Semi-Supervised Semantic Role Labeling via Structural Alignment

Hagen Fürstenau*
Columbia University

Mirella Lapata**
University of Edinburgh

Large-scale annotated corpora are a prerequisite to developing high-performance semantic role labeling systems. Unfortunately, such corpora are expensive to produce, limited in size, and may not be representative. Our work aims to reduce the annotation effort involved in creating resources for semantic role labeling via semi-supervised learning. The key idea of our approach is to find novel instances for classifier training based on their similarity to manually labeled seed instances. The underlying assumption is that sentences that are similar in their lexical material and syntactic structure are likely to share a frame semantic analysis. We formalize the detection of similar sentences and the projection of role annotations as a graph alignment problem, which we solve exactly using integer linear programming. Experimental results on semantic role labeling show that the automatic annotations produced by our method improve performance over using hand-labeled instances alone.

1. Introduction

Recent years have seen growing interest in the **shallow semantic analysis** of natural language text. The term is most commonly used to refer to the automatic identification and labeling of the **semantic roles** conveyed by sentential constituents (Gildea and Jurafsky 2002). Semantic roles themselves have a long-standing tradition in linguistic theory, dating back to the seminal work of Fillmore (1968). They describe the relations that hold between a predicate and its arguments, abstracting over surface syntactic configurations. Consider the following example sentences:

(1) a. The burglar broke the window with a hammer.
b. A hammer broke the window.
c. The window broke.

* Center for Computational Learning Systems, Columbia University, 475 Riverside Drive, Suite 850, New York, NY 10115, USA. E-mail: hagen@ccls.columbia.edu.
(The work reported in this paper was carried out while the author was at Saarland University, Germany.)
** School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, UK. E-mail: mlap@inf.ed.ac.uk.

Submission received: 30 August 2010; revised submission received: 29 April 2011; accepted for publication: 14 June 2011.

Here, the phrase *the window* occupies different syntactic positions—it is the object of *break* in sentences (1a) and (1b), and the subject in (1c)—and yet bears the same semantic role denoting the affected physical object of the breaking event. Analogously, *hammer* is the instrument of *break* both when attested with a prepositional phrase in (1a) and as a subject in (1b). The examples represent **diathesis alternations**[1] (Levin 1993), namely, regular variations in the syntactic expressions of semantic roles, and their computational treatment is one of the main challenges faced by automatic semantic role labelers.

Several theories of semantic roles have been proposed in the literature, differing primarily in the number and type of roles they postulate. These range from Fillmore's (1968) small set of universal roles (e.g., *Agentive*, *Instrumental*, *Dative*) to individual roles for each predicate (Palmer, Gildea, and Kingsbury 2005). Frame semantic theory (Fillmore, Johnson, and Petruck 2003) occupies the middle ground by postulating situations (or *frames*) that can be evoked by different predicates. In this case, roles are not specific to predicates but to frames, and therefore ought to generalize among semantically related predicates. As an example, consider the sentences in Example (2):

(2) a. [Lee]$_{Agent}$ [**punched**]$_{\text{CAUSE\_HARM}}$ [John]$_{Victim}$ [in the eye]$_{Body\_part}$.

b. [A falling rock]$_{Cause}$ [**crushed**]$_{\text{CAUSE\_HARM}}$ [my ankle]$_{Body\_part}$.

c. [She]$_{Agent}$ [**slapped**]$_{\text{CAUSE\_HARM}}$ [him]$_{Victim}$ [hard]$_{Degree}$ [for his change of mood]$_{Reason}$.

d. [Rachel]$_{Agent}$ [**injured**]$_{\text{CAUSE\_HARM}}$ [her friend]$_{Victim}$ [by closing the car door on his left hand]$_{Means}$.

Here, the verbs *punch*, *crush*, *slap*, and *injure* are all **frame evoking elements (FEEs)**, that is, they evoke the CAUSE_HARM frame, which in turn exhibits the frame-specific (or "core") roles *Agent*, *Victim*, *Body_part*, and *Cause*, and the more general ("non-core") roles *Degree*, *Reason*, and *Means*. A frame may be evoked by different lexical items, which may in turn inhabit several frames. For instance, the verb *crush* may also evoke the GRINDING frame, and *slap* the IMPACT frame.

The creation of resources that document the realization of semantic roles in example sentences such as FrameNet (Fillmore, Johnson, and Petruck 2003) and PropBank (Palmer, Gildea, and Kingsbury 2005) has greatly facilitated the development of learning algorithms capable of automatically analyzing the role semantic structure of input sentences. Moreover, the shallow semantic analysis produced by existing systems has been shown to benefit a wide spectrum of applications ranging from information extraction (Surdeanu et al. 2003) and question answering (Shen and Lapata 2007), to machine translation (Wu and Fung 2009) and summarization (Melli et al. 2005).

Most **semantic role labeling (SRL)** systems to date conceptualize the task as a supervised learning problem and rely on role-annotated data for model training. Supervised methods deliver reasonably good performance[2] (F_1 measures in the low 80s on standard test collections for English); however, the reliance on labeled training data, which is both difficult and highly expensive to produce, presents a major obstacle to the widespread application of semantic role labeling across different languages and text genres. And although nowadays corpora with semantic role annotations exist in

1 Sentences (1a) and (1b) illustrate the instrument subject alternation and sentences (1a) and (1c) illustrate the causative/inchoative alternation.

2 We refer the interested reader to the reports on the SemEval-2007 shared task (Baker, Ellsworth, and Erk 2007) for an overview of the state of the art.

other languages (e.g., German, Spanish, Catalan, Chinese, Korean), they tend to be smaller than their English equivalents and of limited value for modeling purposes.

It is also important to note that the performance of supervised systems degrades considerably (by 10%) on out-of-domain data even within English, a language for which two major annotated corpora are available (Pradhan, Ward, and Martin 2008). And this is without taking unseen events into account, which unavoidably affect coverage. The latter is especially an issue for FrameNet (version 1.3) which is still under development, despite being a relatively large resource—it contains almost 140,000 annotated sentences for a total of 502 frames, which are evoked by over 5,000 different lexical units. Coverage issues involve not only lexical units but also missing frames and incompletely exemplified semantic roles.

In this article, we attempt to alleviate some of these problems by using semi-supervised methods that make use of a small number of manually labeled training instances and a large number of *unlabeled* instances. Whereas manually labeled data are expensive to create, unlabeled data are often readily available in large quantities. Our approach aims to improve the performance of a supervised SRL system by enlarging its training set with automatically inferred annotations of unlabeled sentences. The key idea of our approach is to find novel instances for classifier training based on their similarity to manually labeled **seed instances**. The underlying assumption is that sentences that are similar in their lexical material and syntactic structure are likely to share a frame semantic analysis. The annotation of an *unlabeled* sentence can therefore be inferred from a sufficiently similar *labeled* sentence. For example, given the labeled sentence (3) and the unlabeled sentence (4), we wish to recognize that they are lexically and structurally similar; and infer that *thumped* also evokes the IMPACT frame, whereas *the rest of his body* and *against the front of the cage* represent the *Impactor* and *Impactee* roles, respectively.

(3) [His back]$_{Impactor}$ [**thudded**]$_{\text{IMPACT}}$ [against the wall]$_{Impactee}$.

(4) The rest of his body thumped against the front of the cage.

We formalize the detection of similar sentences and the projection of role annotations in graph-theoretic terms by conceptualizing the similarity between labeled and unlabeled sentences as a graph alignment problem. Specifically, we represent sentences as dependency graphs and seek an optimal (structural) alignment between them. Given this alignment, we then project annotations from the labeled onto the unlabeled sentence. Graphs are scored using a function based on *lexical* and *syntactic* similarity which allows us to identify alternations like those presented in Example (1) and more generally to obtain training instances with novel structure and lexical material. We obtain the best scoring graph alignment using integer linear programming, a general-purpose exact optimization framework. Importantly, our approach is not tied to a particular SRL system. We obtain additional annotations irrespective of the architecture or implementation details of the supervised role labeler that uses them. This renders our approach portable across learning paradigms, languages, and domains.

After discussing related work (Section 2), we describe the details of our semi-supervised method (Section 3) and then move on to evaluate its performance (Section 4). We conduct two sets of experiments using data from the FrameNet corpus: In Section 5, we apply our method to increase the training data for *known* predicates, that is, words for which some seed annotations already exist. In Section 6, we focus on the complementary task of creating training instances for *unknown* predicates, that is, words that do not occur in the FrameNet corpus at all. Section 7 concludes the article.

2. Related Work

The lack of annotated data presents an obstacle to developing many natural language applications, especially for resource-poor languages. It is therefore not surprising that previous efforts to reduce the need for semantic role annotation have focused primarily on languages other than English.

Annotation projection is a popular framework for transferring semantic role annotations from one language to another while exploiting the translational and structural equivalences present in parallel corpora. The idea here is to leverage the existing English FrameNet and rely on word or constituent alignments to automatically create an annotated corpus in a new language. Padó and Lapata (2009) transfer semantic role annotations from English onto German and Johansson and Nugues (2006) from English onto Swedish. A different strategy is presented in Fung and Chen (2004), where English FrameNet entries are mapped to concepts listed in HowNet, an on-line ontology for Chinese, without consulting a parallel corpus. Then, Chinese sentences with predicates instantiating these concepts are found in a monolingual corpus and their arguments are labeled with FrameNet roles.

Other work attempts to alleviate the data requirements for semantic role labeling within the same language either by increasing the coverage of existing resources or by inducing role annotations from unlabeled data. Swier and Stevenson (2004) propose a method for bootstrapping a semantic role labeler. Given a verb instance, they first select a frame from VerbNet, a semantic role resource akin to FrameNet and PropBank, and label each argument slot with sets of possible roles. Their algorithm then proceeds iteratively by first making initial unambiguous role assignments, and then successively updating a probability model on which future assignments are based. Gordon and Swanson (2007) attempt to increase the coverage of PropBank. Their approach leverages existing annotations to handle novel verbs. Rather than annotating new sentences that contain novel verbs, they find syntactically similar verbs and use their annotations as surrogate training data.

Much recent work has focused on increasing the coverage of FrameNet, either by generalizing semantic roles across different frames or by determining the frame membership of unknown predicates. Matsubayashi, Okazaki, and Tsujii (2009) propose to exploit the relations between semantic roles in an attempt to overcome the scarcity of frame-specific role annotations. They propose several ways of grouping roles into classes based on the FrameNet role hierarchy, human-understandable descriptors of roles, selectional restrictions, and a FrameNet to VerbNet role mapping. They show that transforming this information into feature functions and incorporating it into supervised learning improves role classification considerably.

The task of relating known frames to unknown predicates is addressed primarily by resorting to WordNet (Fellbaum 1998). For example, Burchardt, Erk, and Frank (2005) apply a word sense disambiguation system to annotate predicates with a WordNet sense and hyponyms of these predicates are then assumed to evoke the same frame. Johansson and Nugues (2007b) treat this problem as an instance of supervised classification. Using a feature representation based also on WordNet, they learn a classifier for each frame, which decides whether an unseen word belongs to the frame or not. Pennacchiotti et al. (2008) create "distributional profiles" for frames. The meaning of each frame is represented by a vector, which is the (weighted) centroid of the vectors representing the predicates that can evoke it. Unknown predicates are then assigned to the most similar frame. They also propose a WordNet-based model that computes the similarity between the synsets representing an unknown predicate and those activated by the

predicates of a frame (see Section 6 for details). Das et al. (2010) represent a departure from the WordNet-based approaches in their use of a latent variable model to allow for the disambiguation of unknown predicates.

Unsupervised approaches to SRL have been few and far between. Abend, Reichart, and Rappoport (2009) propose an algorithm that identifies the arguments of predicates by relying only on part-of-speech annotations, without, however, assigning their semantic roles. In contrast, Grenager and Manning (2006) focus on role induction which they formalize as probabilistic inference in a Bayesian network. Their model defines a joint probability distribution over a verb, its semantic roles, and possible syntactic realizations. More recently, Lang and Lapata (2010) formulate the role induction problem as one of detecting alternations and finding a canonical syntactic form for them. Their model extends the logistic classifier with hidden variables and is trained on parsed output which is used as a noisy target for learning.

Our own work aims to reduce but not entirely eliminate the annotation effort involved in semantic role labeling. We thus assume that a small number of manual annotations is initially available. Our algorithm augments these with unlabeled examples whose roles are inferred automatically. We apply our method in a monolingual setting, and thus do not project annotations between languages but within the same language. Importantly, we acquire new training instances for both known and unknown predicates. Previous proposals extend FrameNet with novel predicates without inducing annotations that exemplify their usage. We represent labeled and unlabeled instances as graphs, and seek to find a globally optimal alignment between their nodes, subject to semantic and structural constraints. Finding similar labeled and unlabeled sentences is reminiscent of paraphrase identification (Qiu, Kan, and Chua 2006; Wan et al. 2006; Das and Smith 2009; Chang et al. 2010), the task of determining whether one sentence is a paraphrase of another. The sentences we identify are not strictly speaking paraphrases (even if the two predicates are similar their arguments often are not); however, the idea of modeling the correspondence structure (or alignment) between parts of the two sentences is also present in the paraphrase identification work (Das and Smith 2009; Chang et al. 2010). Besides machine translation (Matusov, Zens, and Ney 2004; Taskar, Lacoste-Julien, and Klein 2005), methods based on graph alignments have been previously employed for the recognition of semantic entailments (Haghighi, Ng, and Manning 2005; de Marneffe et al. 2007), where an optimization problem similar to ours is solved using approximate techniques (our method is exact) and an alignment scoring function is learned from annotated data (our scoring function does not require extensive supervision). On a related note, de Salvo Braz et al. (2005) model entailments via a subsumption algorithm that operates over concept graphs representing a source S and target T sentence and uses integer linear programming to prove that $S \sqsubseteq T$.

