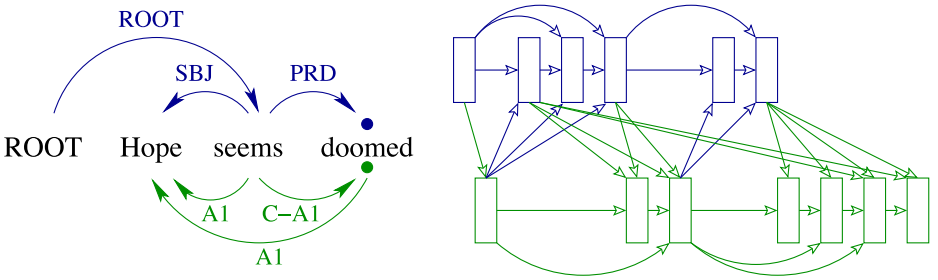

Figure 10
Illustration of the final state of a derivation of a syntactic–semantic structure and the associated ISBN model structure. Only the vectors of latent variables are shown in the model structure.

For the synchronized shift-reduce dependency structure derivations presented in Section 3.1, we distinguish between syntactic states (positions where syntactic decisions are considered, shown in blue, the upper row, in Figure 10) and semantic states (positions where semantic decisions are considered, shown in green, the lower row, in Figure 10). For syntactic states, we assume that the induced latent features primarily relate to the word on the top of the syntactic stack and the word at the front of the queue. Similarly, for semantic states, we assume that the induced latent features primarily relate to the word on the top of the semantic stack and the word at the front of the queue. To decide which previous state's latent features are most relevant to the current decision, we look at these words and words that are structurally local to them in the current partial dependency structure specified by the derivation history. For each such word that we choose as relevant to the current decision, we look for previous states where the stack top or the queue front was the same word. If more than one previous state matches, then the latent vector of the most recent one is used. If no state matches, then no connection is made.

The specific connections between latent vectors that we use in our experiments are specified in Table 2. The second column specifies the relevant word from the current partial dependency structure. The first column specifies what role that word needs to have played at the previous state. For example, the first row indicates edges between the current latent vector and the most recent previous latent vector (if any) that had the same queue front as the current one. The remaining columns distinguish between the cases where the previous and/or current states are for making syntactic and/or semantic decisions, with a "+" indicating that, for the column's state types, the row's relation type is included in the model. For example, the first row indicates that these edges exist within syntactic states, from semantic to syntactic states, within semantic states, and from syntactic to semantic states. As another example, the third cell of the third row indicates that there are edges in the ISBN between the current semantic state and the most recent semantic state where the top of the semantic stack was the same word as the current rightmost dependent of the current top of the semantic stack. Each cell of this table has a distinct weight matrix for the resulting edges in the ISBN, but the same weight matrix is used at each state where the relation applies. Training and testing times asymptotically scale linearly with the number of relations.

In addition to these latent-to-latent edges, the ISBN also conditions latent feature vectors on a set of predefined features extracted from the history of previous decisions. These features are specified in Table 3. They are lexical and syntactic features of the top of the stack and front of the queue, and their respective heads, children, and siblings in the syntactic dependency structure. For the semantic stack, the position immediately

Table 2
Latent-to-latent variable connections. Queue = front of the input queue; Top = top of the stack.

Closest	Current	Syn-Syn	Sem-Syn	Sem-Sem	Syn-Sem
Queue	Queue	+	+	+	+
Top	Top	+	+	+	+
Top	Rightmost right dependent of top	+		+	
Top	Leftmost left dependent of top	+		+	
Top	Head of top	+		+	
Top	Leftmost dependent of queue	+		+	
Queue	Top	+			

Table 3
Predefined features. The syntactic features must be interpreted as applying only to the nodes on the syntactic stack, and the semantic features apply only to the nodes on the semantic stack. Queue = front of the input queue; Top = top of stack; Top−1 = the element immediately below the top of stack. LEX = word; POS = part of speech; DEP = dependency label; FRAMESET = predicate sense.

State	Syntactic step features		
	LEX	POS	DEP
Queue	+	+	
Top	+	+	
Top−1		+	
Head of top	+		
Rightmost dependent of top			+
Leftmost dependent of top			+
Leftmost dependent of queue			+

State	Semantic step features			
	LEX	POS	DEP	FRAMESET
Queue	+	+	+	+
Top	+	+	+	+
Top−1	+	+	+	
Leftmost dependent of queue			+	
Head of top/top−1	+	+	+	
Head of queue	+	+	+	
Rightmost dependent of top/top−1			+	
Leftmost dependent of top/top−1			+	
Left sibling of top/top−1		+	+	
Left sibling of queue		+	+	
Right sibling of top/top−1		+	+	
Right sibling of queue		+	+	

below the top of the stack is also very important, because of the *Swap* operation. To capture the intuition that the set of arguments in a given predicate-argument structure should be learned jointly because of the influence that each argument has on the others, we introduce siblings as features of the node that is being attached. The model distinguishes argument role labels for nominal predicates from argument role labels for verbal predicates.

We investigated the contribution of the features, to test whether all the features indicated in Table 3 are actually useful. We tried several different groups of features. The different groups are as indicated in the table with additional spacing between lines. These groups are to be interpreted inclusively of that group and all preceding groups. So we tried groups of features concerning top, top−1, and front of the queue; features of these elements and also of their heads; features of the nodes and their heads as well as their children; and finally we also added features that make reference to the siblings. We found that the best performing feature set is the most complete. This result confirms linguistic properties of semantic role assignment that would predict that semantic roles benefit from knowledge about siblings. It also confirms that the best results are obtained when assigning SRL jointly to all arguments in a proposition (Toutanova, Haghighi,

and Manning 2008). In all the experiments reported in Section 6, we use the complete feature set.

5. Learning and Parsing

In this section we briefly describe how we estimate the parameters of our model, and how we search for the most probable syntactic–semantic graph given the trained model.

5.1 Learning

We train the ISBN to maximize the fit of the approximate model to the data. Thus, both at parsing time and at training time, the parameters of the model are interpreted according to the feed-forward approximation discussed in Section 4.1.5, and not according to the exact latent variable interpretation of ISBNs. We train these parameters to optimize a maximum likelihood objective function, $\log P(T_d, T_s)$. We use stochastic gradient descent, which requires computing the derivative of the objective function with respect to each parameter, for each training example.

For the feed-forward approximation we use, computation of these derivatives is straightforward, as in neural networks (Rumelhart, Hinton, and Williams 1986). Thus, we use the neural network Backpropagation algorithm for training. The error from all decisions is propagated back through the structure of the graphical model and used to update all parameters in a single pass, so Backpropagation is linear in derivation length. Standard techniques for improving Backpropagation, such as momentum and weight decay regularization, are also used. Momentum makes the gradient descent less stochastic, thereby speeding convergence. Weight decay regularization is equivalent to a Gaussian prior over parameter values, centered at zero. Bias terms are not regularized.

5.2 Parsing

ISBNs define a probability model that does not assume independence between any decision variables, because ISBNs induce latent variables that might capture any such statistical dependency. This property leads to the complexity of complete search being exponential in the number of derivation steps. Fortunately, for many problems, such as natural language parsing, efficient heuristic search methods are possible.

Given a trained ISBN as our probability estimator, we search for the most probable joint syntactic–semantic dependency structure using a best-first search with the search space pruned in two different ways. First, only a fixed beam of the most probable partial derivations are pursued after each word *Shift* operation. That is, after predicting each chunk,[8] we prune the set of partial analyses to some fixed beam width K_1. This width K_1 can be kept small (under 100) without affecting accuracies, and very small beams (under 5) can be used for faster parsing. Even within each chunk (i.e., between *Shift* operations), however, it is hard to use the exhaustive search as each of the K_1 partial analyses can be expanded in an unbounded number of ways. So, we add a second pruning stage. We limit the branching factor at each considered parsing action. That is, for every partial analysis, we consider only K_2 possible next actions. Again this parameter can be kept small (we use 3) without affecting accuracies.

8 See Section 3.1 for our definition of a chunk.

Global constraints (such as uniqueness of certain semantic arguments) are not enforced by the parsing strategy. The power of the ISBN architecture seems to allow the model to learn to enforce these constraints itself, which Merlo and Musillo (2008) found to be adequate. Also, the parsing strategy does not attempt to sum over different derivations for the same structure, and does not try to optimize any measure other than exact match for the complete syntactic–semantic structure.

