

# Coarse Lexical Frame Acquisition at the Syntax–Semantics Interface Using a Latent-Variable PCFG Model

**Laura Kallmeyer\*** & **Behrang Q. Zadeh\***      **Jackie Chi Kit Cheung**  
SFB991, Heinrich-Heine-Universität Düsseldorf      McGill University  
kallmeyer@hhu.de   zadeh@phil.hhu.de   jcheung@cs.mcgill.ca

## Abstract

We present a method for unsupervised lexical frame acquisition at the syntax–semantics interface. Given a set of input strings derived from dependency parses, our method generates a set of clusters that resemble lexical frame structures. Our work is motivated not only by its practical applications (e.g., to build, or expand the coverage of lexical frame databases), but also to gain linguistic insight into frame structures with respect to lexical distributions in relation to grammatical structures. We model our task using a hierarchical Bayesian network and employ tools and methods from latent variable probabilistic context free grammars (L-PCFGs) for statistical inference and parameter fitting, for which we propose a new split and merge procedure. We show that our model outperforms several baselines on a portion of the Wall Street Journal sentences that we have newly annotated for evaluation purposes.

## 1 Introduction

We propose a method for building coarse lexical frames automatically from dependency parsed sentences; i.e., without using any explicit semantic information as training data. The task involves grouping verbs that evoke the same frame (i.e., are considered to be the head of this frame) and further clustering their syntactic arguments into latent semantic roles. Hence, our target structures stand between FrameNet (Ruppenhofer et al., 2016) and PropBank (Palmer et al., 2005) frames. Similar to FrameNet and in contrast to PropBank, we assume a many-to-many relationship between verb types and frame types. But similar to PropBank, we aim to cluster syntactic arguments into general semantic roles instead of frame-specific slot

types in FrameNet. This allows us to generalize across frames concerning semantic roles. As part of this, we study possible ways to automatically generate more abstract lexical-semantic representations from lexicalized dependency structures.

In our task, grouping verb tokens into frames requires not only distinguishing between different senses of verbs, but also identifying a range of lexical relationships (e.g., synonymy, opposite verbs, troponymy, etc.) among them. Hence (as Modi et al., 2012; Green et al., 2004), our problem definition differs from most work on unsupervised fine-grained frame induction using verb sense disambiguation (e.g., Kawahara et al., 2014; Peng et al., 2017). Similarly, forming role clusters yields generalization from several alternate linkings between semantic roles and their syntactic realization. Given, for instance, an occurrence of the verb *pack* and its syntactic arguments, not only do we aim to distinguish different senses of the verb *pack* (e.g., as used to evoke the FILLING frame, or the PLACING frame), but also to group these instances of ‘pack’ with other verbs that evoke the same frame (e.g., to group instances of *pack* that evoke the frame PLACING with instances of verbs *load*, *pile*, *place*, and so on when used to evoke the same PLACING frame).

The motivation for this work is twofold. On the one hand, the frame induction techniques we propose can be useful in the context of applications such as text summarization (Cheung and Penn, 2013), question answering (Frank et al., 2007; Shen and Lapata, 2007), and so on, for languages where we lack a frame-annotated resource for supervised frame induction, or to expand the coverage of already existing resources. On the other hand, we are interested in theoretical linguistic insights into frame structure. In this sense, our

\*Both authors contributed equally to this work.

work is a step towards an empirical investigation of frames and semantic roles including hierarchical relations between them.

We cast the frame induction task as unsupervised learning using an L-PCFG (Johnson, 1998; Matsuzaki et al., 2005; Petrov et al., 2006; Cohen, 2017). As input, our model takes syntactic dependency trees and extracts input strings corresponding to instances of frame expressions, which are subsequently grouped into latent semantic frames and roles using an L-PCFG. We use the inside-outside (i.e., Expectation-Maximization (Dempster et al., 1977; Do and Batzoglou, 2008)) algorithm and a split-merge procedure (Petrov et al., 2006) for dynamically adapting the number of frames and roles to the data, for which we employ new heuristics. As implied, one advantage of the L-PCFGs framework is that we can adapt and reuse statistical inference techniques used for learning PCFGs in syntactic parsing application (e.g., split-merge). Our experiment shows that the method outperforms a number of baselines, including frame grouping by lexical heads and one based on agglomerative clustering.

The main contributions of this paper are a) using L-PCFGs for coarse lexical frame acquisition; b) a new split-merge routine adapted for this task; and, c) a new dataset for evaluating the induced lexical frame-role groupings. In the remainder of the paper, § 2 describes our statistical model and its formalization to an L-PCFG. § 3 describes procedures used for statistical inference. § 4 describes our evaluation dataset and reports results from experiments. § 5 discusses related work followed by a conclusion in § 6.

## 2 From a Latent Model to L-PCFG

We assume that frames and semantic roles are the latent variables of a probabilistic model. Given the probability mass function  $\text{pmf}(F^1, \dots, F^n, R^1 \dots R^k, D^1, \dots, D^m; \mathcal{C}, \theta)$  as our model, we denote latent frames  $F^i, 1 \leq i \leq n$ , and roles  $R^i, 1 \leq i \leq k$  for observations that are annotated syntactically using  $D^i, 1 \leq i \leq m$  in the input corpus  $\mathcal{C}$ . Inspired by Cheung et al. (2013), we approximate the probability of a specific frame  $f$  with head  $v$ , semantic roles  $r_1 \dots r_k$  filled by words  $w_1 \dots w_k$  and corresponding syntactic dependencies  $d_1 \dots d_k$  (under

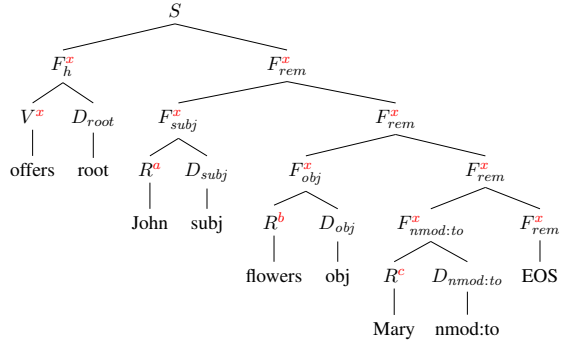


Figure 1: Sample frame structure for (1).

parameters  $\theta$ ) as:

$$p(f) \cdot p(v|f) \prod_{i=1}^k p(d_i|f) \cdot \prod_{i=1}^k p(r_i|f, d_i) \cdot \prod_{i=1}^k p(w_i|r_i). \quad (1)$$

To estimate the parameters of our model, we translate Eq. (1) to an L-PCFG that captures the required conditional and joint probabilities.