3. Method

In this section we describe the general idea behind our semi-supervised algorithm and then move on to present our specific implementation. Given a set L of sentences labeled with FrameNet frames and roles (the **seed corpus**) and a (much larger) set U of unlabeled sentences (the **expansion corpus**), we wish to automatically create a set $X \subset U$ of novel annotated instances. Algorithm 1 describes our approach, which consists of two parts. In the **labeling stage**, annotations are proposed for every unlabeled sentence (lines 1–20), and in the **selection stage**, instances with high quality annotations are chosen to make up the final new corpus (lines 21–26).

Algorithm 1 Given set L of labeled seed sentences and set U of unlabeled sentences, produce expansion set X containing the k nearest neighbors of each seed.

```
Initialize X_l ← ∅ for all l ∈ L
for u ∈ U do
    for relevant target predicate t in u do
        s* ← 0
        for l ∈ relevant seeds from L do
            M ← predicate–argument structure of FEE in l
            N ← predicate–argument structure of t in u
            (σ, s) ← optimal alignment of M and N and its score {cf. Figure 2}
            if (s > s*) and (σ covers all role-bearing nodes in M) then
                l* ← l
                σ* ← σ
                s* ← s
            end if
        end for
        if s* > 0 then
            u′ ← u with annotation projected via σ* from l*
            add (u′, s*) to X_l
        end if
    end for
end for
X ← ∅
for l ∈ L do
    for (u′, s) ∈ X_l do
        add u′ to X if s is among the k highest scores in X_l
    end for
end for
return X
```

In the labeling stage, (almost) every unlabeled sentence $u \in U$ receives an annotation via projection from the seed $l^* \in L$ most similar to it. In theory, this means that each unlabeled sentence u is compared with each labeled seed l. In practice, however, we reduce the number of comparisons by requiring that u and l have identical or at least similar FEEs. This process will yield many sentences for every seed with annotations of varying quality. In default of a better way of distilling high-quality annotations, we use similarity as our criterion in the selection stage. From the annotations originating from a particular seed, we therefore collect the k instances with the highest similarity values. Our selection procedure is guided by the seeds available rather than the corpus from which unlabeled sentences are extracted. This is intended, as the seeds can be used to create a balanced training set or one that exemplifies difficult or rare training instances.

In the remainder of this section, we present the labeling stage of our algorithm in more detail. Section 3.1 formally introduces the notion of semantically labeled dependency graphs and defines the subgraphs M and N representing relevant predicate–argument structures. Section 3.2 formalizes alignments as mappings between graph nodes and defines our similarity score as a function on alignments between labeled and unlabeled dependency graphs. Section 3.3 formulates an integer linear program

(ILP) for finding optimal alignments, and Section 3.4 presents an efficient algorithm for solving this ILP. Finally, Section 3.5 describes how annotations are projected from labeled onto unlabeled graphs.

3.1 Semantically Labeled Dependency Graphs

Seed sentences labeled with role-semantic annotations are represented by dependency graphs. The latter capture grammatical relations between words via directed edges from syntactic heads to their dependents (e.g., from a verb to its subject or from a noun to a modifying adjective). Edges can be labeled to indicate the type of head–dependent relationship (e.g., subject, object, modifier). In our case, dependency graphs are further augmented with FrameNet annotations corresponding to the FEE and its semantic roles.

A dependency graph of the sentence *Old Herkimer blinked his eye and nodded wisely* is shown in Figure 1. Nodes are indicated by rectangles and dependencies by edges (arrows). Solid arrows represent syntactic dependencies (e.g., subject, object), and dashed arrows correspond to FrameNet annotations. Here, *blink* evokes the frame *Body_movement*, *Herkimer* bears the role *Agent*, and *eye* the role *Body_part*.

Unfortunately, FrameNet annotations have not been created with dependency graphs in mind. FEEs and roles are marked as substrings and contain limited syntactic information, distinguishing only the grammatical functions "external argument," "object," and "dependent" for the arguments of verbal FEEs. To obtain dependency graphs with semantic annotations like the one shown in Figure 1, we parse the sentences in the seed corpus with a dependency parser and compare the FrameNet annotations (substrings) to the nodes of the dependency graph. For the FEE, we simply look for a graph node that coincides with the word marked by FrameNet. Analogously, we map

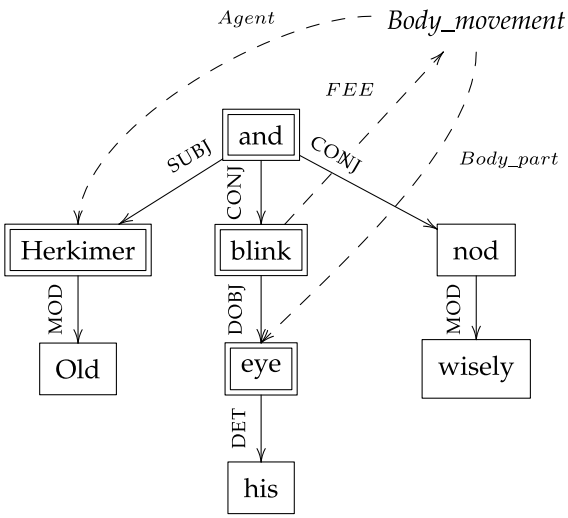


Figure 1
Dependency graph with semantic annotations for the sentence *Old Herkimer blinked his eye and nodded wisely* (taken from the FrameNet corpus). Nodes in the alignment domain are indicated by double frames. Labels in *italics* denote frame roles, and grammatical roles are rendered in small capitals. Annotations are only shown for the predicate *blink*, which evokes the frame *Body_Movement*.

role annotations onto the graph by finding a node with a yield equal to the marked substring, that is, a node that (together with its dominated nodes) represents the words of the role. Our experiments make use of the dependency graphs produced by RASP (Briscoe, Carroll, and Watson 2006), although there is nothing inherent in our approach that assumes this specific parser. Any other dependency parser with broadly similar output could be used instead.

Searching for nodes representing the FEE and its semantic roles may in some cases yield no match. There are two reasons for this—parser errors and role annotations violating syntactic structure. We address this problem heuristically: If no perfect match is found, the closest match is determined based on the number of mismatching characters in the string. We thus compute a mismatch score for the FEE and each role. To make allowances for parser errors, we compute these scores for the *n*-best parses produced by the dependency parser and retain the dependency graph with the lowest mismatch. This mapping procedure is more thoroughly discussed in Fürstenau (2008).

Each sentence in the seed corpus contains annotations for a predicate and its semantic roles. A complex sentence (with many subordinate clauses) will be represented by a large dependency graph, with only a small subgraph corresponding to these annotations. Our method for computing alignments between graphs only considers subgraphs with nodes belonging to the predicate-argument structure in question. This allows us to compare graphs in a computationally efficient manner as many irrelevant alignments are discarded, although admittedly the entire graph may provide useful contextual clues to the labeling problem.

We are now ready to define the **alignment domain** M of a labeled dependency graph. Let p be a node (i.e., word) in the graph corresponding to the FEE. If there are no mismatches between semantic and syntactic arguments, we expect all roles in the graph to be instantiated by syntactic dependents of p. Although this is often the case, it does not always hold—for example, because of the way the dependency parser analyzes raising, control, or coordination structures. We therefore cannot simply define M as the set of direct dependents of the predicate, but also have to consider **complex paths** between p and role-bearing nodes. An example is given in Figure 1, where the role *Agent* is filled by a node that is not dominated by the FEE *blink*; instead, it is connected to *blink* by the complex path (CONJ^{-1}, SUBJ). For a given sentence, we build the set of all such complex paths to any role-bearing node and also include all nodes connected to p by one of these paths. We thus define the subgraph M to contain:

i. the predicate node p

ii. all direct dependents of p, except auxiliaries

iii. all nodes on complex paths from p to any role-bearing node

iv. single direct dependents of any preposition or conjunction node which is in (ii) or end-point of a complex path covered in (iii)

In Figure 1 the nodes in the alignment domain are indicated by double frames.

In an *unlabeled* dependency graph we similarly identify the **alignment range** as the subgraph corresponding to the predicate–argument structure of a **target predicate**. As we do not have any frame semantic analysis for the unlabeled sentence, however, we cannot determine a set of complex paths. We could ignore complex paths altogether and thus introduce a substantial asymmetry into the comparison between a labeled and an unlabeled sentence, as unlabeled sentences would be assumed to be structurally simpler

than labeled ones. This assumption will often be wrong and moreover introduce a bias towards simpler structures for the new annotations. To avoid this, we reuse the set of complex paths from the labeled sentence. Although this is not ideal either (it makes the comparison asymmetrically dependent on the annotation of the labeled sentence) it allows us to compare labeled and unlabeled sentences on a more equal footing. We therefore define the alignment range N in exact analogy to the alignment domain M, the only exception being that complex paths to role-bearing nodes are determined by the labeled partner in the comparison.

3.2 Scoring Graph Alignments

We conceptualize the similarity between subgraphs representing predicate–argument structures as an alignment problem. Specifically, we seek to find an optimal alignment between the alignment domain M of a labeled graph and the alignment range N of an unlabeled sentence. Alignments are scored using a similarity measure that takes syntactic and lexical information into account.

We formalize the **alignment** between M and N as a partial injective function from M to N, that is, a function $\sigma : M \to N \cup \{\epsilon\}$ where $\sigma(x) = \sigma(x') \neq \epsilon$ implies $x = x'$. Here, ϵ denotes a special empty value. We say that $x \in M$ is **aligned** to $x' \in N$ by σ, iff $\sigma(x) = x'$. Correspondingly, a node $x \in M$ with $\sigma(x) = \epsilon$ or a node $x' \in N$ that is not the image of any $x \in M$ is called **unaligned**. Figure 2 shows an example of an alignment

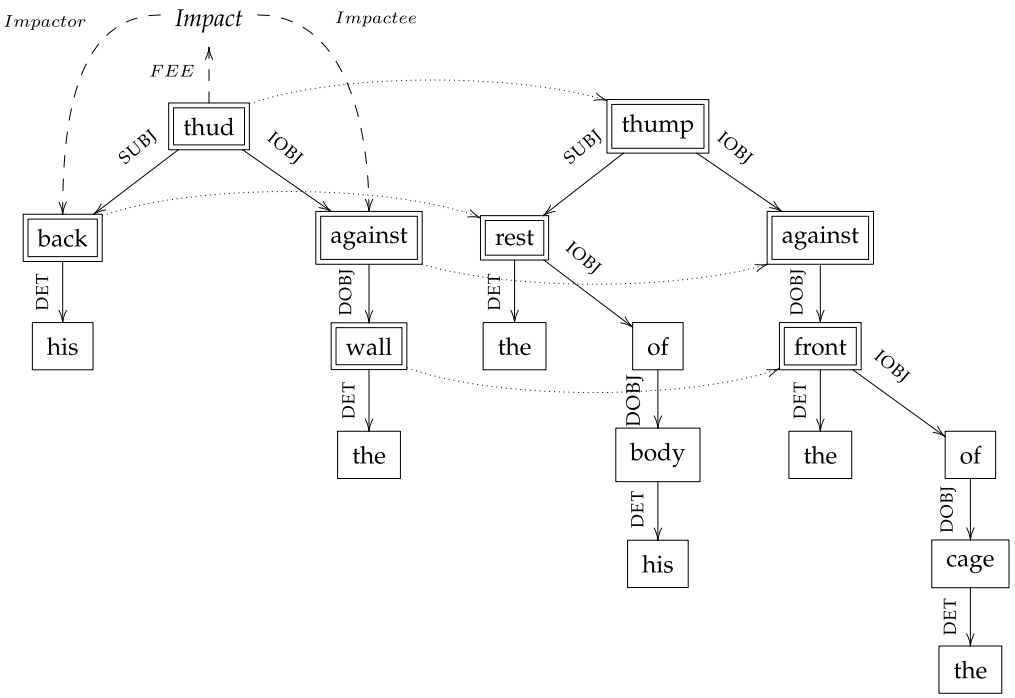


Figure 2
The dotted arrows show aligned nodes in the graphs for the two sentences *His back thudded against the wall* and *The rest of his body thumped against the front of the cage* (graph edges are also aligned to each other). The nodes in the alignment domain and alignment range are indicated by double frames.

between a labeled and an unlabeled dependency graph for the predicates *thud* and *thump*.

Each alignment σ between M and N receives a score, the weighted sum of the lexical similarity between nodes (lex) and syntactic similarity between edges (syn):

$$\text{score}(\sigma) := \frac{1}{C}\left(\sum_{\substack{x\in M\\ \sigma(x)\neq\epsilon}} \text{lex}\,(x,\sigma(x)) + \alpha\sum_{\substack{(x_1,x_2)\in E(M)\\ (\sigma(x_1),\sigma(x_2))\in E(N)}} \text{syn}\left(r_{x_2}^{x_1}, r_{\sigma(x_2)}^{\sigma(x_1)}\right)\right) \tag{1}$$

Here, $E(M)$ and $E(N)$ denote the sets of graph edges between the nodes of M and N, respectively, while $r_{x_2}^{x_1}$ is the label of the edge (x_1, x_2), that is, the grammatical relation between these two nodes.