6. Monolingual and Multilingual Experiments

To test the design of the syntax semantic interface and the use of a latent variable model, we train and evaluate our models on data provided for the CoNLL-2008 shared task on joint learning of syntactic and semantic dependencies for English. Furthermore, we test the cross-linguistic generality of these models on data from the CoNLL-2009 shared task for seven languages.[9]

In our experiments, we use the measures of performance used in the CoNLL-2008 and CoNLL-2009 shared tasks, typical of dependency parsing and semantic role labeling. Syntactic performance is measured by the percentage of correct labeled attachments (LAS in the tables). Semantic performance is indicated by the F-measure on precision and recall on semantic arcs plus predicate sense labels (indicated as Semantic measures in the table). For the CoNLL-2008 scores the predicate sense labeling includes predicate identification, but for the CoNLL-2009 scores predicate identification was given in the task input. The syntactic LAS and the semantic F_1 are then averaged with equal weight to produce an overall score called Macro F_1.[10] When we evaluate the impact of the *Swap* action on crossing arcs, we also calculate precision, recall, and F-measure on pairs of crossing arcs.[11] In our experiments, the statistical significance levels we report are all computed using a stratified shuffling test (Cohen 1995; Yeh 2000) with 10,000 randomized trials.

6.1 Monolingual Experimental Set-up

We start by describing the monolingual English experiments. We train and evaluate our English models on data provided for the CoNLL-2008 shared task on joint learning of syntactic and semantic dependencies. The data is derived by merging a dependency transformation of the Penn Treebank with PropBank and NomBank (Surdeanu et al. 2008). An illustrative example of the kind of labeled structures that we need to parse is given in Figure 3. Training, development, and test data follow the usual partition as sections 02–21, 24, and 23 of the Penn Treebank, respectively. More details and references on the data, on the conversion of the Penn Treebank format to dependencies, and on the experimental set-up are given in Surdeanu et al. (2008).

We set the size of the latent variable vector to 80 units, and the word frequency cut-off to 20, resulting in a vocabulary of only 4,000 words. These two parameters were chosen initially based on previous experience with syntactic dependency parsing (Titov

9 Code and models for the experiments on the CoNLL-2009 shared task data are available at `http://clcl.unige.ch/SOFTWARE.html`.

10 It should be pointed out that, despite the name, this Macro F_1 is not a harmonic mean. Also, this measure does not evaluate the syntactic and semantic parts jointly, hence it does not guarantee coherence of the two parts. In practice, the better the syntactic and semantic parts, the more they will be coherent, as indicated by the exact match measure.

11 In the case of multiple crossings, an arc can be a member of more than one pair.

and Henderson 2007b, 2007d). Additionally, preliminary experiments on the development set indicated that larger cut-offs and smaller dimensionality of the latent variable vector results in a sizable decrease in performance. We did not experiment with decreasing cut-off parameters or increasing the latent space dimensionality beyond these values as it would adversely affect the efficiency of the model. The efficiency of the model is discussed in more detail in Section 6.5.

We use a beam size of 50 to prune derivations after each *Shift* operation, and a branching factor of 3. Larger beam sizes, within a tractable range, did not seem to result in any noticeable improvement in performance on the held-out development set. We compare several experiments in which we manipulate the connectivity of the model and the allowed operations.

6.2 Joint Learning and the Connectivity of the Model

The main idea inspiring our model of parsing syntactic and semantic dependencies is that these two levels of representations are closely correlated and that they should be learned together. Moreover, because the exact nature of these correlations is not always understood or is too complex to annotate explicitly, we learn them through latent variables. Similarly, we argued that the latent representation can act as a shared representation needed for successful multi-task learning.

The first set of monolingual experiments, then, validates the latent-variable model, specifically its pattern of connectivity within levels of representation and across levels. We tested three different connectivity models by performing two ablation studies. In these experiments, we compare the full connectivity and full power of latent variable joint learning to a model where the connections from syntax to semantics, indicated as the Syn-Sem connections in Table 2, were removed, and to a second model where all the connections to the semantic layer—both those coming from syntax and those between semantic decisions, indicated as the Sem-Sem and Syn-Sem connections in Table 2—were removed. While in all these models the connections between the latent vectors specified in Table 2 were modified, the set of explicit features defined in Table 3 was left unchanged. This is a rich set of explicit features that includes features of the syntax relevant to semantic decisions, so, although we expect a degradation, we also expect that it is still possible, to a certain extent, to produce accurate semantic decisions without exploiting latent-to-latent connections. Also, for all these models, parsing searches for the most probable joint analysis of syntactic and semantic dependencies.

Results of these experiments are shown in Table 4, indicating that there is a degradation in performance in the ablated models. Both the differences in the Semantic recall and F_1 scores and the differences in the Macro recall and F_1 scores between the fully connected model (first line) and the model with semantic connections only (second line)

Table 4
Scores on the development set of the CoNLL-2008 shared task (percentages).

	Syntactic	Semantic			Macro		
	LAS	P	R	F_1	P	R	F_1
Fully connected	86.6	79.6	73.1	76.2	83.1	79.9	81.5
No connections syntax to semantics	86.6	79.5	70.9	74.9	83.0	78.8	80.8
No connections to semantics	86.6	79.5	70.1	74.5	83.0	78.3	80.6

are statistically significant at $p = 0.05$. Between the model with no connections from syntax (second line) and the one where all the connections to semantics are removed (third line), the differences between the Semantic recall and F_1 scores and the difference between the Macro F_1 scores are statistically significant at $p = 0.05$.

These results enable us to draw several conclusions. First, the fact that the model with the full connections reaches better performance than the ablated one with no connections from syntax to semantics shows that latent variables do facilitate the joint learning of syntax and semantics (Table 4, first vs. second line). This result shows that joint learning can be beneficial to parsing syntactic and semantic representations. Only the fully connected model allows the learning of the two derivations to influence each other; without the latent-to-latent connections between syntax and semantics, each half of the model can be trained independently of the other. Also, this result cannot be explained as an effect of joint decoding, because both models use a parsing algorithm that maximizes the joint probability. Secondly, the second ablation study indicates that semantic connections do not help much above the presence of a rich set of semantic and syntactic features (Table 4, second vs. third line). Also, the fact that the degradation of the ablated models results mostly in a decrease in recall indicates that, in a situation of more limited information, the system is choosing the safer option of not outputting any label. This is the default option as the semantic annotation is very sparse.

We also find that joint learning does not significantly degrade the accuracy of the syntactic parsing model. To test this, we trained a syntactic parsing model with the same features and the same pattern of interconnections as used for the syntactic states in our joint model. The resulting labeled attachment score was non-significantly better (0.2%) than the score for the joint model. Even if this difference is not noise, it could easily be explained as an effect of joint decoding, rather than joint learning, because decoding with the syntax-only model optimizes just the syntactic probability. Indeed, Henderson et al. (2008) found a larger degradation in syntactic accuracy as a direct result of joint decoding, and even a small improvement in syntactic accuracy as a result of joint learning with semantic roles if decoding optimizes just the syntactic probability, by marginalizing out the semantics during decoding with the joint model.[12]

The standard measures used in the CoNLL-2008 and CoNLL-2009 shared tasks to evaluate semantic performance score semantic arcs independently of one another and ignored the whole propositional argument-structure of the predicates. As suggested in Toutanova, Haghighi, and Manning (2008), such measures are only indirectly relevant to those potential applications of semantic role labeling such as information extraction and question answering that require the whole propositional content associated with a predicate to be recovered in order to be effective.

To address this issue with the standard measures of semantic performance and further clarify the differences in performance between the three distinct connectivity models, we report precision, recall, and F-measure on whole propositions consisting of a predicate and all its core arguments and modifiers. These measures are indicated as Proposition measures in Table 5. According to these measures, a predicted proposition is correct only if it exactly matches a corresponding proposition in the gold-standard data set.

12 This result was for a less interconnected model than the one we use here. This allowed them to compute the marginalization efficiently, whereas this would not be possible in our model. Hence, we did not attempt to perform this type of decoding for our joint model.

Table 5
Proposition scores on the development set of the CoNLL-2008 shared task (percentages).

	Proposition		
	P	R	F_1
Fully connected	49.0	46.5	47.7
No connections syntax to semantics	48.0	44.3	46.1
No connections within semantics	45.8	42.2	43.9

These results are reported in Table 5. The differences in precision, recall, and F_1 are all statistically significant at $p = 0.05$. These results clearly indicate that the connectivity of latent vectors both within representational layers and across them influences the accuracy of recovering the whole propositional content associated with predicates. In particular, our model connecting the latent vectors within the semantic layer significantly improves both the precision and the recall of the predicted propositions over the model where these connections are removed (second vs. third line). Furthermore, the model integrating both the connections from syntax to semantics and the connections within semantics significantly outperforms the model with no connections from syntax to semantics (first vs. second line). Overall, these results suggest that whole propositions are best learned jointly by connecting latent vectors, even when these latent vectors are conditioned on a rich set of predefined features, including semantic siblings.

