Grammatical Annotation Founded on Semantics: A Cognitive Grammar Approach to Grammatical Corpus Annotation

Thesis submitted for the degree of
"Doctor of Philosophy"

By
Omri Abend

Submitted to the Senate of the Hebrew University
June, 2013

This work was carried out under the supervision of:
Prof. Ari Rappoport

Acknowledgments

I was twenty two when I started my studies in the Hebrew University, a young bloke with little acquaintance with the academic world. I was very fortunate to spend most of my BSc years in the math department, a place where I learned the value of an uncompromising academic level and rigour. My studies in the cognitive science department enriched me and exposed me to the fascinating world of computational linguistics.

I met my supervisor, Ari Rappoport, when he gave a talk at the weekly BSc cognitive science seminar. This eventful meeting led to an Amirim project which I conducted under his supervision, followed by an MSc that was too short to be satisfying. Eventually, I found myself his PhD student. After almost eight years of working with Ari I am glad to say that we are still far from reaching our destination. I truly hope we can continue to work together in the future, for I feel I have much I can still learn from him. Ari taught me many valuable lessons academic ones but, as importantly, ones about the practice of building a project. He also taught me how to balance fantasies of perfection with the everyday practice of research. I am much indebted to him for his presence which both endowed me with confidence, and allowed me a sense of independence.

I am also much indebted to Roi Reichart, with whom I have had a very fruitful collaboration for the first few years of my studies, and who has always been a dear friend and tutor. Roi has a remarkable talent for grasping the essence of things in a simple and eloquent way. He is wise far beyond his years, and has a remarkable sense of sensibility. I deeply appreciate and cherish his guiding hand in the first steps of my academic studies and the many meaningful experiences we shared.

I am very grateful to the Azrieli foundation and its staff for generously supporting my research and believing in me. In particular, I would like to thank the foundation's staff Rochelle Avitan, Yula Panai and Yonat Liss for their kind assistance and support.

I have had the good fortune of meeting and collaborating with extraordinary people during my studies. I thank my committee members Amir Globerson and Malka Rappaport-Hovav for their helpful comments and good advice, and to Anna Korhonen for her kind support. I greatly enjoyed working with Roy Schwartz, who always brings wisdom and clarity of thought into the discussion. Tomer Eshet was a wonderful partner in working on the UCCA web-application, and showed remarkable diligence and independence. I am also privileged for working in collaboration with Effi Levi, Dana Rubinstein, Amir Beka, Saggy Herman and Elior Sulem. I would like to thank Oren Tsur for many insightful conversations, and to the late Dmitry Davi-

dov, whose death was a great blow to all of us. I would also like to thank my lab members Eva Kelman, Adi Littman, Elad Dinur and Shulamit Umansky-Pesin for their helping hand, good advice and kind words. Finally, I would like to thank the four UCCA annotators Ayelet Beazley, Henry Brice, Hagit Sheinberger and Meira Yan.

I have much to thank for the love of my friends and family. I would like to convey my deepest gratitude to my close friends Guy Amster, Guy Doulberg, Elad Eban, Itai Greif, Tamar Horev, Shlomit Issar, Ayal Lavi, Zur Luria, Adi and Reshef Meir, Raz Oz, Sonja Pilz, Eviatar Procaccia and Tzachy Rachamim. I would like to send my love to my parents and thank them for their endless love and support, and for encouraging me to pursue my own path. My love is also sent to my siblings Noa, Uri and Roni who are a permanent source of joy in my life. Finally, I would like to thank my beloved partner Na'ama for her love, her kindness and for sharing with me my moments of sorrow and joy.

Abstract

Natural Language Processing (NLP) is concerned with the theoretical and applicative aspects of the automatic analysis of text and speech. Advances in NLP are increasingly noticeable in everyday life, with applications such as machine translation and free-text search engines. Machine learning approaches have dominated NLP in the past two decades, with the *supervised* approach being the most widely used one, despite the significant amounts of manually annotated data it requires.

Semantic structures take a central role in NLP and are widely used in various guises in a multitude of NLP tasks. Examples of semantic structures include *semantic role labeling* structures and *first order logic* semantic formulas. However, despite the major advances in the field, determining what type of data NLP can utilize for learning and predicting elaborate semantic structures remains an open question.

State of the art algorithms for predicting semantic structures often rely on highly elaborate semantic and syntactic training data. Compiling corpora annotated with such elaborate annotation is a difficult task for humans for two reasons. First, constructing an annotation scheme requires detecting and defining the categories and structures that it annotates. Performing this manually is highly challenging, even for an expert, due to the variety and intricateness of linguistic phenomena. Second, annotating text corpora using the resulting schemes generally requires the employment of highly proficient annotators with strong background in linguistics.

This thesis aims to facilitate the annotation process required for learning semantic structure by basing it on two more accessible sources of information: (1) distributional information acquired by automatic pattern recognition methods, (2) semantic information introduced through manual corpus annotation that reflects the meaning of the text as understood by the reader. Relying on these relatively accessible sources of information reduces the need for manual labour and experts, and thereby addresses the two challenges presented above.

I discuss two complementary lines of work. The first explores an *unsupervised* approach that receives only plain text as input, treating language as an ordered collection of semantically-void symbols. I present three novel algorithms that effectively apply this minimalistic approach to tasks at the syntax-semantics interface. The tasks include the induction of part of speech tags (Abend, Reichart and Rappoport, ACL 2010), the identification of verbal arguments (Abend, Reichart and Rappoport, ACL 2009) and their classification into cores and adjuncts (Abend and Rappoport, ACL 2010). All algorithms obtained state of the art results for the unsupervised setting at

their time of publication.

Unsupervised methods are appealing in their attempt to reduce reliance on manual annotation. However, the poverty of the information they use limits their performance. Indeed, the performance of fully unsupervised models still lags considerably behind their supervised counterparts. The second part of this thesis therefore discusses manual semantic representation. It presents Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, ACL 2013; IWCS 2013) — a novel semantic scheme, inspired by leading typological (Dixon, 2005, 2010a,b, 2012) and cognitive linguistics (Langacker, 2008) theories. UCCA covers many of the most important elements and relations present in linguistic utterances using cross-linguistically motivated categories. Notably, it represents the argument structure of various types of predicates and the linkage between them. Much of the information UCCA provides has been shown to be useful for downstream applications, such as machine translation, information extraction and question answering. I also present ongoing work on compiling a UCCA-annotated corpus, currently containing more than 80K annotated tokens. The corpus compilation demonstrates that UCCA can be applied to the large-scale annotation of plain text and indicates that the UCCA scheme, unlike most commonly used syntactic schemes, can be effectively learned by annotators without any background in linguistics. Finally, I briefly discuss how UCCA can be learned using standard supervised structure prediction algorithms widely used in NLP.

To summarize, the contribution of this thesis lies in proposing novel methods for learning and representing semantic structures. Its algorithmic contribution is in presenting three novel algorithms for inducing semantic and grammatical structure from plain text, and its contribution to semantic representation is in presenting an annotation scheme that is accessible to human annotators and not tailored to a specific language or domain.

Contents

Chapter 1

Introduction

Natural Language Processing (NLP) is an interdisciplinary field concerned with the automatic analysis and understanding of language. The field focuses both on real-world applications for text understanding, and on the computational modelling of human language. The study of semantic representation is central to NLP due to its theoretical importance and its applicative value for a wide variety of tasks, ranging from question answering to automatic summarization and machine translation.

Semantic structures come in various forms and degrees of complexity[1]. The simplest forms often reflect one particular aspect of semantics, such as semantic roles (Palmer et al., 2005), coreference (Deemter and Kibble, 2000) or temporal relations (Verhagen et al., 2007). Due to the complexity of general semantic representation, more elaborate semantic schemes are usually restricted to a relatively narrow domain, such as geographical queries (Zelle and Mooney, 1996) or air travel information (Price, 1990). Recently, several attempts have been made to construct more general schemes that are applicable to a wider domain, while still reflecting a wide range of phenomena (Dorr et al., 2010; Basile et al., 2012).

However, despite recent advances, it is still an open question what semantic distinctions are required for NLP applications, and what input is required to learn them. This is the fundamental question this thesis addresses.

Most annotation schemes in NLP today construct their structures on top of a primary syntactic layer that constrains them. For instance, the PropBank scheme for semantic role labeling (Palmer et al., 2005) is built on top of the syntactic trees of the Penn Treebank annotation (Marcus et al., 1993).

[1] In this thesis we use the term *semantic structure* to refer to the semantics of phrases, sentences and texts, and not individual words. The latter is usually considered *lexical semantics*, and is largely besides the scope of this work (see below).

Syntactic schemes are usually focused on distributional regularities, and are committed first and foremost to representing the formal patterns of language, and not necessarily their meaning. For instance, virtually all syntactic annotation schemes are sensitive to the structural difference between "John showered" and "John took a shower", while few are sensitive to the semantic difference between "John took a shower" and "John took my book". Indeed, the annotation of the latter pair is identical under the two most widely used syntactic annotation schemes for English (see Chapter 7).

Representing semantic distinctions more directly can have considerable practical value. Consider machine translation, one of the core tasks of NLP. Representing the similarity between "John took a shower" and "John showered" is directly useful when translating into a target language that does not allow both sentence forms. Question answering applications can benefit from distinguishing between "John took my book" and "John took a shower", as this knowledge would help them recognize "my book" as a much more plausible answer than "a shower" to the question "what did John take?".

The coupling of semantic and syntactic structure in NLP is also apparent in their learning. The dominant approach in NLP for predicting semantic structure applies a syntactic parser as a preprocessing step. Therefore, the training of a semantic structure prediction algorithm typically requires ample amounts of data annotated with syntactic as well as semantic annotation.

Obtaining such elaborate annotations in large magnitudes is difficult for two reasons. First, defining the annotation scheme requires defining the set of syntactic and semantic structures to be annotated. This is not an easy task, even for an expert, given the range and intricateness of phenomena the schemes should cover[2]. Second, the elaborateness of the schemes and their reliance on linguistic theory requires the employment of highly proficient annotators (Marcus et al., 1993; Böhmová et al., 2003).

This thesis proposes a domain-independent approach to semantic representation which relies on accessible sources of information. It explores two main lines of work. The first are unsupervised methods that aim to directly induce semantic and syntactic regularities from large amounts of plain text. The proposed methods rely on little to no domain-specific knowledge, and receive highly accessible input, which primarily consists of large amounts of plain text. The second line proposes a novel scheme for semantic representation — Universal Conceptual Cognitive Annotation (UCCA). UCCA aims to abstract away from specific syntactic forms and to directly express

[2]For a discussion of this point in the context of subcategorization frames, see (Boguraev and Briscoe, 1989).

semantic distinctions. Building on typological theory, UCCA aims to propose a cross-linguistically valid approach. I demonstrate that UCCA can be effectively applied in several domains and that it is accessible to annotators with no background in linguistics. I also present the compilation of a UCCA-annotated corpus.

The rest of the introduction is constructed as follows. It first provides a general survey of semantic and syntactic annotation schemes used in NLP and turns to discussing the unsupervised line of work explored in this thesis. Finally, it presents the UCCA annotation scheme and corpus, and touches on methods for its automatic learning.

Background

Lexical Semantics. Lexical semantics addresses the meaning of words and multi-word expressions, and often views them in the type-level, schematized over their particular instances. Lexical resources come in many forms, and are often termed *lexicons* or *ontologies*. Semantic information for the lexical entries is often given by their (typed) relations to other entities in the lexicon, and by semantic features or meaning components ascribed to them (Vossen, 2003).

WordNet (Miller, 1995) is a large lexical resource for English which focuses on representing taxonomical relations. The EuroWordNet (Vossen, 1998) extends WordNet to other European languages. Verb lexicons, such as COMLEX (Grishman et al., 1994) and VerbNet (Schuler, 2005), often use a combination of the allowed sub-categorization frames as well semantic information. For a somewhat different approach, see FrameNet (below). Many lexical relations can be discovered to a reasonable accuracy using machine learning methods. Examples include (Pantel and Pennacchiotti, 2006; Schulte Im Walde, 2006; Davidov et al., 2007; Sun and Korhonen, 2009).

Lexical resources are used in virtually every field of NLP. Linguistic phenomena tend to demonstrate a Zipfian behavior, where a small number of phenomena account for most instances. This information can be beneficially represented in a lexicon. Indeed, methods that use extensive lexical information have been beneficially applied to spell correction (Agirre et al., 1998), semantic role labeling (Swier and Stevenson, 2004), summarization (Barzilay et al., 1997) and information extraction (Riloff and Jones, 1999). In addition, some of the leading grammatical approaches use rich lexical information, either through elaborate feature structures (Pollard and Sag, 1994) or multi-tags (Joshi and Schabes, 1997; Steedman, 2001).

By and large, the work presented in this thesis is complementary to these efforts. With the possible exception of Chapter 3, most of the work presented

here focuses on the patterns in which words combine to form composite utterances and not on the idiosyncrasies of specific words. Nevertheless, the semantic representation presented in the second part of this thesis supports the representation of increasingly finer distinctions, ultimately reaching a single word granularity. This will be addressed in future work (see Chapter 8).

In some cases it is beneficial to focus on the lexical information of the individual words and to disregard any information about their relative position. Such methods are often termed *bag of words*, and are successfully applied to tasks that do not require deep semantic analysis of the text. Examples include search engines (Croft et al., 2010) and topic models (Blei et al., 2003).

However, bag of words models miss much of the structural information present in linguistic utterances. Examples of such information include the number and identity of events the text describes (e.g., "I have this book to read" vs. "I have to read this book"), the roles of each of the participants in the events (e.g., "man bites dog" vs. "dog bites man"), and the scope of relations (e.g., "he stupidly replied" vs. "he replied stupidly"). Indeed, tasks that require a deeper understanding of the text generally use more elaborate structural information. Examples include machine translation (Yamada and Knight, 2001), question answering (Wang et al., 2007) and information extraction (Banko et al., 2007). Representing and learning linguistic structure is the focus on this work.

Syntactic Representation. Part of Speech (POS) tags are the simplest form of syntactic representation used in NLP. POS tags apply to individual words, and are considered a basic layer of annotation by most theories of syntax. Common categories include nouns, verbs, adjectives, prepositions, adverbs, determiners and conjunctions.

Hierarchical structures apply to sentences and commonly come in the form of either a constituency or a dependency structures. The most commonly used treebank for English is the Penn Treebank constituency annotation (Marcus et al., 1993), and the dependency treebank automatically derived from it (Buchholz and Marsi, 2006). Other commonly-used English treebanks include the Brown corpus (Greene and Rubin, 1971), and the Prague Dependency Treebank (Böhmová et al., 2003).

A constituency structure hierarchically binds words and previously established phrases, resulting in a tree. In the first level of the tree, words are combined into phrases, which may in turn be combined into higher level phrases. Common phrasal categories include noun phrases (NPs), verb phrases (VPs) and prepositional phrases (PPs). The leaves of the tree correspond to the

words of the sentence and are tagged with their POS tags. A dependency structure assigns each word with a head word, which is typically the word that it modifies, or the predicate it is an argument of. For instance, adjectives are dependents of the nouns they modify, and verbs are heads to their arguments. Dependency structures are often represented as trees whose nodes correspond to the words of the sentence (and their POS) and whose edges link head words to their dependents.

More elaborate formalisms aim to better account for a more fine-grained set of phenomena, often for the relations between the syntactic annotation and their corresponding semantic annotations. Examples include Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1981), Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994), Combinatory Categorial Grammar (CCG) (Steedman, 2001) and Tree Adjoining Grammar (TAG) (Joshi and Schabes, 1997).

Syntactic structure reflects first and foremost the syntactic patterns used for constructing phrases and sentences, and only indirectly reflect semantics. The formal patterns are sometimes referred to as "distributional regularities" in the sense that they reflect the distribution of environments in which a word or a phrase may appear. However, semantic and syntactic considerations are tightly coupled, as the formal patterns of language are regularly associated with a semantic interpretation, if only a very schematized and abstract one (Goldberg, 1995).

Semantic Representation. There are many types of semantic representations used in NLP. Due to their richness and variety, most semantic schemes focus on one specific aspect of semantics (e.g., semantic roles), or on a highly restricted domain (e.g., geographical queries). In this section we focus on the schemes most immediately relevant to the work presented in this thesis.

Semantic role labeling schemes aim to identify the arguments of the various predicates that appear in the text and classify them according to their semantic roles. Put differently, given a sentence, SRL structures reflect who did what to whom, when, where and why. For example:

[The surgeon]$_{Agent}$ **operated** [on his colleague]$_{Patient}$ [in the hospital]$_{Location}$

The two most widely used schemes are PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998). PropBank builds its annotation on top of the Penn Treebank annotation, and is generally more coupled with the syntax of its target language (usually English). The set of roles defined by PropBank provides a different set of semantic roles for each verb. Inter-verb consistency is maintained through the indexation of the roles, where the more

agent-like argument is labeled *Arg0* and the more patient-like argument is labeled *Arg1* (see (Dowty, 1991) for a more elaborate discussion of Proto-Agents and Patients)[3].

FrameNet offers a different approach. It uses the notion of frames, which are schematic representations of situations involving various participants and other conceptual roles. The set of semantic roles is defined relative to a frame and is therefore fairly fine-grained. For instance, the above example evokes the Frame *medical interaction scenario*, in which the surgeon receives the role *medic*, the colleague the role *patient* and the hospital the role *medical center*.

In general, the above schemes differ from UCCA in their focus specifically on argument structure phenomena, in contrast to UCCA's attempt to represent a wider range of phenomena.

Complementary lines of work explore the argument structure of other kinds of semantic relations, such as prepositions (Litkowski and Hargraves, 2005; Srikumar and Roth, 2013), commas (Srikumar et al., 2008) and discourse relations (Prasad et al., 2008).

More elaborate semantic schemes are often annotated as a secondary layers on top of a syntactic annotation. The Groningen Meaning Bank (Basile et al., 2012) builds on Combinatory Categorial Grammar (CCG). The Lingo-redwoods corpus (Oepen et al., 2004) uses Minimal Recursion Semantics representations (Copestake et al., 2005) as the semantic representation on top of an HPSG annotated corpus. While constructing a scheme that jointly represents syntax and semantics makes the relation between them explicit, it also mutually constrains the two levels of representation, and increases the annotation costs incurred by such elaborate schemes.

A different line of work aims to form abstract representation divorced from any specific language. Such a representation necessarily abstracts away not only from syntactic variation, but also from the lexicon. Examples of such works include AMR (Banarescu et al., 2013) and IAMTC (Dorr et al., 2010). UCCA differs from these lines of work in two major respects. First, UCCA's representation does not abstract away from the lexicon. Second, UCCA provides a coarse-grained annotation which can be open-endedly refined, in contrast to the fairly elaborate annotations of the above approaches.

A more elaborate comparison of UCCA to other semantic schemes can be found in Chapter 7.

[3]PropBank covers the argument structure of verbs. A similar scheme which covers nouns is NomBank (Meyers et al., 2004).

Unsupervised Learning of Syntactic and Semantic Structure

Unsupervised learning methods attempt to find an underlying structure without relying on annotated data. The algorithms often assume that some hidden structure governs the regularities observed in the unannotated data, and aim to discover this structure. In recent years increasingly sophisticated approaches have been proposed and applied to a wide range of tasks, including parsing (Seginer, 2007), verb clustering (Sun and Korhonen, 2009), induction of POS categories (Abend et al., 2010), lexical semantics (Davidov et al., 2007), and many others.

Despite the challenges that such a minimalist setting poses, unsupervised algorithms carry much potential as they can be applied to any language or genre for which adequate raw text resources are available, and entirely avoid any annotation costs. They also bear theoretical promise for their ability to recover novel, valuable information in textual data and to expose underlying relations between form and various linguistic phenomena. This thesis includes work on the unsupervised learning of three major aspects of syntactic and semantic structure. I will turn to reviewing them in detail.

POS Induction. Part of Speech tags are essential to most theories of grammar and to a great variety of NLP applications. The task of inducing parts of speech from plain text is one of the most frequently tackled in unsupervised NLP. Many different methods were proposed, and the output of these algorithms has been shown effective to subsequent tasks (Finkel and Manning, 2009; Spitkovsky et al., 2011).

In the context of unsupervised learning, a POS is defined in strictly formal terms. Two words are said to share the same category if they may appear in similar syntactic and morphological patterns. For instance, the words "happy", "strong", "brown" and "thin" can be considered to be in the same category by virtue of appearing interchangeably in these patterns:

1. A ____ dog
2. The dog is ____
3. Very ____
4. ____ / ____+er / ____+est

Some linguistic theories indeed view POS categories as purely syntactic in nature, and construe them as constraining the syntactic constructions in which the word may appear (Jackendoff, 1994, p. 68–69). Cognitive linguistics theories on the other hand (e.g., (Croft, 2001)) view constructions as pairings of form and meaning, and underscore the semantic considerations

underlying their definition. Langacker (1987, 1991) goes as far as claiming that grammatical categories, POS categories among them, can be fully characterized semantically.

Several algorithmic approaches were previously applied to this task. Many works addressed it as a type-level task, where a word type (and not each specific instance) is assigned a category. Although different instances of the same word are known to receive different categories in different contexts (e.g., "chair" can be both a noun and a verb), experiments show that on average a great majority of a word's instances belong to the same category (see Chapter 3). Examples of works that apply type-level clustering include (Schütze, 1995; Clark, 2003; Lamar et al., 2010; Christodoulopoulos et al., 2011). Other approaches model the POS assignment task as an instance-level task and use sequential graphical models, such as Hidden Markov Model (HMM) (Johnson, 2007), discriminative sequential models (Smith and Eisner, 2006; Moon et al., 2010) and Bayesian approaches (Goldwater and Griffiths, 2007; Gao and Johnson, 2008).

The feature set for this task is based on distributional and morphological regularities. Distributional features are almost unexceptionally based on the distribution of words appearing immediately before and immediately after the word in question. The morphological representation captures the set of inflections a word has (i.e., the set of forms it appears in), thereby complementing the distributional representation which focuses on the word's neighboring words. Most works use simple features based on terminal letter sequences (e.g., (Smith and Eisner, 2006; Haghighi and Klein, 2006)). More language-general approaches include (Clark, 2003) that models the entire letter sequence as an HMM and uses it for defining a prior distribution, as well as words that use external unsupervised morphological analyzers to derive their morphological features. Examples include (Dasgupta and Ng, 2007; Christodoulopoulos et al., 2011) and the work presented in this thesis. Segmentation models provide strong results on several languages without requiring language-specific tuning (Goldsmith, 2001; Creutz and Lagus, 2005).

Chapter 3 presents a novel algorithm for unsupervised POS induction. The algorithm uses morphological and distributional information, which reflects syntactic but also semantic information. For example, our distributional repsentation is adapted from the unsupervised CCL parser (Seginer, 2007), and reflects the tendency of a word to receive (or "select") a certain modifier or argument[4]. The algorithm is inspired by the cognitive theory of prototypes (Taylor, 2003). The output of the algorithm is used in later chapters as input for an unsupervised system for identifying shallow semantic

[4]Such tendencies are often termed *selectional preferences.*

structures (see below).

The presented algorithm is unique in two respects. First, it employs a novel unsupervised algorithm that discovers prototypes in a fully unsupervised manner and maps the rest of the words according to their association with the prototypes. Haghighi and Klein (2006) used a similar notion of prototypes, but defined them manually. Second, the work uses a distributional representation that has been developed for the purposes of unsupervised parsing, thereby employing the strong link between POS tags and hierarchical syntactic structure. The presented algorithm is also exceptional in using a morphological representation suitable for any affixal language, and based on the notion of *signatures* (Goldsmith, 2001).

We present extensive evaluation using six different evaluation measures and obtain the best reported results on the standard evaluation corpus, comparing against two different gold standard annotations. We also demonstrate the applicability of our algorithm to German, evaluating it against a state of the art tagger and obtaining superior results.

Argument Identification. Semantic role labeling (SRL) is one of the core tasks in NLP and is in wide use in various types of applications, including information extraction (Surdeanu et al., 2003), question answering (Narayanan and Harabagiu, 2004) and summarization (Melli et al., 2004). In addition to its applicative value, the identification of arguments and their classification is a core component in many theories of grammar (Levin and Hovav, 2005). SRL is often tackled in two subsequent stages. In the first stage, the arguments of the given predicate are delineated (argument identification) and in the second they are assigned specific semantic roles (argument classification). It has been shown that these two tasks indeed require somewhat different sets of features (Pradhan et al., 2008).

Chapter 4 presents work that addresses the first and the more challenging stage of the two (Màrquez et al., 2008). Our work is one of the first on unsupervised SRL, and the very first on unsupervised argument identification. Previous works include (Grenager and Manning, 2006), which strictly addressed argument classification (the second stage of the two) and assumed a supervised argument identification, and (Swier and Stevenson, 2004), which additionally employed a verb lexicon. Since the publication of this work in 2009, the topic of unsupervised SRL has attracted considerable attention. For example, Lang and Lapata (2010) addressed the classification of arguments to their semantic roles using a variant of a latent-variable variant of a logistic classifier. Titov and Klementiev (2012) presented a Bayesian non-parametric model for the same task.

Our algorithm works as follows. It first parses the text using a state-of-

the-art unsupervised parser, the CCL parser (Seginer, 2007). It then detects the minimal clause containing the verb, using an unsupervised clause detection algorithm developed for the purposes of this work. While not all verbal arguments are contained in the minimal clause, this has proven to be a reasonable and tractable approximation. Finally, the algorithm filters out spurious arguments by identifying verb-argument pairs that correlate negatively.

We evaluated our algorithm on two languages, English and Spanish, and compared our results to a simpler baseline based on the CCL parser. In both languages we showed a substantial improvement over the baseline.

Core-Adjunct Distinction. The distinction between core arguments and adjuncts is at the basis on most theories of grammar. The distinction separates core arguments, which are obligatory and whose semantic role is highly dependent on the identity of the verb, and adjuncts, which are optional and whose semantic role is independent of the verb. Consider our previous example:

[The surgeon]$_{Agent}$ **operated** [on his colleague]$_{Patient}$ [in the hospital]$_{Location}$

The role of "the surgeon" and "on his colleague" is very much dependent on the identity of the predicate ("operated"), as can be seen by their markedly different roles in the sentence "the surgeon counted on his colleague". "in the hospital" is an adjunct, as it can be used with a similar role with a variety of other verbs, for instance "the surgeon ate his lunch in the hospital".

The core-adjunct distinction was previously addressed by many learning approaches, both supervised (e.g., as part of the SRL task (Màrquez et al., 2008)) and semi-supervised (e.g., (Korhonen, 2002)). However, to the best of my knowledge, all works used supervised parsers in order to construct a syntactic parse tree for the sentence prior to the classification of its arguments into cores and adjuncts. The employment of such tools can greatly bias the results to agree with the manual distinctions incorporated into these parsers through their training data. Indeed, most syntactic annotation schemes reflect the core-adjunct distinction implicitly or explicitly. See Chapter 5 for a more elaborate survey of related work.

This thesis presents the first work that takes a completely unsupervised approach to this task. The presented classifier applies three corpus-based measures that correlate with the core-adjunct distinction and combines them using an ensemble method. It uses the unsupervised POS tagger and unsupervised argument identification algorithm presented in Chapter 3 and Chapter 4, in addition to an unsupervised parser (Seginer, 2007). Our algorithm obtains about 70% prediction accuracy (compared to the chance-level accuracy of 50%) and outperforms several simpler baseline algorithms.