Equation (1) introduces a normalizing factor C whose purpose is to render similarity scores of different pairs of sentences comparable. Without normalization, it would be easier to achieve high similarity to a complex predicate–argument structure than a simpler one, which is counter-intuitive. This can be seen from the fact that the self-similarity of a sentence (i.e., the similarity of a sentence to itself) depends on the number of nodes in M. Assuming that the maximal value for lex and syn is 1 for identical words and grammatical relations, self-similarity is then $|M| + \alpha|E(M)|$ and constitutes an upper bound for the similarity between any two sentences. We could use this term to normalize the similarity score. However, this would only account for unaligned or badly aligned nodes and edges in the labeled sentence while ignoring the unlabeled partner. To obtain a symmetric normalization factor we therefore define:

$$C := \sqrt{|M|\cdot|N|} + \alpha\sqrt{|E(M)|\cdot|E(N)|} \tag{2}$$

C is now symmetric in the two sentences and when introduced in equation (1) leads to self-similarities of 1:

$$\text{score}(\sigma_{\text{self}}) = \frac{1}{\sqrt{|M|^2} + \alpha\sqrt{E(M)^2}}\left(|M|\cdot 1 + \alpha\cdot|E(M)|\cdot 1\right) = 1 \tag{3}$$

Notice that our formulation uses the same score for finding whether there exists an alignment and for evaluating its quality. Consequently, our algorithm will attempt to construct an alignment even if there is none, that is, in cases where the similarity between labeled and unlabeled sentences is low. Our approach is to filter out erroneous alignments by considering only the k nearest neighbors of each seed. Alternatively, we could first establish valid alignments and then score them; we leave this to future work. The employed score is the weighted combination of lexical and syntactic similarity. In our experiments we use cosine similarity in a vector space model of co-occurrence statistics for lex and define syn as a binary function reflecting the identity of grammatical relations (see Section 4 for details). Other measures based on WordNet (e.g., Budanitsky and Hirst 2001) or finer grammatical distinctions are also possible.

3.3 ILP Formulation

We define the similarity of two predicate–argument structures as the maximum score of any alignment σ between them. Intuitively, the alignment score corresponds to the amount of changes required to transform one graph into the other. High scores indicate

high similarity and thus minimal changes. We do not need to formalize such changes, although it would be possible to describe them in terms of substitutions, deletions, and insertions. For our purposes, the alignment scores themselves can be used to indicate whether two graphs are substantially similar to warrant projection of the frame semantic annotations. We do this by finding an optimal alignment, that is, an alignment with the highest score as defined in Equation (1).

To solve this optimization problem efficiently, we recast it as an integer linear program (ILP). The ILP modeling framework has been recently applied to a wide range of natural language processing tasks, demonstrating improvements over more traditional optimization methods. Examples include reluctant paraphrasing (Dras 1999), relation extraction (Roth and tau Yih 2004), semantic role labeling (Punyakanok et al. 2004), concept-to-text generation (Marciniak and Strube 2005; Barzilay and Lapata 2006), dependency parsing (Riedel and Clarke 2006), sentence compression (Clarke and Lapata 2008), and coreference resolution (Denis and Baldridge 2007). Importantly, the ILP approach[3] delivers a globally optimal solution by searching over the entire alignment space without employing heuristics or approximations (see de Marneffe et al. [2007] and Haghighi, Ng, and Manning [2005]). Furthermore, an ILP-based formulation seems well-suited to our problem because the domain of the optimization, namely, the set of partial injective functions from M to N, is discrete. We define arbitrary linear orders on the sets M and N, writing $M = \{n_1, \ldots, n_m\}$ and $N = \{n'_1, \ldots, n'_n\}$ and then introduce binary indicator variables x_{ij} to represent an alignment σ:

$$x_{ij} := \begin{cases} 1 & \text{if } \sigma(n_i) = n'_j \\ 0 & \text{else} \end{cases} \tag{4}$$

Each alignment σ thus corresponds to a distinct configuration of x_{ij} values. In order to ensure that the latter describe a partial injective function, we enforce the following constraints:

1. $\forall_j : \sum_{1 \le i \le m} x_{ij} \le 1$ (Each node in N is aligned to at most one node in M.)
2. $\forall_i : \sum_{1 \le j \le n} x_{ij} \le 1$ (Each node in M is aligned to at most one node in N.)

We can now write Equation (1) in terms of the variables x_{ij} (which capture exactly the same information as the function σ):

$$\text{score}(x) = \frac{1}{C} \left(\sum_{\substack{1 \le i \le m \\ 1 \le j \le n}} \text{lex}\left(n_i, n'_j\right) x_{ij} + \alpha \cdot \sum_{\substack{1 \le i,k \le m \\ 1 \le j,l \le n}} \text{syn}\left(r^{n_i}_{n_k}, r^{n'_j}_{n'_l}\right) x_{ij} x_{kl} \right) \tag{5}$$

Note that Equations (1) and (5) are summations of the same terms.[4] However, Equation (5) is not linear in the variables x_{ij} as it contains products of the form $x_{ij}x_{kl}$.

3 It is outside the scope of this article to provide an introduction to ILP. We refer the interested reader to Winston and Venkataramanan (2003) and Vanderbei (2001) for comprehensive overviews.

4 For convenience, we define $r^{n_1}_{n_2} = \epsilon$ if there is no relation between n_1 and n_2, and assume that syn is 0 if either of its arguments is ϵ.

This can be remedied through the introduction of another set of binary variables y_{ijkl} subject to additional constraints ensuring that $y_{ijkl} = x_{ij}x_{kl}$:

3. $\forall_{i,j,k,l} : y_{ijkl} \leq x_{ij}$
4. $\forall_{i,j,k,l} : y_{ijkl} \leq x_{kl}$
5. $\forall_{i,j,k,l} : y_{ijkl} \geq x_{ij} + x_{kl} - 1$

We also want to make sure that the FEE of the labeled sentence is aligned to the target predicate of the unlabeled sentence. We express this with the following constraint, assuming that the FEE and the target predicate are represented by n_1 and n'_1, respectively:

6. $x_{11} = 1$

We therefore have to solve an ILP in the $mn + m^2n^2$ variables x_{ij} and y_{ijkl}, subject to $m + n + 3m^2n^2 + 1$ constraints (see constraints (1)–(6)), with the objective function:

$$\text{score}(x, y) = \frac{1}{C} \left(\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \text{lex}\left(n_i, n'_j\right) x_{ij} + \alpha \cdot \sum_{\substack{1 \leq i,k \leq m \\ 1 \leq j,l \leq n}} \text{syn}\left(r^{n_i}_{n_k}, r^{n'_j}_{n'_l}\right) y_{ijkl} \right) \tag{6}$$

Exact optimization for the general ILP problem is NP-hard (Cormen, Leiserson, and Rivest 1992). ILPs with a totally unimodular constraint matrix[5] are solvable efficiently, using polynomial time algorithms. In this special case, it can be shown that the optimal solution to the linear program is integral. Unfortunately, our ILP falls outside this class due to the relatively complex structure of our constraints. This can be easily seen when considering the three constraints $x_{11} + x_{12} + \cdots + x_{1m} \leq 1$, $-x_{11} + y_{1112} \leq 0$ and $-x_{12} + y_{1112} \leq 0$. The coefficients of the three variables x_{11}, x_{12}, and y_{1112} in these constraints make up the matrix

$$\begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{pmatrix}$$

The determinant of this matrix is 2 and therefore the complete coefficient matrix of the ILP has a quadratic submatrix with a determinant that is not 0 or ± 1, which means that it is not totally unimodular. Indeed, it has been shown that the structural matching problem is NP-hard (Klau 2009).

3.4 Solving the ILP

There are various techniques for finding the optimal solution of an ILP, such as approximation with error bounds (Klau 2009) or application of the branch-and-bound

5 A matrix A is totally unimodular if every square sub-matrix of A has its determinant equal to 0, $+1$, or -1.

algorithm (Land and Doig 1960). The latter allows for solving an ILP exactly and significantly faster than by naive enumeration. It does this by *relaxing* the integer constraints and solving the resulting LP problem, known as the **LP relaxation**. If the solution of the LP relaxation is integral, then it is the optimal solution. Otherwise, the resulting solution provides an upper bound on the solution for the ILP. The algorithm proceeds by creating two new sub-problems based on the non-integer solution for one variable at a time. These are solved and the process repeats until the optimal integer solution is found. Our alignment problem has only binary variables and is thus an instance of a "pure" 0–1 ILP. For such problems, implicit enumeration can be used to simplify both the braching and bounding components of the branch-and-bound process and to determine efficiently when a node is infeasible. This is achieved by systematically evaluating all possible solutions, without, however, explicitly solving a potentially large number of LPs derived from the relaxation.

To obtain a solution for the ILP in Section 3.3, we could have used any solver that implements the standard branch-and-bound algorithm. To speed up computation time, we have instead modified the branch-and-bound algorithm so as to take into account the special structure of our graph alignment problem. Our own algorithm follows the principles of branch-and-bound but avoids explicit representation of the variables y_{ijkl}, performs early checks of the constraints on the variables x_{ij} on branching, and takes into account some of the constraints on the variables y_{ijkl} for the estimation of lower and therefore more efficient bounds. In the following, we first describe our modified algorithm and then assess its runtime in comparison to a publicly available solver.

Algorithm 2 shows how to find an optimal alignment σ^* with score s^* in pseudocode. σ_0 and σ_1 denote partial solutions, while completions are built in σ. syn^* is the maximum possible value of syn, that is, $\text{syn}^* = 1$ for a binary measure. We initialize σ^* with the trivial solution which aligns n_1 to n'_1 and leaves all other nodes unaligned.[6] This gives a score of $\text{lex}(n_1, n'_1)$. To find better solutions we start with an initial partial alignment σ_0, which contains only the mapping $n_1 \mapsto n'_1$ and leaves the alignments of all other $n \in M$ unspecified. (Note that this is different from the complete alignment σ^* which specifies those nodes as unaligned: $n \mapsto \epsilon$.) As in the general branch-and-bound algorithm, the space of all alignments is searched recursively by branching on the alignment decision for each remaining node. A branch is left as soon as an upper bound on the achievable score indicates that the current best solution cannot be improved within this branch.

Given a partial alignment σ_0 (the initial or any subsequent one) defined on some subset of M, we estimate a suitable bound by extending σ_0 to a complete function σ on all nodes in M: Each of the remaining nodes is aligned to its partner in N maximizing lex. If no positive value can be found for lex, the node is defined as unaligned. We then define the bound s as the score of σ_0 together with the lexical scores of the newly created alignments and a hypothetical syntactic score which assumes that each of the newly considered edges is aligned perfectly, that is, with the maximum value syn^* attainable by syn. (This is a lower bound than the one a naive application of the branch-and-bound algorithm would compute.)

Of course, σ need not fulfill the constraints of the ILP and s need not be an attainable score. It is, however, an upper bound for the score of any valid alignment. If it is not

6 In the description of the algorithm, we use the more intuitive notation $n_i \mapsto n'_j$ to indicate that n_i is aligned to n'_j. Note, however, that this could be equivalently formulated in terms of the ILP variables (i.e., $x_{ij} = 1$), and our algorithm still broadly follows the branch-and-bound procedure for ILPs.

Algorithm 2 Branch-and-bound algorithm to find optimal alignment σ^*.

$\sigma^* \leftarrow \{n_1 \mapsto n'_1, n_2 \mapsto \epsilon, \ldots, n_m \mapsto \epsilon\}$
$s^* \leftarrow \text{lex}(n_1, n'_1)$
Initialize stack with single item $(\{n_1 \mapsto n'_1\}, \text{lex}(n_1, n'_1))$
while stack not empty **do**
    pop $(\sigma_0 = \{n_1 \mapsto n'_1, \ldots, n_k \mapsto \sigma_0(n_k)\}, s_0)$ from stack
    $\sigma \leftarrow \sigma_0$
    $s \leftarrow s_0$
    for i **from** $k+1$ **to** m **do**
        $n' \leftarrow \arg\max_{n' \in N} \text{lex}(n_i, n')$
        if $\text{lex}(n_i, n') > 0$ **then**
            $\sigma \leftarrow \sigma \cup \{n_i \mapsto n'\}$
            $s \leftarrow s + \text{lex}(n_i, n') + \alpha \cdot \text{syn}^* \cdot |\text{neighbors}(n_i)|$
        else
            $\sigma \leftarrow \sigma \cup \{n_i \mapsto \epsilon\}$
        end if
    end for
    if $s > s^*$ **then**
        if $\text{valid}(\sigma, k)$ **and** $\text{syn\_maximizing}(\sigma, k)$ **then**
            $\sigma^* \leftarrow \sigma$
            $s^* \leftarrow s$
        else
            for $n' \in (N - \text{range}(\sigma_0)) \cup \{\epsilon\}$ **do**
                $\sigma_1 \leftarrow \sigma_0 \cup \{n_{k+1} \mapsto n'\}$
                $s_1 \leftarrow s_0 + \text{lex}(n_{k+1}, n')$
                for i **from** 1 **to** k **do**
                    $s_1 \leftarrow s_1 + \alpha \cdot \text{syn}\left(r_{n_i}^{n_{k+1}}, r_{\sigma_0(n_i)}^{n'}\right) + \alpha \cdot \text{syn}\left(r_{n_{k+1}}^{n_i}, r_{n'}^{\sigma_0(n_i)}\right)$
                end for
                push (σ_1, s_1) onto stack
            end for
        end if
    end if
end while
return (σ^*, s^*)

$\text{valid}(\sigma, k) := \forall_{k<i<j\leq m} : \sigma(n_i) \neq \sigma(n_j) \vee \sigma(n_i) = \epsilon$
$\text{syn\_maximizing}(\sigma, k) := \forall_{(n_i,n_j)\in E(M)} : (i \leq k \wedge j \leq k) \vee \left(\text{syn}\left(r_{n_j}^{n_i}, r_{\sigma(n_j)}^{\sigma(n_i)}\right) = \text{syn}^*\right)$

greater than the current best score s^*, we leave the current branch. Otherwise, we check if σ is a valid alignment with score s, that is, if it satisfies the constraints of the ILP and s is its score (which means that the assumptions of perfect syntactic scores were justified). If this is the case, we have a new current optimum and do not need to follow the current branch any more either. If, however, the bound s is greater than the current optimum s^*, but σ violates some constraints or does not achieve a score of s because it contains imperfect syntactic alignments, we have to branch on the decision of how to extend σ_0 by an additional alignment link. We consider the next node with unspecified alignment and recursively apply the algorithm to extensions of σ_0. Each extension σ_1 aligns this node to a partner in N that has thus far been left unaligned. (This simple

check of constraint (1), which extends the general branch-and-bound algorithm, avoids recursion into branches that cannot contain any valid solutions.) The partial score s_1 corresponding to σ_1 is computed by taking into account the consequences of the new alignment to the lexical and syntactic scores.