Table 4 and Table 5 together suggest that although the three models output a similar number of correct argument labels (semantic P column of Table 4), the mistakes are not uniformly distributed across sentences and propositions in the three models. We hypothesize that the ablated models are more often correct on the easy cases, whereas the fully connected model is more able to learn complex regularities.

To test this intuition we develop a measure of sentence complexity, and we disaggregate the accuracy results according to the different levels of complexity. Sentence complexity is measured in two different ways, as the total number of propositions in a sentence, and as the total number of arguments and predicates in the sentence. We also vary the measure of performance: We calculate the F_1 of correct propositions and the usual arguments and predicates semantic F_1 measure.

Results are reported in Figure 11, which plots the F_1 values against the sentence complexity measures. Precision and recall are not reported as they show the same trends. These results confirm that there is a trend for better performance in the complex cases for the full model compared with the other two models. For simpler sentences, the explicit features are apparently adequate to perform at least as well as the full model, and sometimes better. But for complex sentences, the ability to pass information through the latent variables gives the full model an advantage. The effect is robust as it is confirmed for both methods of measuring complexity and both methods of measuring performance.

From this set of experiments and analyses, we can conclude that our system successfully learns a common hidden representation for this multitask learning problem, and thereby achieves important gains from joint parameter estimation. We found these gains only in semantic role labeling. Although the syntactic parses produced were different for the different models, in these experiments the total syntactic accuracy was on average the same across models. This does not imply, however, that joint learning of

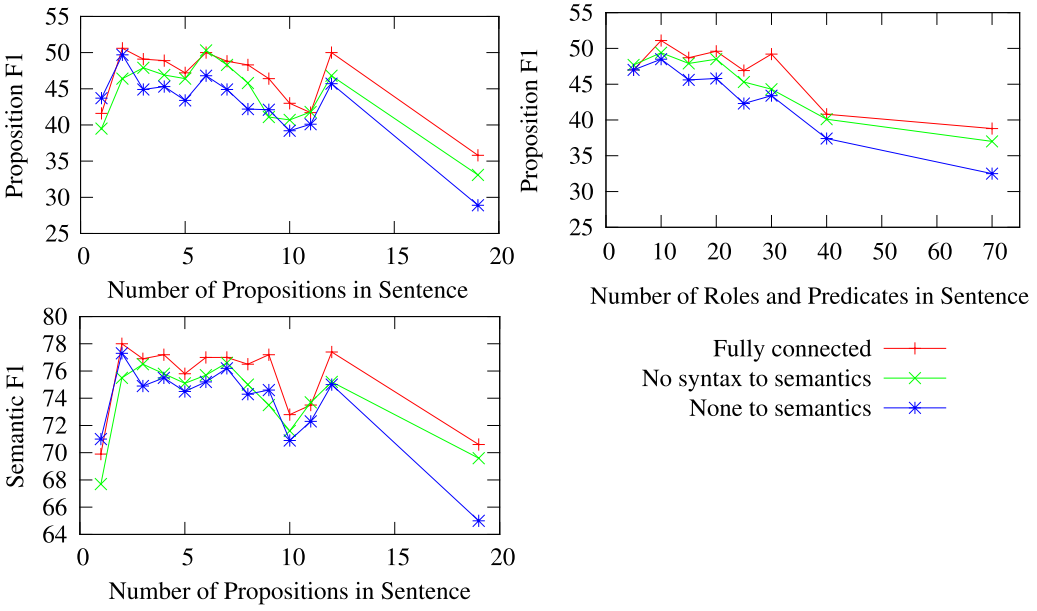

Figure 11
Plots of how the parser accuracy varies as the semantic complexity of sentences vary. The *y*-axis values are calculated by binning sentences according to their *x*-axis values, with the plotted points showing the maximum value of each bin.

the syntactic latent representations was not useful. The fact that adding connections to semantics from the syntactic latent variables results in changes in syntactic parses and large gains in semantic accuracy suggests that joint learning adapts the syntactic latent variables to the needs of semantic parsing decisions.

6.3 Usefulness of the *Swap* Operation

One specific adaptation of our model to processing the specific nature of semantic dependency graphs was the introduction of the new *Swap* action. To test the usefulness of this additional action, we compare several experiments in which we manipulate different variants of on-line planarization techniques for the semantic component of the model. These experiments were run on the development set. The models are listed in Table 6. We compare the use of the *Swap* operation to two baselines. The first baseline (second line) uses Nivre and Nilsson's (2005) HEAD label propagation technique to planarize the syntactic tree, extended to semantic graphs following Henderson et al. (2008). The second baseline is an even simpler baseline that only allows planar graphs, and therefore fails on non-planar graphs (third line). In training, if a model fails to parse an entire sentence, it is still trained on the partial derivation.

The results of these experiments are shown in Table 6. The results are clear. If we look at the left panel of Table 6 (CoNLL Measures), we see that the *Swap* operation performs the best, with this on-line planarization outperforming the extension of Nivre's HEAD technique to semantic graphs (second line) and the simplistic baseline. Clearly, the improvement is due to better recall on the crossing arcs, as shown by the right-hand panel.

Table 6
Scores on the development set (percentages).

	CoNLL Measures			Crossing arcs		
Technique	Syntactic	Semantic	Macro	Semantics		
	LAS	F_1	F_1	P	R	F_1
Swap	86.6	76.2	81.5	61.5	25.6	36.1
HEAD	86.7	73.3	80.1	78.6	2.2	4.2
Planar	85.9	72.8	79.4	undefined	0	undefined

6.4 Monolingual Test Set Results

The previous experiments were both run on the development set. The best performing model used the full set of connections and the *Swap* operation. This model was then tested on the test set from the CoNLL-2008 shared task. Results of all the experiments on the test sets are summarized in Table 7. These results on the complete test set (WSJ+Brown) are compared with some models that participated in the CoNLL-2008 shared task in Table 8. The models listed were chosen among the 20 participating systems either because they had better results or because they learned the two representations jointly, as will be discussed in Section 7.

One comparison in Table 8 that is relevant to the discussion of the properties of our system is the comparison to our own previous model, which did not use the *Swap* operation, but used the HEAD planarization method instead (Henderson et al. 2008). Although the already competitive syntactic performance is not significantly degraded by adding the *Swap* operation, there is a large improvement of 3% on the semantic graphs. This score approaches those of the best systems. As the right-hand panel on crossing arcs indicates, this improvement is due to better recall on crossing arcs.

In this article, we have explored the hypothesis that complex syntactic–semantic representations can be learned jointly and that the complex relationship between these two levels of representation and between the two tasks is better captured through latent variables. Although these experiments clearly indicate that, in our system, joint learning of syntax and semantics performs better than the models without joint learning, four systems in the CoNLL-2008 shared task can report better performance for English than what is described in this article. These results are shown in the CoNLL Measures column of Table 8.

The best performing system learns the two representations separately, with a pipeline of state-of-the-art systems, and then reranks the joint representation in a

Table 7
Scores of the fully connected model on the final testing sets of the CoNLL-2008 shared task (percentages).

	Syntactic	Semantic			Macro		
	LAS	P	R	F_1	P	R	F_1
WSJ	88.4	79.9	75.5	77.6	84.2	82.0	83.0
Brown	80.4	65.9	60.8	63.3	73.1	70.6	71.8
WSJ+Brown	87.5	78.4	73.9	76.1	83.0	80.7	81.8

Table 8
Comparison with other models on the CoNLL-2008 test set (percentages).

	CoNLL Measures			Crossing Arcs		
Model	Synt	Semantic	Macro	Semantics		
	LAS	F_1	F_1	P	R	F_1
Johansson and Nugues (2008b)	89.3	81.6	85.5	67.0	44.5	53.5
Ciaramita et al. (2008)	87.4	78.0	82.7	59.9	34.2	43.5
Che et al. (2008)	86.7	78.5	82.7	56.9	32.4	41.3
Zhao and Kit (2008)	87.7	76.7	82.2	58.5	36.1	44.6
This article	87.5	76.1	81.8	62.1	29.4	39.9
Henderson et al. (2008)	87.6	73.1	80.5	72.6	1.7	3.3
Lluís and Màrquez (2008)	85.8	70.3	78.1	53.8	19.2	28.3

final step (Johansson and Nugues 2008b). Similarly, Che et al. (2008) also implement a pipeline consisting of state-of-the-art components where the final inference stage is performed using Integer Linear Programming to ensure global coherence of the output. The other two better performing systems use ensemble learning techniques (Ciaramita et al. 2008; Zhao and Kit 2008). Comparing our system to these other systems on a benchmark task for English, we can confirm that joint learning is a promising technique, but that on this task it does not outperform reranking or ensemble techniques. Our system's architecture is, however, simpler, in that it consists of a single generative model. We conjecture that the total development time for our system is consequently much lower, if the development time for all the components are added up.