To recap, the first section of this thesis presents novel unsupervised methods for discovering syntactic and semantic structure. Its results demonstrate that much valuable information can be obtained using as little as plain text.

However, the output of such algorithms often differs substantially from analogous manually annotated distinctions. For example, in the task of POS induction, after two decades of research, accuracy is no more than 80% when compared to a manually defined gold standard, even when using a rather a relaxed evaluation measure. In comparison, supervised algorithms obtain more than 95% accuracy on this task (Toutanova et al., 2003).

Part of this gap can be explained by noting that there is more than one way to formulate a grammatical distinction. An unsupervised algorithm that is not exposed to annotated data, may therefore find an equally plausible alternative that substantially differs from the gold standard. Cases of multiple plausible annotations are in fact very frequent (Schwartz et al., 2011).

A manual inspection of the results reveals that this explanation only partially explains the performance gap. Even to the naked eye, the quality of the output of unsupervised algorithms is far from pleasing. This is particularly true in semantic tasks such as argument identification. The reason for that may be found in the poverty of the input unsupervised algorithms rely on, which does not allow them to effectively address the semantic and communicative qualities of language. In the second part of this thesis we develop a semantic annotation scheme that aims to provide the complementary information required to learn semantic structure.

Manual Semantic Annotation

The second part of this thesis (chapters 6 and 7) presents a novel framework for semantic representation called *Universal Conceptual Cognitive Annotation* (UCCA). UCCA aims to provide a domain-general semantic annotation to be used by NLP applications. The main principles of UCCA are as follows:

1. Abstracting away from syntactic variation and directly representing semantics. For instance, in the sentences "John made an appearance" and "John appeared", UCCA would emphasize their semantic similarity, disregarding their syntactic differences.
2. Incorporating distinctions that are difficult to induce automatically. Specifically, argument identification which has been shown to be a particularly difficult task is manually annotated.
3. Domain-generality. UCCA is constructed to be applicable to a wide range of domains and to be able to express a large scope of distinctions.

The scheme is constructed as a multi-layered structure and aims to accommodate a wide spectrum of semantic distinctions required for NLP applications. The foundational layer of UCCA is highly coarse-grained and therefore has the ability to capture coarse-grained similarities between markedly different domains. The layered structure allows the extension of the scheme in order to address the ever expanding needs of the NLP community.

UCCA has several advantages from an NLP perspective. First, coarse-grained semantic distinctions tend to be less domain specific, as languages tend to differ more substantially in terms of their syntactic inventory than in terms of their coarse-grained semantics. Thereby, UCCA addresses one of the core challenges of NLP, i.e., constructing systems that are robust across different domains and languages (e.g., (Blitzer et al., 2006; Reichart and Rappoport, 2007)).

A second advantage is in the accessibility of semantic annotation schemes to annotators with no background in linguistics. Indeed, while syntactic annotation schemes usually require the employment of linguistic experts (Marcus et al., 1993; Böhmová et al., 2003), this thesis shows that in the annotation of UCCA, there is no persistent advantage to annotators with no background in linguistics.

While the learning of UCCA is not the focus of this thesis, there are strong preliminary indications that the UCCA structures can be effectively learned using variants of existing NLP and machine learning methods. The structured prediction of syntactic structures, most often in the form of dependency or constituency trees, is one of the central issues the field addresses (McDonald et al., 2005; Cohen et al., 2013, *inter alia)*. The task of supervised prediction of semantic structure in its various forms is also well studied (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007, *inter alia)*.

Recently, two major developments have been made in the field of semantic parsing which are directly relevant to UCCA. Naradowsky et al. (2012) proposed an algorithm for learning semantic role labeling without assuming manually annotated syntactic data. Instead, they assumed a hidden dependency structure and marginalized over it (see also (Chang et al., 2010)). Their SRL results are comparable to the state of the art results obtained using a supervised syntactic parser, and in some cases even exceed them. This supports the claim advocated by this thesis that in order to best support NLP applications, existing manually annotated schemes are not necessarily optimal (cf. (Schwartz et al., 2012)). A second relevant development is the machine learning methods for learning DAG-based semantic representations (such as UCCA's) (Sagae and Tsujii, 2008; Jones et al., 2012). This shows that it is possible to extend existing parsing technology, which is mostly focused on trees, to DAGs.

Chapter 6 introduces the UCCA framework, its rationale and long-term goals, and provides a detailed account of UCCA's foundational layer. Chapter 7 motivates the use of UCCA from an applicative standpoint and presents a corpus annotated with UCCA's foundational layer. It also discusses the annotation process and demonstrates UCCA's accessibility to annotators with no background in linguistics. The previous work surveyed by the two chapters is somewhat complementary. Chapter 6 focuses on comparable syntactic schemes (notably, dependency schemes), while Chapter 7 focuses on comparable semantic schemes. As these chapters were separately published, they inevitably contain a good deal of overlap, especially in their technical details.

The intellectual roots of UCCA lie in two strands of theoretical linguistics. In its general approach, UCCA builds on Basic Linguistic Theory (BLT) (Dixon, 2005, 2010a,b, 2012), a typological approach to grammar that has been used for the description of a wide variety of languages. Similarly to BLT, UCCA emphasizes semantic criteria for defining grammatical constructions, and is committed to cross-linguistically valid notions.

UCCA is also influenced by the cognitive linguistics tradition (Croft and Cruse, 2004) that relates linguistic phenomena to general non-linguistic cognitive processes and abilities. This influence can be seen in two major respects. First, the notions forming UCCA's annotation scheme are derived from extra-linguistic abilities, such as visual perception. In this we follow previous work on cognitive linguistics (Langacker, 1987, 1991; Talmy, 2000a,b, *inter alia)*, that emphasizes the relation between the grammatical structure of an utterance and its conceptualization. Second, UCCA's motivation to learn syntax automatically given semantic and textual input is in line with much cognitive linguistics work that challenges linguistic nativism, an approach that holds that language is too complex to be learned from experience (Clark and Lappin, 2010).

Summary of Research Goals

The basic goal of this thesis is to characterize the type of input that is required to represent semantics in NLP. The thesis advances the claim that the basic semantic representation required for NLP applications can be founded on two main elements: manually encoded semantic structure that abstracts away from the specific characteristics of individual languages and unsupervised and semi-supervised machine learning methods that statistically learn the mapping between these structures and the text, and generalize it to unseen texts. This proposal stands in contrast to the common approach in NLP today which explicitly represents the syntax of indivual languages and applies supervised methods to learn them.

To support its claim, the thesis explores two complementary approaches.

In its first part, the thesis presents three novel unsupervised algorithms for core syntactic and semantic NLP tasks. Unsupervised algorithms rely on plain text, and can therefore be applied to any domain where large corpora of plain text are available. Such algorithms are also appealing in that they do not require any manual annotation, thereby addressing the almost prohibitive costs required to compile the necessary resources for supervised semantic structure prediction algorithms.

In its second part, the thesis proposes a novel semantic annotation scheme — UCCA. The goal of this scheme is to provide a semantic representation that can be effectively used in a large range of domains and languages, and that can be easily learned by annotators with no background in linguistics.

Chapter 2

Methodology

The great majority of methodological details are given in the individual chapters. I discuss here several additional issues.

Unsupervised Learning Experiments. The three unsupervised algorithms presented in this thesis were evaluated on the standard benchmark corpora for their respective tasks: the Penn Treebank (Marcus et al., 1993) in the case of PoS induction and PropBank (Palmer et al., 2005) in the case of semantic role labeling (Chapters 4 and 5).

The evaluation of induced POS categorizations is notoriously difficult for several reasons. First, there are several POS schemes in common use in English NLP, a difficulty we address by evaluating our results against two common schemes. Second, the evaluation is often dependent on the number of induced clusters, which in most induction algorithms has to be prespecified. We therefore used several several different evaluation methods, each highlighting different qualities of the examined clustering. Third, punctuation marks account for a large portion of the tokens, but have trivial POS tags (usually a special POS tag is given to them). Including punctuation marks in the evaluation therefore artificially boosts the results. To address this issue, we report results both excluding and including punctuation. Finally, there are several mathematical measures for comparing two different clusterings. We use four leading evaluation measures and show superior results over previous works in all of them. A more complete survey of existing approaches to clustering evaluation can be found in Chapter 3.

The evaluation of the SRL algorithms poses less difficulties. For the argument identification (Chapter 4), we use a strict measure and compute the number of predicted arguments matching in their boundaries with the PropBank gold standard. Dividing the number of matches by the total number of predicted arguments yields a *precision* score (P), and dividing it by the total

number of gold standard arguments yields a *recall* score (R). Their harmonic mean (*F–score*) is reported as well. For the evaluation of the core-adjunct classifier (Chapter 5), we use the simple *accuracy* measure the reflects the number of arguments for which the correct label was predicted, again comparing against the PropBank gold standard.

To the best of our knowledge, there are no previous works that tackled the argument identification and core-adjunct classification tasks. In order to assess the quality of our results we compared them against simple baseline algorithms and to partial versions of our algorithms, which include only part of the algorithmic components. By this we were able to ensure that each of the components is required to reach a maximal performance.

One of the major advantages of unsupervised algorithms is their applicability to a wide variety of domains and languages. To test the cross-linguistic applicability of our algorithms, we tested them on German (Chapter 3) and Spanish (Chapter 4). We evaluate against standard resources, the NEGRA corpus for German POS tags (Brants, 1997) and the Semeval semantic annotation for Spanish semantic role labeling (Màrquez et al., 2007).

Manual Semantic Annotation (UCCA). The UCCA annotation scheme is constructed to be highly coarse-grained, while still retaining coverage and interpretability. The development process of the scheme consisted of several iterations, where in each an annotation scheme was proposed and applied to several texts. Cases that did not fall under the set of categories were assembled and modifications were consequently made to the scheme.

We conducted two pilot sessions before embarking in the annotation process. The first included 7 annotators, who volunteered to take part. Prior to the annotation, they were given a short frontal tutorial that lasted roughly two hours. Each annotator was given five short passages (1000 tokens in total) to annotate. We collected feedback from the annotators for the various components of the tutorial, the annotation scheme and the web-application. A second more restricted pilot was conducted two months afterwards. Four annotators (a sub-set of the annotators in the first pilot) participated in this round, and were given 3–6 short passages. The requested feedback mostly focused on the modifications made since the previous pilot.

The annotation of the corpus was conducted by four (different) annotators with varying levels of background in linguistics in order to measure the effect of previous acquaintance on the quality of the annotation. The training process began with a frontal tutorial of 4 hours. The annotators were then given seven passages, one at the time, and received feedback for each passage. A three hour tutorial session was then conducted, and focused on difficult

cases. To conclude their training, two additional training passages were given to each annotator. As this was the first large scale annotation effort with UCCA, the guidelines were somewhat modified during the training period. See Chapter 7 for a more elaborate discussion.

Published in ACL 2010

Chapter 3

Improved Unsupervised POS Induction through Prototype Discovery

Improved Unsupervised POS Induction through Prototype Discovery

Omri Abend[1*] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science, [2]ICNC
Hebrew University of Jerusalem
{omria01|roiri|arir}@cs.huji.ac.il

Abstract

We present a novel fully unsupervised algorithm for POS induction from plain text, motivated by the cognitive notion of prototypes. The algorithm first identifies *landmark* clusters of words, serving as the cores of the induced POS categories. The rest of the words are subsequently mapped to these clusters. We utilize morphological and distributional representations computed in a fully unsupervised manner. We evaluate our algorithm on English and German, achieving the best reported results for this task.

1 Introduction

Part-of-speech (POS) tagging is a fundamental NLP task, used by a wide variety of applications. However, there is no single standard POS tagging scheme, even for English. Schemes vary significantly across corpora and even more so across languages, creating difficulties in using POS tags across domains and for multi-lingual systems (Jiang et al., 2009). Automatic induction of POS tags from plain text can greatly alleviate this problem, as well as eliminate the efforts incurred by manual annotations. It is also a problem of great theoretical interest. Consequently, POS induction is a vibrant research area (see Section 2).

In this paper we present an algorithm based on the theory of prototypes (Taylor, 2003), which posits that some members in cognitive categories are more central than others. These practically define the category, while the membership of other elements is based on their association with the central members. Our algorithm first clusters words based on a fine morphological representation. It then clusters the most frequent words, defining *landmark* clusters which constitute the cores of the categories. Finally, it maps the rest of the words to these categories. The last two stages utilize a distributional representation that has been shown to be effective for unsupervised parsing (Seginer, 2007).

We evaluated the algorithm in both English and German, using four different mapping-based and information theoretic clustering evaluation measures. The results obtained are generally better than all existing POS induction algorithms.

Section 2 reviews related work. Sections 3 and 4 detail the algorithm. Sections 5, 6 and 7 describe the evaluation, experimental setup and results.

2 Related Work

Unsupervised and semi-supervised POS tagging have been tackled using a variety of methods. Schütze (1995) applied latent semantic analysis. The best reported results (when taking into account all evaluation measures, see Section 5) are given by (Clark, 2003), which combines distributional and morphological information with the likelihood function of the Brown algorithm (Brown et al., 1992). Clark's tagger is very sensitive to its initialization. Reichart et al. (2010b) propose a method to identify the high quality runs of this algorithm. In this paper, we show that our algorithm outperforms not only Clark's mean performance, but often its best among 100 runs. Most research views the task as a sequential labeling problem, using HMMs (Merialdo, 1994; Banko and Moore, 2004; Wang and Schuurmans, 2005) and discriminative models (Smith and Eisner, 2005; Haghighi and Klein, 2006). Several

[*] Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

techniques were proposed to improve the HMM model. A Bayesian approach was employed by (Goldwater and Griffiths, 2007; Johnson, 2007; Gao and Johnson, 2008). Van Gael et al. (2009) used the infinite HMM with non-parametric priors. Graça et al. (2009) biased the model to induce a small number of possible tags for each word.

The idea of utilizing seeds and expanding them to less reliable data has been used in several papers. Haghighi and Klein (2006) use POS 'prototypes' that are manually provided and tailored to a particular POS tag set of a corpus. Freitag (2004) and Biemann (2006) induce an initial clustering and use it to train an HMM model. Dasgupta and Ng (2007) generate morphological clusters and use them to bootstrap a distributional model. Goldberg et al. (2008) use linguistic considerations for choosing a good starting point for the EM algorithm. Zhao and Marcus (2009) expand a partial dictionary and use it to learn disambiguation rules. Their evaluation is only at the type level and only for half of the words. Ravi and Knight (2009) use a dictionary and an MDL-inspired modification to the EM algorithm.

Many of these works use a dictionary providing allowable tags for each or some of the words. While this scenario might reduce human annotation efforts, it does not induce a tagging scheme but remains tied to an existing one. It is further criticized in (Goldwater and Griffiths, 2007).

Morphological representation. Many POS induction models utilize morphology to some extent. Some use simplistic representations of terminal letter sequences (e.g., (Smith and Eisner, 2005; Haghighi and Klein, 2006)). Clark (2003) models the entire letter sequence as an HMM and uses it to define a morphological prior. Dasgupta and Ng (2007) use the output of the *Morfessor* segmentation algorithm for their morphological representation. *Morfessor* (Creutz and Lagus, 2005), which we use here as well, is an unsupervised algorithm that segments words and classifies each segment as being a stem or an affix. It has been tested on several languages with strong results.

Our work has several unique aspects. First, our clustering method discovers prototypes in a fully unsupervised manner, mapping the rest of the words according to their association with the prototypes. Second, we use a distributional representation which has been shown to be effective for unsupervised parsing (Seginer, 2007). Third, we use a morphological representation based on signatures, which are sets of affixes that represent a family of words sharing an inflectional or derivational morphology (Goldsmith, 2001).

3 Distributional Algorithm

Our algorithm is given a plain text corpus and optionally a desired number of clusters k. Its output is a partitioning of words into clusters. The algorithm utilizes two representations, distributional and morphological. Although eventually the latter is used before the former, for clarity of presentation we begin by detailing the base distributional algorithm. In the next section we describe the morphological representation and its integration into the base algorithm.

Overview. The algorithm consists of two main stages: landmark clusters discovery, and word mapping. For the former, we first compute a distributional representation for each word. We then cluster the coordinates corresponding to high frequency words. Finally, we define *landmark clusters*. In the word mapping stage we map each word to the most similar landmark cluster.

The rationale behind using only the high frequency words in the first stage is twofold. First, prototypical members of a category are frequent (Taylor, 2003), and therefore we can expect the salient POS tags to be represented in this small subset. Second, higher frequency implies more reliable statistics. Since this stage determines the cores of all resulting clusters, it should be as accurate as possible.

Distributional representation. We use a simplified form of the elegant representation of lexical entries used by the Seginer unsupervised parser (Seginer, 2007). Since a POS tag reflects the grammatical role of the word and since this representation is effective to parsing, we were motivated to apply it to the present task.

Let W be the set of word types in the corpus. The right context entry of a word $x \in W$ is a pair of mappings $r\_int_x : W \rightarrow [0,1]$ and $r\_adj_x : W \rightarrow [0,1]$. For each $w \in W$, $r\_adj_x(w)$ is an adjacency score of w to x, reflecting w's tendency to appear on the right hand side of x.

For each $w \in W$, $r\_int_x(w)$ is an interchangeability score of x with w, reflecting the tendency of w to appear to the left of words that tend to appear to the right of x. This can be viewed as a

similarity measure between words with respect to their right context. The higher the scores the more the words tend to be adjacent/interchangeable.

Left context parameters l_int_x and l_adj_x are defined analogously.

There are important subtleties in these definitions. First, for two words $x, w \in W$, $r\_adj_x(w)$ is generally different from $l\_adj_w(x)$. For example, if w is a high frequency word and x is a low frequency word, it is likely that w appears many times to the right of x, yielding a high $r\_adj_x(w)$, but that x appears only a few times to the left of w yielding a low $l\_adj_w(x)$. Second, from the definition of $r\_int_x(w)$ and $r\_int_w(x)$, it is clear that they need not be equal.

These functions are computed incrementally by a bootstrapping process. We initialize all mappings to be identically 0. We iterate over the words in the training corpus. For every word instance x, we take the word immediately to its right y and update x's right context using y's left context:

$$\forall w \in W: \quad r\_int_x(w) \mathrel{+}= \frac{l\_adj_y(w)}{N(y)}$$

$$\forall w \in W: \quad r\_adj_x(w) \mathrel{+}= \begin{cases} 1 & w = y \\ \frac{l\_int_y(w)}{N(y)} & w \neq y \end{cases}$$

The division by $N(y)$ (the number of times y appears in the corpus before the update) is done in order not to give a disproportional weight to high frequency words. Also, $r\_int_x(w)$ and $r\_adj_x(w)$ might become larger than 1. We therefore normalize them after all updates are performed by the number of occurrences of x in the corpus.

We update l_int_x and l_adj_x analogously using the word z immediately to the left of x. The updates of the left and right functions are done in parallel.

We define the distributional representation of a word type x to be a $4|W| + 2$ dimensional vector v_x. Each word w yields four coordinates, one for each direction (left/right) and one for each mapping type (int/adj). Two additional coordinates represent the frequency in which the word appears to the left and to the right of a stopping punctuation. Of the $4|W|$ coordinates corresponding to words, we allow only $2n$ to be non-zero: the n top scoring among the right side coordinates (those of r_int_x and r_adj_x), and the n top scoring among the left side coordinates (those of l_int_x and l_adj_x). We used $n = 50$.

The distance between two words is defined to be one minus the cosine of the angle between their representation vectors.

Coordinate clustering. Each of our landmark clusters will correspond to a set of high frequency words (HFWs). The number of HFWs is much larger than the number of expected POS tags. Hence we should cluster HFWs. Our algorithm does that by unifying some of the non-zero coordinates corresponding to HFWs in the distributional representation defined above.

We extract the words that appear more than N times per million[1] and apply the following procedure I times (5 in our experiments).

We run average link clustering with a threshold α (AVGLINK$_\alpha$, (Jain et al., 1999)) on these words, in each iteration initializing every HFW to have its own cluster. AVGLINK$_\alpha$ means running the average link algorithm until the two closest clusters have a distance larger than α. We then use the induced clustering to update the distributional representation, by collapsing all coordinates corresponding to words appearing in the same cluster into a single coordinate whose value is the sum of the collapsed coordinates' values. In order to produce a conservative (fine) clustering, we used a relatively low α value of 0.25.

Note that the AVGLINK$_\alpha$ initialization in each of the I iterations assigns each HFW to a separate cluster. The iterations differ in the distributional representation of the HFWs, resulting from the previous iterations.

In our English experiments, this process reduced the dimension of the HFWs set (the number of coordinates that are non-zero in at least one of the HFWs) from 14365 to 10722. The average number of non-zero coordinates per word decreased from 102 to 55.

Since all eventual POS categories correspond to clusters produced at this stage, to reduce noise we delete clusters of less than five elements.

Landmark detection. We define landmark clusters using the clustering obtained in the final iteration of the coordinate clustering stage. However, the number of clusters might be greater than the desired number k, which is an optional parameter of the algorithm. In this case we select a subset of k clusters that best covers the HFW space. We use the following heuristic. We start from the most frequent cluster, and greedily select the clus-

[1] We used $N = 100$, yielding 1242 words for English and 613 words for German.

ter farthest from the clusters already selected. The distance between two clusters is defined to be the average distance between their members. A cluster's distance from a set of clusters is defined to be its minimal distance from the clusters in the set. The final set of clusters $\{L_1, ..., L_k\}$ and their members are referred to as *landmark clusters* and *prototypes*, respectively.

Mapping all words. Each word $w \in W$ is assigned the cluster L_i that contains its nearest prototype:

$$d(w, L_i) = min_{x \in L_i}\{1 - cos(v_w, v_x)\}$$
$$Map(w) = argmin_{L_i}\{d(w, L_i)\}$$

Words that appear less than 5 times are considered as *unknown words*. We consider two schemes for handling unknown words. One randomly maps each such word to a cluster, using a probability proportional to the number of unique known words already assigned to that cluster. However, when the number k of landmark clusters is relatively large, it is beneficial to assign all unknown words to a separate new cluster (after running the algorithm with $k - 1$). In our experiments, we use the first option when k is below some threshold (we used 15), otherwise we use the second.

4 Morphological Model

The morphological model generates another word clustering, based on the notion of a signature. This clustering is integrated with the distributional model as described below.

4.1 Morphological Representation

We use the *Morfessor* (Creutz and Lagus, 2005) word segmentation algorithm. First, all words in the corpus are segmented. Then, for each stem, the set of all affixes with which it appears (its *signature*, (Goldsmith, 2001)) is collected. The morphological representation of a word type is then defined to be its stem's signature in conjunction with its specific affixes[2] (See Figure 1).

We now collect all words having the same representation. For instance, if the words *joined* and *painted* are found to have the same signature, they would share the same cluster since both have the affix '_ed'. The word *joins* does not share the same cluster with them since it has a different affix, '_s'. This results in coarse-grained clusters exclusively defined according to morphology.

[2] A word may contain more than a single affix.

Types	join	joins	joined	joining
Stem	join	join	join	join
Affixes	ϕ	_s	_ed	_ing
Signature	{ϕ,_ed,_s,_ing}			

Figure 1: An example for a morphological representation, defined to be the conjunction of its affix(es) with the stem's signature.

In addition, we incorporate capitalization information into the model, by constraining all words that appear capitalized in more than half of their instances to belong to a separate cluster, regardless of their morphological representation. The motivation for doing so is practical: capitalization is used in many languages to mark grammatical categories. For instance, in English capitalization marks the category of proper names and in German it marks the noun category . We report English results both with and without this modification.

Words that contain non-alphanumeric characters are represented as the sequence of the non-alphanumeric characters they include, e.g., 'vis-à-vis' is represented as *("-", "-")*. We do not assign a morphological representation to words including more than one stem (like *weatherman*), to words that have a null affix (i.e., where the word is identical to its stem) and to words whose stem is not shared by any other word (signature of size 1). Words that were not assigned a morphological representation are included as singletons in the morphological clustering.

4.2 Distributional-Morphological Algorithm

We detail the modifications made to our base distributional algorithm given the morphological clustering defined above.

Coordinate clustering and landmarks. We constrain AVGLINK$_\alpha$ to begin by forming links between words appearing in the same morphological cluster. Only when the distance between the two closest clusters gets above α we remove this constraint and proceed as before. This is equivalent to performing AVGLINK$_\alpha$ separately within each morphological cluster and then using the result as an initial condition for an AVGLINK$_\alpha$ coordinate clustering. The modified algorithm in this stage is otherwise identical to the distributional algorithm.

Word mapping. In this stage words that are not prototypes are mapped to one of the landmark

clusters. A reasonable strategy would be to map all words sharing a morphological cluster as a single unit. However, these clusters are too coarse-grained. We therefore begin by partitioning the morphological clusters into sub-clusters according to their distributional behavior. We do so by applying AVGLINK$_\beta$ (the same as AVGLINK$_\alpha$ but with a different parameter) to each morphological cluster. Since our goal is cluster *refinement*, we use a β that is considerably higher than α (0.9).

We then find the closest prototype to each such sub-cluster (averaging the distance across all of the latter's members) and map it as a single unit to the cluster containing that prototype.

5 Clustering Evaluation

We evaluate the clustering produced by our algorithm using an external quality measure: we take a corpus tagged by gold standard tags, tag it using the induced tags, and compare the two taggings. There is no single accepted measure quantifying the similarity between two taggings. In order to be as thorough as possible, we report results using four known measures, two mapping-based measures and two information theoretic ones.

Mapping-based measures. The induced clusters have arbitrary names. We define two mapping schemes between them and the gold clusters. After the induced clusters are mapped, we can compute a derived accuracy. The **Many-to-1** measure finds the mapping between the gold standard clusters and the induced clusters which maximizes accuracy, allowing several induced clusters to be mapped to the same gold standard cluster. The **1-to-1** measure finds the mapping between the induced and gold standard clusters which maximizes accuracy such that no two induced clusters are mapped to the same gold cluster. Computing this mapping is equivalent to finding the maximal weighted matching in a bipartite graph, whose weights are given by the intersection sizes between matched classes/clusters. As in (Reichart and Rappoport, 2008), we use the Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957) to solve this problem.

Information theoretic measures. These are based on the observation that a good clustering reduces the uncertainty of the gold tag given the induced cluster, and vice-versa. Several such measures exist; we use **V** (Rosenberg and Hirschberg, 2007) and **NVI** (Reichart and Rappoport, 2009), **VI**'s (Meila, 2007) normalized version.

6 Experimental Setup

Since a goal of unsupervised POS tagging is inducing an annotation scheme, comparison to an existing scheme is problematic. To address this problem we compare to three different schemes in two languages. In addition, the two English schemes we compare with were designed to tag corpora contained in our training set, and have been widely and successfully used with these corpora by a large number of applications.

Our algorithm was run with the exact same parameters on both languages: $N = 100$ (high frequency threshold), $n = 50$ (the parameter that determines the effective number of coordinates), $\alpha = 0.25$ (cluster separation during landmark cluster generation), $\beta = 0.9$ (cluster separation during refinement of morphological clusters).

The algorithm we compare with in most detail is (Clark, 2003), which reports the best current results for this problem (see Section 7). Since Clark's algorithm is sensitive to its initialization, we ran it a 100 times and report its average and standard deviation in each of the four measures. In addition, we report the percentile in which our result falls with respect to these 100 runs.