We found this algorithm to be very effective in solving the ILPs arising in our experiments. While its worst case performance is still exponential in the number of aligned nodes and edges, it almost always finds the optimum within a relatively small number of iterations of the outer loop (line 4 in Figure 2). This is also due to the fact that the alignment domain and range are typically not very large. In a realistic application of our method, 70% of the ILPs were solvable with less than 100 iterations, 93% with less than 1,000 iterations, 98.6% with less than 10,000 iterations, and 99.95% with less than 1,000,000 iterations. As the remaining 0.05% of the ILPs may still take an inordinate amount of time, we abort the search at this point. In this case, it is highly likely that the alignment domain and range are large and any resulting alignment would be overly specific and thus not very useful. Aborting at 1,000,000 iterations is also preferable to a time-out based on processing time, as it makes the result deterministic and independent of the specific implementation and hardware. All expansion sets in the experiments described in Sections 5 and 6 were computable within hours on modern hardware and under moderate parallelization, which is trivial to implement over the instances of the unlabeled corpus.

Because our branch-and-bound algorithm performs exact optimization, it could be replaced by any other exact solution algorithm, without affecting our results. To assess its runtime performance further, we compared it to the publicly available `lp_solve`[7] solver which can handle integer variables via the branch-and-bound algorithm. We sampled 100 alignment problems for each problem size (measured in number of nodes in the alignment domain) and determined the average runtime of our algorithm and `lp_solve`. (The latter was run with the option `-time`, which excludes CPU time spent on input parsing). Figure 3 shows how the average time required to solve an ILP varies with the problem size. As can be seen, our algorithm is about one order of magnitude more efficient than the implementation of the general-purpose branch-and-bound algorithm.

3.5 Annotation Projection

Given a labeled graph l, an unlabeled graph u, and an optimal alignment σ between them, it is relatively straightforward to project frame and role information from one to the other. As described in Section 3.1, frame names are associated with the nodes of their FEEs and role names with the nodes of their role filler heads. By definition, all of these nodes are in the alignment range M. It is therefore natural to label $\sigma(x) \in N$ with the role carried by x for each role-bearing node $x \in M$. The only complicating factor is that we have allowed unaligned nodes, that is, nodes with $\sigma(x) = \epsilon$. Although this is useful for ignoring irrelevant nodes in M, we must decide how to treat these when they are role-bearing (note that FEEs are always aligned by constraint (6), so frame names can always be projected).

A possible solution would be to only project roles on nodes x with $\sigma(x) \neq \epsilon$, so that roles associated with unaligned nodes do not show up in the inferred annotation. Unfortunately, allowing such partial projections introduces a systematic bias in favor

7 Version 5.5, available at `http://lpsolve.sourceforge.net/`.

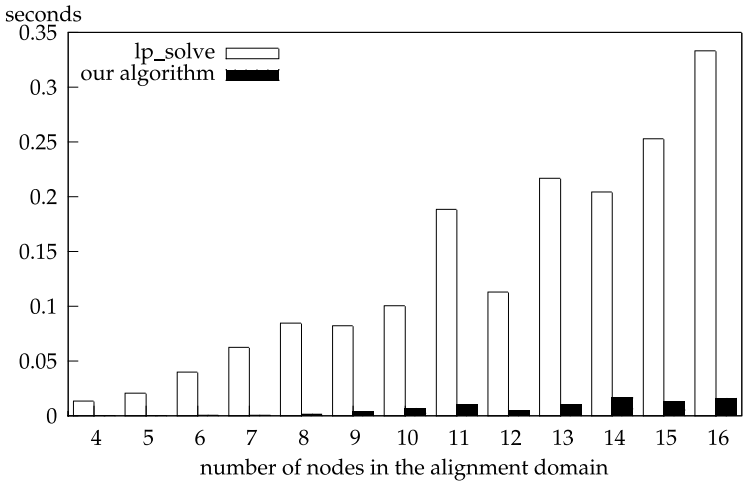


Figure 3
Average time required to solve an ILP as a function of the size of the alignment domain.

of simpler structures. When these new instances are used as a training set for a role labeler, they will bias the classifier towards under-annotating roles and thus decrease performance. We therefore do not want to allow partial projections and demand that $\sigma(x) \neq \epsilon$ for all role-bearing nodes x.

We could incorporate this additional constraint into the ILP by finding a (lower scoring) solution that satisfies it. However, there is no theoretical justification for favoring a lower ranking alignment over the optimal one only because of projection requirements. If lexical and structural measures tell us that a certain alignment is best, we should not dismiss this information, but rather take the contradiction between the optimal alignment and the frame semantic (non-)projectability to indicate that l is not suitable for inferring a labeling of u. There are several possible reasons for this, ranging from idiosyncratic annotations to parser or pre-processing errors. We therefore do not discard the optimal alignment in favor of a lower scoring one, but rather dismiss the seed l as a source of information for inferring a labeling on u. This reflects our precision-oriented approach: If u does not find a better partner among the other seeds, it will be discarded as unsuitable for the expansion set.

4. Experimental Set-up

In this section, we describe the data and supervised semantic role labeler used in our experiments and explain how the free parameters of our method were instantiated. We then move on to present two experiments that evaluate our semi-supervised method.

4.1 Data

In our experiments, we use various subsets of the English FrameNet corpus (version 1.3; Fillmore, Johnson, and Petruck 2003) as seed sets for our semi-supervised method and as test sets in our evaluation. We only consider sentences with verbal FEEs (60,666 in total). Furthermore, we always assume that an oracle identifies the verbal predicate, so recognition of the FEE is not part of our evaluation. Unlabeled sentences for expansion

Table 1
Features used by the frame classifier. Example values for the annotated graph in Figure 1 are given in parentheses.

Feature	Type	Description and example value
target_lemma	atomic	lemma of the target node (*blink*)
frames	set	frames that can be evoked by the target verb ({BODY_MOVEMENT})
voice	binary	voice of the target node (active)
parent_word	set	lemma of the parents of the target node ({*and*})
parent_POS	set	part of speech of the parents of the target node ({CC})
rel_to_parent	set	grammatical relations between the target node and its parents ({CONJ})
parent_has_obj	binary	whether any parents have an outgoing "object" relation (no)
dsubcat	atomic	subcategorization frame, the multi-set of all outgoing relations of the target node (DOBJ)
child_word_set	set	lemma of the children of the target node ({*eye*})
child_dep_set	set	outgoing relations of the target node ({DOBJ})
child_word_dep_set	set	pair (lemma, relation) for the children of the target node ({(*eye*, DOBJ)})

were taken from the British National Corpus (BNC), excluding sentences with manual annotations in FrameNet. The BNC is considerably larger compared with FrameNet, approximately by a factor of 100. Dependency graphs were produced with RASP (Briscoe, Carroll, and Watson 2006). Frame semantic annotations for labeled sentences were merged with their dependency-based representations as described in Section 3.1. Sentences for which this was not possible (mismatch score greater than 0) were excluded from the seed set, but retained in the test sets to allow for unbiased evaluation. For unlabeled BNC sentences, we used an existing RASP-parsed version of the BNC (Andersen et al. 2008).

4.2 Supervised SRL System

A natural way of evaluating the proposed semi-supervised method is by comparing two instantiations of a supervised SRL system, one that is trained solely on FrameNet annotations and one that also uses the additional training instances produced by our algorithm. We will henceforth use the term **unexpanded** to refer to the corpus (and system trained on it) that contains only human-annotated instances, and accordingly, the term **expanded** to describe the corpus (and system) resulting from the application of our method or any other semi-supervised approach that obtains training instances automatically. As our approach is based on dependency graphs, we employed a dependency-based SRL system for evaluation.[8]

We thus implemented a supervised SRL system based on the features proposed by Johansson and Nugues (2007a). Many of these features have been found useful in a number of previous SRL systems, and can be traced back to the seminal work of Gildea and Jurafsky (2002). Our own implementation uses the features listed in Tables 1 and 2 for frame labeling and role labeling, respectively. Atomic features are converted

8 Semantic role labelers that take advantage of dependency information perform comparably to those that rely on phrase structure trees (Johansson 2008).

Table 2
Features used by the role classifiers. Example values for the *Body_part* role of the annotated graph in Figure 1 are given in parentheses.

Feature	Type	Description and example value
target_lemma	atomic	lemma of the FEE (*blink*)
target_POS	atomic	part of speech of the FEE (VVD)
roles	set	roles that can feature in the given frame ({*Agent*, *Body_part*, *Addressee*, ...})
voice	binary	voice of the FEE (active)
parent_word	set	lemma of the parents of the FEE ({*and*})
parent_POS	set	part of speech of the parents of the FEE ({CC})
rel_to_parent	set	grammatical relation between the FEE and its parents ({CONJ})
parent_has_obj	binary	whether any parents have an outgoing "object" relation (no)
dsubcat	atomic	subcategorization frame, multi-set of all outgoing relations of the FEE (DOBJ)
child_dep_set	set	outgoing relations of the FEE ({DOBJ})
arg_word	atomic	lemma of the argument (*eye*)
arg_POS	atomic	part of speech of the argument (NN1)
position	atomic	position of the argument (before, on, or after) in the sentence, relative to the FEE (after)
left_word	atomic	lemma of the word to the left of the argument in the sentence (*his*)
left_POS	atomic	part of speech of the word to the left of the argument in the sentence (APP$)
right_word	atomic	lemma of the word to the right of the argument in the sentence (*and*)
right_POS	atomic	part of speech of the word to the right of the argument in the sentence (CC)
path	atomic	path of grammatical relations between FEE and argument (DOBJ)
function	set	relations between argument and its heads ({DOBJ})

into binary features of the SVM by 1-of-k coding, and for set features each possible set element is represented by its own binary feature. (Features pertaining to parent nodes are set features as we do not require our dependency graphs to be trees and a node can therefore have more than one parent.) We followed a classical pipeline architecture, first predicting a frame name for a given lexical unit, then identifying role-bearing dependency graph nodes, and finally labeling these nodes with specific roles. All three classification stages were implemented as support vector machines, using LIBLINEAR (Fan et al. 2008). The frame classifier is trained on instances of all available predicates, while individual role classifiers are trained for each frame. The one-vs-one strategy (Friedman 1996) was employed for multi-classification.

We evaluate the performance of the SRL system on a test set in terms of *frame accuracy* and *role labeling* F_1. The former is simply the relative number of correctly identified frame names. The latter is based on the familiar measure of labeled F_1 (the harmonic mean of labeled precision and recall). When a frame is labeled incorrectly, however, we assume that its roles are also misclassified. This is in agreement with the notion of frame-specific roles. Moreover, it allows us to compare the performance of different classifiers, which would not be possible if we evaluated role labeling performance on changing test sets, such as the set of only those sentences with correct frame predictions.

The misclassification penalty C for the SVM was optimized on a small training set consisting of five annotated sentences per predicate randomly sampled from FrameNet. We varied C for the frame classification, role recognition, and role classification SVMs

between 0.01 and 10.0 and measured F_1 on a test set consisting of 10% of FrameNet (see Section 5.1). For frame and role classification, we did not observe significant changes in F_1 and therefore maintained the default of $C = 1.0$. For role recognition, we obtained best performance with $C = 0.1$ (F_1 was 38.78% compared to 38.04% with the default $C = 1$), which we subsequently used for all our experiments. All other SVM parameters were left at their default values.