First, we convert input lexicalized dependency parses to a set of strings  $\mathcal{E}$ . Given a verb  $v$  and its dependents  $w_i$  in a dependency parse tree, we build input strings in the form of

$$v \text{ root } w_1 d_1 \dots w_l d_l \text{ EOS},$$

for which we assume  $v$  lexicalizes the head of a frame,  $w_1 \dots w_l$  are the arguments fillers and  $d_1, \dots, d_l$  are the respective dependencies that link these fillers to the head  $v$ ; *EOS* is a special symbol to mark the end of string. For the step from the sentence to input strings, we assume that dependencies are ordered (e.g., *subj* precedes *dobj* and *iobj* and they precede prepositional and complement dependents (i.e., *nmod:\** and *\*comp*).<sup>1</sup> Consider (1) as an example; the corresponding string is the yield of the tree in Fig. 1.

$$(1) \quad \text{John}_{\text{subj}} \text{offer}_{\text{root}} \text{flowers}_{\text{dobj}} \text{to Mary}_{\text{nmod:to}}$$

Given the fixed structure of input strings, we design a CFG that rewrites them to our expected hierarchical frame structure consisting of elements  $F$ ,  $R$ ,  $D$  while capturing the conditional probabilities from Eq. (1). The tree assigning a frame  $F$  of type  $x$  with semantic roles of type  $a, b, c$  to (1) is for

<sup>1</sup>Phrasal arguments are reduced to their syntactic head given by the underlying UD parser. We normalize passive structures by replacing *nsubjpass* with *dobj*. Other syntactic dependents (e.g., *case*, *aux*, *conj*, etc.) are removed. In case of the same dependencies, surface order in the sentence is relevant. If necessary, conjunctions are treated when transforming dependency parses to input strings.

instance given in Fig. 1. More generally, given finite sets of frames  $\mathcal{F}$  and of semantic roles  $\mathcal{R}$ , our underlying CFG  $G = \langle N, T, P, S \rangle$  is as follows:

- $T = T_v \cup T_n \cup \mathcal{D} \cup \{root, EOS\}$ , where  $T_v$  is the set of possible verbal heads,  $T_n$  is the set of possible lexicalizations (fillers) for arguments, and  $\mathcal{D}$  is a finite set of dependency relations; *root* and *EOS* are special symbols.
- $N = \{S\} \cup \{F_h^f \mid f \in \mathcal{F}\} \cup \{F_{rem}^f \mid f \in \mathcal{F}\} \cup \{F_g^f \mid f \in \mathcal{F}, g \in \mathcal{D}\} \cup \{R^r \mid r \in \mathcal{R}\} \cup \{V^f \mid f \in \mathcal{F}\} \cup \{D_g \mid g \in \mathcal{D}\}$ .
- $P$  contains the following rules:
  - $S \rightarrow F_h^f F_{rem}^f$  for all  $f \in \mathcal{F}$ ;
  - $F_h^f \rightarrow V^f D_{root}$  for all  $f \in \mathcal{F}$ ;
  - $F_{rem}^f \rightarrow F_g^f F_{rem}^f$  for all  $f \in \mathcal{F}, g \in \mathcal{D}$ ;
  - $F_{rem}^f \rightarrow EOS$  for all  $f \in \mathcal{F}$ ;
  - $F_g^f \rightarrow R^r D_g$  for all  $f \in \mathcal{F}, r \in \mathcal{R}, g \in \mathcal{D}$ ;
  - $V^f \rightarrow v$  for all  $f \in \mathcal{F}, v \in T_v$ ;
  - $R^r \rightarrow n$  for all  $r \in \mathcal{R}, n \in T_n$ ;
  - $D_g \rightarrow g$  for all  $g \in \mathcal{D} \cup \{root\}$ .

With this grammar, an input string derived from a dependency parsed sentence fully determines the shape of the tree and the node labels are fixed except for the choice of the frame  $f$  and the semantic roles  $r$  of the  $k$  fillers (i.e.,  $x, a, b$ , and  $c$  in Fig. 1).

The probabilities of the rules correspond to the conditional probabilities in Eq. (1). The probability of  $S \rightarrow F_h^f F_{rem}^f$  gives  $p(F = f)$ , the probability of  $V^f \rightarrow v$  gives  $p(V = v \mid F = f)$ , and so on. During the subsequent inside-outside (IO) split-and-merge training procedure, the inventory of frames and roles and the probabilities corresponding to our rules are estimated so that the overall likelihood of observations is maximized.

### 3 Method

This section describes statistical inference methods used for inducing latent frame clusters from input strings. The scenario we used for parameter fitting (split, merge, smoothing, and generating clusters) is described in § 3.1. In § 3.2, we describe our method for computing embedding-based similarity between frames, which we use during the merge process and in our baseline system.

#### 3.1 Parameter Fitting

Given an input corpus parsed into universal dependencies (UD) and converted into a set of input

strings  $\mathcal{E}$ , we instantiate a model  $G$  according to § 2. We set  $|\mathcal{F}| = 1$  and  $|\mathcal{R}| = |\mathcal{D}|$ , and  $\mathcal{D}$ ,  $T_v$  and  $T_n$  are automatically extracted from  $\mathcal{E}$ . Starting from this, we iteratively perform *split-merge* sequences (with an IO parameter estimation in between), and cluster  $\mathcal{E}$  to disjoint subsets  $\mathcal{E}_i$  by finding the most-likely derivations that  $G$  yields. We detail this process in the following subsections.