Punctuation marks are very frequent in corpora and are easy to cluster. As a result, including them in the evaluation greatly inflates the scores. For this reason we do not assign a cluster to punctuation marks and we report results using this policy, which we recommend for future work. However, to be able to directly compare with previous work, we also report results for the full POS tag set. We do so by assigning a singleton cluster to each punctuation mark (in addition to the k required clusters). This simple heuristic yields very high performance on punctuation, scoring (when all other words are assumed perfect tagging) 99.6% (99.1%) 1-to-1 accuracy when evaluated against the English fine (coarse) POS tag sets, and 97.2% when evaluated against the German POS tag set.

For English, we trained our model on the 39832 sentences which constitute sections 2-21 of the PTB-WSJ and on the 500K sentences from the NYT section of the NANC newswire corpus (Graff, 1995). We report results on the WSJ part of our data, which includes 950028 words tokens in 44389 types. Of the tokens, 832629 (87.6%)

English	Fine k=13				Coarse k=13				Fine k=34			
	Prototype Tagger	Clark μ	Clark σ	%	Prototype Tagger	Clark μ	Clark σ	%	Prototype Tagger	Clark μ	Clark σ	%
Many–to–1	**61.0**	55.1	1.6	100	**70.0**	66.9	2.1	94	**71.6**	69.8	1.5	90
	55.5	48.8	1.8	100	**66.1**	62.6	2.3	94	**67.5**	65.5	1.7	90
1–to–1	**60.0**	52.2	1.9	100	**58.1**	49.4	2.9	100	**63.5**	54.5	1.6	100
	54.9	46.0	2.2	100	**53.7**	43.8	3.3	100	**58.8**	48.5	1.8	100
NVI	**0.652**	0.773	0.027	100	**0.841**	0.972	0.036	100	**0.663**	0.725	0.018	100
	0.795	0.943	0.033	100	**1.052**	1.221	0.046	100	**0.809**	0.885	0.022	100
V	**0.636**	0.581	0.015	100	**0.590**	0.543	0.018	100	**0.677**	0.659	0.008	100
	0.542	0.478	0.019	100	**0.484**	0.429	0.023	100	**0.608**	0.588	0.010	98

German	k=17				k=26			
	Prototype Tagger	Clark μ	Clark σ	%	Prototype Tagger	Clark μ	Clark σ	%
Many–to-1	64.6	**64.7**	1.2	41	**68.2**	67.8	1.0	60
	58.9	**59.1**	1.4	40	**63.2**	62.8	1.2	60
1–to–1	**53.7**	52.0	1.8	77	**56.0**	52.0	2.1	99
	48.0	46.0	2.3	78	**50.7**	45.9	2.6	99
NVI	**0.667**	0.675	0.019	66	**0.640**	0.682	0.019	100
	0.819	0.829	0.025	66	**0.785**	0.839	0.025	100
V	**0.646**	0.645	0.010	50	**0.675**	0.657	0.008	100
	0.552	**0.553**	0.013	48	**0.596**	0.574	0.010	100

Table 1: Top: English. Bottom: German. Results are reported for our model (Prototype Tagger), Clark's average score (μ), Clark's standard deviation (σ) and the fraction of Clark's results that scored worse than our model (%). For the mapping based measures, results are accuracy percentage. For $V \in [0,1]$, higher is better. For high quality output, $NVI \in [0,1]$ as well, and lower is better. In each entry, the top number indicates the score when including punctuation and the bottom number the score when excluding it. In English, our results are always better than Clark's. In German, they are almost always better.

are not punctuation. The percentage of unknown words (those appearing less than five times) is 1.6%. There are 45 clusters in this annotation scheme, 34 of which are not punctuation.

We ran each algorithm both with k=13 and k=34 (the number of desired clusters). We compare the output to two annotation schemes: the fine grained PTB WSJ scheme, and the coarse grained tags defined in (Smith and Eisner, 2005). The output of the k=13 run is evaluated both against the coarse POS tag annotation (the *'Coarse k=13'* scenario) and against the full PTB-WSJ annotation scheme (the *'Fine k=13'* scenario). The k=34 run is evaluated against the full PTB-WSJ annotation scheme (the *'Fine k=34'* scenario).

The POS cluster frequency distribution tends to be skewed: each of the 13 most frequent clusters in the PTB-WSJ cover more than 2.5% of the tokens (excluding punctuation) and together 86.3% of them. We therefore chose k=13, since it is both the number of coarse POS tags (excluding punctuation) as well as the number of frequent POS tags in the PTB-WSJ annotation scheme. We chose k=34 in order to evaluate against the full 34 tags PTB-WSJ annotation scheme (excluding punctuation) using the same number of clusters.

For German, we trained our model on the 20296 sentences of the NEGRA corpus (Brants, 1997) and on the first 450K sentences of the DeWAC corpus (Baroni et al., 2009). DeWAC is a corpus extracted by web crawling and is therefore out of domain. We report results on the NEGRA part, which includes 346320 word tokens of 49402 types. Of the tokens, 289268 (83.5%) are not punctuation. The percentage of unknown words (those appearing less than five times) is 8.1%. There are 62 clusters in this annotation scheme, 51 of which are not punctuation.

We ran the algorithms with k=17 and k=26. k=26 was chosen since it is the number of clusters that cover each more than 0.5% of the NEGRA tokens, and in total cover 96% of the (non-punctuation) tokens. In order to test our algorithm in another scenario, we conducted experiments with k=17 as well, which covers 89.9% of the tokens. All outputs are compared against NEGRA's gold standard scheme.

We do not report results for k=51 (where the number of gold clusters is the same as the number of induced clusters), since our algorithm produced only 42 clusters in the landmark detection stage. We could of course have modified the parameters to allow our algorithm to produce 51 clusters. However, we wanted to use the exact same parameters as those used for the English experiments to minimize the issue of parameter tuning.

In addition to the comparisons described above, we present results of experiments (in the 'Fine

	B	B+M	B+C	F(I=1)	F
M-to-1	53.3	54.8	58.2	57.3	**61.0**
1-to-1	50.2	51.7	55.1	54.8	**60.0**
NVI	0.782	0.720	0.710	0.742	**0.652**
V	0.569	0.598	0.615	0.597	**0.636**

Table 2: A comparison of partial versions of the model in the 'Fine k=13' WSJ scenario. M-to-1 and 1-to-1 results are reported in accuracy percentage. Lower NVI is better. *B* is the strictly distributional algorithm, *B+M* adds the morphological model, *B+C* adds capitalization to *B*, *F(I=1)* consists of all components, where only one iteration of coordinate clustering is performed, and *F* is the full model.

	M-to-1	1-to-1	V	VI
Prototype	**71.6**	**63.5**	**0.677**	**2.00**
Clark	69.8	54.5	0.659	2.18
HK	–	41.3	–	–
J	43–62	37–47	–	4.23–5.74
GG	–	–	–	2.8
GJ	–	40–49.9	–	4.03–4.47
VG	–	–	0.54-0.59	2.5–2.9
GGTP-45	65.4	44.5	–	–
GGTP-17	70.2	49.5	–	–

Table 4: Comparison of our algorithms with the recent fully unsupervised POS taggers for which results are reported. The models differ in the annotation scheme, the corpus size and the number of induced clusters (k) that they used. HK: (Haghighi and Klein, 2006), 193K tokens, fine tags, k=45. GG: (Goldwater and Griffiths, 2007), 24K tokens, coarse tags, k=17. J : (Johnson, 2007), 1.17M tokens, fine tags, k=25–50. GJ: (Gao and Johnson, 2008), 1.17M tokens, fine tags, k=50. VG: (Van Gael et al., 2009), 1.17M tokens, fine tags, k=47–192. GGTP-45: (Graça et al., 2009), 1.17M tokens, fine tags, k=45. GGTP-17: (Graça et al., 2009), 1.17M tokens, coarse tags, k=17. Lower VI values indicate better clustering. VI is computed using e as the base of the logarithm. Our algorithm gives the best results.

k=13' scenario) that quantify the contribution of each component of the algorithm. We ran the base distributional algorithm, a variant which uses only capitalization information (i.e., has only one non-singleton morphological class, that of words appearing capitalized in most of their instances) and a variant which uses no capitalization information, defining the morphological clusters according to the morphological representation alone.

7 Results

Table 1 presents results for the English and German experiments. For English, our algorithm obtains better results than Clark's in all measures and scenarios. It is without exception better than the average score of Clark's and in most cases better than the maximal Clark score obtained in 100 runs.

A significant difference between our algorithm and Clark's is that the latter, like most algorithms which addressed the task, induces the clustering

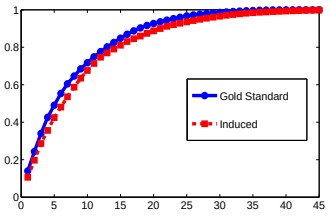

Figure 2: POS class frequency distribution for our model and the gold standard, in the 'Fine k=34' scenario. The distributions are similar.

by maximizing a non-convex function. These functions have many local maxima and the specific solution to which algorithms that maximize them converge strongly depends on their (random) initialization. Therefore, their output's quality often significantly diverges from the average. This issue is discussed in depth in (Reichart et al., 2010b). Our algorithm is deterministic[3].

For German, in the k=26 scenario our algorithm outperforms Clark's, often outperforming even its maximum in 100 runs. In the k=17 scenario, our algorithm obtains a higher score than Clark with probability 0.4 to 0.78, depending on the measure and scenario. Clark's average score is slightly better in the Many-to-1 measure, while our algorithm performs somewhat better than Clark's average in the 1-to-1 and NVI measures.

The DeWAC corpus from which we extracted statistics for the German experiments is out of domain with respect to NEGRA. The corresponding corpus in English, NANC, is a newswire corpus and therefore clearly in-domain with respect to WSJ. This is reflected by the percentage of unknown words, which was much higher in German than in English (8.1% and 1.6%), lowering results.

Table 2 shows the effect of each of our algorithm's components. Each component provides an improvement over the base distributional algorithm. The full coordinate clustering stage (several iterations, F) considerably improves the score over a single iteration (F(I=1)). Capitalization information increases the score more than the morphological information, which might stem from the granularity of the POS tag set with respect to names. This analysis is supported by similar experiments we made in the 'Coarse k=13' scenario (not shown in tables here). There, the decrease in performance was only of 1%–2% in the mapping

[3]The fluctuations inflicted on our algorithm by the random mapping of unknown words are of less than 0.1% .

	Excluding Punctuation				Including Punctuation				Perfect Punctuation			
	M-to-1	1-to-1	NVI	V	M-to-1	1-to-1	NVI	V	M-to-1	1-to-1	NVI	V
Van Gael	59.1	48.4	0.999	0.530	62.3	51.3	0.861	0.591	64.0	54.6	0.820	0.610
Prototype	**67.5**	**58.8**	**0.809**	**0.608**	**71.6**	**63.5**	**0.663**	**0.677**	**71.6**	**63.9**	**0.659**	**0.679**

Table 3: Comparison between the *iHMM: PY-fixed* model (Van Gael et al., 2009) and ours with various punctuation assignment schemes. Left section: punctuation tokens are excluded. Middle section: punctuation tokens are included. Right section: perfect assignment of punctuation is assumed.

based measures and 3.5% in the V measure.

Finally, Table 4 presents reported results for all recent algorithms we are aware of that tackled the task of unsupervised POS induction from plain text. Results for our algorithm's and Clark's are reported for the 'Fine, k=34' scenario. The settings of the various experiments vary in terms of the exact annotation scheme used (coarse or fine grained) and the size of the test set. However, the score differences are sufficiently large to justify the claim that our algorithm is currently the best performing algorithm on the PTB-WSJ corpus for POS induction from plain text[4].

Since previous works provided results only for the scenario in which punctuation is included, the reported results are not directly comparable. In order to quantify the effect various punctuation schemes have on the results, we evaluated the *'iHMM: PY-fixed'* model (Van Gael et al., 2009) and ours when punctuation is excluded, included or perfectly tagged[5]. The results (Table 3) indicate that most probably even after an appropriate correction for punctuation, our model remains the best performing one.

8 Discussion

In this work we presented a novel unsupervised algorithm for POS induction from plain text. The algorithm first generates relatively accurate clusters of high frequency words, which are subsequently used to bootstrap the entire clustering. The distributional and morphological representations that we use are novel for this task.

We experimented on two languages with mapping and information theoretic clustering evaluation measures. Our algorithm obtains the best reported results on the English PTB-WSJ corpus. In addition, our results are almost always better than Clark's on the German NEGRA corpus.

We have also performed a manual error analysis, which showed that our algorithm performs much better on closed classes than on open classes. In order to asses this quantitatively, let us define a random variable for each of the gold clusters, which receives a value corresponding to each induced cluster with probability proportional to their intersection size. For each gold cluster, we compute the entropy of this variable. In addition, we greedily map each induced cluster to a gold cluster and compute the ratio between their intersection size and the size of the gold cluster (mapping accuracy).

We experimented in the 'Fine k=34' scenario. The clusters that obtained the best scores were (brackets indicate mapping accuracy and entropy for each of these clusters) coordinating conjunctions (95%, 0.32), prepositions (94%, 0.32), determiners (94%, 0.44) and modals (93%, 0.45). These are all closed classes.

The classes on which our algorithm performed worst consist of open classes, mostly verb types: past tense verbs (47%, 2.2), past participle verbs (44%, 2.32) and the morphologically unmarked non-3rd person singular present verbs (32%, 2.86). Another class with low performance is the proper nouns (37%, 2.9). The errors there are mostly of three types: confusions between common and proper nouns (sometimes due to ambiguity), unknown words which were put in the unknown words cluster, and abbreviations which were given a separate class by our algorithm. Finally, the algorithm's performance on the heterogeneous adverbs class (19%, 3.73) is the lowest.

Clark's algorithm exhibits[6] a similar pattern with respect to open and closed classes. While his algorithm performs considerably better on adverbs (15% mapping accuracy difference and 0.71 entropy difference), our algorithm scores considerably better on prepositions (17%, 0.77), superlative adjectives (38%, 1.37) and plural proper names (45%, 1.26).

[4] Graça et al. (2009) report very good results for 17 tags in the M-1 measure. However, their 1-1 results are quite poor, and results for the common IT measures were not reported. Their results for 45 tags are considerably lower.

[5] We thank the authors for sending us their data.

[6] Using average mapping accuracy and entropy over the 100 runs.

Naturally, this analysis might reflect the arbitrary nature of a manually design POS tag set rather than deficiencies in automatic POS induction algorithms. In future work we intend to analyze the output of such algorithms in order to improve POS tag sets.

Our algorithm and Clark's are monosemous (i.e., they assign each word exactly one tag), while most other algorithms are polysemous. In order to assess the performance loss caused by the monosemous nature of our algorithm, we took the M-1 greedy mapping computed for the entire dataset and used it to compute accuracy over the monosemous and polysemous words separately. Results are reported for the English 'Fine k=34' scenario (without punctuation). We define a word to be monosemous if more than 95% of its tokens are assigned the same gold standard tag. For English, there are approximately 255K polysemous tokens and 578K monosemous ones. As expected, our algorithm is much more accurate on the monosemous tokens, achieving 76.6% accuracy, compared to 47.1% on the polysemous tokens.

The evaluation in this paper is done at the token level. Type level evaluation, reflecting the algorithm's ability to detect the set of possible POS tags for each word type, is important as well. It could be expected that a monosemous algorithm such as ours would perform poorly in a type level evaluation. In (Reichart et al., 2010a) we discuss type level evaluation at depth and propose type level evaluation measures applicable to the POS induction problem. In that paper we compare the performance of our Prototype Tagger with leading unsupervised POS tagging algorithms (Clark, 2003; Goldwater and Griffiths, 2007; Gao and Johnson, 2008; Van Gael et al., 2009). Our algorithm obtained the best results in 4 of the 6 measures in a margin of 4–6%, and was second best in the other two measures. Our results were better than Clark's (the only other monosemous algorithm evaluated there) on all measures in a margin of 5–21%. The fact that our monosemous algorithm was better than good polysemous algorithms in a type level evaluation can be explained by the prototypical nature of the POS phenomenon (a longer discussion is given in (Reichart et al., 2010a)). However, the quality upper bound for monosemous algorithms is obviously much lower than that for polysemous algorithms, and we expect polysemous algorithms to outperform monosemous algorithms in the future in both type level and token level evaluations.

The skewed (Zipfian) distribution of POS class frequencies in corpora is a problem for many POS induction algorithms, which by default tend to induce a clustering having a balanced distribution. Explicit modifications to these algorithms were introduced in order to bias their model to produce such a distribution (see (Clark, 2003; Johnson, 2007; Reichart et al., 2010b)). An appealing property of our model is its ability to induce a skewed distribution without being explicitly tuned to do so, as seen in Figure 2.

Acknowledgements. We would like to thank Yoav Seginer for his help with his parser.

References

Michele Banko and Robert C. Moore, 2004. *Part of Speech Tagging in Context.* COLING '04.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi and Eros Zanchetta, 2009. *The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora.* Language Resources and Evaluation.

Chris Biemann, 2006. *Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering.* COLING-ACL '06 Student Research Workshop.

Thorsten Brants, 1997. *The NEGRA Export Format.* CLAUS Report, Saarland University.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souze, Jenifer C. Lai and Robert Mercer, 1992. *Class-Based N-Gram Models of Natural Language.* Computational Linguistics, 18(4):467–479.

Alexander Clark, 2003. *Combining Distributional and Morphological Information for Part of Speech Induction.* EACL '03.

Mathias Creutz and Krista Lagus, 2005. *Inducing the Morphological Lexicon of a Natural Language from Unannotated Text.* AKRR '05.

Sajib Dasgupta and Vincent Ng, 2007. *Unsupervised Part-of-Speech Acquisition for Resource-Scarce Languages.* EMNLP-CoNLL '07.

Dayne Freitag, 2004. *Toward Unsupervised Whole-Corpus Tagging.* COLING '04.

Jianfeng Gao and Mark Johnson, 2008. *A Comparison of Bayesian Estimators for Unsupervised Hidden Markov Model POS Taggers.* EMNLP '08.

Yoav Goldberg, Meni Adler and Michael Elhadad, 2008. *EM Can Find Pretty Good HMM POS-Taggers (When Given a Good Start).* ACL '08.

John Goldsmith, 2001. *Unsupervised Learning of the Morphology of a Natural Language.* Computational Linguistics, 27(2):153–198.

Sharon Goldwater and Tom Griffiths, 2007. *Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging.* ACL '07.

João Graça, Kuzman Ganchev, Ben Taskar and Fernando Pereira, 2009. Posterior vs. Parameter Sparsity in Latent Variable Models. *NIPS '09.*

David Graff, 1995. *North American News Text Corpus.* Linguistic Data Consortium. LDC95T21.

Aria Haghighi and Dan Klein, 2006. *Prototype-driven Learning for Sequence Labeling.* HLT–NAACL '06.

Anil K. Jain, Narasimha M. Murty and Patrick J. Flynn, 1999. *Data Clustering: A Review.* ACM Computing Surveys 31(3):264–323.

Wenbin Jiang, Liang Huang and Qun Liu, 2009. *Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging – A Case Study.* ACL '09.

Mark Johnson, 2007. *Why Doesnt EM Find Good HMM POS-Taggers?* EMNLP-CoNLL '07.

Harold W. Kuhn, 1955. *The Hungarian method for the Assignment Problem.* Naval Research Logistics Quarterly, 2:83-97.

Marina Meila, 2007. *Comparing Clustering – an Information Based Distance.* Journal of Multivariate Analysis, 98:873–895.

Bernard Merialdo, 1994. *Tagging English Text with a Probabilistic Model.* Computational Linguistics, 20(2):155–172.

James Munkres, 1957. *Algorithms for the Assignment and Transportation Problems.* Journal of the SIAM, 5(1):32–38.

Sujith Ravi and Kevin Knight, 2009. *Minimized Models for Unsupervised Part-of-Speech Tagging.* ACL '09.

Roi Reichart and Ari Rappoport, 2008. *Unsupervised Induction of Labeled Parse Trees by Clustering with Syntactic Features.* COLING '08.

Roi Reichart and Ari Rappoport, 2009. *The NVI Clustering Evaluation Measure.* CoNLL '09.

Roi Reichart, Omri Abend and Ari Rappoport, 2010a. *Type Level Clustering Evaluation: New Measures and a POS Induction Case Study.* CoNLL '10.

Roi Reichart, Raanan Fattal and Ari Rappoport, 2010b. *Improved Unsupervised POS Induction Using Intrinsic Clustering Quality and a Zipfian Constraint.* CoNLL '10.

Andrew Rosenberg and Julia Hirschberg, 2007. *V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.* EMNLP '07.

Hinrich Schütze, 1995. *Distributional part-of-speech tagging.* EACL '95.

Yoav Seginer, 2007. *Fast Unsupervised Incremental Parsing.* ACL '07.

Noah A. Smith and Jason Eisner, 2005. *Contrastive Estimation: Training Log-Linear Models on Unlabeled Data.* ACL '05.

John R. Taylor, 2003. *Linguistic Categorization: Prototypes in Linguistic Theory, Third Edition.* Oxford University Press.

Jurgen Van Gael, Andreas Vlachos and Zoubin Ghahramani, 2009. *The Infinite HMM for Unsupervised POS Tagging.* EMNLP '09.

Qin Iris Wang and Dale Schuurmans, 2005. *Improved Estimation for Unsupervised Part-of-Speech Tagging.* IEEE NLP–KE '05.

Qiuye Zhao and Mitch Marcus, 2009. *A Simple Unsupervised Learner for POS Disambiguation Rules Given Only a Minimal Lexicon.* EMNLP '09.

Published in ACL 2009

Chapter 4

Unsupervised Argument Identification for Semantic Role Labeling

Unsupervised Argument Identification for Semantic Role Labeling

Omri Abend[1] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science , [2]ICNC
Hebrew University of Jerusalem
{omria01|roiri|arir}@cs.huji.ac.il

Abstract

The task of Semantic Role Labeling (SRL) is often divided into two sub-tasks: verb argument identification, and argument classification. Current SRL algorithms show lower results on the identification sub-task. Moreover, most SRL algorithms are supervised, relying on large amounts of manually created data. In this paper we present an unsupervised algorithm for identifying verb arguments, where the only type of annotation required is POS tagging. The algorithm makes use of a fully unsupervised syntactic parser, using its output in order to detect clauses and gather candidate argument collocation statistics. We evaluate our algorithm on PropBank10, achieving a precision of 56%, as opposed to 47% of a strong baseline. We also obtain an 8% increase in precision for a Spanish corpus. This is the first paper that tackles unsupervised verb argument identification without using manually encoded rules or extensive lexical or syntactic resources.

1 Introduction

Semantic Role Labeling (SRL) is a major NLP task, providing a shallow sentence-level semantic analysis. SRL aims at identifying the relations between the predicates (usually, verbs) in the sentence and their associated arguments.

The SRL task is often viewed as consisting of two parts: argument identification (ARGID) and argument classification. The former aims at identifying the arguments of a given predicate present in the sentence, while the latter determines the type of relation that holds between the identified arguments and their corresponding predicates. The division into two sub-tasks is justified by the fact that they are best addressed using different feature sets (Pradhan et al., 2005). Performance in the ARGID stage is a serious bottleneck for general SRL performance, since only about 81% of the arguments are identified, while about 95% of the identified arguments are labeled correctly (Màrquez et al., 2008).

SRL is a complex task, which is reflected by the algorithms used to address it. A standard SRL algorithm requires thousands to dozens of thousands sentences annotated with POS tags, syntactic annotation and SRL annotation. Current algorithms show impressive results but only for languages and domains where plenty of annotated data is available, e.g., English newspaper texts (see Section 2). Results are markedly lower when testing is on a domain wider than the training one, even in English (see the WSJ-Brown results in (Pradhan et al., 2008)).

Only a small number of works that do not require manually labeled SRL training data have been done (Swier and Stevenson, 2004; Swier and Stevenson, 2005; Grenager and Manning, 2006). These papers have replaced this data with the VerbNet (Kipper et al., 2000) lexical resource or a set of manually written rules and supervised parsers.

A potential answer to the SRL training data bottleneck are unsupervised SRL models that require little to no manual effort for their training. Their output can be used either by itself, or as training material for modern supervised SRL algorithms.

In this paper we present an algorithm for unsupervised argument identification. The only type of annotation required by our algorithm is POS tag-

ging, which needs relatively little manual effort.

The algorithm consists of two stages. As preprocessing, we use a fully unsupervised parser to parse each sentence. Initially, the set of possible arguments for a given verb consists of all the constituents in the parse tree that do not contain that predicate. The first stage of the algorithm attempts to detect the minimal clause in the sentence that contains the predicate in question. Using this information, it further reduces the possible arguments only to those contained in the minimal clause, and further prunes them according to their position in the parse tree. In the second stage we use pointwise mutual information to estimate the collocation strength between the arguments and the predicate, and use it to filter out instances of weakly collocating predicate argument pairs.

We use two measures to evaluate the performance of our algorithm, precision and F-score. Precision reflects the algorithm's applicability for creating training data to be used by supervised SRL models, while the standard SRL F-score measures the model's performance when used by itself. The first stage of our algorithm is shown to outperform a strong baseline both in terms of F-score and of precision. The second stage is shown to increase precision while maintaining a reasonable recall.

We evaluated our model on sections 2-21 of Propbank. As is customary in unsupervised parsing work (e.g. (Seginer, 2007)), we bounded sentence length by 10 (excluding punctuation). Our first stage obtained a precision of 52.8%, which is more than 6% improvement over the baseline. Our second stage improved precision to nearly 56%, a 9.3% improvement over the baseline. In addition, we carried out experiments on Spanish (on sentences of length bounded by 15, excluding punctuation), achieving an increase of over 7.5% in precision over the baseline. Our algorithm increases F–score as well, showing an 1.8% improvement over the baseline in English and a 2.2% improvement in Spanish.

Section 2 reviews related work. In Section 3 we detail our algorithm. Sections 4 and 5 describe the experimental setup and results.

2 Related Work

The advance of machine learning based approaches in this field owes to the usage of large scale annotated corpora. English is the most studied language, using the FrameNet (FN) (Baker et al., 1998) and PropBank (PB) (Palmer et al., 2005) resources. PB is a corpus well suited for evaluation, since it annotates every non-auxiliary verb in a real corpus (the WSJ sections of the Penn Treebank). PB is a standard corpus for SRL evaluation and was used in the CoNLL SRL shared tasks of 2004 (Carreras and Màrquez, 2004) and 2005 (Carreras and Màrquez, 2005).

Most work on SRL has been supervised, requiring dozens of thousands of SRL annotated training sentences. In addition, most models assume that a syntactic representation of the sentence is given, commonly in the form of a parse tree, a dependency structure or a shallow parse. Obtaining these is quite costly in terms of required human annotation.

The first work to tackle SRL as an independent task is (Gildea and Jurafsky, 2002), which presented a supervised model trained and evaluated on FrameNet. The CoNLL shared tasks of 2004 and 2005 were devoted to SRL, and studied the influence of different syntactic annotations and domain changes on SRL results. *Computational Linguistics* has recently published a special issue on the task (Màrquez et al., 2008), which presents state-of-the-art results and surveys the latest achievements and challenges in the field.

Most approaches to the task use a multi-level approach, separating the task to an ARGID and an argument classification sub-tasks. They then use the unlabeled argument structure (without the semantic roles) as training data for the ARGID stage and the entire data (perhaps with other features) for the classification stage. Better performance is achieved on the classification, where state-of-the-art supervised approaches achieve about 81% F-score on the in-domain identification task, of which about 95% are later labeled correctly (Màrquez et al., 2008).