4.3 Lexical and Syntactic Similarity

Our definition of the lexical similarity measure, lex, uses a vector space model of word co-occurrence which we created from a lemmatized version of the BNC. Specifically, we created a semantic space with a context window of five words on either side of the target word and the most common 2,000 context words as vector dimensions. Their values were set to the ratio of the probability of the context word given the target word to the probability of the context word overall. Previous work shows that this configuration is optimal for measuring word similarity (Mitchell and Lapata 2010; Bullinaria and Levy 2007). In our specific setting, lex is then simply the cosine of the angle between the vectors representing any two words.[9]

For the syntactic measure syn we chose the simplest definition possible: $\text{syn}(r, r')$ is 1 if r and r' denote the same grammatical relation ($r = r'$), and 0 otherwise. We also considered more sophisticated definitions based on different degrees of similarity between grammatical relations, but were not able to find parameters performing consistently better than this simple approach.

A crucial parameter in the formulation of our similarity score (see Equation (1)) is the relative weight α of syntactic compared to lexical similarity. Intuitively, both types of information should be taken into account, as favoring one over the other may yield sentences with either similar structure or similar words, but entirely different meaning. This suggests that α should be neither very small nor very large and will ultimately also depend on the specific measures used for lex and syn.

We optimized α on a development set using F_1 score as the objective function. Specifically, we used a random sample of 20% of the FrameNet instances as seed corpus and expanded it with instances from the BNC using different values for α. For each seed sentence, the most similar neighbor was selected (i.e., $k = 1$). We evaluated performance of the role labeler enhanced with automatic annotations on a test set consisting of another random 10% of the FrameNet instances. (These development and test sets were not used in any of the subsequent experiments.) The parameter α ranges between 0 (using only lexical information) and ∞ (using only syntactic information). We therefore performed a grid search on a logarithmic scale, varying $\log \alpha$ between -3 and 3 with steps of size 0.2. We also computed performance in the extreme cases of $\log \alpha = \pm\infty$.

Figure 4 shows the results of the tuning procedure. With the exception of $\alpha = -\infty$ (i.e., ignoring syntactic information) all expansions of the seed corpus lead to better role labelers in terms of F_1. Furthermore, extreme values of α are clearly not as good as values that take both types of information into account. The optimal value according to this tuning experiment is $\log \alpha = -0.6$. Finer tuning of the parameter will most

9 Experiments with off-the-shelf WordNet-based similarity measures did not yield performance superior to the cosine measure (see Fürstenau [2011] for details).

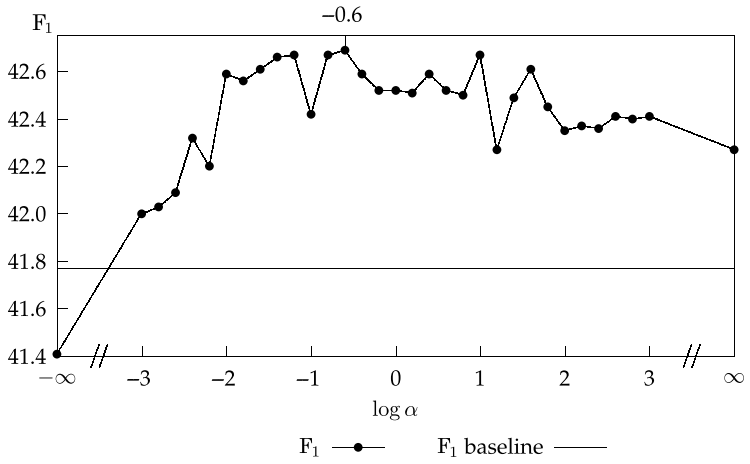


Figure 4
Performance of our method on the development set for different values of the α parameter. The baseline is the performance of a semantic role labeler trained on the seed set.

likely not yield improvements, as the differences in F_1 are already relatively small. We therefore set $\alpha = e^{-0.6} \approx 0.55$ for all further experiments. This means that lex is weighted approximately twice as strongly as syn.

5. Experiment 1: Known Verbs

In this section, we describe a first set of experiments with the aim of automatically creating novel annotation instances for SRL training. We assume that a small number of manually labeled instances are available and apply our method to obtain more annotations for the FEEs attested in the seed corpus. The FEE of the labeled sentence and the target verb of the unlabeled sentence are presumed identical. However, we waive this restriction in Experiment 2, where we acquire annotations for unknown FEEs, that is, predicates for which no manual annotations are available.

5.1 Method

We applied our expansion method to seed corpora of different sizes. A random sample of 60% of the FrameNet instances was used as training set and 10% as test set (the remaining 30% were used as development set for tuning the α parameter). The training set was reduced in size by randomly choosing between 1 and 10 annotated instances per FEE. These reduced sets are our seed corpora. We first trained the supervised SRL system on each of these seed corpora. Next, we used our expansion method to add the k nearest neighbors of each seed instance to the training corpus, with k ranging from 1 to 6, and retrained the SRL classifiers.

We also compared our approach to self-training by selecting k sentences from the unlabeled corpus, labeling them with the baseline classifier trained on the unexpanded corpus (instead of applying our projection method), and then adding these to the training corpus and retraining the classifier. Specifically, we employed three variants of self-training. Firstly, unlabeled sentences were selected for each seed sentence randomly, the only constraint being that both sentences feature the same FEE.

Secondly, new instances were chosen according to a sentence similarity measure shown to be highly competitive on a paraphrase recognition task (Achananuparp, Hu, and Shen 2008). We used the measure proposed in Malik, Subramaniam, and Kaushik (2007), which is a simpler variant of a sentence similarity measure originally described in Mihalcea, Corley, and Strapparava (2006). Given two sentences or more generally text segments T_i and T_j, their similarity is determined as follows:

$$sim(T_i, T_j) = \frac{\sum_{w \in T_i} maxSim(w, T_j) + \sum_{w \in T_j} maxSim(w, T_i)}{|T_i| + |T_j|} \tag{7}$$

where $maxSim(w, T_j)$ is the maximum similarity score between the word w in T_i and any word in T_j with the same part of speech (i.e., noun, verb, adjective). A large number of measures have been proposed in the literature for identifying word-to-word similarities using corpus-based information, a taxonomy such as WordNet (Fellbaum 1998) or a combination of both (see Budanitsky and Hirst [2001] for an overview). Here, we use cosine similarity and the vector space model defined in Section 4.3.

Our third variant of self-training identified new instances according to our own measure (see Section 4.3), which incorporates both lexical and syntactic similarity. The different self-training settings allow us to assess the extent to which the success of our method depends simply on the increase of the training data, the definition of the sentence similarity measure, the alignment algorithm for annotation projection, or their combination.

5.2 Results

Our results are summarized in Figure 5 (and documented exhaustively in the Appendix). Here, we only consider role labeling performance, that is, we use gold-standard

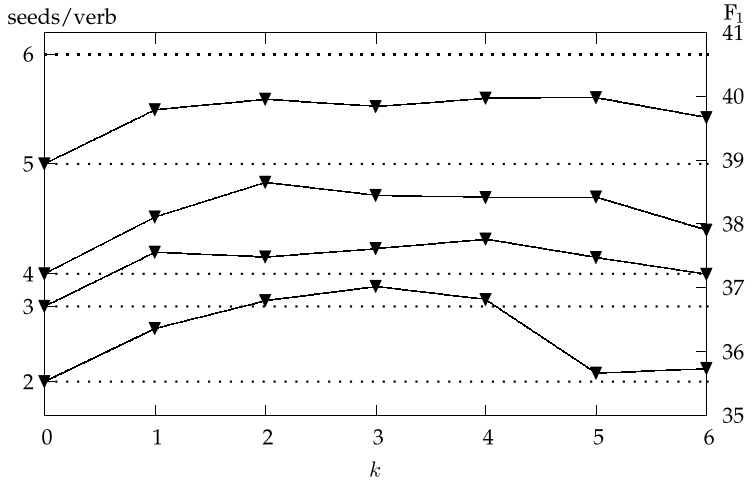


Figure 5
Role labeling F_1 obtained by expanding seed corpora of different sizes: The dotted lines show performance of unexpanded classifiers trained on two to six seed instances per verb. Each solid line starts from such a baseline at $k = 0$ and for $k > 0$ shows the performance obtained by adding the k nearest neighbors of each seed to the respective baseline corpus.

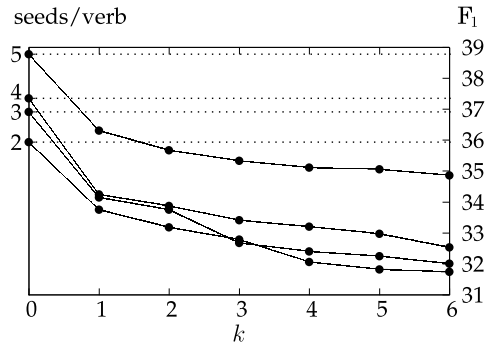
frames of the test set and evaluate the role recognition and classification stages of the classifiers. (Frame labeling accuracy will be evaluated in the following section.) The dotted lines show the performance of unexpanded classifiers trained on two to six seed instances per verb. The solid lines show the performance of our expanded classifiers when the k nearest neighbors (of each seed instance) are added to the training set. So, to give a concrete example, the unexpanded classifier trained on a corpus with two seeds per verb yields an F_1 of 35.94%. When the single nearest neighbors are added, F_1 increases to 36.63%, when the two nearest neighbors are added, F_1 increases to 37.00%, and so on.

As can be seen in Figure 5, most expansions lead to improved SRL performance. All improvements for $1 \leq k \leq 5$ are statistically significant (at $p < 0.05$ and $p < 0.001$) as determined by stratified shuffling (Noreen 1989; see the Appendix for details). The only exception is $k = 5$ for two seeds per FEE. We obtain largest improvements when k ranges between 2 and 4, with a decline in performance for higher values of k. This illustrates the trade-off between acquiring many novel annotations and inevitably introducing noise. For progressively less similar neighbors, the positive effect of the former is out-weighted by the detrimental effect of the latter.

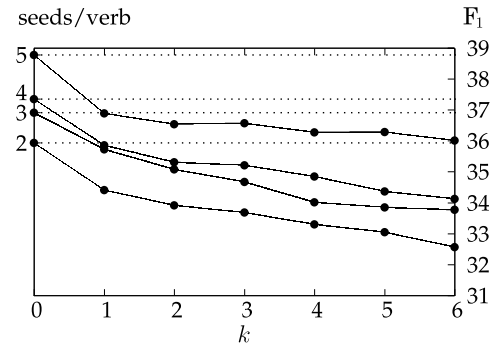
It is also interesting to observe that automatically generated instances often have a positive effect on role labeling performance similar to, or even larger than, manually labeled instances. For example, the corpus with two seeds per FEE, expanded by two, three or four nearest neighbors, leads to better performance than the corpus with three manually labeled seeds; and an expanded version of the five seeds/FEE corpus closes 60% of the gap to the six seeds/FEE corpus. Generally, the positive effect of our expansion method is largest for corpora with only a few seed instances per FEE. The results in Figure 5 may seem low, especially with respect to the state of the art (see the discussion in Section 1). Bear in mind, however, that the semantic role labeler is trained on a small fraction of the available annotated data. This fits well with its intended application to minimize annotation effort when creating resources for new languages or adapting to new domains.

Figure 6 shows the results of self-training. Dotted lines again denote the performance of unexpanded classifiers trained on seed corpora of different sizes (ranging from two to five seeds per verb). The solid lines show the performance of these classifiers expanded with k neighbors. Figures 6(a)–6(c) correspond to different methods for selecting the k-best sentences to add to the seed corpus (i.e., randomly, according to the similarity function presented in Malik, Subramaniam, and Kaushik (2007), and our own similarity measure that takes both syntactic and semantic information into account). In all cases we observe that self-training cannot improve upon the baseline classifier. Randomly selecting new sentences yields the lowest F_1 scores, followed by Malik, Subramaniam, and Kaushik and our own measure. Figure 6(d) compares the three self-training methods in the five seeds per verb setting. These results indicate that the ability to improve labeling performance is not merely due to selecting sentences similar to the seeds. In other words, the graph alignment algorithm is worth the added work as the projection of annotations contributes to achieving better SRL results.

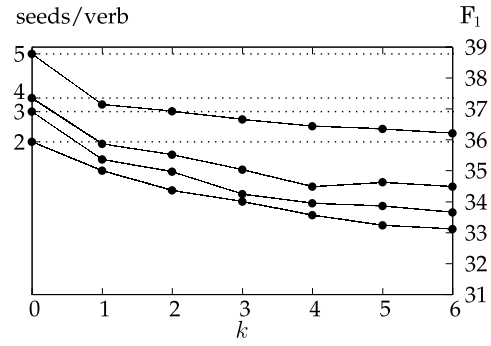
To gain a better understanding of the quality of the annotations inferred by our system, we further analyzed a small sample. Specifically, we randomly selected 100 seed instances from the FrameNet corpus, and used 59,566 instances as the unlabeled expansion corpus, treating their gold standard annotations as unseen (the remaining 1,000 instances were held out as a separate test set, as discussed subsequently). Seed and expansion corpora were thus proportionately similar to those used in our main experiments (where seed instances in the range of [2,092–16,595] were complemented



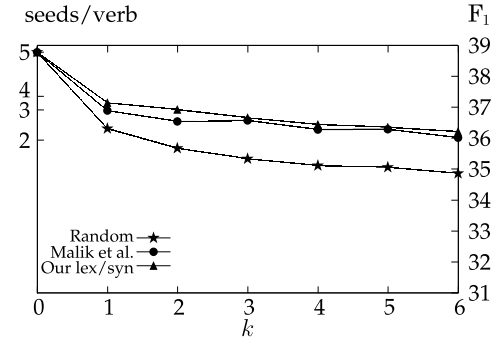
(a) Self-training with randomly chosen instances.