These competitive results, despite using a relatively simple architecture and a relatively small vocabulary, indicate the success of our approach of synchronizing two separate derivations and using latent variables to learn the correlations. This success is achieved despite the model's fairly weak assumptions about the nature of these correlations, thus demonstrating that this architecture is clearly very adaptive and provides a strong form of smoothing. These are important properties, particularly when developing new systems for languages or annotations that have not received the intensive development effort that has English Penn Treebank syntactic parsing and English PropBank semantic role labeling. In the next section, we test the extent of this robustness by using the same approach to build parsers for several languages, and compare against other approaches when they are required to produce systems for multiple languages and annotations.

6.5 Multilingual Experiments

The availability of syntactically annotated corpora for multiple languages (Nivre et al. 2007) has provided a new opportunity for evaluating the cross-linguistic validity of statistical models of syntactic structure. This opportunity has been significantly expanded with the creation and annotation of syntactic and semantic resources in seven languages (Hajič et al. 2009) belonging to several different language families. This data set was released for the CoNLL-2009 shared task.

To evaluate the ability of our model to generalize across languages, we take the model as it was developed for English and apply it directly to all of the six other

languages.[13] The only adaptation of the code was done to handle differences in the data format. Although this consistency across languages was not a requirement of the shared task—individual-language optimization was allowed, and indeed was performed by many teams—the use of latent variables to induce features automatically from the data gives our method the adaptability necessary to perform well across all seven languages, and demonstrates the lack of language specificity in the models.

The data and set-up correspond to the joint task of the closed challenge of the CoNLL-2009 shared task, as described in Hajič et al. (2009).[14] The scoring measures are the same as those for the previous experiments.

We made two modifications to reflect differences in the annotation of these data from the experiments reported in the previous section (based on CoNLL-2008 shared task data). The system was adapted to use two features not provided in the previous shared task: automatically predicted morphological features[15] and features specifying which words were annotated as predicates.[16] Both these features resulted in improved accuracy for all the languages. We also made use of one type of feature that had previously been found not to result in any improvement for English, but resulted in some overall improvement across the languages.[17]

Also, in comparison with previous experiments, the search beam used in the parsing phase was increased from 50 to up to 80, producing a small improvement in the overall development score. The vocabulary frequency cut-off was also changed to 5, from 20. All the development effort to change from the English-only 2008 task to the multilingual 2009 task took about two person-months, mostly by someone who had no previous experience with the system. Most of this time was spent on the differences in the task definition between the 2008 and 2009 shared tasks.

The official results on the testing set and out of domain data are shown in Tables 9, 10, and 11. The best results across systems participating in the CoNLL-2009 shared task are shown in bold. There was only a 0.5% difference between our average macro F_1 score and that of the best system, and there was a 1.29% difference between our score and the fourth ranked system. The differences between our average scores reported in Tables 9, 10, and 11 and the average scores achieved by the other systems participating in the shared task are all statistically significant at $p = 0.05$.

13 An initial report on this work was presented in the CoNLL-2009 Shared Task volume (Gesmundo et al. 2009).

14 The data sets used in this challenge are described in Taulé, Martì, and Recasens (2008) (Catalan and Spanish), Xue and Palmer (2009) (Chinese), Hajič (2004), Čmejrek, Hajič, and Kuboň (2004) (Czech), Surdeanu et al. (2008) (English), Burchardt et al. (2006) (German), and Kawahara, Sadao, and Hasida (2002) (Japanese).

15 Morphological features of a word are not conditionally independent. To integrate them into a generative model, one needs to either make some independence assumptions or model sets of features as atomic feature bundles. In our model, morphological features are treated as an atomic bundle, when computing the probability of the word before shifting the previous word to the stack. When estimating probabilities of future actions, however, we condition latent variables on elementary morphological features of the words.

16 Because the testing data included a specification of which words were annotated as predicates, we constrained the parser's output so as to be consistent with this specification. For rare predicates, if the predicate was not in the parser's lexicon (extracted from the training set), then a frameset was taken from the list of framesets reported in the resources available for the closed challenge. If this information was not available, then a default frameset name was constructed based on the automatically predicted lemma of the predicate.

17 When predicting a semantic arc between the word on the front of the queue and the word on the top of the stack, these features explicitly specify any syntactic dependency already predicted between the same two words.

Table 9
The three main scores for our system. Rank indicates ranking in the CoNLL 2009 shared task. Best results across systems are marked in **bold**.

	Rank	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
Macro F_1	3	82.14	82.66	76.15	83.21	86.03	79.59	84.91	82.43
Syntactic LAS	1	**85.77**	**87.86**	76.11	**80.38**	88.79	87.29	92.34	**87.64**
Semantic F_1	3	78.42	77.44	76.05	86.02	83.24	71.78	77.23	77.19

Table 10
Semantic precision and recall and macro precision and recall for our system. Rank indicates ranking in the CoNLL-2009 shared task. Best results across systems are marked in **bold**.

	Rank	Ave	Catalan	Chinese	Czech	English	German	Japanese	Spanish
semantic Prec	3	81.60	79.08	80.93	87.45	84.92	75.60	83.75	79.44
semantic Rec	3	75.56	75.87	71.73	**84.64**	81.63	68.33	71.65	75.05
macro Prec	2	83.68	83.47	78.52	83.91	86.86	81.44	88.05	83.54
macro Rec	3	80.66	**81.86**	73.92	**82.51**	85.21	77.81	81.99	81.35

Table 11
Results on out-of-domain for our system. Rank indicates ranking in the CoNLL-2009 shared task. Best results across systems are marked in **bold**.

	Rank	Ave	Czech-ood	English-ood	German-ood
Macro F_1	3	75.93	**80.70**	75.76	71.32
Syntactic LAS	2	78.01	**76.41**	80.84	76.77
Semantic F_1	3	73.63	84.99	70.65	65.25

Despite the good results, a more detailed analysis of the source of errors seems to indicate that our system is still having trouble with crossing dependencies, even after the introduction of the *Swap* operation. In Table 8, our recall on English crossing semantic dependencies is relatively low. Some statistics that illustrate the nature of the input and could explain some of the errors are shown in Table 12. As can be observed, semantic representations often have many more crossing arcs than syntactic ones, and they often do not form a fully connected tree, as each proposition is represented by an independent treelet. We observe that, with the exception of German, we do relatively well on those languages that do not have crossing arcs, such as Catalan and Spanish, or have even large amounts of crossing arcs that can be parsed with the *Swap* operation, such as Czech. As indicated in Table 12, only 2% of Czech sentences are unparsable, despite 16% requiring the *Swap* action.

6.6 Experiments on Training and Parsing Speed

The training and parsing times for our models are reported in Table 13, using the same meta-parameters (discussed subsequently) as for the accuracies reported in the previous section, which optimize accuracy at the expense of speed. Training times are mostly affected by data-set size, which increases the time taken for each iteration. This is not only because the full training set must be processed, but also because a larger data set

Table 12
For each language, percentage of training sentences with crossing arcs in syntax and semantics, with semantic arcs forming a tree, and which were not parsable using the *Swap* action, as well as the performance of our system in the CoNLL-2009 shared task by syntactic accuracy and semantic F_1.

	Syntactic crossings	Semantic crossings	Semantic tree	Not parsable	Macro F_1	LAS (rank)	Sem F_1 (rank)
Catalan	0.0	0.0	61.4	0.0	82.7	87.9 (1)	77.4 (2)
Chinese	0.0	28.0	28.6	9.5	76.1	76.1 (4)	76.1 (4)
Czech	22.4	16.3	6.1	1.8	83.2	80.4 (1)	86.0 (2)
English	7.6	43.9	21.4	3.9	83.2	88.8 (3)	83.2 (4)
German	28.1	1.3	97.4	0.0	79.6	87.3 (2)	71.8 (5)
Japanese	0.9	38.3	11.2	14.4	84.9	92.3 (2)	77.2 (4)
Spanish	0.0	0.0	57.1	0.0	82.4	87.6 (1)	77.2 (2)

Table 13
Parsing and training times for different languages, run on a 3.4 GHz machine with 16 GB of memory. Parsing times computed on the test set. Indicators of SRL complexity provided for comparison.