There have been several exceptions to the standard architecture described in the last paragraph. One suggestion poses the problem of SRL as a sequential tagging of words, training an SVM classifier to determine for each word whether it is inside, outside or in the beginning of an argument (Hacioglu and Ward, 2003). Other works have integrated argument classification and identification into one step (Collobert and Weston, 2007), while others went further and combined the former two along with parsing into a single model (Musillo

and Merlo, 2006).

Work on less supervised methods has been scarce. Swier and Stevenson (2004) and Swier and Stevenson (2005) presented the first model that does not use an SRL annotated corpus. However, they utilize the extensive verb lexicon VerbNet, which lists the possible argument structures allowable for each verb, and supervised syntactic tools. Using VerbNet along with the output of a rule-based chunker (in 2004) and a supervised syntactic parser (in 2005), they spot instances in the corpus that are very similar to the syntactic patterns listed in VerbNet. They then use these as seed for a bootstrapping algorithm, which consequently identifies the verb arguments in the corpus and assigns their semantic roles.

Another less supervised work is that of (Grenager and Manning, 2006), which presents a Bayesian network model for the argument structure of a sentence. They use EM to learn the model's parameters from unannotated data, and use this model to tag a test corpus. However, ARGID was not the task of that work, which dealt solely with argument classification. ARGID was performed by manually-created rules, requiring a supervised or manual syntactic annotation of the corpus to be annotated.

The three works above are relevant but incomparable to our work, due to the extensive amount of supervision (namely, VerbNet and a rule-based or supervised syntactic system) they used, both in detecting the syntactic structure and in detecting the arguments.

Work has been carried out in a few other languages besides English. Chinese has been studied in (Xue, 2008). Experiments on Catalan and Spanish were done in SemEval 2007 (Màrquez et al., 2007) with two participating systems. Attempts to compile corpora for German (Burdchardt et al., 2006) and Arabic (Diab et al., 2008) are also underway. The small number of languages for which extensive SRL annotated data exists reflects the considerable human effort required for such endeavors.

Some SRL works have tried to use unannotated data to improve the performance of a base supervised model. Methods used include bootstrapping approaches (Gildea and Jurafsky, 2002; Kate and Mooney, 2007), where large unannotated corpora were tagged with SRL annotation, later to be used to retrain the SRL model. Another approach used similarity measures either between verbs (Gordon and Swanson, 2007) or between nouns (Gildea and Jurafsky, 2002) to overcome lexical sparsity. These measures were estimated using statistics gathered from corpora augmenting the model's training data, and were then utilized to generalize across similar verbs or similar arguments.

Attempts to substitute full constituency parsing by other sources of syntactic information have been carried out in the SRL community. Suggestions include posing SRL as a sequence labeling problem (Màrquez et al., 2005) or as an edge tagging problem in a dependency representation (Hacioglu, 2004). Punyakanok et al. (2008) provide a detailed comparison between the impact of using shallow vs. full constituency syntactic information in an English SRL system. Their results clearly demonstrate the advantage of using full annotation.

The identification of arguments has also been carried out in the context of automatic subcategorization frame acquisition. Notable examples include (Manning, 1993; Briscoe and Carroll, 1997; Korhonen, 2002) who all used statistical hypothesis testing to filter a parser's output for arguments, with the goal of compiling verb subcategorization lexicons. However, these works differ from ours as they attempt to characterize the behavior of a verb type, by collecting statistics from various instances of that verb, and not to determine which are the arguments of specific verb instances.

The algorithm presented in this paper performs unsupervised clause detection as an intermediate step towards argument identification. Supervised clause detection was also tackled as a separate task, notably in the CoNLL 2001 shared task (Tjong Kim Sang and Dèjean, 2001). Clause information has been applied to accelerating a syntactic parser (Glaysher and Moldovan, 2006).

3 Algorithm

In this section we describe our algorithm. It consists of two stages, each of which reduces the set of argument candidates, which a-priori contains all consecutive sequences of words that do not contain the predicate in question.

3.1 Algorithm overview

As pre-processing, we use an unsupervised parser that generates an unlabeled parse tree for each sen-

tence (Seginer, 2007). This parser is unique in that it is able to induce a bracketing (unlabeled parsing) from raw text (without even using POS tags) achieving state-of-the-art results. Since our algorithm uses millions to tens of millions sentences, we must use very fast tools. The parser's high speed (thousands of words per second) enables us to process these large amounts of data.

The only type of supervised annotation we use is POS tagging. We use the taggers MXPOST (Ratnaparkhi, 1996) for English and TreeTagger (Schmid, 1994) for Spanish, to obtain POS tags for our model.

The first stage of our algorithm uses linguistically motivated considerations to reduce the set of possible arguments. It does so by confining the set of argument candidates only to those constituents which obey the following two restrictions. First, they should be contained in the minimal clause containing the predicate. Second, they should be k-th degree cousins of the predicate in the parse tree. We propose a novel algorithm for clause detection and use its output to determine which of the constituents obey these two restrictions.

The second stage of the algorithm uses pointwise mutual information to rule out constituents that appear to be weakly collocating with the predicate in question. Since a predicate greatly restricts the type of arguments with which it may appear (this is often referred to as "selectional restrictions"), we expect it to have certain characteristic arguments with which it is likely to collocate.

3.2 Clause detection stage

The main idea behind this stage is the observation that most of the arguments of a predicate are contained within the minimal clause that contains the predicate. We tested this on our development data – section 24 of the WSJ PTB, where we saw that 86% of the arguments that are also constituents (in the gold standard parse) were indeed contained in that minimal clause (as defined by the tree label types in the gold standard parse that denote a clause, e.g., S, SBAR). Since we are not provided with clause annotation (or any label), we attempted to detect them in an unsupervised manner. Our algorithm attempts to find sub-trees within the parse tree, whose structure resembles the structure of a full sentence. This approximates the notion of a clause.

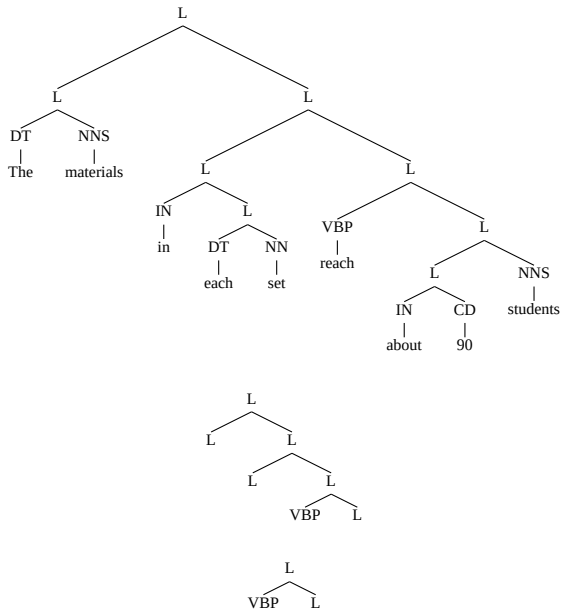

Figure 1: An example of an unlabeled POS tagged parse tree. The middle tree is the ST of 'reach' with the root as the encoded ancestor. The bottom one is the ST with its parent as the encoded ancestor.

Statistics gathering. In order to detect which of the verb's ancestors is the minimal clause, we score each of the ancestors and select the one that maximizes the score. We represent each ancestor using its *Spinal Tree* (ST). The ST of a given verb's ancestor is obtained by replacing all the constituents that do not contain the verb by a leaf having a label. This effectively encodes all the k-th degree cousins of the verb (for every k). The leaf labels are either the word's POS in case the constituent is a leaf, or the generic label "L" denoting a non-leaf. See Figure 1 for an example.

In this stage we collect statistics of the occurrences of STs in a large corpus. For every ST in the corpus, we count the number of times it occurs in a form we consider to be a clause (positive examples), and the number of times it appears in other forms (negative examples).

Positive examples are divided into two main types. First, when the ST encodes the root ancestor (as in the middle tree of Figure 1); second, when the ancestor complies to a clause lexico-syntactic pattern. In many languages there is a small set of lexico-syntactic patterns that mark a clause, e.g. the English 'that', the German 'dass' and the Spanish 'que'. The patterns which were used in our experiments are shown in Figure 2.

For each verb instance, we traverse over its an-

English
TO + VB. The constituent starts with "to" followed by a verb in infinitive form.
WP. The constituent is preceded by a Wh-pronoun.
That. The constituent is preceded by a "that" marked by an "IN" POS tag indicating that it is a subordinating conjunction.
Spanish
CQUE. The constituent is preceded by a word with the POS "CQUE" which denotes the word "que" as a conjunction.
INT. The constituent is preceded by a word with the POS "INT" which denotes an interrogative pronoun.
CSUB. The constituent is preceded by a word with one of the POSs "CSUBF", "CSUBI" or "CSUBX", which denote a subordinating conjunction.

Figure 2: The set of lexico-syntactic patterns that mark clauses which were used by our model.

cestors from top to bottom. For each of them we update the following counters: $sentence(ST)$ for the root ancestor's ST, $pattern_i(ST)$ for the ones complying to the i-th lexico-syntactic pattern and $negative(ST)$ for the other ancestors[1].

Clause detection. At test time, when detecting the minimal clause of a verb instance, we use the statistics collected in the previous stage. Denote the ancestors of the verb with $A_1 \dots A_m$. For each of them, we calculate $clause(ST_{A_j})$ and $total(ST_{A_j})$. $clause(ST_{A_j})$ is the sum of $sentence(ST_{A_j})$ and $pattern_i(ST_{A_j})$ if this ancestor complies to the i-th pattern (if there is no such pattern, $clause(ST_{A_j})$ is equal to $sentence(ST_{A_j})$). $total(ST_{A_j})$ is the sum of $clause(ST_{A_j})$ and $negative(ST_{A_j})$.

The selected ancestor is given by:

$$(1)\ A_{max} = argmax_{A_j} \frac{clause(ST_{A_j})}{total(ST_{A_j})}$$

An ST whose $total(ST)$ is less than a small threshold[2] is not considered a candidate to be the minimal clause, since its statistics may be unreliable. In case of a tie, we choose the lowest constituent that obtained the maximal score. If there is only one verb in the sentence[3] or if $clause(ST_{A_j}) = 0$ for every $1 \leq j \leq m$, we choose the top level constituent by default to be the minimal clause containing the verb. Otherwise, the minimal clause is defined to be the yield of the selected ancestor.

Argument identification. For each predicate in the corpus, its argument candidates are now defined to be the constituents contained in the minimal clause containing the predicate. However, these constituents may be (and are) nested within each other, violating a major restriction on SRL arguments. Hence we now prune our set, by keeping only the siblings of all of the verb's ancestors, as is common in supervised SRL (Xue and Palmer, 2004).

3.3 Using collocations

We use the following observation to filter out some superfluous argument candidates: since the arguments of a predicate many times bear a semantic connection with that predicate, they consequently tend to collocate with it.

We collect collocation statistics from a large corpus, which we annotate with parse trees and POS tags. We mark arguments using the argument detection algorithm described in the previous two sections, and extract all (predicate, argument) pairs appearing in the corpus. Recall that for each sentence, the arguments are a subset of the constituents in the parse tree.

We use two representations of an argument: one is the POS tag sequence of the terminals contained in the argument, the other is its head word[4]. The predicate is represented as the conjunction of its lemma with its POS tag.

Denote the number of times a predicate x appeared with an argument y by n_{xy}. Denote the total number of (predicate, argument) pairs by N. Using these notations, we define the following quantities: $n_x = \Sigma_y n_{xy}$, $n_y = \Sigma_x n_{xy}$, $p(x) = \frac{n_x}{N}$, $p(y) = \frac{n_y}{N}$ and $p(x,y) = \frac{n_{xy}}{N}$. The pointwise mutual information of x and y is then given by:

[1] If while traversing the tree, we encounter an ancestor whose first word is preceded by a coordinating conjunction (marked by the POS tag "CC"), we refrain from performing any additional counter updates. Structures containing coordinating conjunctions tend not to obey our lexico-syntactic rules.

[2] We used 4 per million sentences, derived from development data.

[3] In this case, every argument in the sentence must be related to that verb.

[4] Since we do not have syntactic labels, we use an approximate notion. For English we use the Bikel parser default head word rules (Bikel, 2004). For Spanish, we use the left-most word.

(2) $PMI(x,y) = log\frac{p(x,y)}{p(x)\cdot p(y)} = log\frac{n_{xy}}{(n_x \cdot n_y)/N}$

PMI effectively measures the ratio between the number of times x and y appeared together and the number of times they were expected to appear, had they been independent.

At test time, when an (x,y) pair is observed, we check if $PMI(x,y)$, computed on the large corpus, is lower than a threshold α for either of x's representations. If this holds, for at least one representation, we prune all instances of that (x,y) pair. The parameter α may be selected differently for each of the argument representations.

In order to avoid using unreliable statistics, we apply this for a given pair only if $\frac{n_x \cdot n_y}{N} > r$, for some parameter r. That is, we consider $PMI(x,y)$ to be reliable, only if the denominator in equation (2) is sufficiently large.

4 Experimental Setup

Corpora. We used the PropBank corpus for development and for evaluation on English. Section 24 was used for the development of our model, and sections 2 to 21 were used as our test data. The free parameters of the collocation extraction phase were tuned on the development data. Following the unsupervised parsing literature, multiple brackets and brackets covering a single word are omitted. We exclude punctuation according to the scheme of (Klein, 2005). As is customary in unsupervised parsing (e.g. (Seginer, 2007)), we bounded the lengths of the sentences in the corpus to be at most 10 (excluding punctuation). This results in 207 sentences in the development data, containing a total of 132 different verbs and 173 verb instances (of the non-auxiliary verbs in the SRL task, see 'evaluation' below) having 403 arguments. The test data has 6007 sentences containing 1008 different verbs and 5130 verb instances (as above) having 12436 arguments.

Our algorithm requires large amounts of data to gather argument structure and collocation patterns. For the statistics gathering phase of the clause detection algorithm, we used 4.5M sentences of the NANC (Graff, 1995) corpus, bounding their length in the same manner. In order to extract collocations, we used 2M sentences from the British National Corpus (Burnard, 2000) and about 29M sentences from the Dmoz corpus (Gabrilovich and Markovitch, 2005). Dmoz is a web corpus obtained by crawling and cleaning the URLs in the Open Directory Project (dmoz.org). All of the above corpora were parsed using Seginer's parser and POS-tagged by MXPOST (Ratnaparkhi, 1996).

For our experiments on Spanish, we used 3.3M sentences of length at most 15 (excluding punctuation) extracted from the Spanish Wikipedia. Here we chose to bound the length by 15 due to the smaller size of the available test corpus. The same data was used both for the first and the second stages. Our development and test data were taken from the training data released for the SemEval 2007 task on semantic annotation of Spanish (Màrquez et al., 2007). This data consisted of 1048 sentences of length up to 15, from which 200 were randomly selected as our development data and 848 as our test data. The development data included 313 verb instances while the test data included 1279. All corpora were parsed using the Seginer parser and tagged by the "TreeTagger" (Schmid, 1994).

Baselines. Since this is the first paper, to our knowledge, which addresses the problem of unsupervised argument identification, we do not have any previous results to compare to. We instead compare to a baseline which marks all k-th degree cousins of the predicate (for every k) as arguments (this is the second pruning we use in the clause detection stage). We name this baseline the ALL COUSINS baseline. We note that a random baseline would score very poorly since any sequence of terminals which does not contain the predicate is a possible candidate. Therefore, beating this random baseline is trivial.

Evaluation. Evaluation is carried out using standard SRL evaluation software[5]. The algorithm is provided with a list of predicates, whose arguments it needs to annotate. For the task addressed in this paper, non-consecutive parts of arguments are treated as full arguments. A match is considered each time an argument in the gold standard data matches a marked argument in our model's output. An unmatched argument is an argument which appears in the gold standard data, and fails to appear in our model's output, and an excessive argument is an argument which appears in our model's output but does not appear in the gold standard. Precision and recall are defined accordingly. We report an F-score as well (the harmonic mean of precision and recall). We do not attempt

[5] http://www.lsi.upc.edu/∼srlconll/soft.html#software.

to identify multi-word verbs, and therefore do not report the model's performance in identifying verb boundaries.

Since our model detects clauses as an intermediate product, we provide a separate evaluation of this task for the English corpus. We show results on our development data. We use the standard parsing F-score evaluation measure. As a gold standard in this evaluation, we mark for each of the verbs in our development data the minimal clause containing it. A minimal clause is the lowest ancestor of the verb in the parse tree that has a syntactic label of a clause according to the gold standard parse of the PTB. A verb is any terminal marked by one of the POS tags of type verb according to the gold standard POS tags of the PTB.

5 Results

Our results are shown in Table 1. The left section presents results on English and the right section presents results on Spanish. The top line lists results of the clause detection stage alone. The next two lines list results of the full algorithm (clause detection + collocations) in two different settings of the collocation stage. The bottom line presents the performance of the ALL COUSINS baseline.

In the "Collocation Maximum Precision" setting the parameters of the collocation stage (α and r) were generally tuned such that maximal precision is achieved while preserving a minimal recall level (40% for English, 20% for Spanish on the development data). In the "Collocation Maximum F-score" the collocation parameters were generally tuned such that the maximum possible F-score for the collocation algorithm is achieved.

The best or close to best F-score is achieved when using the clause detection algorithm alone (59.14% for English, 23.34% for Spanish). Note that for both English and Spanish F-score improvements are achieved via a precision improvement that is more significant than the recall degradation. F-score maximization would be the aim of a system that uses the output of our unsupervised ARGID by itself.

The "Collocation Maximum Precision" achieves the best precision level (55.97% for English, 21.8% for Spanish) but at the expense of the largest recall loss. Still, it maintains a reasonable level of recall. The "Collocation Maximum F-score" is an example of a model that provides a precision improvement (over both the baseline and the clause detection stage) with a relatively small recall degradation. In the Spanish experiments its F-score (23.87%) is even a bit higher than that of the clause detection stage (23.34%).

The full two–stage algorithm (clause detection + collocations) should thus be used when we intend to use the model's output as training data for supervised SRL engines or supervised ARGID algorithms.

In our algorithm, the initial set of potential arguments consists of constituents in the Seginer parser's parse tree. Consequently the fraction of arguments that are also constituents (81.87% for English and 51.83% for Spanish) poses an upper bound on our algorithm's recall. Note that the recall of the ALL COUSINS baseline is 74.27% (45.75%) for English (Spanish). This score emphasizes the baseline's strength, and justifies the restriction that the arguments should be k-th cousins of the predicate. The difference between these bounds for the two languages provides a partial explanation for the corresponding gap in the algorithm's performance.

Figure 3 shows the precision of the collocation model (on development data) as a function of the amount of data it was given. We can see that the algorithm reaches saturation at about 5M sentences. It achieves this precision while maintaining a reasonable recall (an average recall of 43.1% after saturation). The parameters of the collocation model were separately tuned for each corpus size, and the graph displays the maximum which was obtained for each of the corpus sizes.

To better understand our model's performance, we performed experiments on the English corpus to test how well its first stage detects clauses. Clause detection is used by our algorithm as a step towards argument identification, but it can be of potential benefit for other purposes as well (see Section 2). The results are 23.88% recall and 40% precision. As in the ARGID task, a random selection of arguments would have yielded an extremely poor result.

6 Conclusion

In this work we presented the first algorithm for argument identification that uses neither supervised syntactic annotation nor SRL tagged data. We have experimented on two languages: English and Spanish. The straightforward adaptability of un-

	English (Test Data)			Spanish (Test Data)		
	Precision	Recall	F1	Precision	Recall	F1
Clause Detection	52.84	67.14	**59.14**	18.00	33.19	23.34
Collocation Maximum F–score	54.11	63.53	58.44	20.22	29.13	**23.87**
Collocation Maximum Precision	**55.97**	40.02	46.67	**21.80**	18.47	20.00
ALL COUSINS baseline	46.71	74.27	57.35	14.16	45.75	21.62

Table 1: Precision, Recall and F1 score for the different stages of our algorithm. Results are given for English (PTB, sentences length bounded by 10, left part of the table) and Spanish (SemEval 2007 Spanish SRL task, right part of the table). The results of the collocation (second) stage are given in two configurations, Collocation Maximum F-score and Collocation Maximum Precision (see text). The upper bounds on Recall, obtained by taking all arguments output by our unsupervised parser, are 81.87% for English and 51.83% for Spanish.

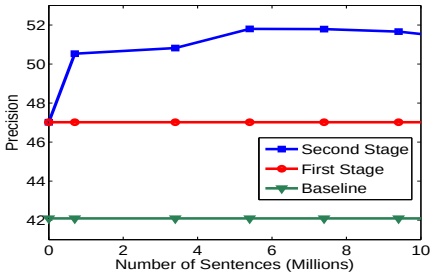

Figure 3: The performance of the second stage on English (squares) vs. corpus size. The precision of the baseline (triangles) and of the first stage (circles) is displayed for reference. The graph indicates the maximum precision obtained for each corpus size. The graph reaches saturation at about 5M sentences. The average recall of the sampled points from there on is 43.1%. Experiments were performed on the English development data.

supervised models to different languages is one of their most appealing characteristics. The recent availability of unsupervised syntactic parsers has offered an opportunity to conduct research on SRL, without reliance on supervised syntactic annotation. This work is the first to address the application of unsupervised parses to an SRL related task.

Our model displayed an increase in precision of 9% in English and 8% in Spanish over a strong baseline. Precision is of particular interest in this context, as instances tagged by high quality annotation could be later used as training data for supervised SRL algorithms. In terms of F–score, our model showed an increase of 1.8% in English and of 2.2% in Spanish over the baseline.

Although the quality of unsupervised parses is currently low (compared to that of supervised approaches), using great amounts of data in identifying recurring structures may reduce noise and in addition address sparsity. The techniques presented in this paper are based on this observation, using around 35M sentences in total for English and 3.3M sentences for Spanish.

As this is the first work which addressed unsupervised ARGID, many questions remain to be explored. Interesting issues to address include assessing the utility of the proposed methods when supervised parses are given, comparing our model to systems with no access to unsupervised parses and conducting evaluation using more relaxed measures.

Unsupervised methods for syntactic tasks have matured substantially in the last few years. Notable examples are (Clark, 2003) for unsupervised POS tagging and (Smith and Eisner, 2006) for unsupervised dependency parsing. Adapting our algorithm to use the output of these models, either to reduce the little supervision our algorithm requires (POS tagging) or to provide complementary syntactic information, is an interesting challenge for future work.

References

Collin F. Baker, Charles J. Fillmore and John B. Lowe, 1998. *The Berkeley FrameNet Project.* ACL-COLING '98.

Daniel M. Bikel, 2004. *Intricacies of Collins' Parsing Model.* Computational Linguistics, 30(4):479–511.

Ted Briscoe, John Carroll, 1997. *Automatic Extraction of Subcategorization from Corpora.* Applied NLP 1997.

Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Pad and Manfred Pinkal, 2006 *The SALSA Corpus: a German Corpus Resource for Lexical Semantics.* LREC '06.

Lou Burnard, 2000. *User Reference Guide for the British National Corpus.* Technical report, Oxford University.

Xavier Carreras and Lluìs Màrquez, 2004. *Introduction to the CoNLL–2004 Shared Task: Semantic Role Labeling.* CoNLL '04.

Xavier Carreras and Lluìs Màrquez, 2005. *Introduction to the CoNLL–2005 Shared Task: Semantic Role Labeling.* CoNLL '05.

Alexander Clark, 2003. *Combining Distributional and Morphological Information for Part of Speech Induction.* EACL '03.

Ronan Collobert and Jason Weston, 2007. *Fast Semantic Extraction Using a Novel Neural Network Architecture.* ACL '07.

Mona Diab, Aous Mansouri, Martha Palmer, Olga Babko-Malaya, Wajdi Zaghouani, Ann Bies and Mohammed Maamouri, 2008. *A pilot Arabic PropBank.* LREC '08.

Evgeniy Gabrilovich and Shaul Markovitch, 2005. *Feature Generation for Text Categorization using World Knowledge.* IJCAI '05.

Daniel Gildea and Daniel Jurafsky, 2002. *Automatic Labeling of Semantic Roles.* Computational Linguistics, 28(3):245–288.

Elliot Glaysher and Dan Moldovan, 2006. *Speeding Up Full Syntactic Parsing by Leveraging Partial Parsing Decisions.* COLING/ACL '06 poster session.

Andrew Gordon and Reid Swanson, 2007. *Generalizing Semantic Role Annotations across Syntactically Similar Verbs.* ACL '07.

David Graff, 1995. *North American News Text Corpus.* Linguistic Data Consortium. LDC95T21.

Trond Grenager and Christopher D. Manning, 2006. *Unsupervised Discovery of a Statistical Verb Lexicon.* EMNLP '06.

Kadri Hacioglu, 2004. *Semantic Role Labeling using Dependency Trees.* COLING '04.

Kadri Hacioglu and Wayne Ward, 2003. *Target Word Detection and Semantic Role Chunking using Support Vector Machines.* HLT-NAACL '03.

Rohit J. Kate and Raymond J. Mooney, 2007. *Semi-Supervised Learning for Semantic Parsing using Support Vector Machines.* HLT–NAACL '07.

Karin Kipper, Hoa Trang Dang and Martha Palmer, 2000. *Class-Based Construction of a Verb Lexicon.* AAAI '00.

Dan Klein, 2005. *The Unsupervised Learning of Natural Language Structure.* Ph.D. thesis, Stanford University.

Anna Korhonen, 2002. *Subcategorization Acquisition.* Ph.D. thesis, University of Cambridge.

Christopher D. Manning, 1993. *Automatic Acquisition of a Large Subcategorization Dictionary.* ACL '93.

Lluìs Màrquez, Xavier Carreras, Kenneth C. Litkowski and Suzanne Stevenson, 2008. *Semantic Role Labeling: An introdution to the Special Issue.* Computational Linguistics, 34(2):145–159

Lluìs Màrquez, Jesus Gimènez Pere Comas and Neus Català, 2005. *Semantic Role Labeling as Sequential Tagging.* CoNLL '05.

Lluìs Màrquez, Lluis Villarejo, M. A. Martì and Mariona Taulè, 2007. *SemEval–2007 Task 09: Multilevel Semantic Annotation of Catalan and Spanish.* The 4th international workshop on Semantic Evaluations (SemEval '07).

Gabriele Musillo and Paula Merlo, 2006. *Accurate Parsing of the proposition bank.* HLT-NAACL '06.

Martha Palmer, Daniel Gildea and Paul Kingsbury, 2005. *The Proposition Bank: A Corpus Annotated with Semantic Roles.* Computational Linguistics, 31(1):71–106.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin and Daniel Jurafsky, 2005. *Support Vector Learning for Semantic Argument Classification.* Machine Learning, 60(1):11–39.

Sameer Pradhan, Wayne Ward, James H. Martin, 2008. *Towards Robust Semantic Role Labeling.* Computational Linguistics, 34(2):289–310.

Adwait Ratnaparkhi, 1996. *Maximum Entropy Part-Of-Speech Tagger.* EMNLP '96.

Helmut Schmid, 1994. *Probabilistic Part-of-Speech Tagging Using Decision Trees* International Conference on New Methods in Language Processing.

Yoav Seginer, 2007. *Fast Unsupervised Incremental Parsing.* ACL '07.