##### 3.1.1 The IO Algorithm

As a solution to sparsity of observations, we modify the IO algorithm slightly. We adapt the procedures described in (Eisner, 2016) with the exception that for computing inside and outside probabilities, instead of mapping terminals to nonterminals using an exact matching of the right-hand-sides of the rules (and respectively their assigned parameters), we use embedding-based similarities. I.e., for computing inside probabilities, given symbol  $a$  as input, instead of considering  $A \rightarrow_\theta as$  rewrite rules and updating the parse chart only by asserting  $\theta$  in it, we also consider  $B \rightarrow_\theta bs$  in which instead of  $\theta$  we assert  $\alpha \times \theta_s$  in the IO table, where  $\alpha$  is the  $r^2$  coefficient correlations of embeddings for  $a$  and  $bs$ . During the outside procedure,  $\theta_s$  are updated proportionally w.r.t. to  $\alpha_s$  used during the inside parameter estimation procedure.

##### 3.1.2 Split

We alter the splitting procedure from (Klein and Manning, 2003a; Petrov et al., 2006) for our application. In (Klein and Manning, 2003b; Petrov et al., 2006), during split, a non-terminal symbol (which represents a random variable in the underlying probabilistic model) is split and its related production rules are duplicated *independently of its parent, or sibling nodes*. We can apply such a context-independent split only to the  $R^r$  non-terminals but the  $F_{rem}^f$ s must split *dependently* w.r.t. their sibling nodes that define the frame structure. Therefore, to split frame  $x$  to two frames  $y$  and  $z$ , we replace the entire set  $\{S \rightarrow F_h^x F_{rem}^x, F_h^x \rightarrow V^x D_{root}, F_{rem}^x \rightarrow EOS, F_{rem}^x \rightarrow F_g^x F_{rem}^x, F_g^x \rightarrow R^r D_g, \text{ and } V^x \rightarrow v\}$  with two similar sets where  $x$  gets replaced with  $y$  and  $z$ , respectively. The parameters for the new rules are set to half of the value of the parameters of the rules that they originated from with the addition (or subtraction) of a random  $\epsilon$  (e.g., 1e-7) to break symmetry.

Moreover, in our application, training the split grammar on the whole input is ineffective and at

a certain point, computationally intractable. The problem is due to the scope of the split grammar and the large portion of input strings  $\mathcal{E}$  that they span. Splitting a frame’s rules, unlike parameter fitting for syntactic rules, increases the number of possible derivations for all  $\mathcal{E}$ , to the extent that after a number of split iterations the computation of the derivations becomes intractable even for a small  $\mathcal{E}$  of short length. We address this problem by using a new strategy for splitting: we not only split the grammar, but also the input training data.

Before each split, we cluster input strings  $\mathcal{E}$  to clusters  $\mathcal{E}_i$  that  $G$  gives (§ 3.1.4) at that point. For input strings in each cluster  $\mathcal{E}_i$ , we instantiate a new  $G_i$  and perform parameter fitting and splitting independently of other  $\mathcal{E}_i$ s. The corresponding probabilistic  $G_i$  is initialized by assigning random parameters to its rules and then smoothing them (§ 3.1.3) by the fitted parameters for  $G$ . We apply the aforementioned process several times, until the number of independently generated clusters is at least twice as large as  $|T_v|$ . At the end of each split iteration, we collect the elicited  $\mathcal{E}_i$  clusters (and their respective  $G_i$ s) for the next merge process. Given the independence assumption between roles and frames, pre-terminals that rewrite roles are split similar to (Petrov et al., 2006).

### 3.1.3 Smoothing

We apply a notion of smoothing by interpolating parameters that are obtained in the  $n-1$ th iteration of split-merge with parameters that are randomly initialized at the beginning of each split-merge iteration and, as mentioned earlier in § 3.1.2, when deriving new  $G_i$ s from  $G$ : For each rule in  $G_i$  or  $G^m$  with parameters  $\theta$  (i.e., the  $G$  instantiated for the next split-merge iteration), we smooth  $\theta$ s using  $\theta = \alpha\theta + (1 - \alpha)\theta_{n-1}$ , where  $\theta_{n-1}$  is the already known and fitted parameter for the corresponding rule in  $G$ . We choose  $\alpha = 0.1$ .

### 3.1.4 Generating Clusters from $G$

After fitting parameters of  $G$ , the frame structure for an input string is given by its most-likely viterbi derivation with respect to  $G$ . The verb which is rewritten by  $F_h^f$  is placed to frame-type/cluster  $f$ . Similarly, lexical items that are argument fillers are assigned to type/cluster  $r$  where  $r$  is the structural annotation for pre-terminal  $R^r$  that rewrite them. For example, assuming Fig 1 is the most likely derivation for (1), the verb ‘offer’ is categorized as frame  $x$  and its arguments as

roles  $a$ ,  $b$ , and  $c$ .

### 3.1.5 Merge

The model resulting from the split process generates a relatively large number of ‘homogeneous’ clusters that are ‘incomplete’. A subsequent merge process unifies these homogeneous-but-incomplete clusters to achieve a clustering that is both homogeneous and complete. To this end, we use heuristics which are based on both the estimated loss in likelihood from merging two symbols that span the same input sequence (as proposed previously in Petrov et al. (2006)) as well as the ‘discriminative similarities’ between the obtained clusters.

**Merge by likelihoods does not work:** The heuristics for merge in (Petrov et al., 2006) (i.e., minimizing the loss in training likelihood using ‘locally’ estimated inside and outside probabilities) are based on the assumptions that a) pre-terminals appearing in different places in derivations are nearly independent, and b) that their approximation (according to the method proposed by Petrov et al. (2006)) requires less computation than computing full derivation trees. However, neither of these hold in our case: a) most pre-terminals in our model are dependent on each other and, b) to compute the loss in likelihood from a cluster merge requires computation of full derivation trees (given the interdependence between pre-terminals that define frame structures). More importantly, in our application, the outside probabilities for clusters are always 1.0 and differences in the sum of inside probabilities is often negligible since input strings are spanned more-or-less by the same set of derivations. For these reasons (i.e., computation cost and the lack of sufficient statistics), the ‘estimated loss in likelihood’ heuristics is a futile method for guiding the merge process in our application. We resolve this problem by leveraging discriminative similarities between the obtained frame clusters and proposing a hybrid method.

**Our merge approach:** In the beginning of a merge process, we conflate  $G_i$ s that are obtained from the previous split procedure to form a  $G$  that spans all input strings. Where applicable, we set parameters of rules in  $G$  to the arithmetic mean of corresponding ones obtained from the split process and normalize them such that sum of the parameters of rules with the same pre-terminal is 1.0.