(b) Self-training with instances selected according to the similarity measure proposed in Malik et al. (2007).



(c) Self-training with new instances chosen according to our own similarity measure (see Equation (1)).



(d) Direct comparison between the three self-training methods for corpus with five seeds per verb.

Figure 6
Role labeling F_1 with self-training; dotted lines show the performance of unexpanded classifiers trained on two to five seed instances per verb. Each solid line starts from such a baseline at $k = 0$ and for $k > 0$ shows the performance obtained by adding k sentences with the same FEE to the respective baseline corpus.

with approximately 6 million unlabeled BNC sentences). For each of the 100 seeds, we projected annotations to their nearest neighbors according to our algorithm, and compared their quality to the held-out gold standard. Figure 7 reports labeled F_1 for the sets of d-th neighbors. Unlike the neighbors used in our previous experiments, these are mutually exclusive. In other words, the set for $d = 1$ includes only the first most similar neighbors, for $d = 2$ the second most similar neighbors, and so on. As expected, we observe decreasing quality for more distant neighbors, falling from 44.24% for $d = 1$ to 20.53% for $d = 12$.

Next, we examined how the quality of the novel annotations impacts the semi-supervised learning task when these are used as additional training data. As in our previous experiments, we trained the system on the 100 seed sentences alone to obtain an "unexpanded" baseline and on several "expanded" versions containing the seeds and one of the $d = 1, \ldots, 12$ sets. The resulting role labeling systems were evaluated on the 1,000 held-out test sentences mentioned previously. As shown in Figure 7, performance increases for intermediate values of d and then progressively decreases for larger values. The performance of the expanded classifiers corresponds closely to the quality of the projected annotations (or lack thereof). We observe substantial gains

for the sets $d = 1, \ldots, 6$ compared to the baseline role labeler. The latter achieves an F_1 of 9.06% which increases to 12.82% for $d = 1$ neighbors, to 11.61% for $d = 2$ neighbors, and so on. In general, improvements in semantic role labeling occur when the projected annotations maintain an F_1 quality in the range of [40–30%]. When F_1 drops below 30%, improvements are relatively small and finally disappear.

We also manually inspected the projected annotations in the set of first neighbors (i.e., $d = 1$). Of these, 33.3% matched the gold standard exactly, 55.5% received the right frame but showed one or more role labeling errors, and 11.1% were labeled with an incorrect frame. We further analyzed sentences with incorrect roles and found that for 22.5% of them this was caused by parser errors, whereas another 42.5% could not have received a correct annotation in the first place by any alignment, because there was no node in the dependency graph whose yield exactly corresponded to the annotated substring of the gold standard. This was again due to parser errors or to FrameNet specific idiosyncrasies (e.g., the fact that roles may span more than one constituent). For 35.0% of these sentences, the incorrect roles were genuine failures of our projection algorithm. Some of these failures are due to subtle role distinctions (e.g., *Partner1* and *Partners* for the frame FORMING_RELATIONSHIPS), whereas others require detailed linguistic knowledge which the parser does not capture either by mistake or by design. For example, seed sentences without overtly realized subjects (such as imperatives) can lead to incomplete annotations, missing on the *Agent* role.

In total, we found that parser errors contributed to 45.8% of the erroneous annotations. The remaining errors range from minor problems, which could be fixed by more careful preprocessing or more linguistically aware features in the similarity function, to subtle distinctions in the FrameNet annotation, which are not easily addressed by computational methods. As parsing errors are the main source of projection errors, one would expect improvements in semantic role labeling with more accurate parsers. We leave a detailed comparison of dependency parsers and their influence on our method to future work, however. Moreover, our results demonstrate that some mileage can be

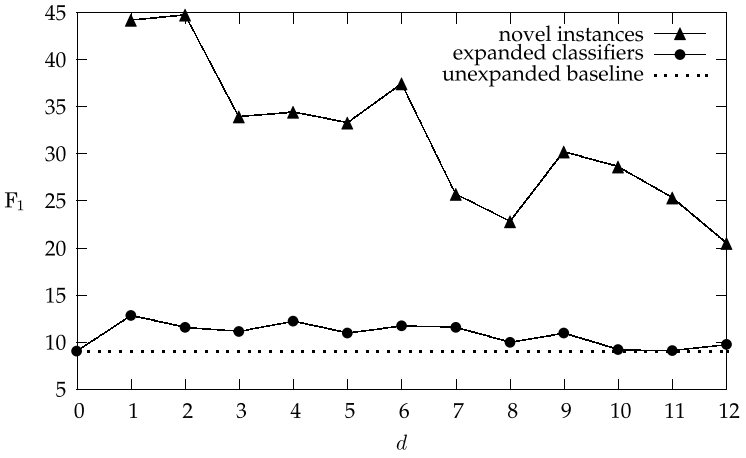


Figure 7
Evaluation of annotations projected onto the d-th neighbors of 100 randomly chosen seed sentences. The quality of the novel annotations is evaluated *directly* against held-out gold-standard data, and *indirectly* when these are used as training data for an SRL system. In both cases, we measure labeled F_1.

gained from annotation projection in spite of parser noise. In fact, comparison with self-training indicates that annotation projection is a major contributor to performance improvements.

6. Experiment 2: Unknown Verbs

In this section, we describe a second set of experiments, where our method is applied to acquire novel instances for unknown FEEs, that is, predicates for which no manually labeled instances are available. Unknown predicates present a major obstacle to existing supervised SRL systems. Labeling performance on such predicates is typically poor due to the lack of specific training material for learning (Baker, Ellsworth, and Erk 2007).

6.1 Method

To simulate frame and role labeling for unknown FEEs, we divided the set of verbal FEEs in FrameNet into two sets, namely, "known" and "unknown." All annotations of verbs marked as "unknown" made up the test set, and the annotations for the "known" verbs were the seed corpus (in both cases excluding the 30% of FrameNet used as development set). To get a balanced division, we sorted all verbal predicates by their number of annotated sentences and marked every fifth verb in the resulting list (i.e., 20% of the verbs) as "unknown," the rest as "known." We used our expansion algorithm to automatically produce labeled instances for unknown verbs, selecting the most similar neighbor of each seed sentence ($k = 1$). Then we trained the SRL system on both the seeds and the new annotations and tested it on the held-out instances of the "unknown" verbs.

6.2 Frame Candidates

So far we have made the simplifying assumption (see Experiment 1, Section 5) that the FEE of the labeled sentence and the target verb of the unlabeled sentence are identical. This assumption is not strictly necessary in our framework; however, it reduces computational effort and ensures precision that is higher than would be expected when comparing arbitrary pairs of verbs. When acquiring novel instances for unseen FEEs, it is no longer possible to consider identical verbs. The vast majority of seeds, however, will be inappropriate for a given unlabeled sentence, because their predicates relate to different situations. So, in order to maintain high precision, and to make expansions computationally feasible, we must first identify the seeds that might be relevant for a sentence featuring an unknown predicate. In the following, we propose two methods for determining frame candidates for an unknown verb, one using vector-based similarity and one that takes WordNet information into account. As we shall see, WordNet-based similarity yields significantly better results, but its application is restricted to languages or domains with similar resources.

6.2.1 Vector-based Method. To associate unknown FEEs with known frames, Pennacchiotti et al. (2008) make use of a simple co-occurrence-based semantic space similar to the one we used to define the lexical measure lex. They represent each FEE v by a vector $\vec{v}$ and

then compute a vector representation $\vec{f}$ for a frame f as the weighted centroid of the vectors of all words evoking it:

$$\vec{f} = \sum_{v \in f} w_{vf} \vec{v} \tag{8}$$

The weight w_{vf} is operationalized as the relative frequency of v among the FEEs evoking f, counted over the corpus used in building the vector space. The (cosine) similarity between the unknown target $\vec{v}_0$ and each frame vector $\vec{f}$ produces an ordering of frames, the n-best of which are considered frame candidates.

$$\text{sim}_V(v_0, f) = \cos\left(\vec{v}_0, \vec{f}\right) \tag{9}$$

6.2.2 WordNet-based Method. In addition to the vector-based approach, Pennacchiotti et al. (2008) propose a method that is based on WordNet (Fellbaum 1998) and treats nouns, verbs, and adjectives differently. Given a frame and an unknown verb v_0, they count the number of FEEs that are co-hyponyms of v_0 in WordNet. If the number of co-hyponyms exceeds a threshold τ,[10] then the frame is considered a candidate for v_0.

In our experiments, we found this method to perform poorly. This suggests that the improvements reported in Pennacchiotti et al. (2008) are due to their more refined treatment of nouns, which are not considered in our set-up. We thus follow the basic idea of measuring relatedness between an unknown verb v_0 and the set of lexical units of a frame, and propose a measure based on counts of synonyms, hypernyms, hyponyms, and co-hyponyms in WordNet. We define:

$$\text{sim}_W(v_0, f) = \sum_{v \in F} r(v_0, v) \tag{10}$$

where $r(v, v')$ is 1 if v and v' are synonyms, $\frac{1}{2}$ if one is a hypernym of the other, $\frac{1}{4}$ if they are co-hyponyms, and 0 otherwise. These numbers were chosen heuristically to represent different degrees of relatedness in WordNet. Relations more distant than co-hyponymy did not improve performance, as the verb hierarchy in WordNet is shallow. It therefore seems unlikely that much could be gained by refining the measure r, for example, by incorporating traditional WordNet similarity measures (e.g., Budanitsky and Hirst 2001).

6.2.3 Method Comparison. To evaluate which of the methods just described performs best, we used a leave-one-out procedure over the FrameNet predicates marked as "known" in our experimental set-up. Specifically, we set aside one predicate at a time and use all remaining predicates to predict its frame candidates. The resulting candidates are then compared to the true frames evoked by the predicate. (We do not consider "unknown" predicates here as these are reserved for evaluating the expansion method as a whole.) For the vector-based method we also explore an *unweighted* variant, setting all $w_{vf} = 1$.

Evaluation results are summarized in Figure 8, which shows the proportion of predicates for which at least one frame candidate is among the true evokable frames

10 Set to $\tau = 2$ according to personal communication.

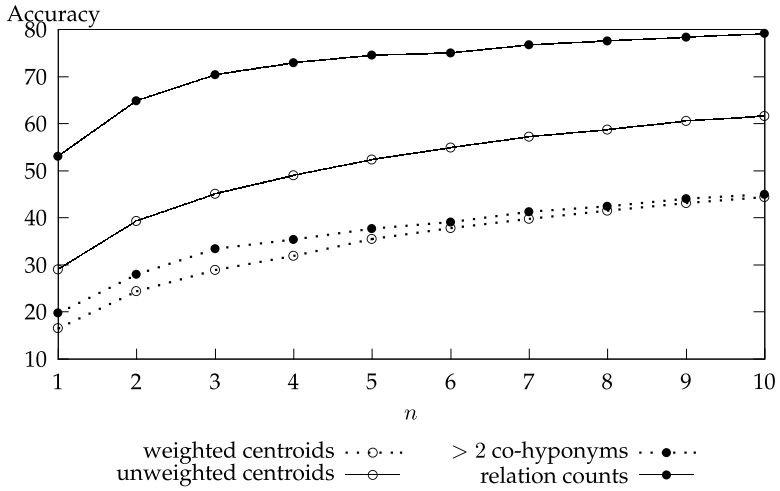


Figure 8
Frame labeling accuracy out of n frame candidates; open circles indicate vector-based similarity; black circles indicate WordNet-based similarity.

(when considering up to 10 best candidates).[11] As can be seen, performance increases by a large margin when unweighted centroids are considered instead of weighted ones. Apparently, the stabilizing effect of the centroid computation, which allows common meaning aspects of the predicates to reinforce each other and reduces the effect of spurious word senses, is more pronounced when all predicates are weighted equally. Figure 8 also shows that a WordNet-based approach that takes into account various kinds of semantic relations is superior to vector-based methods and to Pennachiotti et al.'s (2008) original proposal based only on co-hyponyms. All subsequent experiments will identify frame candidates using our WordNet-based definition (Equation (10)).

6.3 Results

Evaluation results of our approach on unknown verbs are summarized in Figure 9. Frame labeling accuracy is shown in Figure 9(a) and role labeling performance in Figure 9(b).

As far as frame labeling accuracy is concerned, we compare a semantic role labeler trained on additional annotations produced by our method against a baseline classifier trained on known verbs only. Both expanded and unexpanded classifiers choose frames from the same sets of candidates, which is also the set of frames that the expansion algorithm is considering. We could have let the unexpanded classifier select among the entire set of FrameNet frames (more than 500 in total). This would perform poorly, however, and our evaluation would conflate the effect of additional training material with the effect of restricting the set of possible frame predictions to likely candidates.

11 Note that although our evaluation is similar to Pennacchiotti et al. (2008) the numbers are not strictly comparable due to differences in the test sets, as well as the fact that they consider FEEs across parts of speech (not only verbs) and omit infrequent predicates.