Noah A. Smith and Jason Eisner, 2006. *Annealing Structural Bias in Multilingual Weighted Grammar Induction.* ACL '06.

Robert S. Swier and Suzanne Stevenson, 2004. *Unsupervised Semantic Role Labeling.* EMNLP '04.

Robert S. Swier and Suzanne Stevenson, 2005. *Exploiting a Verb Lexicon in Automatic Semantic Role Labelling.* EMNLP '05.

Erik F. Tjong Kim Sang and Hervé Déjean, 2001. *Introduction to the CoNLL-2001 Shared Task: Clause Identification.* CoNLL '01.

Nianwen Xue and Martha Palmer, 2004. *Calibrating Features for Semantic Role Labeling.* EMNLP '04.

Nianwen Xue, 2008. *Labeling Chinese Predicates with Semantic Roles.* Computational Linguistics, 34(2):225–255.

Published in ACL 2010

Chapter 5

Fully Unsupervised Core-Adjunct Argument Classification

Fully Unsupervised Core-Adjunct Argument Classification

Omri Abend*
Institute of Computer Science
The Hebrew University
omria01@cs.huji.ac.il

Ari Rappoport
Institute of Computer Science
The Hebrew University
arir@cs.huji.ac.il

Abstract

The core-adjunct argument distinction is a basic one in the theory of argument structure. The task of distinguishing between the two has strong relations to various basic NLP tasks such as syntactic parsing, semantic role labeling and subcategorization acquisition. This paper presents a novel unsupervised algorithm for the task that uses no supervised models, utilizing instead state-of-the-art syntactic induction algorithms. This is the first work to tackle this task in a fully unsupervised scenario.

1 Introduction

The distinction between core arguments (henceforth, cores) and adjuncts is included in most theories on argument structure (Dowty, 2000). The distinction can be viewed syntactically, as one between obligatory and optional arguments, or semantically, as one between arguments whose meanings are predicate dependent and independent. The latter (cores) are those whose function in the described event is to a large extent determined by the predicate, and are obligatory. Adjuncts are optional arguments which, like adverbs, modify the meaning of the described event in a predictable or predicate-independent manner.

Consider the following examples:

1. The surgeon operated [on his colleague].
2. Ron will drop by [after lunch].
3. Yuri played football [in the park].

The marked argument is a core in 1 and an adjunct in 2 and 3. Adjuncts form an independent semantic unit and their semantic role can often be inferred independently of the predicate (e.g., [after lunch] is usually a temporal modifier). Core roles are more predicate-specific, e.g., [on his colleague] has a different meaning with the verbs 'operate' and 'count'.

Sometimes the same argument plays a different role in different sentences. In (3), [in the park] places a well-defined situation (Yuri playing football) in a certain location. However, in "The troops are based [in the park]", the same argument is obligatory, since being based requires a place to be based in.

Distinguishing between the two argument types has been discussed extensively in various formulations in the NLP literature, notably in PP attachment, semantic role labeling (SRL) and subcategorization acquisition. However, no work has tackled it yet in a fully unsupervised scenario. Unsupervised models reduce reliance on the costly and error prone manual multi-layer annotation (POS tagging, parsing, core-adjunct tagging) commonly used for this task. They also allow to examine the nature of the distinction and to what extent it is accounted for in real data in a theory-independent manner.

In this paper we present a fully unsupervised algorithm for core-adjunct classification. We utilize leading fully unsupervised grammar induction and POS induction algorithms. We focus on prepositional arguments, since non-prepositional ones are generally cores. The algorithm uses three measures based on different characterizations of the core-adjunct distinction, and combines them using an ensemble method followed by self-training. The measures used are based on selectional preference, predicate-slot collocation and argument-slot collocation.

We evaluate against PropBank (Palmer et al., 2005), obtaining roughly 70% accuracy when evaluated on the prepositional arguments and more than 80% for the entire argument set. These results are substantially better than those obtained by a non-trivial baseline.

* Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

Section 2 discusses the core-adjunct distinction. Section 3 describes the algorithm. Sections 4 and 5 present our experimental setup and results.

2 Core-Adjunct in Previous Work

PropBank. PropBank (PB) (Palmer et al., 2005) is a widely used corpus, providing SRL annotation for the entire WSJ Penn Treebank. Its core labels are predicate specific, while adjunct (or modifiers under their terminology) labels are shared across predicates. The adjuncts are subcategorized into several classes, the most frequent of which are locative, temporal and manner[1].

The organization of PropBank is based on the notion of diathesis alternations, which are (roughly) defined to be alternations between two subcategorization frames that preserve meaning or change it systematically. The frames in which each verb appears were collected and sets of alternating frames were defined. Each such set was assumed to have a unique set of roles, named 'role-set'. These roles include all roles appearing in any of the frames, except of those defined as adjuncts.

Adjuncts are defined to be optional arguments appearing with a wide variety of verbs and frames. They can be viewed as fixed points with respect to alternations, i.e., as arguments that do not change their place or slot when the frame undergoes an alternation. This follows the notions of optionality and compositionality that define adjuncts.

Detecting diathesis alternations automatically is difficult (McCarthy, 2001), requiring an initial acquisition of a subcategorization lexicon. This alone is a challenging task tackled in the past using supervised parsers (see below).

FrameNet. FrameNet (FN) (Baker et al., 1998) is a large-scale lexicon based on frame semantics. It takes a different approach from PB to semantic roles. Like PB, it distinguishes between core and non-core arguments, but it does so for each and every frame separately. It does not commit that a semantic role is consistently tagged as a core or a non-core across frames. For example, the semantic role 'path' is considered core in the 'Self Motion' frame, but as non-core in the 'Placing' frame. Another difference is that FN does not allow any type of non-core argument to attach to a given frame. For instance, while the 'Getting' frame allows a 'Duration' non-core argument, the 'Active Perception' frame does not.

PB and FN tend to agree in clear (prototypical) cases, but to differ in others. For instance, both schemes would tag "Yuri played football [in the park]" as an adjunct and "The commander placed a guard [in the park]" as a core. However, in "He walked [into his office]", the marked argument is tagged as a directional adjunct in PB but as a 'Direction' core in FN.

Under both schemes, non-cores are usually confined to a few specific semantic domains, notably time, place and manner, in contrast to cores that are not restricted in their scope of applicability. This approach is quite common, e.g., the COBUILD English grammar (Willis, 2004) categorizes adjuncts to be of manner, aspect, opinion, place, time, frequency, duration, degree, extent, emphasis, focus and probability.

Semantic Role Labeling. Work in SRL does not tackle the core-adjunct task separately but as part of general argument classification. Supervised approaches obtain an almost perfect score in distinguishing between the two in an in-domain scenario. For instance, the confusion matrix in (Toutanova et al., 2008) indicates that their model scores 99.5% accuracy on this task. However, adaptation results are lower, with the best two models in the CoNLL 2005 shared task (Carreras and Màrquez, 2005) achieving 95.3% (Pradhan et al., 2008) and 95.6% (Punyakanok et al., 2008) accuracy in an adaptation between the relatively similar corpora WSJ and Brown.

Despite the high performance in supervised scenarios, tackling the task in an unsupervised manner is not easy. The success of supervised methods stems from the fact that the predicate-slot combination (slot is represented in this paper by its preposition) strongly determines whether a given argument is an adjunct or a core (see Section 3.4). Supervised models are provided with an annotated corpus from which they can easily learn the mapping between predicate-slot pairs and their core/adjunct label. However, induction of the mapping in an unsupervised manner must be based on inherent core-adjunct properties. In addition, supervised models utilize supervised parsers and POS taggers, while the current state-of-the-art in unsupervised parsing and POS tagging is considerably worse than their supervised counterparts.

This challenge has some resemblance to un-

[1] PropBank annotates modals and negation words as modifiers. Since these are not arguments in the common usage of the term, we exclude them from the discussion in this paper.

supervised detection of multiword expressions (MWEs). An important MWE sub-class is that of phrasal verbs, which are also characterized by verb-preposition pairs (Li et al., 2003; Sporleder and Li, 2009) (see also (Boukobza and Rappoport, 2009)). Both tasks aim to determine semantic compositionality, which is a highly challenging task.

Few works addressed unsupervised SRL-related tasks. The setup of (Grenager and Manning, 2006), who presented a Bayesian Network model for argument classification, is perhaps closest to ours. Their work relied on a supervised parser and a rule-based argument identification (both during training and testing). Swier and Stevenson (2004, 2005), while addressing an unsupervised SRL task, greatly differ from us as their algorithm uses the VerbNet (Kipper et al., 2000) verb lexicon, in addition to supervised parses. Finally, Abend et al. (2009) tackled the argument identification task alone and did not perform argument classification of any sort.

PP attachment. PP attachment is the task of determining whether a prepositional phrase which immediately follows a noun phrase attaches to the latter or to the preceding verb. This task's relation to the core-adjunct distinction was addressed in several works. For instance, the results of (Hindle and Rooth, 1993) indicate that their PP attachment system works better for cores than for adjuncts.

Merlo and Esteve Ferrer (2006) suggest a system that jointly tackles the PP attachment and the core-adjunct distinction tasks. Unlike in this work, their classifier requires extensive supervision including WordNet, language-specific features and a supervised parser. Their features are generally motivated by common linguistic considerations. Features found adaptable to a completely unsupervised scenario are used in this work as well.

Syntactic Parsing. The core-adjunct distinction is included in many syntactic annotation schemes. Although the Penn Treebank does not explicitly annotate adjuncts and cores, a few works suggested mapping its annotation (including function tags) to core-adjunct labels. Such a mapping was presented in (Collins, 1999). In his Model 2, Collins modifies his parser to provide a core-adjunct prediction, thereby improving its performance.

The Combinatory Categorial Grammar (CCG) formulation models the core-adjunct distinction explicitly. Therefore, any CCG parser can be used as a core-adjunct classifier (Hockenmaier, 2003).

Subcategorization Acquisition. This task specifies for each predicate the number, type and order of obligatory arguments. Determining the allowable subcategorization frames for a given predicate necessarily involves separating its cores from its allowable adjuncts (which are not framed). Notable works in the field include (Briscoe and Carroll, 1997; Sarkar and Zeman, 2000; Korhonen, 2002). All these works used a parsed corpus in order to collect, for each predicate, a set of hypothesized subcategorization frames, to be filtered by hypothesis testing methods.

This line of work differs from ours in a few aspects. First, all works use manual or supervised syntactic annotations, usually including a POS tagger. Second, the common approach to the task focuses on syntax and tries to identify the entire frame, rather than to tag each argument separately. Finally, most works address the task at the verb type level, trying to detect the allowable frames for each type. Consequently, the common evaluation focuses on the quality of the allowable frames acquired for each verb type, and not on the classification of specific arguments in a given corpus. Such a token level evaluation was conducted in a few works (Briscoe and Carroll, 1997; Sarkar and Zeman, 2000), but often with a small number of verbs or a small number of frames. A discussion of the differences between type and token level evaluation can be found in (Reichart et al., 2010).

The core-adjunct distinction task was tackled in the context of child language acquisition. Villavicencio (2002) developed a classifier based on preposition selection and frequency information for modeling the distinction for locative prepositional phrases. Her approach is not entirely corpus based, as it assumes the input sentences are given in a basic logical form.

The study of prepositions is a vibrant research area in NLP. A special issue of *Computational Linguistics*, which includes an extensive survey of related work, was recently devoted to the field (Baldwin et al., 2009).

3 Algorithm

We are given a (predicate, argument) pair in a test sentence, and we need to determine whether the argument is a core or an adjunct. Test arguments are assumed to be correctly bracketed. We are allowed to utilize a training corpus of raw text.

3.1 Overview

Our algorithm utilizes statistics based on the (predicate, slot, argument head) (PSH) joint distribution (a slot is represented by its preposition). To estimate this joint distribution, PSH samples are extracted from the training corpus using unsupervised POS taggers (Clark, 2003; Abend et al., 2010) and an unsupervised parser (Seginer, 2007). As current performance of unsupervised parsers for long sentences is low, we use only short sentences (up to 10 words, excluding punctuation). The length of test sentences is not bounded. Our results will show that the training data accounts well for the argument realization phenomena in the test set, despite the length bound on its sentences. The sample extraction process is detailed in Section 3.2.

Our approach makes use of both aspects of the distinction – obligatoriness and compositionality. We define three measures, one quantifying the obligatoriness of the slot, another quantifying the selectional preference of the verb to the argument and a third that quantifies the association between the head word and the slot irrespective of the predicate (Section 3.3).

The measures' predictions are expected to coincide in clear cases, but may be less successful in others. Therefore, an ensemble-based method is used to combine the three measures into a single classifier. This results in a high accuracy classifier with relatively low coverage. A self-training step is now performed to increase coverage with only a minor deterioration in accuracy (Section 3.4).

We focus on prepositional arguments. Non-prepositional arguments in English tend to be cores (e.g., in more than 85% of the cases in PB sections 2–21), while prepositional arguments tend to be equally divided between cores and adjuncts. The difficulty of the task thus lies in the classification of prepositional arguments.

3.2 Data Collection

The statistical measures used by our classifier are based on the (predicate, slot, argument head) (PSH) joint distribution. This section details the process of extracting samples from this joint distribution given a raw text corpus.

We start by parsing the corpus using the Seginer parser (Seginer, 2007). This parser is unique in its ability to induce a bracketing (unlabeled parsing) from raw text (without even using POS tags) with strong results. Its high speed (thousands of words per second) allows us to use millions of sentences, a prohibitive number for other parsers.

We continue by tagging the corpus using Clark's unsupervised POS tagger (Clark, 2003) and the unsupervised Prototype Tagger (Abend et al., 2010)[2]. The classes corresponding to prepositions and to verbs are manually selected from the induced clusters[3]. A preposition is defined to be any word which is the first word of an argument and belongs to a prepositions cluster. A verb is any word belonging to a verb cluster. This manual selection requires only a minute, since the number of classes is very small (34 in our experiments). In addition, knowing what is considered a preposition is part of the task definition itself.

Argument identification is hard even for supervised models and is considerably more so for unsupervised ones (Abend et al., 2009). We therefore confine ourselves to sentences of length not greater than 10 (excluding punctuation) which contain a single verb. A sequence of words will be marked as an argument of the verb if it is a constituent that does not contain the verb (according to the unsupervised parse tree), whose parent is an ancestor of the verb. This follows the pruning heuristic of (Xue and Palmer, 2004) often used by SRL algorithms.

The corpus is now tagged using an unsupervised POS tagger. Since the sentences in question are short, we consider every word which does not belong to a closed class cluster as a head word (an argument can have several head words). A closed class is a class of function words with relatively few word types, each of which is very frequent. Typical examples include determiners, prepositions and conjunctions. A class which is not closed is open. In this paper, we define closed classes to be clusters in which the ratio between the number of word tokens and the number of word types ex-

[2] Clark's tagger was replaced by the Prototype Tagger where the latter gave a significant improvement. See Section 4.

[3] We also explore a scenario in which they are identified by a supervised tagger. See Section 4.

ceeds a threshold T[4].

Using these annotation layers, we traverse the corpus and extract every (predicate, slot, argument head) triplet. In case an argument has several head words, each of them is considered as an independent sample. We denote the number of times that a triplet occurred in the training corpus by $N(p, s, h)$.

3.3 Collocation Measures

In this section we present the three types of measures used by the algorithm and the rationale behind each of them. These measures are all based on the PSH joint distribution.

Given a (predicate, prepositional argument) pair from the test set, we first tag and parse the argument using the unsupervised tools above[5]. Each word in the argument is now represented by its word form (without lemmatization), its unsupervised POS tag and its depth in the parse tree of the argument. The last two will be used to determine which are the head words of the argument (see below). The head words themselves, once chosen, are represented by the lemma. We now compute the following measures.

Selectional Preference (SP). Since the semantics of cores is more predicate dependent than the semantics of adjuncts, we expect arguments for which the predicate has a strong preference (in a specific slot) to be cores.

Selectional preference induction is a well-established task in NLP. It aims to quantify the likelihood that a certain argument appears in a certain slot of a predicate. Several methods have been suggested (Resnik, 1996; Li and Abe, 1998; Schulte im Walde et al., 2008).

We use the paradigm of (Erk, 2007). For a given predicate slot pair (p, s), we define its preference to the argument head h to be:

$$SP(p, s, h) = \sum_{h' \in Heads} Pr(h'|p, s) \cdot sim(h, h')$$

$$Pr(h|p, s) = \frac{N(p, s, h)}{\Sigma_{h'} N(p, s, h')}$$

$sim(h, h')$ is a similarity measure between argument heads. $Heads$ is the set of all head words. This is a natural extension of the naive (and sparse) maximum likelihood estimator $Pr(h|p, s)$, which is obtained by taking $sim(h, h')$ to be 1 if $h = h'$ and 0 otherwise.

The similarity measure we use is based on the slot distributions of the arguments. That is, two arguments are considered similar if they tend to appear in the same slots. Each head word h is assigned a vector where each coordinate corresponds to a slot s. The value of the coordinate is the number of times h appeared in s, i.e. $\Sigma_{p'} N(p', s, h)$ (p' is summed over all predicates). The similarity measure between two head words is then defined as the cosine measure of their vectors.

Since arguments in the test set can be quite long, not every open class word in the argument is taken to be a head word. Instead, only those appearing in the top level (depth = 1) of the argument under its unsupervised parse tree are taken. In case there are no such open class words, we take those appearing in depth 2. The selectional preference of the whole argument is then defined to be the arithmetic mean of this measure over all of its head words. If the argument has no head words under this definition or if none of the head words appeared in the training corpus, the selectional preference is undefined.

Predicate-Slot Collocation. Since cores are obligatory, when a predicate persistently appears with an argument in a certain slot, the arguments in this slot tends to be cores. This notion can be captured by the $(predicate, slot)$ joint distribution. We use the Pointwise Mutual Information measure (PMI) to capture the slot and the predicate's collocation tendency. Let p be a predicate and s a slot, then:

$$PS(p, s) = PMI(p, s) = log \frac{Pr(p, s)}{Pr(s) \cdot Pr(p)} =$$

$$= log \frac{N(p, s) \Sigma_{p', s'} N(p', s')}{\Sigma_{s'} N(p, s') \Sigma_{p'} N(p', s)}$$

Since there is only a meager number of possible slots (that is, of prepositions), estimating the $(predicate, slot)$ distribution can be made by the maximum likelihood estimator with manageable sparsity.

In order not to bias the counts towards predicates which tend to take more arguments, we define here $N(p, s)$ to be the number of times the (p, s) pair occurred in the training corpus, irrespective of the number of head words the argument had (and not e.g., $\Sigma_h N(p, s, h)$). Argu-

[4] We use sections 2–21 of the PTB WSJ for these counts, containing 0.95M words. Our T was set to 50.

[5] Note that while current unsupervised parsers have low performance on long sentences, arguments, even in long sentences, are usually still short enough for them to operate well. Their average length in the test set is 5.1 words.

ments with no prepositions are included in these counts as well (with $s = NULL$), so not to bias against predicates which tend to have less non-prepositional arguments.

Argument-Slot Collocation. Adjuncts tend to belong to one of a few specific semantic domains (see Section 2). Therefore, if an argument tends to appear in a certain slot in many of its instances, it is an indication that this argument tends to have a consistent semantic flavor in most of its instances. In this case, the argument and the preposition can be viewed as forming a unit on their own, independent of the predicate with which they appear. We therefore expect such arguments to be adjuncts.

We formalize this notion using the following measure. Let p, s, h be a predicate, a slot and a head word respectively. We then use[6]:

$$AS(s,h) = 1 - Pr(s|h) = 1 - \frac{\Sigma_{p'} N(p', s, h)}{\Sigma_{p',s'} N(p', s', h)}$$

We select the head words of the argument as we did with the selectional preference measure. Again, the AS of the whole argument is defined to be the arithmetic mean of the measure over all of its head words.

Thresholding. In order to turn these measures into classifiers, we set a threshold below which arguments are marked as adjuncts and above which as cores. In order to avoid tuning a parameter for each of the measures, we set the threshold as the median value of this measure in the test set. That is, we find the threshold which tags half of the arguments as cores and half as adjuncts. This relies on the prior knowledge that prepositional arguments are roughly equally divided between cores and adjuncts[7].

3.4 Combination Model

The algorithm proceeds to integrate the predictions of the weak classifiers into a single classifier. We use an ensemble method (Breiman, 1996). Each of the classifiers may either classify an argument as an adjunct, classify it as a core, or abstain. In order to obtain a high accuracy classifier, to be used for self-training below, the ensemble classifier only tags arguments for which none of the classifiers abstained, i.e., when sufficient information was available to make all three predictions. The prediction is determined by the majority vote.

The ensemble classifier has high precision but low coverage. In order to increase its coverage, a self-training step is performed. We observe that a predicate and a slot generally determine whether the argument is a core or an adjunct. For instance, in our development data, a classifier which assigns all arguments that share a predicate and a slot their most common label, yields 94.3% accuracy on the pairs appearing at least 5 times. This property of the core-adjunct distinction greatly simplifies the task for supervised algorithms (see Section 2).

We therefore apply the following procedure: (1) tag the training data with the ensemble classifier; (2) for each test sample x, if more than a ratio of α of the training samples sharing the same predicate and slot with x are labeled as cores, tag x as core. Otherwise, tag x as adjunct.

Test samples which do not share a predicate and a slot with any training sample are considered out of coverage. The parameter α is chosen so half of the arguments are tagged as cores and half as adjuncts. In our experiments α was about 0.25.

4 Experimental Setup

Experiments were conducted in two scenarios. In the '*SID*' (supervised identification of prepositions and verbs) scenario, a gold standard list of prepositions was provided. The list was generated by taking every word tagged by the preposition tag ('*IN*') in at least one of its instances under the gold standard annotation of the WSJ sections 2–21. Verbs were identified using MXPOST (Ratnaparkhi, 1996). Words tagged with any of the verb tags, except of the auxiliary verbs ('have', 'be' and 'do') were considered predicates. This scenario decouples the accuracy of the algorithm from the quality of the unsupervised POS tagging.

In the '*Fully Unsupervised*' scenario, prepositions and verbs were identified using Clark's tagger (Clark, 2003). It was asked to produce a tagging into 34 classes. The classes corresponding to prepositions and to verbs were manually identified. Prepositions in the test set were detected with 84.2% precision and 91.6% recall.

The prediction of whether a word belongs to an open class or a closed was based on the output of the Prototype tagger (Abend et al., 2010). The Prototype tagger provided significantly more ac-

[6] The conditional probability is subtracted from 1 so that higher values correspond to cores, as with the other measures.

[7] In case the test data is small, we can use the median value on the training data instead.

curate predictions in this context than Clark's.

The 39832 sentences of PropBank's sections 2–21 were used as a test set without bounding their lengths[8]. Cores were defined to be any argument bearing the labels 'A0' – 'A5', 'C-A0' – 'C-A5' or 'R-A0' – 'R-A5'. Adjuncts were defined to be arguments bearing the labels 'AM', 'C-AM' or 'R-AM'. Modals ('AM-MOD') and negation modifiers ('AM-NEG') were omitted since they do not represent adjuncts.

The test set includes 213473 arguments, 45939 (21.5%) are prepositional. Of the latter, 22442 (48.9%) are cores and 23497 (51.1%) are adjuncts. The non-prepositional arguments include 145767 (87%) cores and 21767 (13%) adjuncts. The average number of words per argument is 5.1.

The NANC (Graff, 1995) corpus was used as a training set. Only sentences of length not greater than 10 excluding punctuation were used (see Section 3.2), totaling 4955181 sentences. 7673878 (5635810) arguments were identified in the '*SID*' ('*Fully Unsupervised*') scenario. The average number of words per argument is 1.6 (1.7).

Since this is the first work to tackle this task using neither manual nor supervised syntactic annotation, there is no previous work to compare to. However, we do compare against a non-trivial baseline, which closely follows the rationale of cores as obligatory arguments.

Our *Window Baseline* tags a corpus using MXPOST and computes, for each predicate and preposition, the ratio between the number of times that the preposition appeared in a window of W words after the verb and the total number of times that the verb appeared. If this number exceeds a certain threshold β, all arguments having that predicate and preposition are tagged as cores. Otherwise, they are tagged as adjuncts. We used 18.7M sentences from NANC of unbounded length for this baseline. W and β were fine-tuned against the test set[9].

We also report results for partial versions of the algorithm, starting with the three measures used (selectional preference, predicate-slot collocation and argument-slot collocation). Results for the ensemble classifier (prior to the bootstrapping stage) are presented in two variants: one in which the ensemble is used to tag arguments for which all three measures give a prediction (the '*Ensemble(Intersection)*' classifier) and one in which the ensemble tags all arguments for which at least one classifier gives a prediction (the '*Ensemble(Union)*' classifier). For the latter, a tie is broken in favor of the core label. The '*Ensemble(Union)*' classifier is not a part of our model and is evaluated only as a reference.

In order to provide a broader perspective on the task, we compare the measures in the basis of our algorithm to simplified or alternative measures. We experiment with the following measures:

1. *Simple SP* – a selectional preference measure defined to be $Pr(head|slot, predicate)$.

2. *Vast Corpus SP* – similar to '*Simple SP*' but with a much larger corpus. It uses roughly 100M arguments which were extracted from the web-crawling based corpus of (Gabrilovich and Markovitch, 2005) and the British National Corpus (Burnard, 2000).

3. *Thesaurus SP* – a selectional preference measure which follows the paradigm of (Erk, 2007) (Section 3.3) and defines the similarity between two heads to be the Jaccard affinity between their two entries in Lin's automatically compiled thesaurus (Lin, 1998)[10].

4. *Pr(slot|predicate)* – an alternative to the used predicate-slot collocation measure.

5. *PMI(slot, head)* – an alternative to the used argument-slot collocation measure.

6. *Head Dependence* – the entropy of the predicate distribution given the slot and the head (following (Merlo and Esteve Ferrer, 2006)):

$$HD(s,h) = -\Sigma_p Pr(p|s,h) \cdot log(Pr(p|s,h))$$

Low entropy implies a core.

For each of the scenarios and the algorithms, we report accuracy, coverage and effective accuracy. Effective accuracy is defined to be the accuracy obtained when all out of coverage arguments are tagged as adjuncts. This procedure always yields a classifier with 100% coverage and therefore provides an even ground for comparing the algorithms' performance.

We see accuracy as important on its own right since increasing coverage is often straightforward given easily obtainable larger training corpora.

[8]The first 15K arguments were used for the algorithm's development and therefore excluded from the evaluation.

[9]Their optimal value was found to be W=2, β=0.03. The low optimal value of β is an indication of the noisiness of this technique.

[10]Since we aim for a minimally supervised scenario, we used the proximity-based version of his thesaurus which does not require parsing as pre-processing. http://webdocs.cs.ualberta.ca/∼lindek/Downloads/sims.lsp.gz

		Collocation Measures				Ensemble + Cov.	
		Sel. Preference	Pred-Slot	Arg-Slot	Ensemble(I)	Ensemble(U)	E(I) + ST
SID Scenario	Accuracy	65.6	64.5	72.4	**74.1**	68.7	70.6
	Coverage	35.6	77.8	44.7	33.2	**88.1**	74.2
	Eff. Acc.	56.7	64.8	58.8	58.8	67.8	**68.4**
Fully Unsupervised Scenario	Accuracy	62.6	61.1	69.4	**70.6**	64.8	68.8
	Coverage	24.8	59.0	38.7	22.8	**74.2**	56.9
	Eff. Acc.	52.6	57.5	55.8	53.8	61.0	**61.4**

Table 1: Results for the various models. Accuracy, coverage and effective accuracy are presented in percents. Effective accuracy is defined to be the accuracy resulting from labeling each out of coverage argument with an adjunct label. The rows represent the following models (left to right): selectional preference, predicate-slot collocation, argument-slot collocation, '*Ensemble(Intersection)*', '*Ensemble(Union)*' and the '*Ensemble(Intersection)*' followed by self-training (see Section 3.4). '*Ensemble(Intersection)*' obtains the highest accuracy. The ensemble + self-training obtains the highest effective accuracy.