	Average	Catalan	Chinese	Czech	English	German	Japanese	Spanish
Training time (hours, full set)	21.28	12.76	33.31	46.27	22.91	14.58	5.02	14.12
(sec, per word per iteration)	0.0033	0.0032	0.0043	0.0048	0.0026	0.0021	0.0030	0.0030
Parsing time (sec, per sentence)	4.415	3.257	11.119	6.985	5.443	0.805	1.006	2.293
(sec, per word per beam)	0.0032	0.0019	0.0049	0.0041	0.0028	0.0013	0.0037	0.0020
Training words	542,657	390,302	609,060	652,393	958,167	648,677	112,555	427,442
Parsing words per sentence	22.4	30.8	28.2	16.8	25.0	16.0	26.4	30.4
SRL complexity (% predicates)	20.6	9.6	16.9	63.5	18.7	2.7	22.8	10.3
(% crossing)	18.3	0.0	28.0	16.3	43.9	1.3	38.3	0.0

tends to result in more parameters to train, including larger vocabulary sizes. Also, larger data sets tend to result in more iterations of training, which further increases training times. Normalizing for data-set size and number of iterations (second row of Table 13), we get fairly consistent speeds across languages. The remaining differences are correlated with the number of parameters in the model, and with the proportion of words which are predicates in the SRL annotation, shown in the bottom panel of Table 13.

Parsing times are more variable, even when normalizing for the number of sentences in the data set, as shown in the third row of Table 13. As discussed earlier, this is in part an effect of the different beam widths used for different languages, and the different distributions of sentence lengths. If we divide times by beam width and by average sentence length (fourth row of Table 13), we get more consistent numbers, but still with a lot of variation.[18] These differences are in part explained by the relative complexity of the SRL annotation in the different languages. They are correlated with both the percentage of words that are predicates and the percentage of sentences that

18 Dividing by the average square of the sentence length does not result in more consistent values than dividing by the average length.

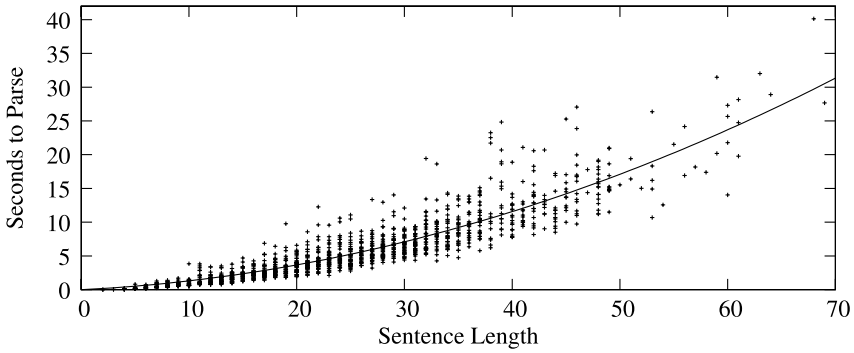

Figure 12
The parsing time for each sentence in the English development set, with a search beam width of 80, plotted against its length (up to length 70). The curve is the best fitting quadratic function with zero intercept.

have crossing arcs in the SRL, shown in the bottom panel of Table 13. Crossing arcs result in increased parsing times because choosing when to apply the *Swap* action is difficult and complicates the search space.

As discussed in Section 5.2, the parsing strategy prunes to a fixed beam of alternatives only after the shifting of each word, and between shifts it only constrains the branching factor of the search. Because of this second phase, parsing time is quadratic as a function of sentence length. As a typical example, the distribution of parsing times for English sentences is shown in Figure 12. The function of sentence length that best fits this distribution of seconds per sentence is the quadratic function $0.078n + 0.0053n^2$, also shown in Figure 12. In this function, the linear factor is 15 times larger than the quadratic factor. Fitting a cubic function does not account for any more variance than this quadratic function. The best fitting function for Catalan is $-0.0040n + 0.0031n^2$, for Chinese is $0.16n + 0.0057n^2$, for Czech is $-0.00068n + 0.012n^2$, for German is $0.020n + 0.0015n^2$, for Japanese is $-0.0013n + 0.0011n^2$, and for Spanish is $0.0083n + 0.0018n^2$. As with English, Chinese and German have larger linear terms, but the second-order term dominates for Catalan, Czech, Japanese, and Spanish. It is not clear what causes these differences in the shape of the curve.

One of the characteristics of our model is that it makes no independence assumptions and deals with the large space of alternatives by pruning. The size of the pruning beam determines speed and accuracy. Figure 13 shows how the accuracy of the parser degrades as we speed it up by decreasing the search beam used in parsing, for each language's development set. For some languages, a slightly smaller search beam is actually more accurate, and we used this smaller beam when running the given evaluations on the testing set. But in each case the beam was set to maximize accuracy at the expense of speed, without considering beam widths greater than 80. For some languages, in particular Czech and Chinese, the accuracy increase from a larger beam is relatively large. It is not clear whether this is due to the language, the annotation, or our definition of derivations. For smaller beams the trade-off of accuracy versus words-per-second is roughly linear. Comparing parsing time per word directly to beam width, there is also a linear relationship, with a zero intercept.[19]

19 When discussing timing, we use "word" to refer to any token in the input string, including punctuation.

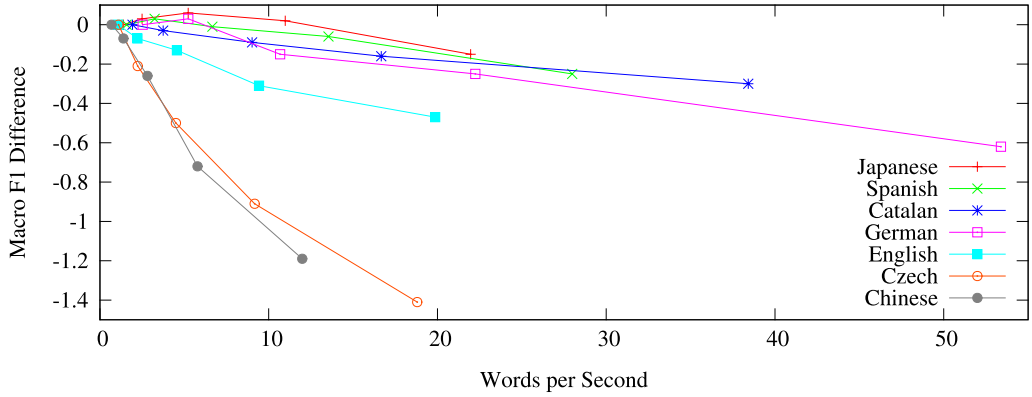

Figure 13
Difference in development set macro F_1 as the search beam is decreased from 80 to 40, 20, 10, and 5, plotted against parser speed.

It is possible to increase both parsing and training speeds, potentially at the expense of some loss in parsing accuracy, by decreasing the size of the latent variable vectors, and by increasing the vocabulary frequency threshold. For all the results reported in this section, all languages used a latent vector size of 80 and a vocabulary frequency threshold of 5, which were set to be large enough not to harm accuracy. Figure 14 summarizes the speed–accuracy trade-off for parsing English as these parameters are varied. Training times were more variable due to differences in the number of iterations and the decreases tended to be smaller. As Figure 14 shows, some speed-up can be achieved with little change in accuracy by using smaller latent vectors and smaller vocabularies, but the accuracy quickly drops when these parameters are set too low. For this data set, there is actually a small increase in accuracy with a small decrease in the vocabulary size, probably due to smoothing effects, but this trend is limited and variable. In contrast, much larger efficiency gains can be achieved by reducing the search beam width. Varying the parameters together produced a range of similar

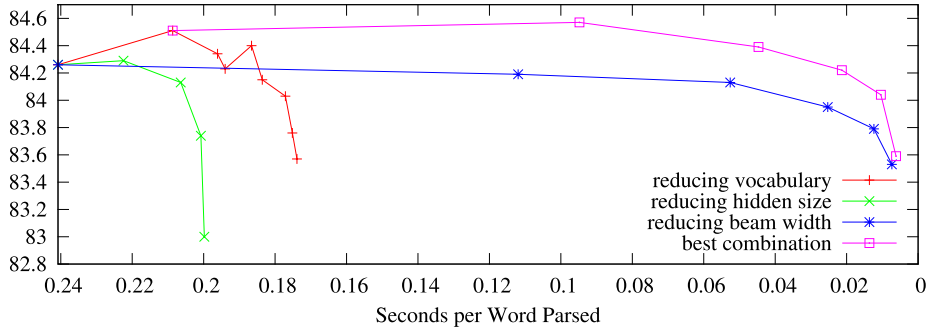

Figure 14
The parsing speed-accuracy trade-off when changing the meta-parameters of the model, on the English CoNLL-2009 development set. The vocabulary frequency threshold is increased from 5 to 10, 20, 30, 40, 60, 80, 120, and 160. The latent vector size is reduced from 80 to 70, 60, 50, and 40. The search beam width is reduced from 80 to 40, 20, 10, 5, and 3. The best combination keeps the vocabulary frequency threshold at 10 and reduces the search beam width as above.

curves, bounded by the "best combination" shown. These experiments achieved a 96% reduction in parsing time with an absolute reduction in parsing accuracy of only 0.2%, which is not generally considered a meaningful difference. This results in a parsing speed of 0.010 seconds per word. All other things being equal, both training and parsing times asymptotically scale quadratically with the latent vector size, due to the latent-to-latent connections in the model. Training and parsing times asymptotically scale linearly with vocabulary size, and vocabulary size can be expected to increase superlinearly with the value of the frequency threshold.