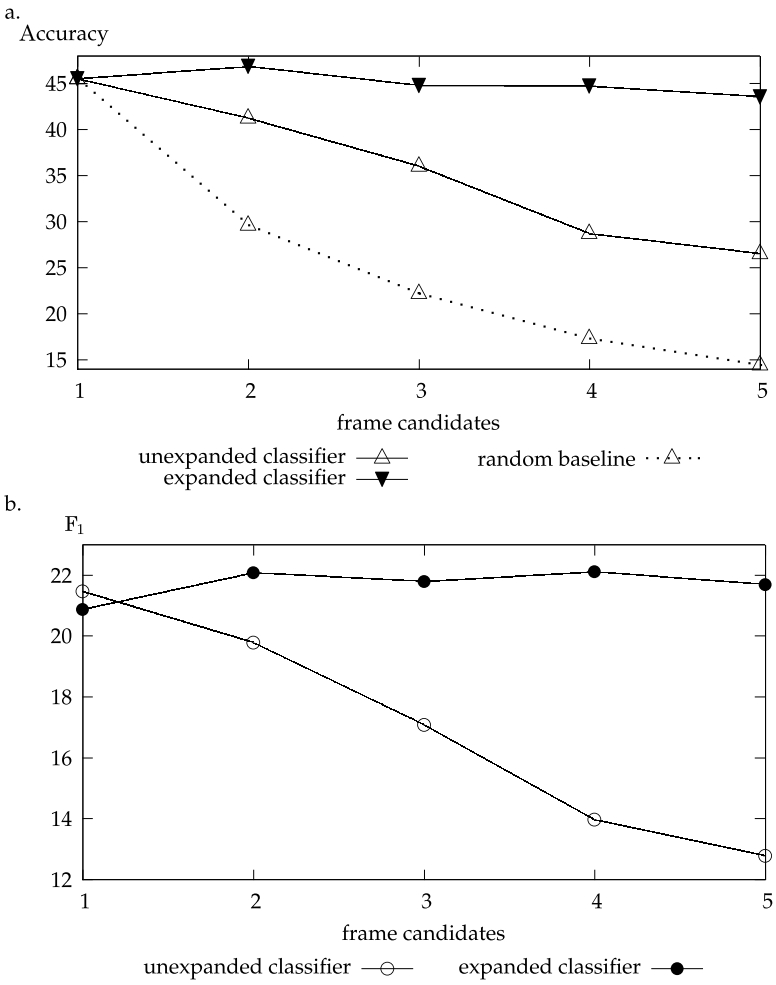


Figure 9
Frame labeling accuracy (a) and role labeling performance (b); comparison between unexpanded and expanded classifiers and random baseline; frame candidates selected based on WordNet.

We also show the accuracy of a simple baseline labeler, which chooses one of the k candidate frames at random.

As illustrated in Figure 9(a), both expanded and unexpanded classifiers outperform the random baseline by a wide margin. This indicates that the SRL system is indeed able to generalize to unknown predicates, even without specific training data. The expanded classifier is in turn consistently better than the unexpanded one for all numbers of frame candidates (x axis). The case where only one frame candidate ($k = 1$) is considered deserves a special mention. Here, a predicate is assigned the frame most similar to it, irrespectively of its sentential context. In other words, all instances of the predicate are assigned the same frame, without any attempt at disambiguation. In this case, both expanded and unexpanded classifiers obtain the same performance. Although the unexpanded classifier does not improve over and above this type-based frame labeling approach, however, the expanded classifier yields significantly better results for two candidates ($p < 0.01$ with McNemar's test). This means that the additional

training material enables the classifier to successfully favor lower scoring candidates over higher-scoring ones based on sentential context.

Figure 9(b) shows our results for the role labeling task. We again compare expanded and unexpanded classifiers. Note that there is no obvious random baseline for the complex task of predicting role spans and their labels, however. Again, we observe that the expanded classifier outperforms the unexpanded one, save the artificial case of one candidate where it yields slightly lower results. In this configuration, our expansion framework cannot account for FEEs that are polysemous by selecting among different frames, and as a result role labeling performance is compromised. For two candidates the expanded classifier yields significantly better results than this token-based approach ($p < 0.05$ with stratified shuffling). For three, four, and five candidates, performance is also numerically better, but the results do not reach statistical significance. This shows that the expanded classifier is not only able to correctly select lower scoring frame candidates for unknown verbs, but also to accurately label their roles. The overall scale of our F_1 scores might seem low. This is due to both the difficulty of the task of predicting fine-grained sense distinctions for verbs without specific training data, and the comprehensive evaluation measure, which takes into account all three stages of the SRL system: frame labeling, role recognition, and role classification.

Incidentally, we should point out that similar tendencies are observed when using vector-based similarity for identifying the frame candidates. Although overall classifier performance is worse, results are qualitatively similar: The expanded classifiers outperform the unexpanded ones, and obtain best frame accuracy and labeled F_1 with two candidates. Performance also significantly improves compared to selecting a frame randomly or defaulting to the first candidate (we summarize these results in the Appendix).

7. Conclusions

We have presented a novel semi-supervised approach for reducing the annotation effort involved in creating resources for semantic role labeling. Our method automatically produces training instances from an unlabeled corpus. The key idea is to project annotations from labeled sentences onto similar unlabeled ones. We formalize the projection task as a graph alignment problem. Specifically, we optimize alignments between dependency graphs under an objective function that takes both lexical and structural similarity into account. The optimization problem is solved exactly by an integer linear program.

Experimental results show that the additional training instances produced by our method significantly improve role labeling performance of a supervised SRL system on predicates for which only a few or no manually labeled training instances are available. In the latter case, we first determine suitable frame candidates, improving over similar methods proposed in the literature. Comparison with a self-training approach shows that the improvements attained with our method are not merely a side effect of additional training data. Rather, by identifying sentences that are structurally and lexically similar to the labeled seeds we are able to acquire qualitatively novel annotations. Our experiments make use of relatively simple similarity measures, which could be improved in future work. Incorporating a notion of selectional preferences would allow for finer-grained distinctions in computing argument similarities. Analogously, our definition of syntactic similarity could be refined by considering grammar formalisms

with richer syntactic categories such as Combinatory Categorial Grammar (Steedman 2000).

Possible extensions to the work presented in this article are many and varied. For example, we could combine our approach with cross-lingual annotation projection (Johansson and Nugues 2006; Padó and Lapata 2009). For languages without any role semantic resources, initial annotations could be obtained by cross-lingual projection and then extended with our semi-supervised method. Another application of our framework would be in domain adaptation, where a supervised model is trained on a seed corpus, and then unlabeled data from a target domain is used to select new instances and thus train a new semantic role labeler for the given domain. As our algorithm produces novel annotated sentences, it could also be used to reduce annotation effort by offering automatically labeled sentences to humans to inspect and correct. The experiments presented here are limited to verbal categories and focus solely on English. In the future, we would like to examine whether our approach generalizes to other syntactic categories such as nouns, adjectives, and prepositions. An obvious extension also involves experiments with other languages. Experiments on the SALSA corpus (Burchardt et al. 2006) show that similar improvements can be obtained for German (Fürstenau 2011).

Finally, the general formulation of our expansion framework allows its application to other tasks. Deschacht and Moens (2009) adapt our approach to augment subsets of the PropBank corpus and observe improvements over a supervised system for a small seed corpus. They also show that defining the lexical similarity measure in terms of Jensen–Shannon divergence instead of cosine similarity can additionally improve performance. Another possibility would be to employ our framework for the acquisition of paraphrases, for example, by extending the multiple-sequence alignment approach of Barzilay and Lee (2003) with our notion of graph alignments. Finally, it would be interesting to investigate how to reduce the dependency on full syntactic analyses, for example, by employing shallow parsers or chunkers.

Appendix: Detailed Experimental Results

In this appendix, we give complete results for the expansion experiments discussed in Sections 5 and 6. Asterisks in the tables indicate levels of significance. For simplicity, we only present two levels of significance, $p < 0.05$ with a single asterisk (*) and $p < 0.001$ with two asterisks (**). Significance tests for exact match and frame labeling accuracy were performed using McNemar's test. We used stratified shuffling Noreen (1989) to examine whether differences in labeled F_1 were significant.[12]

Experiments on Known Predicates. The following table shows the performance of expanded classifiers when [1–6] automatically generated nearest neighbors (NN) are added to seed corpora containing [1–10] manually labeled sentences per verb. We report precision (Prec), recall (Rec), their harmonic mean (F_1), and exact match (ExMatch; the proportion of sentences that receive entirely correct frame and role annotations). Some of these results were visualized in Figure 5.

12 We used the `sigf` tool (Padó 2006).

Training set	Size	Prec (%)	Rec (%)	F_1 (%)	ExMatch (%)
1 seed/verb	2,092	40.74	23.69	29.96	6.38
+ 1-NN	3,297	40.52	24.23	30.33	6.81 *
+ 2-NN	4,481	40.29	24.99	30.85 *	6.97 *
+ 3-NN	5,649	39.52	25.02	30.64 *	7.35 **
+ 4-NN	6,803	39.52	25.39	30.92 *	7.30 **
+ 5-NN	7,947	39.04	25.34	30.73 *	7.12 *
+ 6-NN	9,076	38.40	25.16	30.40	6.89
2 seeds/verb	4,105	45.22	29.81	35.94	9.40
+ 1-NN	6,500	45.09	30.84	36.63 *	10.19 **
+ 2-NN	8,850	44.82	31.50	37.00 **	10.32 **
+ 3-NN	11,157	44.65	31.85	37.18 **	10.32 **
+ 4-NN	13,423	43.99	31.94	37.01 **	10.15 *
+ 5-NN	15,652	42.64	31.23	36.05	9.73
+ 6-NN	17,846	42.57	31.36	36.11	9.63
3 seeds/verb	6,021	45.03	31.29	36.92	9.81
+ 1-NN	9,492	44.78	32.45	37.63 *	10.35 *
+ 2-NN	12,874	44.15	32.69	37.57 *	10.37 *
+ 3-NN	16,179	43.90	33.00	37.68 *	10.68 *
+ 4-NN	19,424	43.60	33.36	37.80 *	10.35
+ 5-NN	22,609	43.15	33.26	37.56 *	10.50 *
+ 6-NN	25,734	42.72	33.17	37.34	10.45 *
4 seeds/verb	7,823	44.42	32.21	37.35	9.48
+ 1-NN	12,321	44.45	33.31	38.09 *	10.20 **
+ 2-NN	16,688	44.26	34.13	38.54 **	10.40 **
+ 3-NN	20,944	43.71	34.20	38.37 **	10.72 **
+ 4-NN	25,098	43.37	34.35	38.34 **	10.57 **
+ 5-NN	29,166	43.25	34.45	38.35 *	10.67 **
+ 6-NN	33,142	42.48	34.24	37.92	10.40 *
5 seeds/verb	9,515	45.45	33.81	38.78	10.35
+ 1-NN	15,026	45.47	34.90	39.49 *	10.95 *
+ 2-NN	20,363	45.03	35.39	39.63 *	11.42 **
+ 3-NN	25,533	44.56	35.51	39.53 *	11.56 **
+ 4-NN	30,576	44.44	35.78	39.64 *	11.70 **
+ 5-NN	35,494	44.22	35.94	39.65 *	11.72 **
+ 6-NN	40,286	43.74	35.83	39.39 *	11.49 **
6 seeds/verb	11,105	46.50	35.44	40.22	10.95
+ 1-NN	17,553	46.05	36.11	40.48	11.56 *
+ 2-NN	23,779	45.71	36.67	40.70	12.07 **
+ 3-NN	29,787	45.16	36.83	40.57	11.92 **
+ 4-NN	35,623	44.82	36.92	40.49	11.80 *
+ 5-NN	41,310	44.60	36.91	40.40	12.13 **
+ 6-NN	46,851	44.07	36.86	40.14	12.02 **
8 seeds/verb	13,999	47.60	37.29	41.82	12.25
+ 1-NN	22,115	47.08	37.71	41.88	12.48
+ 2-NN	29,907	46.45	38.01	41.81	12.64
+ 3-NN	37,400	46.01	38.11	41.69	12.69
+ 4-NN	44,656	45.55	38.12	41.51	12.78
+ 5-NN	51,705	45.53	38.38	41.65	13.22 *
+ 6-NN	58,562	45.00	38.24	41.34	13.34 **
10 seeds/verb	16,595	48.97	39.02	43.43	13.73
+ 1-NN	26,180	48.24	39.55	43.47	14.01
+ 2-NN	35,336	47.11	39.32	42.86	13.80
+ 3-NN	44,113	46.69	39.45	42.77	13.85
+ 4-NN	52,602	46.18	39.31	42.47	13.63
+ 5-NN	60,827	46.22	39.76	42.75	13.68
+ 6-NN	68,791	45.69	39.58	42.42	13.95

Experiments on Unknown Predicates. In the following, we show the performance of unexpanded and expanded classifiers when selecting among [1–5] frame candidates generated by the WordNet-based method. We report frame labeling accuracy, role labeling performance, and exact match scores. Asterisks indicate that the expanded classifier is significantly better than an unexpanded classifier choosing among the same number of candidates. For frame labeling accuracy, we additionally provide the results of the random baseline and an upper bound, which always chooses the correct frame if it is among the candidates. Some of these results were shown in Figure 9.