	Selectional Preference Measures				Pred-Slot Measures			Arg-Slot Measures		
	SP*	S. SP	V.C. SP	Lin SP	PS*	Pr(s|p)	Window	AS*	PMI(s, h)	HD
Acc.	**65.6**	41.6	44.8	49.9	**64.5**	58.9	64.1	**72.4**	67.5	67.4
Cov.	35.6	36.9	**45.3**	36.7	77.8	77.8	**92.6**	44.7	44.7	44.7
Eff. Acc.	**56.7**	48.2	47.7	51.3	64.8	60.5	**65.0**	**58.8**	56.6	56.6

Table 2: Comparison of the measures used by our model to alternative measures in the '*SID*' scenario. Results are in percents. The sections of the table are (from left to right): selectional preference measures, predicate-slot measures, argument-slot measures and head dependence. The measures are (left to right): SP*, Simple SP, Vast Corpus SP, Lin SP, PS*, Pr(slot|predicate), Window Baseline, AS*, PMI(slot, head) and Head Dependence. The measures marked with $*$ are the ones used by our model. See Section 4.

Another reason is that a high accuracy classifier may provide training data to be used by subsequent supervised algorithms.

For completeness, we also provide results for the entire set of arguments. The great majority of non-prepositional arguments are cores (87% in the test set). We therefore tag all non-prepositional as cores and tag prepositional arguments using our model. In order to minimize supervision, we distinguish between the prepositional and the non-prepositional arguments using Clark's tagger.

Finally, we experiment on a scenario where even argument identification on the test set is not provided, but performed by the algorithm of (Abend et al., 2009), which uses neither syntactic nor SRL annotation but does utilize a supervised POS tagger. We therefore run it in the '*SID*' scenario. We apply it to the sentences of length at most 10 contained in sections 2–21 of PB (11586 arguments in 6007 sentences). Non-prepositional arguments are invariably tagged as cores and out of coverage prepositional arguments as adjuncts.

We report labeled and unlabeled recall, precision and F-scores for this experiment. An unlabeled match is defined to be an argument that agrees in its boundaries with a gold standard argument and a labeled match requires in addition that the arguments agree in their core/adjunct label. We also report labeling accuracy which is the ratio between the number of labeled matches and the number of unlabeled matches[11].

5 Results

Table 1 presents the results of our main experiments. In both scenarios, the most accurate of the three basic classifiers was the argument-slot collocation classifier. This is an indication that the collocation between the argument and the preposition is more indicative of the core/adjunct label than the obligatoriness of the slot (as expressed by the predicate-slot collocation).

Indeed, we can find examples where adjuncts, although optional, appear very often with a certain verb. An example is 'meet', which often takes a temporal adjunct, as in 'Let's meet [in July]'. This is a semantic property of 'meet', whose syntactic expression is not obligatory.

All measures suffered from a comparable deterioration of accuracy when moving from the '*SID*' to the '*Fully Unsupervised*' scenario. The deterioration in coverage, however, was considerably lower for the argument-slot collocation.

The '*Ensemble(Intersection)*' model in both cases is more accurate than each of the basic classifiers alone. This is to be expected as it combines the predictions of all three. The self-training step significantly increases the ensemble model's cov-

[11] Note that the reported unlabeled scores are slightly lower than those reported in the 2009 paper, due to the exclusion of the modals and negation modifiers.

	Precision	Recall	F-score	lAcc.
Unlabeled	50.7	66.3	57.5	–
Labeled	42.4	55.4	48.0	83.6

Table 3: Unlabeled and labeled scores for the experiments using the unsupervised argument identification system of (Abend et al., 2009). Precision, recall, F-score and labeling accuracy are given in percents.

erage (with some loss in accuracy), thus obtaining the highest effective accuracy. It is also more accurate than the simpler classifier '*Ensemble(Union)*' (although the latter's coverage is higher).

Table 2 presents results for the comparison to simpler or alternative measures. Results indicate that the three measures used by our algorithm (leftmost column in each section) obtain superior results. The only case in which performance is comparable is the window baseline compared to the Pred-Slot measure. However, the baseline's score was obtained by using a much larger corpus and a careful hand-tuning of the parameters[12].

The poor performance of *Simple SP* can be ascribed to sparsity. This is demonstrated by the median value of 0, which this measure obtained on the test set. Accuracy is only somewhat better with a much larger corpus (*Vast Corpus SP*). The *Thesaurus SP* most probably failed due to insufficient coverage, despite its applicability in a similar supervised task (Zapirain et al., 2009).

The Head Dependence measure achieves a relatively high accuracy of 67.4%. We therefore attempted to incorporate it into our model, but failed to achieve a significant improvement to the overall result. We expect a further study of the relations between the measures will suggest better ways of combining their predictions.

The obtained effective accuracy for the entire set of arguments, where the prepositional arguments are automatically identified, was 81.6%.

Table 3 presents results of our experiments with the unsupervised argument identification model of (Abend et al., 2009). The unlabeled scores reflect performance on argument identification alone, while the labeled scores reflect the joint performance of both the 2009 and our algorithms. These results, albeit low, are potentially beneficial for unsupervised subcategorization acquisition. The accuracy of our model on the entire set (prepositional argument subset) of correctly identified arguments was 83.6% (71.7%). This is somewhat higher than the score on the entire test set ('*SID*' scenario), which was 83.0% (68.4%), probably due to the bounded length of the test sentences in this case.

[12]We tried about 150 parameter pairs for the baseline. The average of the five best effective accuracies was 64.3%.

6 Conclusion

We presented a fully unsupervised algorithm for the classification of arguments into cores and adjuncts. Since most non-prepositional arguments are cores, we focused on prepositional arguments, which are roughly equally divided between cores and adjuncts. The algorithm computes three statistical measures and utilizes ensemble-based and self-training methods to combine their predictions.

The algorithm applies state-of-the-art unsupervised parser and POS tagger to collect statistics from a large raw text corpus. It obtains an accuracy of roughly 70%. We also show that (somewhat surprisingly) an argument-slot collocation measure gives more accurate predictions than a predicate-slot collocation measure on this task. We speculate the reason is that the head word disambiguates the preposition and that this disambiguation generally determines whether a prepositional argument is a core or an adjunct (somewhat independently of the predicate). This calls for a future study into the semantics of prepositions and their relation to the core-adjunct distinction. In this context two recent projects, *The Preposition Project* (Litkowski and Hargraves, 2005) and *PrepNet* (Saint-Dizier, 2006), which attempt to characterize and categorize the complex syntactic and semantic behavior of prepositions, may be of relevance.

It is our hope that this work will provide a better understanding of core-adjunct phenomena. Current supervised SRL models tend to perform worse on adjuncts than on cores (Pradhan et al., 2008; Toutanova et al., 2008). We believe a better understanding of the differences between cores and adjuncts may contribute to the development of better SRL techniques, in both its supervised and unsupervised variants.

References

Omri Abend, Roi Reichart and Ari Rappoport, 2009. *Unsupervised Argument Identification for Semantic Role Labeling.* ACL '09.

Omri Abend, Roi Reichart and Ari Rappoport, 2010. *Improved Unsupervised POS Induction through Prototype Discovery.* ACL '10.

Collin F. Baker, Charles J. Fillmore and John B. Lowe, 1998. *The Berkeley FrameNet Project.* ACL-COLING '98.

Timothy Baldwin, Valia Kordoni and Aline Villavicencio, 2009. *Prepositions in Applications: A Survey and Introduction to the Special Issue.* Computational Linguistics, 35(2):119–147.

Ram Boukobza and Ari Rappoport, 2009. *Multi-Word Expression Identification Using Sentence Surface Features.* EMNLP '09.

Leo Breiman, 1996. *Bagging Predictors.* Machine Learning, 24(2):123–140.

Ted Briscoe and John Carroll, 1997. *Automatic Extraction of Subcategorization from Corpora.* Applied NLP '97.

Lou Burnard, 2000. *User Reference Guide for the British National Corpus.* Technical report, Oxford University.

Xavier Carreras and Lluìs Màrquez, 2005. *Introduction to the CoNLL–2005 Shared Task: Semantic Role Labeling.* CoNLL '05.

Alexander Clark, 2003. *Combining Distributional and Morphological Information for Part of Speech Induction.* EACL '03.

Michael Collins, 1999. *Head-driven statistical models for natural language parsing.* Ph.D. thesis, University of Pennsylvania.

David Dowty, 2000. *The Dual Analysis of Adjuncts and Complements in Categorial Grammar.* Modifying Adjuncts, ed. Lang, Maienborn and Fabricius–Hansen, de Gruyter, 2003.

Katrin Erk, 2007. *A Simple, Similarity-based Model for Selectional Preferences.* ACL '07.

Evgeniy Gabrilovich and Shaul Markovitch, 2005. *Feature Generation for Text Categorization using World Knowledge.* IJCAI '05.

David Graff, 1995. *North American News Text Corpus.* Linguistic Data Consortium. LDC95T21.

Trond Grenager and Christopher D. Manning, 2006. *Unsupervised Discovery of a Statistical Verb Lexicon.* EMNLP '06.

Donald Hindle and Mats Rooth, 1993. *Structural Ambiguity and Lexical Relations.* Computational Linguistics, 19(1):103–120.

Julia Hockenmaier, 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar.* Ph.D. thesis, University of Edinburgh.

Karin Kipper, Hoa Trang Dang and Martha Palmer, 2000. *Class-Based Construction of a Verb Lexicon.* AAAI '00.

Anna Korhonen, 2002. *Subcategorization Acquisition.* Ph.D. thesis, University of Cambridge.

Hang Li and Naoki Abe, 1998. *Generalizing Case Frames using a Thesaurus and the MDL Principle.* Computational Linguistics, 24(2):217–244.

Wei Li, Xiuhong Zhang, Cheng Niu, Yuankai Jiang and Rohini Srihari, 2003. *An Expert Lexicon Approach to Identifying English Phrasal Verbs.* ACL '03.

Dekang Lin, 1998. *Automatic Retrieval and Clustering of Similar Words.* COLING–ACL '98.

Ken Litkowski and Orin Hargraves, 2005. *The Preposition Project.* ACL-SIGSEM Workshop on "The Linguistic Dimensions of Prepositions and Their Use in Computational Linguistic Formalisms and Applications".

Diana McCarthy, 2001. *Lexical Acquisition at the Syntax-Semantics Interface: Diathesis Alternations, Subcategorization Frames and Selectional Preferences.* Ph.D. thesis, University of Sussex.

Paula Merlo and Eva Esteve Ferrer, 2006. *The Notion of Argument in Prepositional Phrase Attachment.* Computational Linguistics, 32(3):341–377.

Martha Palmer, Daniel Gildea and Paul Kingsbury, 2005. *The Proposition Bank: A Corpus Annotated with Semantic Roles.* Computational Linguistics, 31(1):71–106.

Sameer Pradhan, Wayne Ward and James H. Martin, 2008. *Towards Robust Semantic Role Labeling.* Computational Linguistics, 34(2):289–310.

Vasin Punyakanok, Dan Roth and Wen-tau Yih, 2008. *The Importance of Syntactic Parsing and Inference in Semantic Role Labeling.* Computational Linguistics, 34(2):257–287.

Adwait Ratnaparkhi, 1996. *Maximum Entropy Part-Of-Speech Tagger.* EMNLP '96.

Roi Reichart, Omri Abend and Ari Rappoport, 2010. *Type Level Clustering Evaluation: New Measures and a POS Induction Case Study.* CoNLL '10.

Philip Resnik, 1996. *Selectional constraints: An information-theoretic model and its computational realization.* Cognition, 61:127–159.

Patrick Saint-Dizier, 2006. *PrepNet: A Multilingual Lexical Description of Prepositions.* LREC '06.

Anoop Sarkar and Daniel Zeman, 2000. *Automatic Extraction of Subcategorization Frames for Czech.* COLING '00.

Sabine Schulte im Walde, Christian Hying, Christian Scheible and Helmut Schmid, 2008. *Combining EM Training and the MDL Principle for an Automatic Verb Classification Incorporating Selectional Preferences.* ACL '08.

Published in IWCS 2013

Chapter 6

UCCA: A Semantics-based Grammatical Annotation Scheme

UCCA: A Semantics-based Grammatical Annotation Scheme

Omri Abend* and Ari Rappoport
Institute of Computer Science
Hebrew University of Jerusalem
{omria01|arir}@cs.huji.ac.il

Abstract

Syntactic annotation is an indispensable input for many semantic NLP applications. For instance, Semantic Role Labelling algorithms almost invariably apply some form of syntactic parsing as preprocessing. The categories used for syntactic annotation in NLP generally reflect the formal patterns used to form the text. This results in complex annotation schemes, often tuned to one language or domain, and unintuitive to non-expert annotators. In this paper we propose a different approach and advocate substituting existing syntax-based approaches with semantics-based grammatical annotation. The rationale of this approach is to use manual labor where there is no substitute for it (i.e., annotating semantics), leaving the detection of formal regularities to automated statistical algorithms. To this end, we propose a simple semantic annotation scheme, UCCA for Universal Conceptual Cognitive Annotation. The scheme covers many of the most important elements and relations present in linguistic utterances, including verb-argument structure, optional adjuncts such as adverbials, clause embeddings, and the linkage between them. The scheme is supported by extensive typological cross-linguistic evidence and accords with the leading Cognitive Linguistics theories.

1 Introduction

Syntactic annotation is used as scaffolding in a wide variety of NLP applications. Examples include Machine Translation (Yamada and Knight, 2001), Semantic Role Labeling (SRL) (Punyakanok et al., 2008) and Textual Entailment (Yuret et al., 2010). Syntactic structure is represented using a combinatorial apparatus and a set of categories assigned to the linguistic units it defines. The categories are often based on distributional considerations and reflect the formal patterns in which that unit may occur.

The use of distributional categories leads to intricate annotation schemes. As languages greatly differ in their inventory of constructions, such schemes tend to be tuned to one language or domain. In addition, the complexity of the schemes requires highly proficient workforce for its annotation. For example, the Penn Treebank project (PTB) (Marcus et al., 1993) used linguistics graduates as annotators.

In this paper we propose a radically different approach to grammatical annotation. Under this approach, only semantic distinctions are manually annotated, while distributional regularities are induced using statistical algorithms and without any direct supervision. This approach has four main advantages. First, it facilitates manual annotation that would no longer require close acquaintance with syntactic theory. Second, a data-driven approach for detecting distributional regularities is less prone to errors and to the incorporation of implicit biases. Third, as distributional regularities need not be manually annotated, they can be arbitrarily intricate and fine-grained, beyond the capability of a human annotator to grasp and apply. Fourth, it is likely that semantic tasks that rely on syntactic information would be better served by using a semantics-based scheme.

We present UCCA (Universal Conceptual Cognitive Annotation), an annotation scheme for encoding semantic information. The scheme is designed as a multi-layer structure that allows extending it open-endedly. In this paper we describe the foundational layer of UCCA that focuses on grammatically-relevant information. Already in this layer the scheme covers (in a coarse-grained level) major semantic

*Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

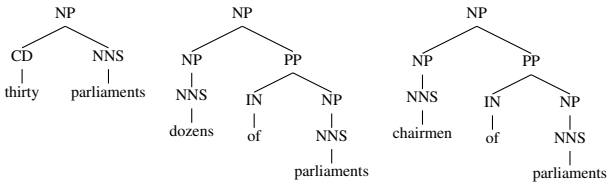

Figure 1: Demonstrating the difference between distributional and semantic representations. The central example is formally more similar to the example on the right, but semantically more similar to the example on the left.

phenomena including verbal and nominal predicates and their arguments, the distinction between core arguments and adjuncts, adjectives, copula clauses, and relations between clauses.

This paper provides a detailed description of the foundational layer of UCCA. To demonstrate UCCA's value over existing approaches, we examine two major linguistic phenomena: relations between clauses (linkage) and the distinction between core arguments and adjuncts. We show that UCCA provides an intuitive coarse-grained analysis in these cases.

UCCA's category set is strongly influenced by "Basic Linguistic Theory" (BLT) (Dixon, 2005, 2010), a theoretical framework used for the description of a great variety of languages. The semantic approach of BLT allows it to draw similarities between constructions, both within and across languages, that share a similar meaning. UCCA takes a similar approach.

The UCCA project includes the compilation of a large annotated corpus. The first distribution of the corpus, to be released in 2013, will consist of about 100K tokens, of which 10K tokens have already been annotated. The annotation of the corpus is carried out mostly using annotators with little to no linguistic background. Details about the corpus and its compilation are largely besides the scope of this paper.

The rest of the paper is constructed as follows. Section 2 explains the basic terms of the UCCA framework. Section 3 presents UCCA's foundational layer. Specifically, Section 3.1 describes the annotation of simple argument structures, Section 3.2 delves into more complex cases, Section 3.3 discusses the distinction between core arguments and adjuncts, Section 3.4 discusses linkages between different structures and Section 3.5 presents a worked-out example. Section 4 describes relevant previous work.

2 UCCA: Basic Terms

Distributional Regularities and Semantic Distinctions. One of the defining characteristics of UCCA is its emphasis on representing semantic distinctions rather than distributional regularities. In order to exemplify the differences between the two types of representations, consider the phrases "dozens of parliaments", "thirty parliaments" and "chairmen of parliaments". Their PTB annotations are presented in Figure 1. The annotation of "dozens of parliaments" closely resembles that of "chairmen of parliaments", and is considerably different from that of "thirty parliaments". A more semantically-motivated representation would have probably emphasized the similarity between "thirty" and "dozens of" and the semantic dissimilarity between "dozens" and "chairmen".

Formalism. UCCA's semantic representation consists of an inventory of relations and their arguments. We use the term *terminals* to refer to the atomic meaning-bearing units. UCCA's foundational layer treats words and fixed multi-word expressions as its terminals, but this definition can easily be extended to include morphemes. The basic formal elements of UCCA are called *units*. A unit may be either (i) a terminal or (ii) several elements that are jointly viewed as a single entity based on conceptual/cognitive considerations. In most cases, a non-terminal unit will simply be comprised of a single relation and its arguments, although in some cases it may contain secondary relations as well (see below). Units can be used as arguments in other relations, giving rise to a hierarchical structure.

UCCA is a multi-layered formalism, where each layer specifies the relations it encodes. For example, consider "big dogs love bones" and assume we wish to encode the relations given by "big" and "love". "big" has a single argument ("dogs"), while "love" has two ("big dogs" and "bones"). Therefore, the units of the sentence are the terminals (always units), "big dogs" and "big dogs love bones". The latter

Abb.	Category	Short Definition
Scene Elements		
P	Process	The main relation of a Scene that evolves in time (usually, action or movement).
S	State	The main relation of a Scene that does not evolve in time.
A	Participant	A participant in a Scene in a broad sense (including locations, abstract entities and Scenes serving as arguments).
D	Adverbial	A secondary relation in a Scene (including temporal relations).
Elements of Non-Scene Relations		
E	Elaborator	A relation (which is not a State or a Process) which applies to a single argument.
N	Connector	A relation (which is not a State or a Process) which applies to two or more arguments.
R	Relator	A secondary relation that pertains to a specific entity and relates it to some super-ordinate relation.
C	Center	An argument of a non-Scene relation.
Inter-Scene Relations		
L	Linker	A relation between Scenes (e.g., temporal, logical, purposive).
H	Parallel Scene	A Scene linked to other Scenes by a Linker.
G	Ground	A relation between the speech event and the described Scene.
Other		
F	Function	Does not introduce a relation or participant. Required by some structural pattern.

Table 1: The complete set of categories in UCCA's foundational layer.

two are units by virtue of corresponding to a relation along with its arguments.

We can compactly annotate the unit structure using a directed graph. Each unit is represented as a node, and descendants of non-terminal units are the sub-units comprising it. Non-terminal nodes in the graph only represent the fact that their descendant units form a unit, and hence do not bear any features. Edges bear labels (or more generally feature sets) that express the descendant unit's role in the relation represented by the parent unit. Therefore, the internal structure of the unit is represented by its outbound edges and their features, while the roles a unit plays in relations it participates in are represented by its inbound edges. Figure 2(a) presents the graph representation for the above example "big dogs love bones". The labels on the figure's edges are explained in Section 3.

Extendability. Extendability is a necessary feature for an annotation scheme given the huge number of features required to formally represent semantics, and the ever-expanding range of distinctions used by the NLP community. UCCA's formalism can be easily extended with new annotation layers introducing new types of semantic distinctions and refining existing types. For example, a layer that represents semantic roles can refine a coarse-grained layer that only distinguishes between arguments and adjuncts. A layer that represents coreference relations between textual entities can be built on top of a more basic layer that simply delineates those entities.

3 The Foundational Layer of UCCA

This section presents an in-depth description of the foundational set of semantic distinctions encoded by UCCA. The three desiderata for this layer are: (i) covering the entire text, so each terminal is a part of at least one unit, (ii) representing argument structure phenomena of both verbal and nominal predicates, (iii) representing relations between argument structures (linkage). Selecting argument structures and their inter-relations as the basic objects of annotation is justified both by their centrality in many approaches for grammatical representation (see Section 4), and their high applicative value, demonstrated by the extensive use of SRL in NLP applications.

Each unit in the foundational layer is annotated with a single feature, which will be simply referred to as its *category*[1]. In the following description, the category names appear *italicized* and accompanied by an abbreviation. The categories are described in detail below and are also summarized in Table 1.

[1]Future extensions of UCCA will introduce more elaborate feature structures.

3.1 Simple Scene Structure

The most basic notion in this layer is the *Scene*. A Scene can either describe some movement or action, or otherwise a temporally persistent state. A Scene usually has a temporal and a spatial dimension. It may be specific to a particular time and place, but may also describe a schematized event which jointly refers to many occurrences of that event in different times and locations. For example, the Scene "elephants eat plants" is a schematized event, which presumably occurs each time an elephant eats a plant. This definition is similar to the definition of a clause in BLT. We avoid the term "clause" due to its syntactic connotation, and its association specifically with verbal rather than nominal predicates.

Every Scene contains one main relation, which is marked as a *Process (P)* if the Scene evolves in time, or otherwise as a *State (S)*. The main relation in an utterance is its "anchor", its most conceptually important aspect of meaning. We choose to incorporate the Process-State distinction in the foundational layer because of its centrality, but it is worth noting this distinction is not necessary for the completeness of the scheme.

A Scene contains one or more *Participants (A)*, which can be either concrete or abstract. Embedded Scenes are also considered Participants (see Section 3.4). Scenes may also include secondary relations, which are generally marked as *Adverbials (D)* using the standard linguistic term. Note that for brevity, we do not designate Scene units as such, as this information can be derived from the categories of its sub-units (i.e., a unit is a Scene if it has a P or an S as a sub-unit).

As an example, consider "Woody generally rides his bike home". The sentence contains a single Scene with three A's: "Woody", "his bike" and "home". It also contains a D: "generally" (see Figure 2(b)).

Non-Scene Relations. Not all relation words evoke a Scene. We distinguish between several types of non-Scene relations. *Elaborators (E)* apply to a single argument, while *Connectors (N)* are relations that apply to two or more entities in a way that highlights the fact that they have a similar feature or type. The arguments of non-Scene relations are marked as *Centers (C)*.

For example, in the expression "hairy dog", "hairy" is an E, and "dog" is a C. In "John and Mary", "John" and "Mary" are C's, while "and" is an N. Determiners are considered E's in the foundational layer, as they relate to a single argument.

Finally, any other type of relation between two or more units that does not evoke a Scene is a *Relator (R)*. R's have two main varieties. In one, R's relate a single entity to other relations or entities in the same context. For instance, in "I saw cookies in the jar", "in" relates "the jar" to the rest of the Scene. In the other, R's relate two units pertaining to different aspects of the same entity. For instance, in "bottom of the sea", "of" relates "bottom" and "the sea", two units that ultimately refer to the same entity.

As for notational conventions, in the first case we place the R inside the boundaries of the unit it relates (so "in the jar" would be an A in "I saw cookies in the jar"). In the second case, we place the R as a sibling of the related units (so "bottom", "of" and "sea" would all be siblings in "bottom of the sea").

Function Units. Some terminals do not refer to a participant or relation. They function only as a part of the construction they are situated in. We mark such terminals as *Function (F)*. Function units usually cannot be substituted by any other word. For example, in the sentence "it is likely that John will come tomorrow", the "it" does not refer to any specific entity or relation and is therefore an F.

Words whose meaning is not encoded in the foundational layer of annotation are also considered F's. For instance, auxilliary verbs in English ("have", "be" and "do") are marked as F's in the foundational layer of UCCA, as features such as voice or tense are not encoded in this layer.

Consider the sentence "John broke the jar lid". It describes a single Scene, where "broke" is the main (non-static) relation. The Participants are "John" and "the jar lid". "the jar lid" contains a part-whole relation, where "jar" describes the whole, and "lid" specifies the part. In such cases, UCCA annotates the "part" as an E and the "whole" as a C. The determiner "the" is also annotated as an E. In more refined layers of annotation, special categories will be devoted to annotating part-whole relations and the semantic relations described by determiners. Figure 2(c) presents the annotation of this example.

3.2 Beyond Simple Scenes

Nominal Predicates. The foundational layer of UCCA annotates the argument structure of nominal predicates much in the same fashion as that of verbal predicates. This accords with the standard practice in several NLP resources, which tend to use the same formal devices for annotating nominal and verbal argument structure (see, e.g., NomBank (Meyers et al., 2004) and FrameNet (Baker et al., 1998)). For example, consider "his speech against the motion". "speech" evokes a Scene that evolves in time and is therefore a P. The Scene has two Participants, namely "his" and "against the motion".