We go through an iterative process: Using the method proposed in § 3.2 below, the frame instances in the clusters are converted to tensors and similarities among them are computed. Motivated by the assumption that split clusters are homogeneous, for every pair of clusters  $c^x$  and  $c^y$  ( $x \neq y$ ) with instances  $a_i \in c^x$  and  $b_j \in c^y$ , we find  $\arg \max_{i,j} \text{sim}(a_i, b_j)$  and  $\arg \min_{i,j} \text{sim}(a_i, b_j)$  ( $\text{sim}$  is given by Eq. 2 below) and calculate their harmonic mean as the similarity  $s_c$  between  $c^x$  and  $c^y$ . Cluster pairs are sorted in a descending order by  $s_c$ . Given a threshold  $\delta$ , for all  $s_c(c^x, c^y) > \delta$ , their corresponding production rules (i.e., the similar set of rules mentioned in the split procedure) are merged and their parameters are updated to the arithmetic mean of their origin rules.

Parameters for this new merged  $G$  are updated through a few IO iterations (in an incremental fashion (Liang and Klein, 2009)), and finally  $G$  is used to obtain a new clustering. The process is repeated for this newly obtained clustering until all the resulting cluster-wise  $s_c$  similarities are less than a threshold  $\beta$ .

Computing all derivations for each input string is time consuming and makes the merge process computationally expensive, particularly in the first few iterations. We resolve this issue using a stratified random sampling and by performing the aforementioned iterations only on a random subset of input strings in each cluster. Each cluster in the output of the split process is taken as a stratum and its size is reduced by 90% by applying a random sampling; this random sampling is updated in each iteration (we use a similar strategy for parameter estimation, i.e., we update samples in each estimation iteration). This process reduces the required time for merge drastically without hurting the overall outcome of the merge process. It is worth to mention that after merging clusters  $c^x$  and  $c^y$ , the output does not necessarily contain a cluster  $c^x \cup c^y$ . Instead, the resulting clustering reflects the effect of merging the rules that rewrite  $c^x$  and  $c^y$  in the whole model.

To merge  $R^r$  categories, we use the merge method from (Petrov et al., 2006) based on the obtained likelihoods. After merging frame clusters, we reduce the number of  $R^r$ s by 50%. Since our method for merging role categories is similar to (Petrov et al., 2006), we do not describe it here.

### 3.2 Similarity Between Frame Instances

When necessary (such as during merge), we compute embedding-based similarities between frame instances similar to methods proposed in (Mitchell and Lapata, 2008; Clark, 2013). We build a  $n$ -dimensional embedding for each word appearing in our input strings from large web corpora. Each frame instance is then represented using a  $(m + 1, n)$ -tensor, in which  $m$  is the total number of argument types/clusters given by our model at its current stage and  $n$  is the dimensionality of the embeddings that represent words that fill these arguments. To this, we add the embedding for the verb that lexicalizes the head of the frame, which gives us the final  $(m + 1, n)$ -tensor.

For two frame-instances represented by tensors  $a$  and  $b$ , the similarity for their arguments is

$$\text{sim-arg}(a, b) = \frac{1}{k} \sum_{i=1}^m r^2(a_{v^i}, b_{v^i}),$$

in which  $v^i$ s are embeddings for the  $i$ th argument filler ( $\sum_{j=1}^n v_j^i \neq 0$ ),  $r^2$  is the coefficient of determination, and  $k = \sum_i [r^2(a_{v^i}, b_{v^i}) \neq 0]$ . If an argument is lexicalized by more than one filler, we replace  $r^2$  with the arithmetic mean of  $r^2$ s computed over each distinct pair of fillers. The overall  $\text{sim}$  between  $a$  and  $b$  is:

$$\text{sim}(a, b) = w_1 \cdot r^2(a_{v^h}, b_{v^h}) + w_2 \cdot \text{sim-arg}(a, b), \quad (2)$$

where  $v^h$ s are the embeddings for the lexical heads (i.e., verbs), and  $w_1$  and  $w_2$  are two hyper-parameters which can be tuned. For instance, for two hypothetical structures of  $F^a$ : [Head:travel, [Arg<sup>1</sup>:John, Arg<sup>2</sup>:London]] and  $F^b$ : [Head:walk, [Arg<sup>1</sup>:Mary, Arg<sup>3</sup>:home]], the similarity between  $F^a$  and  $F^b$  is  $w_1 r^2(\text{travel}, \text{walk}) + w_2 r^2(\text{John}, \text{Mary})$ , given that the all these vectors have at least one nonzero component. During merge we use  $w_1 = w_2 = 0.50$ .

We build our lexical embeddings of dimensionality  $n = 900$  using the hash-based embedding learning technique proposed in (QasemiZadeh and Kallmeyer, 2017); before using these embeddings, we weight them using positive pointwise mutual information. During evaluation, this combination of PPMI-weighted hash-based embeddings and the  $r^2$  estimator consistently yielded better results than using other popular choices such as the co-

sine of word2vec vectors. We associate this observation to the imbalanced frequency of the usages of lexical items in our experiments in the corpora used to train embeddings (i.e., an English web corpus (Schäfer, 2015) and PTB’s WSJ).

## 4 Experiments and Results

### 4.1 Dataset

We derive our data for evaluation from the PTB’s WSJ sections parsed (using Schuster and Manning, 2016) to the enhanced UD format. We augment these sentences with semantic role annotations obtained from Prague Semantic Dependencies (PSD) (Cinkova et al., 2012) from the SDP resource (Oepen et al., 2016). Using EngVallex (Cinková et al., 2014) and SemLink (Bonial et al., 2013), we semi-automatically annotate verbs with FrameNet frames (Baker et al., 1998). We choose 1k random sentences and manually verify the semi-automatic mappings to eventually build our evaluation dataset of approximately 5k instances (all). From this data, we use a random subset of 200 instances (dev) during the development and for parameter tuning (see Table 1 for detailed statistics).

| Set | FT | FI    | V   | AT | AI     |
|-----|----|-------|-----|----|--------|
| all | 27 | 5,324 | 169 | 13 | 10,523 |
| dev | 15 | 200   | 35  | 7  | 393    |

Table 1: Gold Data: FT, FI, V, AT, and AI denote the number of frame types, instances, distinct verb heads, argument types, and argument instances

For these gold instances, we extract input strings from their UD parses according § 2. Since we discard verbs without syntactic arguments, use automatic parses, and do not distinguish arguments from adjuncts, the input strings do not exactly match the gold data argument structures. We report results *only* for the portion of the gold data that appears in the extracted input strings. Table 2 reports the statistics for the induced input strings and their agreement with the gold data (in terms of precision and recall).