7. Related Work

In this article, we report on a joint generative history-based model to predict the most likely derivation of a dependency parser for both syntactic and semantic dependencies. In answer to the first question raised in the Introduction, we provide a precise proposal for the interface between syntactic dependencies and semantic roles dependencies, based on a weak synchronization of meaningful subsequences of the two derivations. We also propose a novel operation for semantic dependency derivations. In answer to the second question raised in the Introduction, we investigate issues related to the joint learning of syntactic and semantic dependencies. To train a joint model of their synchronized derivations, we make use of latent variable models of parsing and of estimation methods adapted to these models. Both these contributions have a rich context of related work that is discussed further here.

7.1 The Syntactic–Semantic Interface

The main feature of our proposal about the syntactic–semantic interface is based on the observation that the syntactic and the semantic representations are not isomorphic. We propose therefore a weak form of synchronization based on derivation subsequences. These synchronized subsequences encompass decisions about the left side of each individual word.

Other work has investigated the complex issue of the syntax–semantics interface. Li, Zhou, and Ng (2010) systematically explore different levels of integration of phrase-structure syntactic parsing and SRL for Chinese. Although the syntactic representations are too different for a direct comparison to our Chinese results, they provide results of general interest. Li, Zhou, and Ng compare two models of tight coupling of syntax and semantics and show that both joint approaches improve performance compared to a strong n-best pipeline approach. The first model interleaves SRL labeling at each completed constituent of a bottom–up multi-pass parser, inspired by Ratnaparkhi's (1999) model. This model thus learns the conditional probability of each individual semantic role assignment, conditioned on the whole portion of the syntactic structure that is likely to affect the assignment (as indicated by the fact that the value of the features is the same as when the whole tree is available). This model improves on the n-best pipeline model, although the improvement on parsing is not significant. A second model manages the harder task of improving the syntactic score, but requires feature selection from the SRL task. These best-performing features are then added to the syntactic parser by design. Although these results confirm the intuition that syntactic and semantic information influence each other, they also, like ours, find that it is not trivial to develop systems that actually succeed in exploiting this intuitively obvious

correlation. Li, Zhou, and Ng's approach is also different from ours in that they do not attempt to induce common representations useful for both tasks or for many languages, and as such cannot be regarded as multi-task, nor as multilingual, learning.

Synchronous grammars provide an elegant way to handle multiple levels of representation. They have received much attention because of their applications in syntax-based statistical machine translation (Galley et al. 2004; Chiang 2005; Nesson and Shieber 2008) and semantic parsing (Wong and Mooney 2006, 2007). Results indicate that these techniques are among the best both in machine translation and in the database query domain. Our method differs from those techniques that use a synchronous grammar, because we do not rewrite pairs of synchronized non-terminals, but instead synchronize chunks of derivation sequences. This difference is in part motivated by the fact that the strings for our two structures are perfectly aligned (being the same string), so synchronizing on the chunks of derivations associated with individual words eliminates any further alignment issues.

We have also proposed novel derivations for semantic dependency structures, which are appropriate for the relatively unconstrained nature of these graphs. Our *Swap* operation differs from the reordering that occurs in synchronous grammars in that its goal is to uncross arcs, rather than to change the order of the target string. The switching of elements of the semantic structure used in Wong and Mooney (2007) is more similar to the word reordering technique of Hajičová et al. (2004) than to our *Swap* operation, because the reordering occurs before, rather than during, the derivation. The notion of planarity has been widely discussed in many works cited herein, and in the dependency parsing literature. Approaches to dealing with non-planar graphs belong to two conceptual groups: those that manipulate the graph, either by pre-processing or by post-processing (Hall and Novak 2005; McDonald and Pereira 2006), and those that adapt the algorithm to deal with non-planarity. Among the approaches that, like ours, devise an algorithm to deal with non-planarity, Yngve (1960) proposed a limited manipulation of registers to handle discontinuous constituents, which guaranteed that parsing/generation could be performed with a stack of very limited depth. An approach to non-planar parsing that is more similar to ours has been proposed in Attardi (2006). Attardi's dependency parsing algorithm adds six new actions that allow this algorithm to parse any type of non-planar tree. Our *Swap* action is related to Attardi's actions *Left2* and *Right2*, which create dependency arcs between the second element on the stack and the front of the input queue. In the Attardi algorithm, every attachment to an element below the top of the stack requires the use of one of the new actions, whose frequency is much lower than the normal attachment actions, and therefore harder to learn. This contrasts with the *Swap* action, which handles reordering with a single action, and the normal attachment operations are used to make all attachments to the reordered word. Though much simpler, this single action can handle the vast majority of crossing arcs that occur in the data. Nivre (2008, 2009) presents the formal properties of a *Swap* action for dependency grammars that enables parsing of non-planar structures. The formal specifications of this action are different from the specifications of the action proposed here. Nivre's action can swap terminals repeatedly and move them down to an arbitrary point into the stack. This *Swap* action can potentially generate word orders that cannot be produced by only swapping the two top-most elements in the stack. When defining the oracle parsing order for training, however, Nivre (2008, 2009) assumes that the dependency structure can be planarized by changing the order of words. This is not true for many of the semantic dependency graphs, because they are not trees. More recently, Gómez-Rodríguez and Nivre (2010) proposed to use two stacks to parse non-planar graphs. Though the resulting automata is probably expressive enough to

handle complex semantic structures, predicting decisions in this representation can be a challenging task.

7.2 Multi-task Learning and Latent Variables

In answer to the second question raised in the Introduction, we investigate issues related to the joint learning of syntactic and semantic dependencies for these synchronized derivations.

To train a joint model of these synchronized derivations, we make use of a latent variable model of parsing. The ISBN architecture induces latent feature representations of the derivations, which are used to discover correlations both within and between the two derivations. This is the first application of ISBNs to a multi-task learning problem. The automatic induction of features is particularly important for modeling the correlations between the syntactic and semantic structures, because our prior knowledge about the nature of these correlations is relatively weak compared to the correlations within each single structure.

Other joint models do not perform as well as our system. In Lluís and Màrquez (2008), a fully joint model is developed that learns the syntactic and semantic dependencies together as a single structure whose factors are scored using a combination of syntactic and semantic scores. This differentiates their approach from our model, which learns two separate structures, one for syntax and one for semantics, and relies on latent variables to represent the interdependencies between them. It is not clear whether it is this difference in the way the models are parametrized or the difference in the estimation techniques used that gives us better performance, but we believe it is the former. These experimental results may be explained by theoretical results demonstrating that pipelines can be preferable to joint learning when no shared hidden representation is learned (Roth, Small, and Titov 2009). Previous work on joint phrase-structure parsing and semantic role labeling also suggests that joint models of these two tasks can achieve competitive results when latent representations are induced to inform both tasks, as shown in Musillo and Merlo (2006) and Merlo and Musillo (2008).

The relevance of latent representations to joint modeling of NLP tasks is further demonstrated by Collobert and Weston (2007, 2008). They propose a deep learning architecture to solve a task closely related to semantic role labeling. This task is defined as a tagging task: Those words in a sentence that correspond to an argument of a predicate are all tagged with the semantic role label assigned to that argument and those words that do not correspond to any argument of a predicate are tagged with the null label. The accuracy for this sequence labeling task is defined as the proportion of correctly tagged words. The learning architecture of Collobert and Weston (2008) is designed to jointly learn word features across a variety of related tasks. Large gains in accuracy for this semantic role tagging task are obtained when word features are jointly learned with other tasks such as part-of-speech tagging, chunking, and language modeling that are annotated on the same training data. Direct comparison with their work is problematic as we focused in this article on the supervised setting and a different form of semantic role labeling (predicting its dependency representation).[20] Note, however, that our model can be potentially extended to induce a latent word

20 More recent work (Collobert et al. 2011) has evaluated a similar multi-task learning model in terms of standard SRL evaluation measures, where they reach 74% F_1 on the CoNLL-2005 data set without using syntactic information and 76% F_1 when they exploit a syntactic parse.

representation shared across different tasks by introducing an additional layer of latent variables, as for Collobert and Weston (2008).