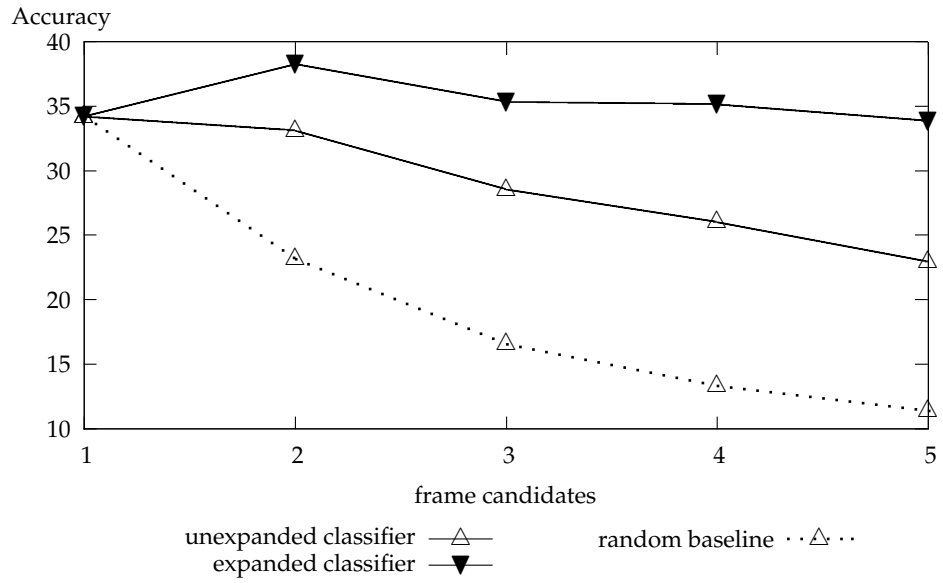
	Frame labeling accuracy (%)			
Candidates	Random	Unexpanded	Expanded	Upper bound
1	45.50	45.50	45.50	45.50
2	29.61	41.24	46.89 **	59.23
3	22.20	36.02	44.82 **	66.60
4	17.31	28.75	44.75 **	69.23
5	14.45	26.56	43.58 **	72.25

	Unexpanded (%)				Expanded (%)			
Candidates	Prec	Rec	F_1	ExMatch	Prec	Rec	F_1	ExMatch
1	24.77	18.94	21.47	6.54	23.61	18.72	20.88	6.56
2	22.52	17.63	19.78	5.87	24.60	20.05	22.09 **	7.02 **
3	19.52	15.20	17.09	5.04	24.23	19.79	21.79 **	7.24 **
4	16.18	12.31	13.98	4.02	24.59	20.09	22.11 **	7.26 **
5	14.78	11.27	12.78	3.77	24.12	19.70	21.69 **	7.44 **

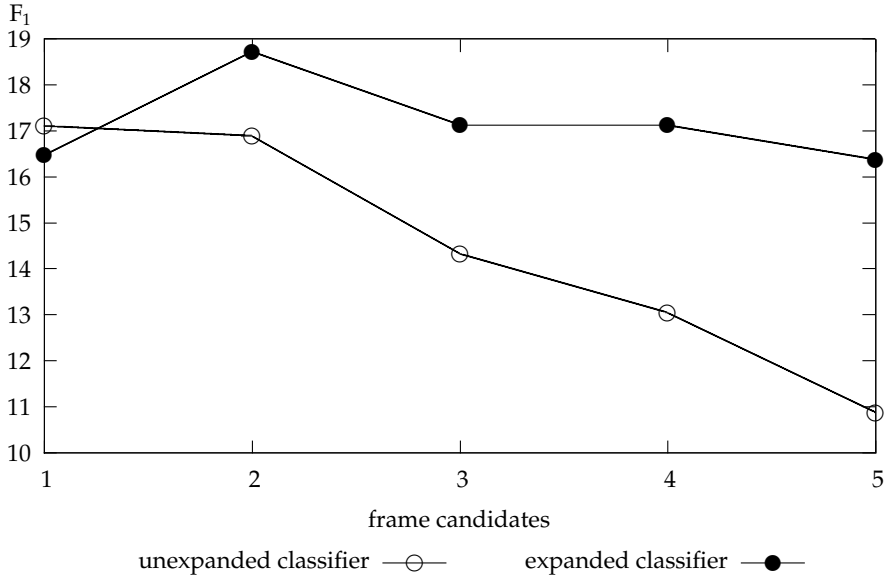
For two candidates, the expanded classifier also performs significantly better than the best unexpanded classifier (i.e., the one given only one candidate) in terms of frame labeling accuracy, F_1, and exact match ($p < 0.05$). In terms of exact match, it also performs significantly better for three candidates ($p < 0.05$), four candidates ($p < 0.05$), and five candidates ($p < 0.001$).

Vector-based frame candidates. The following graphs show the performance of the unexpanded and expanded classifiers when frame candidates are selected by the vector-based method. The expanded classifiers significantly outperform the unexpanded ones in terms of frame labeling accuracy and role labeling F_1 for [2–5] candidates ($p < 0.001$). For two candidates, frame labeling accuracy and role labeling F_1 also significantly improve compared with the type-based approach of always choosing the first candidate ($p < 0.001$). For three candidates performance is significantly better only in terms of frame labeling accuracy ($p < 0.05$) but not F_1.

a.



b.



Acknowledgments

We are grateful to the anonymous referees, whose feedback helped to substantially improve this article. Special thanks are due to Richard Johansson for his help with the re-implementation of his semantic role labeler and Manfred Pinkal for insightful comments and suggestions. We acknowledge the support of EPSRC (Lapata; grant GR/T04540/01) and DFG (Fürstenau; IRTG 715 and project PI 154/9-3).

References

Abend, Omri, Roi Reichart, and Ari Rappoport. 2009. Unsupervised argument identification for semantic role labeling. 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 28–36, Singapore.

Achananuparp, Palakorn, Xiaohua Hu, and Xiajiong Shen. 2008. The evaluation of

sentence similarity measures. In *Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery*, pages 305–316, Turin.

Andersen, Øistein E., Julien Nioche, Ted Briscoe, and John Carroll. 2008. The BNC parsed with RASP4UIMA. In *Proceedings of LREC*, pages 865–869, Marrakech.

Baker, Collin F., Michael Ellsworth, and Katrin Erk. 2007. SemEval-2007 Task 19: Frame Semantic structure extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 99–104, Prague.

Barzilay, Regina and Mirella Lapata. 2006. Aggregation via set partitioning for natural language generation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 359–366, New York, NY.

Barzilay, Regina and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 16–23, Edmonton.

Briscoe, Ted, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 77–80, Sydney.

Budanitsky, Alexander and Graeme Hirst. 2001. Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the ACL Workshop on WordNet and other Lexical Resources*, pages 29–34, Pittsburgh, PA.

Bullinaria, John A. and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39:510–526.

Burchardt, Aljoscha, Katrin Erk, and Anette Frank. 2005. A WordNet detour to FrameNet. In *Proceedings of the GLDV GermaNet II Workshop*, pages 408–421, Bonn.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: A German corpus resource for lexical semantics. In *Proceedings of LREC*, pages 969–974, Genoa.

Chang, Ming-Wei, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 429–437, Los Angeles, CA.

Clarke, James and Mirella Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:273–381.

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1992. *Introduction to Algorithms*. The MIT Press, Cambridge, MA.

Das, Dipanjan, Nathan Schneider, Desai Chen, and Noah A. Smith. 2010. Probabilistic Frame-Semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956, Los Angeles, CA.

Das, Dipanjan and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. 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 468–476, Singapore.

de Marneffe, Marie-Catherine, Trond Grenager, Bill MacCartney, Daniel Cer, Daniel Ramage, Chloé Kiddon, and Christopher D. Manning. 2007. Aligning semantic graphs for textual inference and machine reading. In *AAAI Spring Symposium at Stanford*.

de Salvo Braz, Rodrigo, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. 2005. An inference model for semantic entailment in natural language. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 1043–1049, Pittsburgh, PA.

Denis, Pascal and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, NY.

Deschacht, Koen and Marie-Francine Moens. 2009. Semi-supervised semantic role labeling using the Latent Words Language

Model. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 21–29, Singapore.
Dras, Mark. 1999. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text*. Ph.D. thesis, Macquarie University, Sydney.
Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.
Fillmore, Charles J. 1968. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart & Winston, New York, NY, pages 1–88.
Fillmore, Charles J., Christopher R. Johnson, and Miriam R. L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235–250.
Friedman, Jerome H. 1996. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University.
Fung, Pascale and Benfeng Chen. 2004. BiFrameNet: Bilingual Frame Semantics resource construction by cross-lingual induction. In *Proceedings of COLING 2004*, pages 931–937, Geneva.
Fürstenau, Hagen. 2008. Enriching frame semantic resources with dependency graphs. In *Proceedings of LREC*, pages 1478–1484, Marrakech.
Fürstenau, Hagen. 2011. *Semi-supervised Semantic Role Labeling via Graph Alignment*, volume 32 of *Saarbrücken Dissertations in Computational Linguistics and Language Technology*. German Research Center for Artificial Intelligence and Saarland University, Saarbrücken, Germany.
Gildea, Daniel and Dan Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.
Gordon, Andrew and Reid Swanson. 2007. Generalizing semantic role annotations across syntactically similar verbs. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 192–199, Prague.
Grenager, Trond and Christopher D. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Sydney.
Haghighi, Aria D., Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via graph matching. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 387–394, Vancouver.
Johansson, Richard. 2008. *Dependency-based Semantic Analysis of Natural-language Text*. Ph.D. thesis, Department of Computer Science, Lund University, Sweden.
Johansson, Richard and Pierre Nugues. 2006. A FrameNet-based semantic role labeler for Swedish. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 436–443, Sydney.
Johansson, Richard and Pierre Nugues. 2007a. Syntactic representations considered for frame-semantic analysis. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT 2007)*, Bergen.
Johansson, Richard and Pierre Nugues. 2007b. Using WordNet to extend FrameNet coverage. In *Proceedings of the NODALIDA-2007 Workshop FRAME 2007: Building Frame Semantics Resources for Scandinavian and Baltic Languages*, pages 27–30, Tartu.
Klau, Gunnar W. 2009. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 10 (Suppl 1):S59.
Land, Ailsa H. and Alison G. Doig. 1960. An automatic method for solving discrete programming problems. *Econometrica*, 28(3):497–520.
Lang, Joel 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*, pages 939–947, Los Angeles, CA.
Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.
Malik, Rahul, L. Venkata Subramaniam, and Saroj Kaushik. 2007. Automatically selecting answer templates to respond to customer emails. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1659–1664, Hyderabad.
Marciniak, Tomasz and Michael Strube. 2005. Beyond the pipeline: Discrete optimization in NLP. In *Proceedings of*

the 9th Conference on Computational Natural Language Learning (CoNLL-2005), pages 136–143, Ann Arbor, MI.

Matsubayashi, Yuichiroh, Naoaki Okazaki, and Jun'ichi Tsujii. 2009. A comparative study on generalization of semantic roles in FrameNet. 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 19–27, Singapore.

Matusov, Evgeny, Richard Zens, and Hermann Ney. 2004. Symmetric word alignments for statistical matching translation. In *Proceedings of COLING 2004*, pages 219–225, Geneva.

Melli, Gabor, Yang Wang, Yudong Liu, Mehdi M. Kashani, Zhongmin Shi, Baohua Gu, Anoop Sarkar, and Fred Popowich. 2005. Description of SQUASH, the SFU question answering summary handler for the DUC-2005 summarization task. In *Proceedings of the HLT-EMNLP Document Understanding Workshop*, Vancouver.

Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 775–780, Boston, MA.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, (34):1388–1429.

Noreen, Eric. 1989. *Computer-intensive Methods for Testing Hypotheses: An Introduction*. New York, Wiley.

Padó, Sebastian, 2006. *User's guide to `sigf`: Significance testing by approximate randomization*. Available at: `www.nlpado.de/~sebastian/software/sigf.shtml`.

Padó, Sebastian and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Pennacchiotti, Marco, Diego De Cao, Roberto Basili, Danilo Croce, and Michael Roth. 2008. Automatic induction of FrameNet lexical units. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 457–465, Honolulu, HI.

Pradhan, Sameer S., Wayne Ward, and James H. Martin. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34(2):289–310.

Punyakanok, Vasin, Dan Roth, Wen-tau Yih, and Dav Zimak. 2004. Semantic role labeling via integer linear programming inference. In *Proceedings of COLING 2004*, pages 1346–1352, Geneva.

Qiu, Long, Min-Yen Kan, and Tat-Seng Chua. 2006. Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 18–26, Sydney.

Riedel, Sebastian and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 129–137, Sydney.

Roth, Dan and Wen tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the 8th Conference on Computational Natural Language Learning*, pages 1–8, Boston, MA.

Shen, Dan 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)*, pages 12–21, Prague.

Steedman, Mark. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

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 8–15, Sapporo.

Swier, Robert S. and Suzanne Stevenson. 2004. Unsupervised semantic role labelling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 95–102, Bacelona.

Taskar, Ben, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 73–80, Vancouver.

Vanderbei, Robert J. 2001. *Linear Programming: Foundations and Extensions*. Berlin, Springer.

Wan, Stephen, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the "para-farce" out of paraphrase. In *Proceedings of the 2006 Australasian Language Technology Workshop*, pages 131–138, Sydney.

Winston, Wayne L. and Munirpallam Venkataramanan. 2003. *Introduction to Mathematical Programming: Applications and Algorithms* (4th edition). Pacific Grove, CA, Duxbury Press.

Wu, Dekai and Pascale Fung. 2009. Semantic roles for SMT: A hybrid two-pass model. In *Proceedings of HLT-NAACL, Companion Volume: Short Papers*, pages 13–16, Boulder, CO.