Input strings are hard to cluster in the sense that a) all the frames are lexicalized by at least two different verb lemmas, b) many verbs lexicalizes at least two different types of frames, c) verb lemmas that lexicalize a frame have long-tailed distributions, i.e., a large proportion of instances of a frame are realized at surface structure by one

| Set | FT | FI    | $R_F$ | V   | GR | AI     | AIG   | $P_A$ | $R_A$ |
|-----|----|-------|-------|-----|----|--------|-------|-------|-------|
| all | 27 | 4,984 | 0.94  | 167 | 56 | 10,893 | 7,305 | 0.67  | 0.76  |
| dev | 15 | 191   | 0.95  | 34  | 24 | 450    | 277   | 0.62  | 0.76  |

Table 2: Input strings extracted from the UD parses: GR, AIG,  $R_F$ ,  $R_A$ , and  $P_A$  denote, respectively, the number of distinct grammatical relations, syntactic arguments that are a semantic role in the gold data, recall for frame and arguments, and precision for arguments. The remaining symbols are the same as Table 1.

lemma while in the remaining instances the frame is evoked by different lemmas, and d) last but not least, the frame types themselves have long-tailed distribution. Table 3 shows examples of frames and verb lemmas that lexicalize them; in the table, the most frequent lemma for each frame type is italicized.

### 4.2 Evaluation Measures

We evaluate our method’s performance on a) clustering input strings to frame types, and b) clustering syntactic arguments to semantic role types. To this end, we report the harmonic mean of BCubed precision and recall (BCF) (Bagga and Baldwin, 1998), and purity (PU), inverse purity (IPU) and their harmonic mean (FPU) (Steinbach et al., 2000) as figures of merit. These measures reflect a notion of similarity between the distribution of instances in the obtained clusters and the gold/evaluation data based on certain criteria and alone may lack sufficient information for a fair understanding of the system’s performance. While PU and IPU are easy to interpret (by establishing an analogy between them and precision and recall in classification tasks), they may be deceiving under certain conditions (as explained by Amigó et al., 2009, under the notions of homogeneity, completeness, rag bag, and ‘size vs. quantity’ constraints). Reporting BCF alongside FPU ensures that these pitfalls are not overlooked when our system’s output are compared quantitatively with the baselines.

### 4.3 Baselines

As baselines, we report the standard all-in-one-class clustering (ALLIN1) and the one-cluster-per-instance (1CPERI) baselines, as well as the random baseline ( $R_n$ ) in which instances are randomly partitioned into  $n$  clusters ( $n$  being the number of generated clusters in our sys-

| Frame                    | #T   | #V | {Examples of verbs occurrences}  |
|--------------------------|------|----|--|
| ADORNING                 | 26   | 10 | { <i>fill</i> :8, <i>cover</i> :4, <i>adorn</i> :2 ... }                           |
| PLACING                  | 121  | 21 | { <i>place</i> :62, <i>pack</i> :3, <i>wrap</i> :1... }                            |
| FILLING                  | 35   | 12 | { <i>fill</i> :14, <i>pack</i> :6, <i>cover</i> :3, <i>wrap</i> :2 ... }           |
| ACTIVITY_START           | 290  | 2  | { <i>begin</i> :182, <i>start</i> :108}  |
| PROCESS_START            | 188  | 2  | { <i>begin</i> :143, <i>start</i> :45}   |
| CHANGE_POSITION_ON_SACLE | 1259 | 17 | { <i>fall</i> =356, <i>rise</i> =271, <i>drop</i> =135, <i>decline</i> =119, ... } |

Table 3: Examples of frames in our evaluation set and verbs that evoke them; #T and #V denote the total number of instances for the frame and the number of distinct verb lemmas that evoke them, respectively.

tem’s output). Moreover, for frame type clustering, we report the one-cluster-per-lexical-head baseline (1C<sub>PER</sub>HEAD). For role clustering, we report the additional one-cluster-per-syntactic-category baseline (1C<sub>PER</sub>GR). Similar to the most-frequent-sense baseline in word sense induction and disambiguation problems, the latter 1C<sub>PER</sub>HEAD and 1C<sub>PER</sub>GR are particularly hard to beat given the heavy-tailed distribution of lexical items in frame and role categories.

For both subtasks, an additional baseline from (Modi et al., 2012) and (Titov and Klementiev, 2012) could be an interesting comparison to our method with the state of the art in frame head clustering and unsupervised semantic role labeling, particularly given that (Titov and Klementiev, 2012) and respectively (Modi et al., 2012) employ Gibbs sampling for statistical inference, whereas we use the IO algorithm. We are, unfortunately, not able to access codes for Modi et al. (2012) and the system in Titov and Klementiev (2012) relies on features that are engineered for treebanks in the format and formalisms set for the CoNLL-2008 shared-task. As explained by Oepen et al. (2016), mapping to (and from) formalisms used in CoNLL-2008 (from-to) those proposed in SDP-PSD (used in this paper) is a nontrivial task. We expect that an automatic conversion from our data to the CoNLL-2008 format as an input for (Titov and Klementiev, 2012) would not reflect the best performance of their method. Nonetheless, we report the result from this experiment (marked as TK-URL) later in this section, not as a baseline, but to confirm (Oepen et al., 2016).

Lastly, as an extra baseline for frame type clustering, we report performance of a HAC method. The HAC method is described below (§ 4.3.1).