Latent variable models that induce complex representations without estimating them from equally complex annotated data have also been shown to be relevant to single-structure prediction NLP tasks such as phrase-structure syntactic parsing (Matsuzaki, Miyao, and Tsujii 2005; Prescher 2005; Petrov et al. 2006; Liang et al. 2007). Latent representations of syntactic structures are induced by decorating the non-terminal symbols in the syntactic trees with hidden variables. The values of these hidden variables thus refine the non-terminal labels, resulting in finer-grained probabilistic context-free grammars than those that can be read off treebanks. Work by Petrov et al. (2006) shows that state-of-the-art results can be achieved when the space of grammars augmented with latent annotations is searched with the split-merge heuristics. In contrast, our ISBN latent variable models do not require heuristics to control the complexity of the augmented grammars or to search for predictive latent representations. Furthermore, probabilistic context-free grammars augmented with latent annotations do impose context-free independence assumptions between the latent labels, contrary to our models. Finally, our ISBN models have been successfully applied to both phrase-structure and dependency parsing. State-of-the-art results on unlexicalized dependency parsing have recently been achieved with latent variable probabilistic context-free grammars (Musillo and Merlo 2008; Musillo 2010). These latent variable grammars are compact and interpretable from a linguistic perspective, and they integrate grammar transforms that constrain the flow of latent information, thereby drastically limiting the space of latent annotations. For example, they encode the notion of X-bar projection in their constrained latent variables.

8. Conclusions and Future Work

The proposed joint model achieves competitive performance on both syntactic and semantic dependency parsing for several languages. Our experiments also demonstrate the benefit of joint learning of syntax and semantics. We believe that this success is due to both the linguistically appropriate design of the synchronous parsing model and the flexibility and power of the machine learning method.

This joint model of syntax and semantics has recently been applied to problems where training data with gold annotation is available only for the syntactic side, while the semantic role side is produced by automatic annotations, projected from a different language (Van der Plas, Merlo, and Henderson 2011). The results show that joint learning can improve the quality of the semantic annotation in this setting, thereby extending the range of techniques available for tasks and languages for which no annotation exists.

The success of the proposed model also suggests that the same approach should be applicable to other complex structured prediction tasks. In particular, this extension of the ISBN architecture to weakly synchronized syntactic–semantic derivations is also applicable to other problems where two independent, but related, representations are being learned, such as syntax-based statistical machine translation.

Acknowledgments

The authors would particularly like to thank Andrea Gesmundo for his help with the CoNLL-2009 shared task. The research leading to these results has received funding from the EU FP7 programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSIC project: www.classic-project.org), and from the Swiss NSF under grants 122643 and 119276.

References

Ando, Rie Kubota and Tong Zhang. 2005a. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1,817–1,853.

Ando, Rie Kubota and Tong Zhang. 2005b. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL2005)*, pages 1–9, Ann Arbor, MI.

Argyriou, Andreas, Theodoros Evgeniou, and Massimiliano Pontil. 2006. Multi-task feature learning. In *NIPS*, pages 41–48, Vancouver.

Attardi, Giuseppe. 2006. Experiments with a multilanguage non-projective dependency parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-2006)*, pages 166–170, New York, NY.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics (ACL-COLING'98)*, pages 86–90, Montreal.

Basili, Roberto, Diego De Cao, Danilo Croce, Bonaventura Coppola, and Alessandro Moschitti. 2009. Cross-language frame semantics transfer in bilingual corpora. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 332–345, Mexico City.

Black, E., F. Jelinek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos. 1993. Towards history-based grammars: Using richer models for probabilistic parsing. In *Proceedings of the 31st Meeting of the Association for Computational Linguistics*, pages 31–37, Columbus, OH.

Bohnet, Bernd and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1,455–1,465, Jeju Island, July.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Pado, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of LREC 2006*, Genoa.

Carreras, Xavier and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164, Ann Arbor, MI.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, Seattle, WA.

Che, Wanxiang, Zhenghua Li, Yuxuan Hu, Yongqiang Li, Bing Qin, Ting Liu, and Sheng Li. 2008. A cascaded syntactic and semantic dependency parsing system. In *Proceedings of CONLL 2008*, pages 238–242, Manchester.

Chen, Enhong, Liu Shi, and Dawei Hu. 2008. Probabilistic model for syntactic and semantic dependency parsing. In *Proceedings of the 12th Conference on Computational Natural Language Learning: Shared Task*, CoNLL '08, pages 263–267, Manchester.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, MI.

Choi, J. D. and M. Palmer. 2010. Retrieving correct semantic boundaries in dependency structure. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 91–99, Uppsala.

Ciaramita, Massimiliano, Giuseppe Attardi, Felice Dell'Orletta, and Mihai Surdeanu. 2008. DeSRL: A linear-time semantic role labeling system. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 258–262, Manchester.

Čmejrek, Martin, Jan Hajič, and Vladislav Kuboň. 2004. Prague Czech-English dependency treebank: Syntactically annotated resources for machine translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1,597–1,600, Lisbon.

Cohen, P. R. 1995. *Empirical Methods for Artificial Intelligence*. MIT Press, Cambridge, MA.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*.

Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.
Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.
Collobert, Ronan and Jason Weston. 2007. Fast semantic extraction using a novel neural network architecture. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 560–567, Prague.
Collobert, Ronan and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, Helsinki.
Dowty, David. 1991. Thematic proto-roles and argument selection. *Language*, 67(3):547–619.
Fillmore, Charles J. 1968. The case for case. In Bach E. and Harms R. T., editors, *Universals in Linguistic Theory*. Holt, Rinehart, and Winston, New York, pages 1–88.
Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu, Susan Dumais, and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, MA.
Gao, Qin and Stephan Vogel. 2011. Corpus expansion for statistical machine translation with semantic role label substitution rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 294–298, Portland, OR.
Garg, Nikhil and James Henderson. 2011. Temporal restricted Boltzmann machines for dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 11–17, Portland, OR.
Ge, Ruifang and Raymond Mooney. 2009. Learning a compositional semantic parser using an existing syntactic parser. 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 611–619, Singapore.
Ge, Ruifang and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 9–16, Ann Arbor, MI.
Gesmundo, Andrea, James Henderson, Paola Merlo, and Ivan Titov. 2009. A latent variable model of synchronous syntactic-semantic parsing for multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 37–42, Boulder, CO.
Ghahramani, Zoubin. 1998. Learning dynamic Bayesian networks. In C. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*. Springer-Verlag, Berlin, pages 168–197.
Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.
Gómez-Rodríguez, Carlos and Joakim Nivre. 2010. A transition-based parser for 2-planar dependency structures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL 2010, pages 1,492–1,501, Uppsala.
Hajič, J., J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtskỳ, and M. Ševčíková-Razímová. 2006. Prague dependency treebank 2.0. *Linguistic Data Consortium*, Philadelphia, PA.
Hajič, Jan. 2004. Complex corpus annotation: The Prague dependency treebank. In *Linguistic Data Consortium*, Bratislava, Slovakia. Jazykovedný ústav Ľ. Štúra, SAV.
Hajič, Jan, 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, CO.
Hajičová, Eva, Jiří Havelka, Petr Sgall, Kateřina Veselá, and Daniel Zeman. 2004. Issues of projectivity in the Prague dependency treebank. In *Prague Bulletin of Mathematical Linguistics*, pages 5–22, Prague.
Hall, Johan and Joakim Nivre. 2008. Parsing discontinuous phrase structure with grammatical functions. In *Proceedings of*

the 6th International Conference on Natural Language Processing (GoTAL 2008), pages 169–180, Gothenburg.

Hall, Keith and Vaclav Novak. 2005. Corrective modeling for non-projective dependency parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology (IWPT'05)*, pages 42–52, Vancouver.

Hatori, Jun, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1,045–1,053, Jeju Island.

Hedegaard, Steffen and Jakob Grue Simonsen. 2011. Lost in translation: Authorship attribution using frame semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 65–70, Portland, OR.

Henderson, James. 2003. Inducing history representations for broad coverage statistical parsing. In *Proceedings of the Joint Meeting of the North American Chapter of the Association for Computational Linguistics and the Human Language Technology Conference*, pages 103–110, Edmonton.

Henderson, James, Paola Merlo, Gabriele Musillo, and Ivan Titov. 2008. A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proceedings of CoNLL 2008*, pages 178–182, Manchester.

Henderson, James and Ivan Titov. 2010. Incremental sigmoid belief networks for grammar learning. *Journal of Machine Learning Research*, 11(Dec):3,541–3,570.

Johansson, Richard and Pierre Nugues. 2007. Extended constituent-to-dependency conversion in English. In *Proceedings of NODALIDA 2007*, pages 105–112, Gothenburg.

Johansson, Richard and Pierre Nugues. 2008a. Dependency-based semantic role labeling of PropBank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78, Honolulu, HI.

Johansson, Richard and Pierre Nugues. 2008b. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of CoNLL 2008*, pages 183–187, Manchester.