Multiple Parents. In general, a unit may participate in more than one relation. To this end, UCCA allows a unit to have multiple parents. Recall that in UCCA, a non-terminal node represents a relation, and its descendants are the sub-units comprising it. A unit's category is a label over the edge connecting

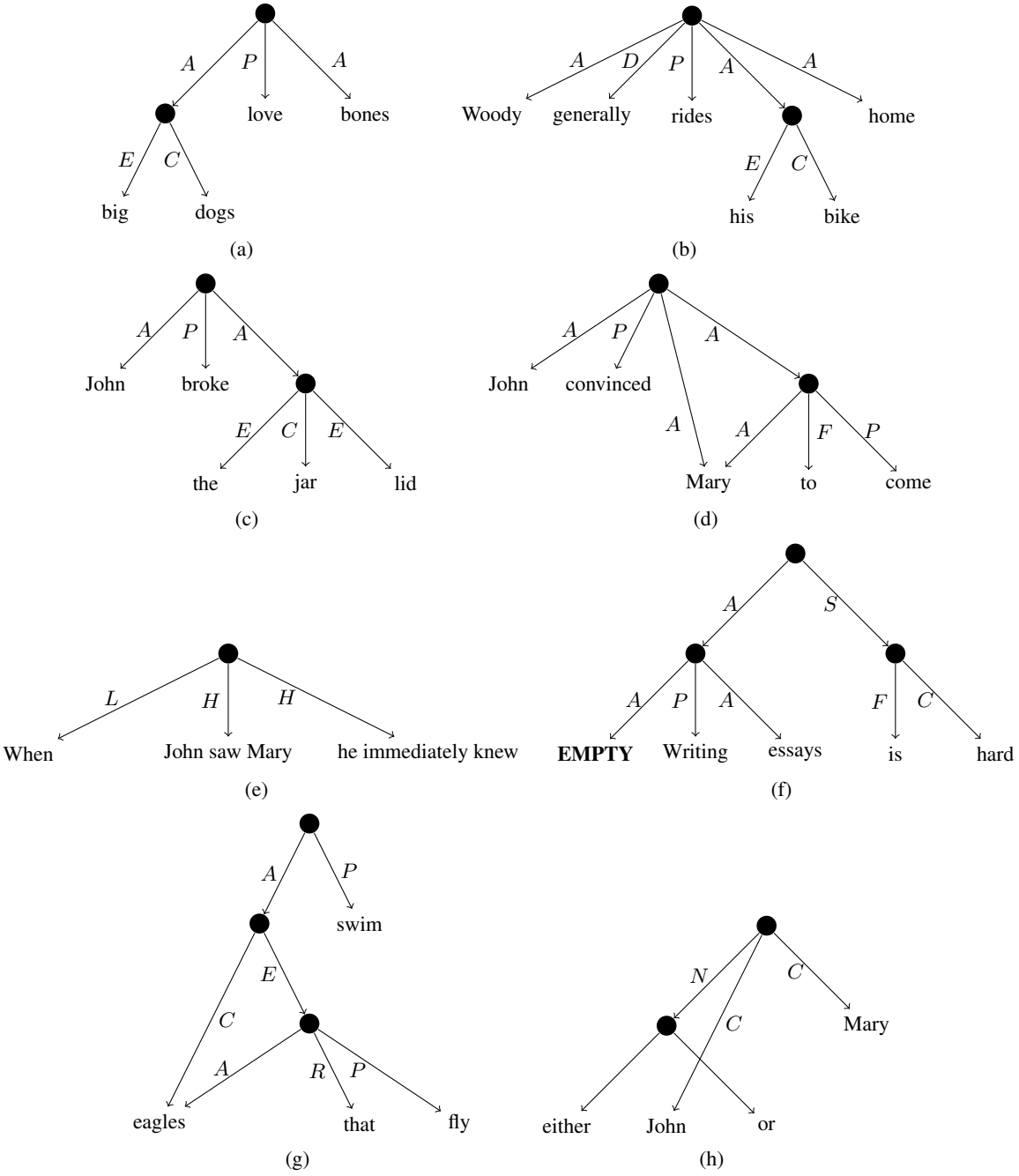

Figure 2: Examples of UCCA annotations.

it to its parent, that reflects the unit's role in the parent relation. A unit that participates in several relations (i.e., has several parents) may thus receive different categories in each of these relations.

For example, consider the sentence "John convinced Mary to come". The relation "convinced" has "John", "Mary" and "Mary to come" as Participants (Scenes may also be Participants, see below). The relation "come" has one Participant, namely "Mary". The resulting graph is presented in Figure 2(d).

The use of multiple parents leads to overlaps between the terminals of different units. It is sometimes convenient to define one of the terminal's parents as its base parent and the others as remote parents. In this paper we do not make this distinction.

Implicit Units. In some cases a relation or argument are clearly described in the text, but do not appear in it overtly. Formally, this results in a unit X that lacks one or more of its descendants. We distinguish between two cases. If that argument or relation corresponds to a unit Y that is placed in some other point in the text, we simply assign that Y as a descendant of X (using UCCA's capacity to represent multiple parents). Otherwise, if this argument or relation never appears in the text, we add an empty leaf node and assign it as X's descendant. We call such units *"Implicit Units"*. Other than not corresponding to any stretch of text, an implicit unit is similar to any other unit.

As an example, consider the sentence "Writing essays is hard". The participant who writes the essays is clearly present in the interpretation of the sentence, but never appears explicitly in the text. It is therefore considered an implicit A in this Scene (see Figure 2(f))[2].

3.3 The Core-Adjunct Distinction

The distinction between core arguments and adjuncts is central in most formalisms of grammar. Despite its centrality, the distinction lacks clear theoretical criteria for defining it, resulting in many borderline cases. This has been a major source of difficulty for establishing clear annotation guidelines. Indeed, the PTB describes the core-adjunct distinction as "very difficult" for the annotators, resulting in a significant slowdown of the annotation Process (Marcus et al., 1993).

Dowty (2003) claims that the pre-theoretic notions underlying the core-adjunct distinction are a conjunction of syntactic and semantic considerations. The syntactic distinction separates "optional elements" (adjuncts), and "obligatory elements" (cores). The semantic criterion distinguishes elements that "modify" or restrict the meaning of the head (adjuncts) and elements that are required by the meaning of the head, without which its meaning is incomplete (cores). A related semantic criterion distinguishes elements that have a similar semantic content with different predicates (adjuncts), and elements whose role is highly predicate-dependent (cores).

Consider the following opposing examples: (i) "Woody walked **quickly**" and (ii) "Woody cut **the cake**". "quickly" meets both the syntactic and the semantic criteria for an adjunct: it is optional and it serves to restrict the meaning of "walked". It also has a similar semantic content when appearing with different verbs ("walk quickly", "eat quickly", "talk quickly" etc.). "the cake" meets both the syntactic and the semantic criteria for a core: it is obligatory, and completes the meaning of "cut". However, many other cases are not as obvious. For instance, in "he walked **into his office**", the boldfaced argument is a core according to Framenet, but an adjunct according to PropBank (Abend and Rappoport, 2010).

The core-adjunct distinction in UCCA is translated into the distinction between D's (Adverbials) and A's (Participants). UCCA is a semantic scheme and therefore the syntactic criterion of "obligatoriness" is not applicable, and is instead left to be detected by statistical means. Instead, UCCA defines A's as units that introduce a new participant to the Scene and D's as units that add more information to the Scene without introducing a participant.

Revisiting our earlier examples, in "Woody cut the cake", "the cake" introduces a new participant and is therefore an A, while in "Woody walked quickly", "quickly" does not introduce a new participant and is therefore a D. In the more borderline example "Woody walked into his office", "into his office" is clearly an A under UCCA's criteria, as it introduces a new participant, namely "his office".

[2]Note the internal structure of the unit "is hard". The semantically significant sub-unit ("hard") is a C, while the other sub-unit ("is"), which does not convey relevant semantic information, is marked as an F. In general, if a unit has a single sub-unit which contributes virtually all relevant semantic information, that unit is marked as a C while all other units are marked as F's.

Note that locations in UCCA are almost invariably A's, as they introduce a new participant, namely the location. Consider "Woody walked in the park". "in the park" introduces the participant "the park" and is therefore an A. Unlike many existing approaches (including the PTB), UCCA does not distinguish between obligatory locations (e.g., "based in Europe") and optional locations (e.g., "walked in the park"), as this distinction is mostly distributional in nature and can be detected by automatic means.

Two cases which do not easily fall into either side of this distinction are subordinated clauses and temporal relations. Subordinated clauses are discussed as part of a general discussion of linkage in Section 3.4. The treatment of temporal relations requires a more fine-grained layer of representation. For the purposes of the foundational layer, we follow common practice and mark them as D's.

3.4 Linkage

Linkage in UCCA refers to the relation between Scenes. Scenes are invariably units, as they include a relation along with all its arguments. The category of the Scene units is determined by the relation they are situated in, as is the case with any other unit. The foundational layer takes a coarse-grained approach to inter-Scene relations and recognizes three types of linkage. This three-way distinction is adopted from Basic Linguistic Theory and is valid cross-linguistically.

First, a Scene can be a Participant in another Scene, in which case the Scene is marked as an A. For example, consider "writing essays is hard". It contains a main temporally static relation (S) "is hard" and an A "writing essays". The sentence also contains another Scene "writing essays", which has an implicit A (the one writing) and an explicit A ("essays"). See Figure 2(f) for the annotation of this Scene (note the empty node corresponding to the implicit unit).

Second, a Scene may serve as an Elaborator of some unit in another Scene, in which case the Scene is marked as an E. For instance, "eagles that fly swim". There are two Scenes in this sentence: (1) one whose main relation is "swim" and its A is "eagles that fly", (2) and another Scene whose main relation is "fly", and whose A is "eagles". See Figure 2(g) for the annotation graph of this sentence.

The third type of linkage covers inter-Scene relations that are not covered above. In this case, we mark the unit specifying the relation between the Scenes as a *Linker (L)* and its arguments as *Parallel Scenes (H)*. The Linker and the Parallel Scenes are positioned in a flat structure, which represents the linkage relation. For example, consider "When John saw Mary, he immediately knew" (Figure 2(e)). The sentence is composed of two Scenes "John saw Mary" and "he immediately knew" marked by H's and linked by the L "when". More fine-grained layers of annotation can represent the coreference relation between "John" and "he", as well as a more refined typology of linkages, distinguishing, e.g., temporal, logical and purposive linkage types.

UCCA does not allow annotating a Scene as an Adverbial within another Scene. Instead it represents temporal, manner and other relations between Scenes often represented as Adverbials (or sub-ordinate clauses), as linked Scenes. For instance, the sentence "I'm here because I wanted to visit you" is annotated as two Parallel Scenes ("I'm here" and "I wanted to visit you"), linked by the Linker "because".

Linkage is handled differently in other NLP resources. SRL formalisms, such as FrameNet and PropBank, consider a predicate's argument structure as the basic annotation unit and do not represent linkage in any way. Syntactic annotation schemes (such as the PTB) consider the sentence to be the basic unit for annotation and refrain from annotating inter-sentential relations, which are addressed only as part of the discourse level. However, units may establish similar relations between sentences as those expressed within a sentence. Another major difference between UCCA and other grammatical schemes is that UCCA does not recognize any type of subordination between clauses except for the cases where one clause serves as an Elaborator or as a Participant in another clause (see above discussion). In all other cases, linkage is represented by the identity of the Linker and, in future layers, by more fine-grained features assigned to the linkage structure.

Ground. Some units express the speaker's opinion of a Scene, or otherwise relate the Scene to the speaker, the hearer or the speech event. Examples include "in my opinion", "surprisingly" and "rumor has it". In principle, such units constitute a Scene in their own right, whose participants (minimally including the speaker) are implicit. However, due to their special characteristics, we choose to designate

a special category for such cases, namely *Ground (G)*. For example, "Surprisingly" in "Surprisingly, Mary didn't come to work today" is a G linked to the Scene "Mary didn't come to work today".

Note that the distinction between G's and fully-fledged Scenes is a gradient one. Consider the above example and compare it to "I think Mary didn't come today" and "John thinks Mary didn't come today". While "John thinks" in the last example is clearly not a G, "I think" is a more borderline case. Gradience is a central phenomenon in all forms of grammatical representation, including UCCA. However, due to space limitations, we defer the discussion of UCCA's treatment of gradience to future work.

3.5 Worked-out Example

Consider the following sentence[3]:

> After her parents' separation in 1976, Jolie and her brother lived with their mother, who gave up acting to focus on raising her children.

There are four Scenes in this sentence, with main relations "separation", "lived", "gave up acting" and "focus on raising". Note that "gave up acting" and "focus on raising" are composed of two relations, one central and the other dependent. UCCA annotates such cases as a single P. A deeper discussion of these issues can be found in (Dixon, 2005; Van Valin, 2005).

The Linkers are "after" (linking "separation" and "lived"), and "to" (linking "gave up acting" and "focus on raising"). The unit "who gave up acting to focus on raising her children" is an E, and therefore "who" is an R. We start with the top-level structure and continue by analyzing each Scene separately (non-Scene relations are not analyzed in this example):

- "After$_L$ [her parents' separation in 1976]$_H$, [Jolie and her brother lived with their mother, [who$_R$ [gave up acting]$_H$ to$_L$ [focus on raising her children]$_H$]$_E$]$_H$"
- "[her parents']$_A$ separation$_P$ [in 1976]$_D$"
- "[Jolie and her brother]$_A$ lived$_P$ [with their mother who abandoned ... children]$_A$"
- "mother$_A$ ... [gave up acting]$_P$"
- "mother$_A$ ... [focus on raising]$_P$ [her children]$_A$"

4 Previous Work

Many grammatical annotation schemes have been proposed over the years in an attempt to capture the richness of grammatical phenomena. In this section, we focus on approaches that provide a sizable corpus of annotated text. We put specific emphasis on English corpora, which is the most studied language and the focus language of this paper.

Semantic Role Labeling Schemes. The most prominent schemes to SRL are FrameNet (Baker et al., 1998), PropBank (Palmer et al., 2005) and VerbNet (Schuler, 2005) for verbal predicates and NomBank for nominal predicates (Meyers et al., 2004). They share with UCCA their focus on semantically-motivated rather than distributionally-motivated distinctions. However, unlike UCCA, they annotate each predicate separately, yielding shallow representations which are hard to learn directly without using syntactic parsing as preprocessing (Punyakanok et al., 2008). In addition, UCCA has a wider coverage than these projects, as it addresses both verbal, nominal and adjectival predicates.

Recently, the *Framenet Constructicon* project (Fillmore et al., 2010) extended FrameNet to more complex constructions, including a representation of relations between argument structures. However, the project is admittedly devoted to constructing a lexical resource focused on specific cases of interest, and does not attempt to provide a fully annotated corpus of naturally occurring text. The foundational layer of UCCA can be seen as being complementary to Framenet and Framenet Constructicon, as the UCCA foundational layer focuses on a high coverage, coarse-grained annotation, while Framenet focuses on more fine-grained distinctions at the expense of coverage. In addition, the projects differ in terms of their approach to linkage.

[3]Taken from "Angelina Jolie" article in Wikipedia (http://http://en.wikipedia.org/wiki/Angelina_Jolie).

Penn Treebank. The most influential syntactic annotation in NLP is probably the PTB. The PTB has spawned much subsequent research both in treebank compilation and in parsing technology. However, despite its tremendous contribution to NLP, the corpus today does not meet the community's needs in two major respects. First, it is hard to extend, both with new distinctions and with new sentences (due to its complex annotation that requires expert annotators). Second, its interface with semantic applications is far from trivial. Even in the syntactically-oriented semantic task of argument identification for SRL, results are of about 85% F-score for the in-domain scenario (Màrquez et al., 2008; Abend et al., 2009).

Dependency Grammar. An alternative approach to syntactic representation is Dependency Grammar. This approach is widely used in NLP today due to its formal and conceptual simplicity, and its ability to effectively represent fundamental semantic relations, notably predicate-argument and head-modifier relations. UCCA is similar to dependency grammar both in terms of their emphasis on representing predicate-argument relations and in terms of their formal definition[4]. The formal similarity is reflected in that they both place features over the graph's edges rather than over its nodes, and in that they both form a directed graph. In addition, neither formalism imposes contiguity (or projectivity in dependency terms) on its units, which facilitates their application to languages with relatively free word order.

However, despite their apparent similarity, the formalisms differ in several major respects. Dependency grammar uses graphs where each node is a word. Despite the simplicity and elegance of this approach, it leads to difficulties in the annotation of certain structures. We discuss three such cases: structures containing multiple heads, units with multiple parents and empty units. Cases where there is no clear dependency annotation are a major source of difficulty in standardizing, evaluating and creating clear annotation guidelines for dependency annotation (Schwartz et al., 2011). UCCA provides a natural solution in all of these cases, as is hereby detailed.

First, UCCA rejects the assumption that every structure has a unique head. Formally, instead of selecting a single head whose descendants are (the heads of) the argument units, UCCA introduces a new node for each relation, whose descendants are all the sub-units comprising that relation, including the predicate and its arguments. The symmetry between the descendants is broken through the features placed on the edges.

Consider coordination structures as an example. The difficulty of dependency grammar to capture such structures is exemplified by the 8 possible annotations in current use in NLP (Ivanova et al., 2012). In UCCA, all elements of the coordination (i.e., the conjunction along with its conjuncts) are descendants of a mutual parent, where only their categories distinguish between their roles. For instance, in "John and Mary", "John", "Mary" and "and" are all listed under a joint parent. Discontiguous conjunctions (such as "**either** John **or** Mary") are also handled straightforwardly by placing "either" and "or" under a single parent, which in turn serves as a Connector (Figure 2(h)). Note that the edges between "either" and "or" and their mutual parent have no category labels, since the unit "either ... or" is considered an unanalyzable terminal. A related example is inter-clause linkage, where it is not clear which clause should be considered the head of the other. See the discussion of UCCA's approach with respect to clause subordination in Section 3.4.

Second, a unit in UCCA can have multiple parents if it participates in multiple relations. Multiple parents are already found in the foundational layer (see, e.g., Figure 2(d)), and will naturally multiply with the introduction of new annotation layers introducing new relations. This is prohibited in standard dependency structures.

Third, UCCA allows implicit units, i.e., units that do not have any corresponding stretch of text. The importance of such "empty" nodes has been previously recognized in many formalisms for grammatical representation, including the PTB.

At a more fundamental level, the difference between UCCA and most dependency structures used in NLP is the latter's focus on distributional regularities. One example for this is the fact the most widely used scheme for English dependency grammar is automatically derived from the PTB. Another

[4]Dependency structures appear in different contexts in various guises. Those used in NLP are generally trees in which each word has at most one head and whose nodes are the words of the sentence along with a designated root node (Ivanova et al., 2012). We therefore restrict our discussion to dependency structures that follow these restrictions.

example is the treatment of fixed expressions, such as phrasal verbs and idioms. In these cases, several words constitute one unanalyzable semantic unit, and are treated by UCCA as such. However, they are analyzed up to the word level by most dependency structures. Finally, a major divergence of UCCA from standard dependency representation is UCCA's multi-layer structure that allows for the extension of the scheme with new distinctions.

Linguistically Expressive Grammars. Numerous approaches to grammatical representation in NLP have set to provide a richer grammatical representation than the one provided by the common phrase structure and dependency structures. Examples include Combinatory Categorial Grammar (CCG) (Steedman, 2001), Tree Adjoining Grammar (TAG) (Joshi and Schabes, 1997), Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1981) and Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag, 1994). One of the major motivations for these approaches is to provide a formalism for encoding both semantic and distributional distinctions and the interface between them. UCCA diverges from these approaches in its focus on annotating semantic information, leaving distributional regularities to be detected automatically.

A great body of work in formal semantics focuses on compositionality, i.e., how the meaning of a unit is derived from its syntactic structure along with the meaning of its sub-parts. Compositionality forms a part of the mapping between semantics and distribution, and is therefore modeled statistically by UCCA. A more detailed comparison between the different approaches is not directly relevant to this paper.

5 Conclusion

In this paper we proposed a novel approach to grammatical representation. Under this approach, only semantic distinctions are manually annotated, while distributional regularities are detected by automatic means. This approach greatly facilitates manual annotation of grammatical phenomena, by focusing the manual labor on information that can only be annotated manually.

We presented UCCA, a multi-layered semantic annotation scheme for representing a wide variety of semantic information in varying granularities. In its foundational layer, the scheme encodes verbal and nominal argument structure, copula clauses, the distinction between core arguments and adjuncts, and the relations between different predicate-argument structures. The scheme is based on basic, coarse-grained semantic notions, supported by cross-linguistic evidence.

Preliminary results show that the scheme can be learned quickly by non-expert annotators. Concretely, our annotators, including some with no linguistic background in linguistics, have reached a reasonable level of proficiency after a training period of 30 to 40 hours. Following the training period, our annotators have been found to make only occasional errors. These few errors are manually corrected in a later review phase. Preliminary experiments also show that the scheme can be applied to several languages (English, French, German) using the same basic set of distinctions.

Two important theoretical issues were not covered this paper due to space considerations. One is UCCA's treatment of cases where there are several analyses that do not exclude each other, each highlighting a different aspect of meaning of the analyzed utterance (termed *Conforming Analyses*). The other is UCCAs treatment of cases where a unit of one type is used in a relation that normally receives a sub-unit of a different type. For example, in "John's kick saved the game", "John's kick" describes an action but is used as a subject of "saved", a slot usually reserved for animate entities. Both of these issues will be discussed in future works.

Current efforts are devoted to creating a corpus of annotated text in English. The first distribution of the corpus consisting of about 100K tokens, of which 10K tokens have already been annotated, will be released during 2013. A parallel effort is devoted to constructing a statistical analyzer, trained on the annotated corpus. Once available, the analyzer will be used to produce UCCA annotations that will serve as input to NLP applications traditionally requiring syntactic preprocessing. The value of UCCA for applications and the learning algorithms will be described in future papers.

References

Abend, O. and A. Rappoport (2010). Fully unsupervised core-adjunct argument classification. In ACL '10.

Abend, O., R. Reichart, and A. Rappoport (2009). Unsupervised Argument identification for semantic role labeling. In ACL-IJCNLP '09.

Baker, C., C. Fillmore, and J. Lowe (1998). The berkeley framenet project. In ACL-COLING '98.

Dixon, R. (2005). A Semantic Approach To English Grammar. Oxford University Press.

Dixon, R. (2010). Basic Linguistic Theory: Grammatical Topics, Volume 2. Oxford University Press.

Dowty, D. (2003). The dual analysis of adjuncts/complements in categorial grammar. Modifying Adjuncts.

Fillmore, C., R. Lee-Goldman, and R. Rhodes (2010). The framenet constructicon. Sign-based Construction Grammar. CSLI Publications, Stanford.

Ivanova, A., S. Oepen, L. Øvrelid, and D. Flickinger (2012). Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In LAW '12.

Joshi, A. and Y. Schabes (1997). Tree-adjoining grammars. Handbook Of Formal Languages 3.

Kaplan, R. and J. Bresnan (1981). Lexical-Functional Grammar: A Formal System For Grammatical Representation. Massachusetts Institute Of Technology, Center For Cognitive Science.

Marcus, M., M. Marcinkiewicz, and B. Santorini (1993). Building a large annotated corpus of english: The penn treebank. Computational Linguistics 19(2).

Màrquez, L., X. Carreras, K. Litkowski, and S. Stevenson (2008). Semantic role labeling: An introduction to the special issue. Computational Linguistics 34(2).

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman (2004). Annotating noun argument structure for nombank. In LREC '04.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The proposition bank: An annotated corpus of semantic roles. Computational Linguistics 31(1).

Pollard, C. and I. Sag (1994). Head-driven Phrase Structure Grammar. University Of Chicago Press.

Punyakanok, V., D. Roth, and W. Yih (2008). The importance of syntactic parsing and inference in semantic role labeling. Computational Linguistics 34(2).

Schuler, K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon. Ph. D. thesis, University of Pennsylvania.

Schwartz, R., O. Abend, R. Reichart, and A. Rappoport (2011). Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In ACL-NAACL '11.

Steedman, M. (2001). The Syntactic Process. MIT Press.

Van Valin, R. (2005). Exploring The Syntax-semantics Interface. Cambridge University Press.

Yamada, K. and K. Knight (2001). A syntax-based statistical translation model. In ACL '01.

Yuret, D., A. Han, and Z. Turgut (2010). Semeval-2010 task 12: Parser evaluation using textual entailments. The SemEval-2010 Evaluation Exercises On Semantic Evaluation '10.

Published in ACL 2013

Chapter 7

Universal Conceptual Cognitive Annotation (UCCA)

Universal Conceptual Cognitive Annotation (UCCA)

Omri Abend[*]
Institute of Computer Science
The Hebrew University
omria01@cs.huji.ac.il

Ari Rappoport
Institute of Computer Science
The Hebrew University
arir@cs.huji.ac.il

Abstract

Syntactic structures, by their nature, reflect first and foremost the formal constructions used for expressing meanings. This renders them sensitive to formal variation both within and across languages, and limits their value to semantic applications. We present UCCA, a novel multi-layered framework for semantic representation that aims to accommodate the semantic distinctions expressed through linguistic utterances. We demonstrate UCCA's portability across domains and languages, and its relative insensitivity to meaning-preserving syntactic variation. We also show that UCCA can be effectively and quickly learned by annotators with no linguistic background, and describe the compilation of a UCCA-annotated corpus.

1 Introduction

Syntactic structures are mainly committed to representing the formal patterns of a language, and only indirectly reflect semantic distinctions. For instance, while virtually all syntactic annotation schemes are sensitive to the structural difference between (a) "John took a shower" and (b) "John showered", they seldom distinguish between (a) and the markedly different (c) "John took my book". In fact, the annotations of (a) and (c) are identical under the most widely-used schemes for English, the Penn Treebank (PTB) (Marcus et al., 1993) and CoNLL-style dependencies (Surdeanu et al., 2008) (see Figure 1).

[*] Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

Underscoring the semantic similarity between (a) and (b) can assist semantic applications. One example is machine translation to target languages that do not express this structural distinction (e.g., both (a) and (b) would be translated to the same German sentence "John duschte"). Question Answering applications can also benefit from distinguishing between (a) and (c), as this knowledge would help them recognize "my book" as a much more plausible answer than "a shower" to the question "what did John take?".

This paper presents a novel approach to grammatical representation that annotates semantic distinctions and aims to abstract away from specific syntactic constructions. We call our approach *Universal Conceptual Cognitive Annotation (UCCA)*. The word "cognitive" refers to the type of categories UCCA uses and its theoretical underpinnings, and "conceptual" stands in contrast to "syntactic". The word "universal" refers to UCCA's capability to accommodate a highly rich set of semantic distinctions, and its aim to ultimately provide all the necessary semantic information for learning grammar. In order to accommodate this rich set of distinctions, UCCA is built as a multi-layered structure, which allows for its open-ended extension. This paper focuses on the foundational layer of UCCA, a coarse-grained layer that represents some of the most important relations expressed through linguistic utterances, including argument structure of verbs, nouns and adjectives, and the inter-relations between them (Section 2).

UCCA is supported by extensive typological cross-linguistic evidence and accords with the leading Cognitive Linguistics theories. We build primarily on Basic Linguistic Theory (BLT) (Dixon, 2005; 2010a; 2010b; 2012), a typological approach to grammar successfully used for the de-

scription of a wide variety of languages. BLT uses semantic similarity as its main criterion for categorizing constructions both within and across languages. UCCA takes a similar approach, thereby creating a set of distinctions that is motivated cross-linguistically. We demonstrate UCCA's relative insensitivity to paraphrasing and to cross-linguistic variation in Section 4.

UCCA is exceptional in (1) being a semantic scheme that abstracts away from specific syntactic forms and is not defined relative to a specific domain or language, (2) providing a coarse-grained representation which allows for open-ended extension, and (3) using cognitively-motivated categories. An extensive comparison of UCCA to existing approaches to syntactic and semantic representation, focusing on the major resources available for English, is found in Section 5.

This paper also describes the compilation of a UCCA-annotated corpus. We provide a quantitative assessment of the annotation quality. Our results show a quick learning curve and no substantial difference in the performance of annotators with and without background in linguistics. This is an advantage of UCCA over its syntactic counterparts that usually need annotators with extensive background in linguistics (see Section 3).

We note that UCCA's approach that advocates automatic learning of syntax from semantic supervision stands in contrast to the traditional view of generative grammar (Clark and Lappin, 2010).