#### 4.3.1 A Baseline HAC Method

To build a frame clustering using HAC, we begin by initializing one cluster per instance and itera-

tively merge the pair of clusters with the lowest distance, using average-link cluster distance. For two clusters  $A$  and  $B$ , we define their distance as:

$$\text{Dis-Cl}(A, B) = \frac{1}{l(l-1)} \sum_{f_i \in A} \sum_{f_j \in B} 1 - \text{sim}(f_i, f_j),$$

in which  $\text{sim}(f_i, f_j)$  is given by Eq. 2, and  $l = |A| + |B|$ . We iteratively update the distance matrix and agglomerate clusters until we reach a single cluster. During iterations, we keep track of the merges/linkages which we later use to flatten the built hierarchy into  $q$  clusters. To set our baseline, by constraining  $w_1 + w_2 = 1$  in Eq. (2), we build cluster hierarchies for different  $w_1$  and  $w_2$  (starting with  $w_1 = 0.0$ ,  $w_2 = 1 - w_1$  and gradually increasing  $w_1$  by 0.1 until  $w_1 = 1.0$ ) and find  $w_1, w_2$ , and  $q$  that yield the ‘best’ clustering according to the BCF metric ( $w_1 = 0.8, w_2 = 0.2, q = 140$ ). For this baseline, the argument types are defined by their syntactic relation to their heads, e.g., subj, dobj, and so on.

## 4.4 Results

Since our method involves stochastic decisions, its performance varies slightly in each run. Hence, we report the mean and the standard deviation of the obtained performances from 4 independent runs. The reported results are based on the output of the system after 7 split and merge iterations. After tuning parameters on the dev set, we choose  $\delta = 0.55$  during merge, and in each inner-merge iteration subtract  $\delta$  by 0.01 until  $\delta < \beta = 0.42$ .

**Quantitative Comparison with Baselines** Tables 4 and 5 show the results for clustering input strings to frame types and semantic roles, respectively. On frame type clustering, our method (denoted by L-PCFG) outperforms all the baselines. FPU and BCF for our system are simultaneously higher than all the baselines, which verifies that the output contains a small proportion of ‘rag bag’ clusters. The system, however, tends to generate

| Method                    | #C     | PU         | IPU        | FPU         | BCF          |
|---------------------------|--------|------------|------------|-------------|--------------|
| ALLINI                    | 1      | 22.35      | <b>100</b> | 36.54       | 17.43        |
| ICPERI                    | 4984   | <b>100</b> | 0.54       | 1.08        | 1.08         |
| ICPERHEAD                 | 167    | 94.38      | 59.59      | 73.06       | 63.53        |
| R <sub>235</sub>          | 235    | 24.7       | 2.03       | 3.75        | 1.79         |
| HAC                       | 140    | 75.07      | 65.52      | 69.97       | 61.74        |
| <b>L-PCFG (Avg.)</b>      | 230.25 | 86.2       | 73.64      | <b>79.4</b> | <b>71.29</b> |
| <b>L-PCFG (Std. Dev.)</b> | ±6.24  | ±3.07      | ±0.96      | ±1.17       | ±1.49        |

Table 4: Results for head groupings: #C denotes the number of induced clusters by each method/baseline; the last two rows reports the average and the standard deviation for the obtained results using the L-PCFG model. The remaining abbreviations are introduced in § 4.2 and 4.3.

| Method                    | #C    | PU         | IPU        | FPU          | BCF          |
|---------------------------|-------|------------|------------|--------------|--------------|
| ALLINI                    | 1     | 47.73      | <b>100</b> | 64.62        | 55.43        |
| ICPERI                    | 7257  | <b>100</b> | 0.18       | 0.36         | 0.36         |
| ICPERGR                   | 32    | 92.89      | 79.83      | <b>85.86</b> | <b>76.71</b> |
| R <sub>24</sub>           | 24    | 47.73      | 5.36       | 9.64         | 7.79         |
| TK-URL                    | 333   | 85.7       | 15.01      | 25.54        | 11.71        |
| <b>L-PCFG (Avg.)</b>      | 24    | 90.36      | 79.25      | 84.44        | 74.65        |
| <b>L-PCFG (Std. Dev.)</b> | ±5.29 | ±0.33      | ±1.31      | ±0.61        | ±1.38        |

Table 5: Results on clustering of syntactic arguments to semantic roles.

many incomplete yet homogeneous clusters (as we discuss below). With respect to roles, however, the method’s performance and its output remains very similar to the syntactic baseline (BCF=97.3).

**What is in the clusters?** The ability of the system to successfully cluster instances varies from one gold frame category to another one. The most problematic cases are the frame types ACTIVITY\_START and PROCESS\_START, as well as PLACING. While the system put instances of ‘start’ and ‘begin’ verbs in one cluster, it fails to distinguish between ACTIVITY\_START and PROCESS\_START. Regarding the PLACING frame, the system places verbs that evoke this frame in different clusters; each cluster consists of instances from one verb lemma. In our opinion, this is due to the frequent idiomatic usages of these verbs, e.g., ‘to lay claim’, ‘to place listing’, ‘to position oneself as’ and so on. This being said, however, the system is capable of distinguishing between different readings of polysemous verbs, e.g., instances of the verb ‘pack’ that evoke the FILLING frame end up in different clusters than those evoking the PLACING frame. Additionally, for a number of frame types, we observe that the system can successfully group synonymous (and opposite) verbs that evoke the same frame into one cluster: representative examples are the instances of

the CHANGE\_POSITION\_ON\_A\_SCALE frame that are evoked by different verb lemmas such as ‘decline’, ‘drop’, ‘fall’, ‘gain’, ‘jump’, ‘rise’, . . . , which all end up in one cluster. The output of the system also contains a large number of small clusters (consisting of only one or two instances): we observe that these instances are usually those with wrong (and incomplete) dependency parses.

## 5 Related Work

Our work differs from most work on word sense induction (WSI), e.g. (Goyal and Hovy, 2014; Lau et al., 2012; Manandhar et al., 2010; Van de Cruys and Apidianaki, 2011), in that not only do we discern different senses of a lexical item but also we group the induced senses into more general meaning categories (i.e., FrameNet’s grouping). Hence, our model must be able to capture lexical relationships other than polysemy, e.g., synonymy, antonymy (opposite verbs), troponymy, etc.. However, our method can be adapted to WSI, too. Firstly, we can assume that word senses are ‘incompatible’ and thus they necessarily evoke different frames; subsequently, the induced frame clusters can be seen directly as clusters of word senses. Otherwise, the proposed method can be adapted for WSI by altering its initialization, e.g., by building one-model-at-a-time for each word form (i.e., simply altering the input).