Kawahara, Daisuke, Hongo Sadao, and Koiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, pages 2,008–2,013, Las Palmas.

Kipper, K., A. Korhonen, N. Ryant, and M. Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.

Kwiatkowski, Tom, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1,512–1,523, Edinburgh.

Lang, Joel and Mirella Lapata. 2011. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1,117–1,126.

Levin, Beth. 1993. *English Verb Classes and Alternations*. University of Chicago Press, Chicago, Illinois.

Levin, Lori. 1986. *Operations on Lexical Form: Unaccusative Rules in Germanic Languages*. Ph.D. thesis, Massachussetts Institute of Technology, Cambridge, MA.

Li, Junhui, Guodong Zhou, and Hwee Tou Ng. 2010. Joint syntactic and semantic parsing of Chinese. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1,108–1,117, Uppsala.

Liang, Percy, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical Dirichlet processes. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697, Prague.

Liu, Ding and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 716–724, Beijing.

Lluís, Xavier and Lluís Màrquez. 2008. A joint model for parsing syntactic and semantic dependencies. In *Proceedings of CONLL 2008*, pages 188–192, Manchester.

Lo, Chi-kiu and Dekai Wu. 2011. MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 220–229, Portland, Oregon.

MacKay, David J. C. 2003. Exact marginalization in graphs. In David J. C. MacKay, editor, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, UK, pages 334 – 340.

Marcus, Mitch, Beatrice Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Màrquez, Lluís, Xavier Carreras, Kenneth C. Litkowski, and Suzanne Stevenson. 2008. Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, 34(2):145–159.

Matsuzaki, Takuya, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 75–82.

McDonald, Ryan. 2006. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph.D. thesis, Department of Computer Science, University of Pennsylvania.

McDonald, Ryan T. and Fernando C. N. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, EACL 2006, pages 81–88, Trento.

Merlo, Paola and Gabriele Musillo. 2008. Semantic parsing for high-precision semantic role labelling. In *Proceedings of CONLL 2008*, pages 101–104, Manchester.

Merlo, Paola and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408.

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank project: An interim report. In A. Meyers, editor, *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, MA.

Miller, George, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. 1990. Five papers on Wordnet. Technical report, Cognitive Science Laboratory, Princeton University. CSL Report 43, Princeton, NJ.

Miller, S., H. Fox, L. Ramshaw, and R. Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, NAACL 2000, pages 226–233, Seattle.

Morante, Roser, Vincent Van Asch, and Antal van den Bosch. 2009. Dependency parsing and semantic role labeling as a single task. In *Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing*, pages 275–280, Borovets.

Moschitti, Alessandro, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question-answer classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 776–783, Prague.

Musillo, Gabriele and Paola Merlo. 2005. Lexical and structural biases for function parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology (IWPT'05)*, pages 83–92, Vancouver.

Musillo, Gabriele and Paola Merlo. 2006. Accurate semantic parsing of the Proposition Bank. In *Proceedings of the North American Conference for Computational Linguistics, Companion Volume: Short Papers*, pages 101–104, New York, NY.

Musillo, Gabriele and Paola Merlo. 2008. Unlexicalised hidden variable models of split dependency grammars. In *Proceedings of the Annual Conference for Computational Linguistics (ACL'08)*, pages 213–216, Columbus, Ohio.

Musillo, Gabriele Antonio. 2010. *Latent Variable Transforms for Dependency Parsing*. Ph.D. thesis, Department of Computer Science, University of Geneva, Switzerland.

Neal, Radford. 1992. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113.

Nesson, Rebecca and Stuart Shieber. 2008. Synchronous vector-tag for natural language syntax and semantics. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 9)*, Tübingen.

Nivre, Joakim. 2006. *Inductive Dependency Parsing*. Springer, Berlin.

Nivre, Joakim. 2008. Sorting out dependency parsing. In *Proceedings of GoTAL 2008*, pages 16–27, Gothenburg.

Nivre, Joakim. 2009. Non-projective dependency parsing in expected linear time. 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 351–359, Suntec, Singapore.

Nivre, Joakim, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning*, CoNLL 2004, pages 49–56, Boston, MA.

Nivre, Joakim, Johan Hall, Jens Nilsson, Gulsen Eryigit, and Svetoslav Marinov. 2006. Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL 2006, pages 221–225, New York, NY.

Nivre, Joakim, Marco Kuhlmann, and Johan Hall. 2009. An improved oracle for dependency parsing with online reordering. In *Proceedings of the 11th International Conference on Parsing Technologies*, IWPT '09, pages 73–76, Paris.

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

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31:71–105.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and 21st International Conference on Computational Linguistics*, ACL-COLING 2006, pages 403–440, Sydney.

Pradhan, Sameer, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. Ontonotes: A unified relational semantic representation. In *International Conference on Semantic Computing (ICSC 2007)*, pages 405–419, Prague.

Prescher, Detlef. 2005. Head-driven PCFGs with latent-head statistics. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 115–124, Vancouver.

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

Ratnaparkhi, Adwait. 1999. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–175.

Roth, Dan, Kevin Small, and Ivan Titov. 2009. Sequential learning of classifiers for structured prediction problems. In *AISTATS 2009 : Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *JMLR : Workshop and Conference Proceedings*, pages 440–447, Clearwaters, FL.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing, Vol 1*. MIT Press, Cambridge, MA, pages 318–362.

Sallans, Brian. 2002. *Reinforcement Learning for Factored Markov Decision Processes*. Ph.D. thesis, University of Toronto, Toronto, Canada.

Saul, Lawrence K., Tommi Jaakkola, and Michael I. Jordan. 1996. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76.

Surdeanu, Mihai, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 45–52. Sapporo.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, pages 159–177.

Taulé, Mariona, M. Antònia Martì, and Marta Recasens. 2008. Ancora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, pages 797–782, Marrakech.

Thompson, Cynthia A., Roger Levy, and Christopher D. Manning. 2003. A generative model for semantic role labeling. In *Proceedings of the 14th European Conference on Machine Learning*, ECML 2003, pages 397–408, Dubrovnik.

Titov, Ivan and James Henderson. 2007a. Constituent parsing with Incremental Sigmoid Belief Networks. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL 2007, pages 632–639, Prague.

Titov, Ivan and James Henderson. 2007b. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 947–951, Prague.

Titov, Ivan and James Henderson. 2007c. Incremental Bayesian networks for structure prediction. In *Proceedings of the 24th International Conference on Machine Learning*, ICML 2007, pages 887–894, Corvallis, OR.

Titov, Ivan and James Henderson. 2007d. A latent variable model for generative dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 144–155, Prague.

Titov, Ivan, James Henderson, Paola Merlo, and Gabriele Musillo. 2009. Online graph planarisation for synchronous parsing of semantic and syntactic dependencies. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1,562–1,567, Pasadena, CA.

Titov, Ivan and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, ACL 2011, pages 1,445–1,455, Portland, OR.

Titov, Ivan and Alexandre Klementiev. 2012. A Bayesian approach to unsupervised semantic role induction. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, Avignon.

Toutanova, Kristina, Aria Haghighi, and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Tsarfaty, Reut, Khalil Sima'an, and Remko Scha. 2009. An alternative to head-driven approaches for parsing a (relatively) free word-order language. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 842–851, Singapore.

Van der Plas, Lonneke, James Henderson, and Paola Merlo. 2009. Domain adaptation with artificial data for semantic parsing of speech. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 125–128, Boulder, Colorado.

Van der Plas, Lonneke, Paola Merlo, and James Henderson. 2011. Scaling up automatic cross-lingual semantic role annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 299–304, Portland, OR.

Wong, Yuk Wah and Raymond Mooney. 2006. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 439–446, New York, NY.

Wong, Yuk Wah and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967, Prague.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Wu, Dekai, Marianna Apidianaki, Marine Carpuat, and Lucia Specia, editors. 2011. In *Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*. ACL, Portland, Oregon, June.

Wu, Dekai and Pascale Fung. 2009. Semantic roles for SMT: A hybrid two-pass model. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, NAACL-Short '09, pages 13–16, Boulder, CO.

Xue, Nianwen and Martha Palmer. 2009. Adding semantic roles to the Chinese treebank. *Natural Language Engineering*, 15:143–172, January.

Yeh, Alexander. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th International Conference in Computational Linguistics (COLING 2000)*, pages 947–953, Saarbruecken.

Yngve, Victor H. 1960. A model and a hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104(5):444–466.

Zettlemoyer, Luke and Michael Collins. 2007. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687, Prague.

Zhao, Hai and Chunyu Kit. 2008. Parsing syntactic and semantic dependencies with two single-stage maximum entropy models. In *Proceedings of CONLL 2008*, pages 203–207, Manchester.