Despite similarities between our method and those proposed previously to address unsupervised semantic role induction (Carreras and Marquez, 2005; Lang and Lapata, 2010, 2011; Titov and Klementiev, 2012; Swier and Stevenson, 2004), our method differs from them in that we attempt to include frame head grouping information for inducing roles associated to them. In other words, these methods leave out the problem of sense/frame grouping in their models.

Our work differs in objective from methods for unsupervised template induction in information extraction (IE) (e.g., MUC-style frames in Chambers and Jurafsky (2009, 2011) and its later refinements such as (Chambers, 2013; Cheung et al., 2013; Balasubramanian et al., 2013), and in a broader sense attempts towards ontology learning and population from text (Cimiano et al., 2005)). Our focus is on lexicalized elementary syntactic structures, identifying lexical semantic relationships, and thereby finding salient patterns in syntax–semantic interface. However, in IE



tasks the aim is to build structured summaries of text. Therefore, the pre-and post-processing in these induction models are often more complex/different than our method (e.g., they require anaphora resolution, identifying discourse relations, etc.). Lastly, we deal with a broader set of verbs and domains and more general frame definitions than these methods.

As stated earlier, Modi et al. (2012) propose the most similar work to ours. They adapt (Titov and Klementiev, 2012) to learn FrameNet-style head and role groupings. Modi et al. (2012) assume roles to be frame-specific, while our role clusters are defined independently of frame groupings (as expressed in Eq. 1). Last, with respect to research such as (Pennacchiotti et al., 2008; Green et al., 2004) in which lexical resources such as WordNet are used (in supervised or unsupervised settings) to refine and extend existing frame repositories such as FrameNet, our model learns and bootstraps a frame repository from text annotated only with syntactic structure in an unsupervised way.

## 6 Conclusion

We proposed an unsupervised method for coarse-lexical frame induction from dependency parsed sentences using L-PCFG. We converted lexicalized dependency trees of sentences to a set of input strings of fixed, predetermined structure consisting of a verbal head, its arguments and their syntactic dependencies. We then use a CFG model (subsequently L-PCFG) to shape/capture frame structures from these strings. We adapted EM parameter estimation techniques from PCFG while relaxing independence assumptions, including appropriate methods for splitting and merging frames and semantic roles and using word embeddings for better generalization. In empirical evaluations, our model outperforms several baselines.

## Acknowledgments

We would like to thank Dr. Curt Anderson and Dr. Rainer Osswald for valuable comments on the paper and analysis of frame clusters. The work described in this paper is funded by the Deutsche Forschungsgemeinschaft (DFG) through the ‘Collaborative Research Centre 991 (CRC 991): The Structure of Representations in Language, Cognition, and Science’.

## References

- Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.* 12(4):461–486. <https://doi.org/10.1007/s10791-008-9066-8>.
- Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING ’98, pages 79–85. <https://doi.org/10.3115/980451.980859>.
- Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL ’98, pages 86–90. <https://doi.org/10.3115/980845.980860>.
- Niranjana Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1721–1731. <http://www.aclweb.org/anthology/D13-1178>.
- Claire Bonial, Kevin Stowe, and Martha Palmer. 2013. Renewing and revising SemLink. In *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*. Association for Computational Linguistics, Pisa, Italy, pages 9–17. <http://www.aclweb.org/anthology/W13-5503>.
- X. Carreras and L. Marquez. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*. pages 152–164. <http://www.aclweb.org/anthology/W05-0620>.
- Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*. volume 13, pages 1797–1807. <https://aclweb.org/anthology/D13/D13-1185.pdf>.
- Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. 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*. Association for Computational Linguistics, Suntec, Singapore, pages 602–610. <http://www.aclweb.org/anthology/P/P09/P09-1068>.

- Nathanael Chambers and Dan Jurafsky. 2011. **Template-based information extraction without the templates.** In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 976–986. <http://dl.acm.org/citation.cfm?id=2002472.2002595>.
- Jackie Chi Kit Cheung and Gerald Penn. 2013. **Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain.** In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Sofia, Bulgaria, pages 1233–1242. <http://www.aclweb.org/anthology/P13-1121>.
- J.C.K. Cheung, H. Poon, and L. Vanderwende. 2013. **Probabilistic frame induction.** In *Proceedings of NAACL-HLT*. pages 837–846. <http://www.aclweb.org/anthology/N13-1104>.
- Philipp Cimiano, Andreas Hotho, and Steffen Staab. 2005. **Learning concept hierarchies from text corpora using formal concept analysis.** *J. Artif. Int. Res.* 24(1):305–339. <http://dl.acm.org/citation.cfm?id=1622519.1622528>.
- Silvie Cinková, Eva Fučíková, Jana Šindlerová, and Jan Hajič. 2014. **EngVallex - English valency lexicon.** LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University. <http://hdl.handle.net/11858/00-097C-0000-0023-4337-2>.
- Silvie Cinkova, Marie Mikulova, Lucie Mladova, Anja Nedoluzko, Petr Pajas, Jarmila Panevova, Jiri Semecky, Jana Sindlerova, Zdenka Uresova, Zdenek Zabokrtsky, Jiri Semecky, Jana Sindlerova, Josef Toman, Zdenka Uresova, and Zdenek Zabokrtsky. 2012. **Annotation of English on the tectogrammatical level: Reference book.** <https://ufal.mff.cuni.cz/techrep/tr35.pdf>.
- Stephen Clark. 2013. *Quantum Physics and Linguistics*, Oxford University Press, chapter Type-Driven Syntax and Semantics for Composing Meaning Vectors. <https://doi.org/http://dx.doi.org/10.1093/acprof:oso/9780199646296.003.0013>.
- Shay Cohen. 2017. **Latent-variable PCFGs: Background and applications.** In *Proceedings of the 15th Meeting on the Mathematics of Language*. Association for Computational Linguistics, London, UK, pages 47–58. <http://www.aclweb.org/anthology/W17-3405>.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. **Maximum likelihood from incomplete data via the EM algorithm.** *Journal of the Royal Statistical Society* 39(1):1–21.
- Chuong B. Do and Serafim Batzoglou. 2008. **What is the expectation maximization algorithm?** *Nature Biotechnology* <https://doi.org/10.1038/nbt1406>.
- Jason Eisner. 2016. **Inside-outside and forward-backward algorithms are just backprop.** In *Proceedings of the EMNLP Workshop on Structured Prediction for NLP*. Austin, TX.
- Anette Frank, Hans-Ulrich Krieger, Feiyu Xu, Hans Uszkoreit, Berthold Crysmann, Brigitte Jrg, and Ulrich Schfer. 2007. **Question answering from structured knowledge sources.** *Journal of Applied Logic* 5(1):20 – 48. **Questions and Answers: Theoretical and Applied Perspectives.** <https://doi.org/https://doi.org/10.1016/j.jal.2005.12.006>.
- Kartik Goyal and Eduard Hovy. 2014. **Unsupervised word sense induction using distributional statistics.** In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 1302–1310. <http://www.aclweb.org/anthology/C14-1123>.
- Rebecca Green, Bonnie J. Dorr, and Philip Resnik. 2004. **Inducing frame semantic verb classes from wordnet and LDOCE.** In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '04. <https://doi.org/10.3115/1218955.1219003>.
- Mark Johnson. 1998. **PCFG models of linguistic tree representations.** *Comput. Linguist.* 24(4):613–632. <http://dl.acm.org/citation.cfm?id=972764.972768>.
- Daisuke Kawahara, Daniel Peterson, Octavian Popescu, and Martha Palmer. 2014. **Inducing example-based semantic frames from a massive amount of verb uses.** In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Gothenburg, Sweden, pages 58–67. <http://www.aclweb.org/anthology/E14-1007>.
- Dan Klein and Christopher D. Manning. 2003a. **A\* parsing: Fast exact Viterbi parse selection.** In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '03, pages 40–47. <https://doi.org/10.3115/1073445.1073461>.
- Dan Klein and Christopher D. Manning. 2003b. **Accurate unlexicalized parsing.** In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL

- '03, pages 423–430. <https://doi.org/10.3115/1075096.1075150>.
- Joel Lang and Mirella Lapata. 2010. **Unsupervised induction of semantic roles**. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pages 939–947. <http://dl.acm.org/citation.cfm?id=1857999.1858135>.
- Joel Lang and Mirella Lapata. 2011. **Unsupervised semantic role induction via split-merge clustering**. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 1117–1126. <http://dl.acm.org/citation.cfm?id=2002472.2002614>.
- Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. **Word sense induction for novel sense detection**. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 591–601. <http://www.aclweb.org/anthology/E12-1060>.
- Percy Liang and Dan Klein. 2009. **Online EM for unsupervised models**. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '09, pages 611–619. <http://dl.acm.org/citation.cfm?id=1620754.1620843>.
- Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. **SemEval-2010 Task 14: Word sense induction & disambiguation**. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, Uppsala, Sweden, pages 63–68. <http://www.aclweb.org/anthology/S10-1011>.
- Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. **Probabilistic CFG with latent annotations**. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '05, pages 75–82. <https://doi.org/10.3115/1219840.1219850>.
- Jeff Mitchell and Mirella Lapata. 2008. **Vector-based models of semantic composition**. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, pages 236–244. <http://www.aclweb.org/anthology/P08-1028>.
- Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. 2012. **Unsupervised induction of frame-semantic representations**. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. Association for Computational Linguistics, Montréal, Canada, pages 1–7. <http://www.aclweb.org/anthology/W12-1901>.
- Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Zdenka Uresova. 2016. **Towards comparability of linguistic graph banks for semantic parsing**. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.
- Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. **The Proposition Bank: An annotated corpus of semantic roles**. *Computational Linguistics* 31(1):71–106.
- Haoruo Peng, Snigdha Chaturvedi, and Dan Roth. 2017. **A joint model for semantic sequences: Frames, entities, sentiments**. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Association for Computational Linguistics, pages 173–183. <https://doi.org/10.18653/v1/K17-1019>.
- Marco Pennacchiotti, Diego De Cao, Roberto Basili, Danilo Croce, and Michael Roth. 2008. **Automatic induction of framenet lexical units**. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 457–465. <http://dl.acm.org/citation.cfm?id=1613715.1613773>.
- Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. **Learning accurate, compact, and interpretable tree annotation**. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. <http://aclweb.org/anthology/P06-1055>.
- Behrang QasemiZadeh and Laura Kallmeyer. 2017. **HHU at SemEval-2017 Task 2: Fast hash-based embeddings for semantic word similarity assessment**. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, pages 250–255. <https://doi.org/10.18653/v1/S17-2039>.
- Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. *FrameNet II: Extended Theory and Practice*. ICSI, Berkeley.

- Roland Schäfer. 2015. [Processing and querying large web corpora with the COW14 architecture](#). In Piotr Baski, Hanno Biber, Evelyn Breiteneder, Marc Kupietz, Harald Lngen, and Andreas Witt, editors, *Proceedings of Challenges in the Management of Large Corpora 3 (CMLC-3)*. UCREL, IDS, Lancaster. <http://rolandschaefer.net/?p=749>.
- Sebastian Schuster and Christopher D. Manning. 2016. [Enhanced english universal dependencies: An improved representation for natural language understanding tasks](#). In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA), Paris, France.
- Dan Shen and Mirella Lapata. 2007. [Using semantic roles to improve question answering](#). In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 12–21. <http://www.aclweb.org/anthology/D07-1002>.
- M. Steinbach, G. Karypis, and V. Kumar. 2000. [A comparison of document clustering techniques](#). In *KDD Workshop on Text Mining*. <http://citeseer.ist.psu.edu/steinbach00comparison.html>.
- Robert S. Swier and Suzanne Stevenson. 2004. [Unsupervised semantic role labelling](#). In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*. Association for Computational Linguistics, Barcelona, Spain, pages 95–102. <http://www.aclweb.org/anthology/W/W04/W04-3213.pdf>.
- Ivan Titov and Alexandre Klementiev. 2012. [A Bayesian approach to unsupervised semantic role induction](#). In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*. pages 12–22. <http://aclweb.org/anthology/E/E12/E12-1003.pdf>.
- Tim Van de Cruys and Marianna Apidianaki. 2011. [Latent semantic word sense induction and disambiguation](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 1476–1485. <http://www.aclweb.org/anthology/P11-1148>.