2 The UCCA Scheme

2.1 The Formalism

UCCA uses directed acyclic graphs (DAGs) to represent its semantic structures. The atomic meaning-bearing units are placed at the leaves of the DAG and are called *terminals*. In the foundational layer, terminals are words and multi-word chunks, although this definition can be extended to include arbitrary morphemes.

The nodes of the graph are called *units*. A unit may be either (i) a terminal or (ii) several elements jointly viewed as a single entity according to some semantic or cognitive consideration. In many cases, a non-terminal unit is comprised of a single relation and the units it applies to (its arguments), although in some cases it may also contain secondary relations. Hierarchy is formed by using units as arguments or relations in other units.

Categories are annotated over the graph's edges, and represent the descendant unit's role in forming the semantics of the parent unit. Therefore, the internal structure of a unit is represented by its outbound edges and their categories, while the roles a unit plays in the relations it participates in are represented by its inbound edges.

We note that UCCA's structures reflect a single interpretation of the text. Several discretely different interpretations (e.g., high vs. low PP attachments) may therefore yield several different UCCA annotations.

UCCA is a multi-layered formalism, where each layer specifies the relations it encodes. The question of which relations will be annotated (equivalently, which units will be formed) is determined by the layer in question. For example, consider "John kicked his ball", and assume our current layer encodes the relations expressed by "kicked" and by "his". In that case, the unit "his" has a single argument[1] ("ball"), while "kicked" has two ("John" and "his ball"). Therefore, the units of the sentence are the terminals (which are always units), "his ball" and "John kicked his ball". The latter two are units by virtue of expressing a relation along with its arguments. See Figure 2(a) for a graph representation of this example.

For a brief comparison of the UCCA formalism with other dependency annotations see Section 5.

2.2 The UCCA Foundational Layer

The foundational layer is designed to cover the entire text, so that each word participates in at least one unit. It focuses on argument structures of verbal, nominal and adjectival predicates and the inter-relations between them. Argument structure phenomena are considered basic by many approaches to semantic and grammatical representation, and have a high applicative value, as demonstrated by their extensive use in NLP.

The foundational layer views the text as a collection of *Scenes*. A Scene can describe some movement or action, or a temporally persistent state. It generally has a temporal and a spatial dimension, which can be specific to a particular time and place, but can also describe a schematized event which refers to many events by highlighting a common meaning component. For example, the Scene "John loves bananas" is a schematized event, which refers to John's disposition towards bananas without making any temporal or spatial

[1] The anaphoric aspects of "his" are not considered part of the current layer (see Section 2.3).

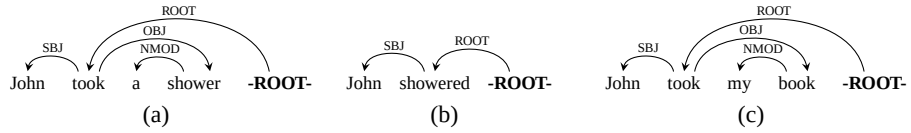

Figure 1: CoNLL-style dependency annotations. Note that (a) and (c), which have different semantics but superficially similar syntax, have the same annotation.

Abb.	Category	Short Definition
		Scene Elements
P	**Process**	The main relation of a Scene that evolves in time (usually an action or movement).
S	**State**	The main relation of a Scene that does not evolve in time.
A	**Participant**	A participant in a Scene in a broad sense (including locations, abstract entities and Scenes serving as arguments).
D	**Adverbial**	A secondary relation in a Scene (including temporal relations).
		Elements of Non-Scene Units
C	**Center**	Necessary for the conceptualization of the parent unit.
E	**Elaborator**	A non-Scene relation which applies to a single Center.
N	**Connector**	A non-Scene relation which applies to two or more Centers, highlighting a common feature.
R	**Relator**	All other types of non-Scene relations. Two varieties: (1) Rs that relate a C to some super-ordinate relation, and (2) Rs that relate two Cs pertaining to different aspects of the parent unit.
		Inter-Scene Relations
H	**Parallel Scene**	A Scene linked to other Scenes by regular linkage (e.g., temporal, logical, purposive).
L	**Linker**	A relation between two or more Hs (e.g., "when", "if", "in order to").
G	**Ground**	A relation between the speech event and the uttered Scene (e.g., "surprisingly", "in my opinion").
		Other
F	**Function**	Does not introduce a relation or participant. Required by the structural pattern it appears in.

Table 1: The complete set of categories in UCCA's foundational layer.

specifications. The definition of a Scene is motivated cross-linguistically and is similar to the semantic aspect of the definition of a "clause" in Basic Linguistic Theory[2].

Table 1 provides a concise description of the categories used by the foundational layer[3]. We turn to a brief description of them.

Simple Scenes. Every Scene contains one main relation, which is the anchor of the Scene, the most important relation it describes (similar to frame-evoking lexical units in FrameNet (Baker et al., 1998)). We distinguish between static Scenes, that describe a temporally persistent state, and processual Scenes that describe a temporally evolving event, usually a movement or an action. The main relation receives the category *State (S)* in static and *Process (P)* in processual Scenes. We note that the S-P distinction is introduced here mostly for practical purposes, and that both categories can be viewed as sub-categories of the more abstract category Main Relation.

A Scene contains one or more *Participants (A)*. This category subsumes concrete and abstract participants as well as embedded Scenes (see below). Scenes may also contain secondary relations, which are marked as *Adverbials (D)*.

The above categories are indifferent to the syntactic category of the Scene-evoking unit, be it a verb, a noun, an adjective or a preposition. For instance, in the Scene "The book is in the garden", "is in" is the S, while "the book" and "the garden" are As. In "Tomatoes are red", the main static relation is "are red", while "Tomatoes" is an A.

The foundational layer designates a separate set of categories to units that do not evoke a Scene. *Centers (C)* are the sub-units of a non-Scene unit that are necessary for the unit to be conceptualized and determine its semantic type. There can be one or more Cs in a non-Scene unit[4].

Other sub-units of non-Scene units are categorized into three types. First, units that apply to a single C are annotated as *Elaborators (E)*. For instance, "big" in "big dogs" is an E, while "dogs" is a C. We also mark determiners as Es in this coarse-grained layer[5]. Second, relations that relate two or

[2] As UCCA annotates categories on its edges, Scene nodes bear no special indication. They can be identified by examining the labels on their outgoing edges (see below).

[3] Repeated here with minor changes from (Abend and Rappoport, 2013), which focuses on the categories themselves.

[4] By allowing several Cs we avoid the difficulties incurred by the common single head assumption. In some cases the Cs are inferred from context and can be implicit.

[5] Several Es that apply to a single C are often placed in

more Cs, highlighting a common feature or role (usually coordination), are called *Connectors (N)*. See an example in Figure 2(b).

Relators (R) cover all other types of relations between two or more Cs. Rs appear in two main varieties. In one, Rs relate a single entity to a super-ordinate relation. For instance, in "I heard noise in the kitchen", "in" relates "the kitchen" to the Scene it is situated in. In the other, Rs relate two units pertaining to different aspects of the same entity. For instance, in "bottom of the sea", "of" relates "bottom" and "the sea", two units that refer to different aspects of the same entity.

Some units do not introduce a new relation or entity into the Scene, and are only part of the formal pattern in which they are situated. Such units are marked as *Functions (F)*. For example, in the sentence "it is customary for John to come late", the "it" does not refer to any specific entity or relation and is therefore an F.

Two example annotations of simple Scenes are given in Figure 2(a) and Figure 2(b).

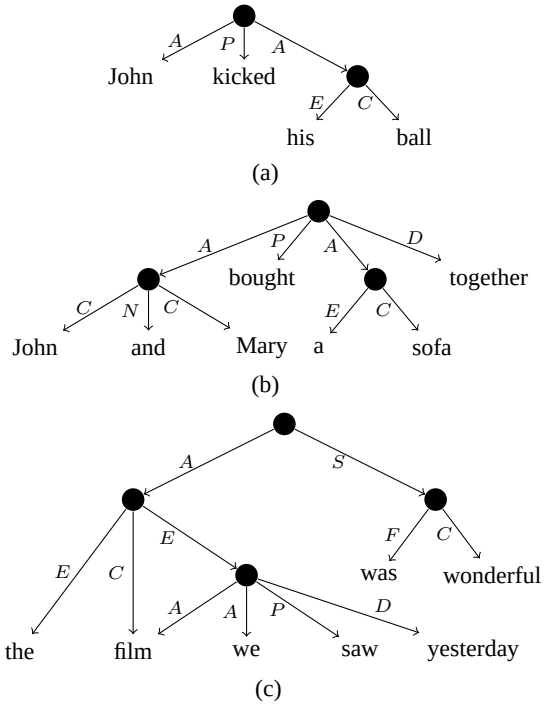

Figure 2: Examples of UCCA annotation graphs.

More complex cases. UCCA allows units to participate in more than one relation. This is a natural requirement given the wealth of distinctions UCCA is designed to accommodate. Already in the foundational layer of UCCA, the need arises for multiple parents. For instance, in "John asked Mary to join him", "Mary" is a Participant of both the "asking" and the "joining" Scenes.

In some cases, an entity or relation is prominent in the interpretation of the Scene, but is not mentioned explicitly anywhere in the text. We mark such entities as *Implicit Units*. Implicit units are identical to terminals, except that they do not correspond to a stretch of text. For example, "playing games is fun" has an implicit A which corresponds to the people playing the game.

UCCA annotates inter-Scene relations (linkage) and, following Basic Linguistic Theory, distinguishes between three major types of linkage. First, a Scene can be an A in another Scene. For instance, in "John said he must leave", "he must leave" is an A inside the Scene evoked by "said". Second, a Scene may be an E of an entity in another Scene. For instance, in "the film we saw yesterday was wonderful", "film we saw yesterday" is a Scene that serves as an E of "film", which is both an A in the Scene and the Center of an A in the Scene evoked by "wonderful" (see Figure 2(c)).

A third type of linkage covers all other cases, e.g., temporal, causal and conditional inter-Scene relations. The linked Scenes in such cases are marked as *Parallel Scenes (H)*. The units specifying the relation between Hs are marked as *Linkers (L)*[6]. As with other relations in UCCA, Linkers and the Scenes they link are bound by a unit.

Unlike common practice in grammatical annotation, linkage relations in UCCA can cross sentence boundaries, as can relations represented in other layers (e.g., coreference). UCCA therefore annotates texts comprised of several paragraphs and not individual sentences (see Section 3).

Example sentences. Following are complete annotations of two abbreviated example sentences from our corpus (see Section 3).

"Golf became a passion for his oldest daughter:
she took daily lessons and became very good,
reaching the Connecticut Golf Championship."

This sentence contains four Scenes, evoked by "became a passion", "took daily lessons", "became very good" and "reaching". The individual Scenes are annotated as follows:

1. "Golf$_A$ [became$_E$ a$_E$ passion$_C$]$_P$ [for$_R$ his$_E$ oldest$_E$ daughter$_C$]$_A$"

a flat structure. In general, the coarse-grained foundational layer does not try to resolve fine scope issues.

[6] It is equally plausible to include Linkers for the other two linkage types. This is not included in the current layer.

2. "she$_A$ [took$_F$ [daily$_E$ lessons$_C$]$_C$]$_P$"
3. "she$_A$ ... [became$_E$ [very$_E$ good$_C$]$_C$]$_S$"
4. "she$_A$ ... reaching$_P$ [the$_E$ Connecticut$_E$ Golf$_E$ Championship$_C$]$_A$"

There is only one explicit Linker in this sentence ("and"), which links Scenes (2) and (3). None of the Scenes is an A or an E in the other, and they are therefore all marked as Parallel Scenes. We also note that in the case of the light verb construction "took lessons" and the copula clauses "became good" and "became a passion", the verb is not the Center of the main relation, but rather the following noun or adjective. We also note that the unit "she" is an A in Scenes (2), (3) and (4).

We turn to our second example:

"Cukor encouraged the studio to accept her demands."

This sentence contains three Scenes, evoked by "encouraged", "accept" and "demands":

1. Cukor$_A$ encouraged$_P$ [the$_E$ studio$_C$]$_A$ [to$_R$ [accept her demands]$_C$]$_A$
2. [the studio]$_A$ ... accept$_P$ [her demands]$_A$
3. her$_A$ demands$_P$ **IMP**$_A$

Scenes (2) and (3) act as Participants in Scenes (1) and (2) respectively. In Scene (2), there is an implicit Participant which corresponds to whatever was demanded. Note that "her demands" is a Scene, despite being a noun phrase.

2.3 UCCA's Multi-layered Structure

Additional layers may refine existing relations or otherwise annotate a complementary set of distinctions. For instance, a refinement layer can categorize linkage relations according to their semantic types (e.g., temporal, purposive, causal) or provide tense distinctions for verbs. Another immediate extension to UCCA's foundational layer can be the annotation of coreference relations. Recall the example "John kicked his ball". A coreference layer would annotate a relation between "John" and "his" by introducing a new node whose descendants are these two units. The fact that this node represents a coreference relation would be represented by a label on the edge connecting them to the coreference node.

There are three common ways to extend an annotation graph. First, by adding a relation that relates previously established units. This is done by introducing a new node whose descendants are the related units. Second, by adding an intermediate unit between a parent unit and some of its sub-units. For instance, consider "he replied foolishly" and "he foolishly replied". A layer focusing on Adverbial scope may refine the flat Scene structure assigned by the foundational layer, expressing the scope of "foolishly" over the relation "replied" in the first case, and over the entire Scene in the second. Third, by adding sub-units to a terminal. For instance, consider "gave up", an expression which the foundational layer considers atomic. A layer that annotates tense can break the expression into "gave" and "up", in order to annotate "gave" as the tense-bearing unit.

Although a more complete discussion of the formalism is beyond the scope of this paper, we note that the formalism is designed to allow different annotation layers to be defined and annotated independently of one another, in order to facilitate UCCA's construction through a community effort.

	Passage #					
	1	2	3	4	5	6
# Sents.	8	20	23	14	13	15
# Tokens	259	360	343	322	316	393
ITA	67.3	74.1	71.2	73.5	77.8	81.1
Vs. Gold	72.4	76.7	75.5	75.7	79.5	84.2
Correction	**93.7**					

Table 2: The upper part of the table presents the number of sentences and the number of tokens in the first passages used for the annotator training. The middle part presents the average F-scores obtained by the annotators throughout these passages. The first row presents the average F-score when comparing the annotations of the different annotators among themselves. The second row presents the average F-score when comparing them to a "gold standard". The bottom row shows the average F-score between an annotated passage of a trained annotator and its manual correction by an expert. It is higher due to *conforming analyses* (see text). All F-scores are in percents.

3 A UCCA-Annotated Corpus

The annotated text is mostly based on English Wikipedia articles for celebrities. We have chosen this genre as it is an inclusive and diverse domain, which is still accessible to annotators from varied backgrounds.

For the annotation process, we designed and implemented a web application tailored for UCCA's annotation. A sample of the corpus containing roughly 5K tokens, as well as the annotation application can be found in our website[7].

UCCA's annotations are not confined to a single sentence. The annotation is therefore carried out in passages of 300-400 tokens. After its an-

[7] www.cs.huji.ac.il/~omria01

notation, a passage is manually corrected before being inserted into the repository.

The section of the corpus annotated thus far contains 56890 tokens in 148 annotated passages (average length of 385 tokens). Each passage contains 450 units on average and 42.2 Scenes. Each Scene contains an average of 2 Participants and 0.3 Adverbials. 15% of the Scenes are static (contain an S as the main relation) and the rest are dynamic (containing a P). The average number of tokens in a Scene (excluding punctuation) is 10.7. 18.3% of the Scenes are Participants in another Scene, 11.4% are Elaborator Scenes and the remaining are Parallel Scenes. A passage contains an average of 11.2 Linkers.

Inter-annotator agreement. We employ 4 annotators with varying levels of background in linguistics. Two of the annotators have no background in linguistics, one took an introductory course and one holds a Bachelor's degree in linguistics. The training process of the annotators lasted 30–40 hours, which includes the time required for them to get acquainted with the web application. As this was the first large-scale trial with the UCCA scheme, some modifications to the scheme were made during the annotator's training. We therefore expect the training process to be even faster in later distributions.

There is no standard evaluation measure for comparing two grammatical annotations in the form of labeled DAGs. We therefore converted UCCA to constituency trees[8] and, following standard practice, computed the number of brackets in both trees that match in both span and label. We derive an F-score from these counts.

Table 2 presents the inter-annotator agreement in the training phase. The four annotators were given the same passage in each of these cases. In addition, a "gold standard" was annotated by the authors of this paper. The table presents the average F-score between the annotators, as well as the average F-score when comparing to the gold standard. Results show that although it represents complex hierarchical structures, the UCCA scheme is learned quickly and effectively.

We also examined the influence of prior linguistic background on the results. In the first passage there was a substantial advantage to the annotators who had prior training in linguistics. The obtained F-scores when comparing to a gold standard, ordered decreasingly according to the annotator's acquaintance with linguistics, were 78%, 74.4%, 69.5% and 67.8%. However, this performance gap quickly vanished. Indeed, the obtained F-scores, again compared to a gold standard and averaged over the next five training passages, were (by the same order) 78.6%, 77.3%, 79.2% and 78%.

This is an advantage of UCCA over other syntactic annotation schemes that normally require highly proficient annotators. For instance, both the PTB and the Prague Dependency Treebank (Böhmová et al., 2003) employed annotators with extensive linguistic background. Similar findings to ours were reported in the PropBank project, which successfully employed annotators with various levels of linguistic background. We view this as a major advantage of semantic annotation schemes over their syntactic counterparts, especially given the huge amount of manual labor required for large syntactic annotation projects.

The UCCA interface allows for multiple non-contradictory ("conforming") analyses of a stretch of text. It assumes that in some cases there is more than one acceptable option, each highlighting a different aspect of meaning of the analyzed utterance (see below). This makes the computation of inter-annotator agreement fairly difficult. It also suggests that the above evaluation is excessively strict, as it does not take into account such conforming analyses. To address this issue, we conducted another experiment where an expert annotator corrected the produced annotations. Comparing the corrected versions to the originals, we found that F-scores are typically in the range of 90%–95%. An average taken over a sample of passages annotated by all four annotators yielded an F-score of 93.7%.

It is difficult to compare the above results to the inter-annotator agreement of other projects for two reasons. First, many existing schemes are based on other annotation schemes or heavily rely on automatic tools for providing partial annotations. Second, some of the most prominent annotation projects do not provide reliable inter-annotator agreement scores (Artstein and Poesio, 2008).

A recent work that did report inter-annotator agreement in terms of bracketing F-score is an annotation project of the PTB's noun phrases with more elaborate syntactic structure (Vadas and Cur-

[8] In cases a unit had multiple parents, we discarded all but one of its incoming edges. This resulted in discarding 1.9% of the edges. We applied a simple normalization procedure to the resulting trees.

ran, 2011). They report an agreement of 88.3% in a scenario where their two annotators worked separately. Note that this task is much more limited in scope than UCCA (annotates noun phrases instead of complete passages in UCCA; uses 2 categories instead of 12 in UCCA). Nevertheless, the obtained inter-annotator agreement is comparable.

Disagreement examples. Here we discuss two major types of disagreements that recurred in the training process. The first is the distinction between Elaborators and Centers. In most cases this distinction is straightforward, particularly where one sub-unit determines the semantic type of the parent unit, while its siblings add more information to it (e.g., "truck$_E$ company$_C$" is a type of a company and not of a truck). Some structures do not nicely fall into this pattern. One such case is with apposition. In the example "the Fox drama Glory days", both "the Fox drama" and "Glory days" are reasonable candidates for being a Center, which results in disagreements.

Another case is the distinction between Scenes and non-Scene relations. Consider the example "[John's portrayal of the character] has been described as ...". The sentence obviously contains two scenes, one in which John portrays a character and another where someone describes John's doings. Its internal structure is therefore "John's$_A$ portrayal$_P$ [of the character]$_A$". However, the syntactic structure of this unit leads annotators at times into analyzing the subject as a non-Scene relation whose C is "portrayal".

Static relations tend to be more ambiguous between a Scene and a non-Scene interpretation. Consider "Jane Smith (née Ross)". It is not at all clear whether "née Ross" should be annotated as a Scene or not. Even if we do assume it is a Scene, it is not clear whether the Scene it evokes is her Scene of birth, which is dynamic, or a static Scene which can be paraphrased as "originally named Ross". This leads to several conforming analyses, each expressing a somewhat different conceptualization of the Scene. This central notion will be more elaborately addressed in future work.

We note that all of these disagreements can be easily resolved by introducing an additional layer focusing on the construction in question.

4 UCCA's Benefits to Semantic Tasks

UCCA's relative insensitivity to syntactic forms has potential benefits for a wide variety of semantic tasks. This section briefly demonstrates these benefits through a number of examples.

Recall the example "John took a shower" (Section 1). UCCA annotates the sentence as a single Scene, with a single Participant and a processual main relation: "John$_A$ [took$_F$ [a$_E$ shower$_C$]$_C$]$_P$". The paraphrase "John showered" is annotated similarly: "John$_A$ showered$_P$". The structure is also preserved under translation to other languages, such as German ("John$_A$ duschte$_P$", where "duschte" is a verb), or Portuguese "John$_A$ [tomou$_F$ banho$_C$]$_P$" (literally, John took shower). In all of these cases, UCCA annotates the example as a Scene with an A and a P, whose Center is a word expressing the notion of showering.

Another example is the sentence "John does not have any money". The foundational layer of UCCA annotates negation units as Ds, which yields the annotation "John$_A$ [does$_F$]$_S$- not$_D$ [have$_C$]-$_S$ [any$_E$ money$_C$]$_A$" (where "does ... have" is a discontiguous unit)[9]. This sentence can be paraphrased as "John$_A$ has$_P$ no$_D$ money$_A$". UCCA reflects the similarity of these two sentences, as it annotates both cases as a single Scene which has two Participants and a negation. A syntactic scheme would normally annotate "no" in the second sentence as a modifier of "money", and "not" as a negation of "have".

The value of UCCA's annotation can again be seen in translation to languages that have only one of these forms. For instance, the German translation of this sentence, "John$_A$ hat$_S$ kein$_D$ Geld$_A$", is a literal translation of "John has no money". The Hebrew translation of this sentence is "eyn le john kesef" (literally, "there-is-no to John money"). The main relation here is therefore "eyn" (there-is-no) which will be annotated as S. This yields the annotation "eyn$_S$ [le$_R$ John$_C$]$_A$ kesef$_A$".

The UCCA annotation in all of these cases is composed of two Participants and a State. In English and German, the negative polarity unit is represented as a D. The negative polarity of the Hebrew "eyn" is represented in a more detailed layer.

As a third example, consider the two sentences "There are children playing in the park" and "Children are playing in the park". The two sentences have a similar meaning but substantially different syntactic structures. The first contains two clauses, an existential main clause (headed by "there are")

[9]The foundational layer places "not" in the Scene level to avoid resolving fine scope issues (see Section 2).

and a subordinate clause ("playing in the park"). The second contains a simple clause headed by "playing". While the parse trees of these sentences are very different, their UCCA annotation in the foundational layer differ only in terms of Function units: "Children$_A$ [are$_F$ playing$_C$]$_P$ [in$_R$ the$_E$ park$_C$]$_A$" and "There$_F$ are$_F$ children$_A$ [playing]$_P$ [in$_R$ the$_E$ park$_C$]$_A$"[10].

Aside from machine translation, a great variety of semantic tasks can benefit from a scheme that is relatively insensitive to syntactic variation. Examples include text simplification (e.g., for second language teaching) (Siddharthan, 2006), paraphrase detection (Dolan et al., 2004), summarization (Knight and Marcu, 2000), and question answering (Wang et al., 2007).

5 Related Work

In this section we compare UCCA to some of the major approaches to grammatical representation in NLP. We focus on English, which is the most studied language and the focus of this paper.

Syntactic annotation schemes come in many forms, from lexical categories such as POS tags to intricate hierarchical structures. Some formalisms focus particularly on syntactic distinctions, while others model the syntax-semantics interface as well (Kaplan and Bresnan, 1981; Pollard and Sag, 1994; Joshi and Schabes, 1997; Steedman, 2001; Sag, 2010, *inter alia*). UCCA diverges from these approaches in aiming to abstract away from specific syntactic forms and to only represent semantic distinctions. Put differently, UCCA advocates an approach that treats syntax as a hidden layer when learning the mapping between form and meaning, while existing syntactic approaches aim to model it manually and explicitly.

UCCA does not build on any other annotation layers and therefore implicitly assumes that semantic annotation can be learned directly. Recent work suggests that indeed structured prediction methods have reached sufficient maturity to allow direct learning of semantic distinctions. Examples include Naradowsky et al. (2012) for semantic role labeling and Kwiatkowski et al. (2010) for semantic parsing to logical forms. While structured prediction for the task of predicting tree structures is already well established (e.g., (Suzuki et al., 2009)), recent work has also successfully tackled the task of predicting semantic structures in the form of DAGs (Jones et al., 2012).

The most prominent annotation scheme in NLP for English syntax is the Penn Treebank. Many syntactic schemes are built or derived from it. An increasingly popular alternative to the PTB are dependency structures, which are usually represented as trees whose nodes are the words of the sentence (Ivanova et al., 2012). Such representations are limited due to their inability to naturally represent constructions that have more than one head, or in which the identity of the head is not clear. They also face difficulties in representing units that participate in multiple relations. UCCA proposes a different formalism that addresses these problems by introducing a new node for every relation (cf. (Sangati and Mazza, 2009)).

Several annotated corpora offer a joint syntactic and semantic representation. Examples include the Groningen Meaning bank (Basile et al., 2012), Treebank Semantics (Butler and Yoshimoto, 2012) and the Lingo Redwoods treebank (Oepen et al., 2004). UCCA diverges from these projects in aiming to abstract away from syntactic variation, and is therefore less coupled with a specific syntactic theory.

A different strand of work addresses the construction of an interlingual representation, often with a motivation of applying it to machine translation. Examples include the UNL project (Uchida and Zhu, 2001), the IAMTC project (Dorr et al., 2010) and the AMR project (Banarescu et al., 2012). These projects share with UCCA their emphasis on cross-linguistically valid annotations, but diverge from UCCA in three important respects. First, UCCA emphasizes the notion of a multi-layer structure where the basic layers are maximally coarse-grained, in contrast to the above works that use far more elaborate representations. Second, from a theoretical point of view, UCCA differs from these works in aiming to represent conceptual semantics, building on works in Cognitive Linguistics (e.g., (Langacker, 2008)). Third, unlike interlingua that generally define abstract representations that may correspond to several different texts, UCCA incorporates the text into its structure, thereby facilitating learning.

Semantic role labeling (SRL) schemes bear similarity to the foundational layer, due to their focus on argument structure. The leading SRL ap-

[10] The two sentences are somewhat different in terms of their information structure (Van Valin Jr., 2005), which is represented in a more detailed UCCA layer.

proaches are PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004) on the one hand, and FrameNet (Baker et al., 1998) on the other. At this point, all these schemes provide a more fine-grained set of categories than UCCA.

PropBank and NomBank are built on top of the PTB annotation, and provide for each verb (PropBank) and noun (NomBank), a delineation of their arguments and their categorization into semantic roles. Their structures therefore follow the syntax of English quite closely. UCCA is generally less tailored to the syntax of English (e.g., see secondary verbs (Dixon, 2005)).

Furthermore, PropBank and NomBank do not annotate the internal structure of their arguments. Indeed, the construction of the commonly used semantic dependencies derived from these schemes (Surdeanu et al., 2008) required a set of syntactic head percolation rules to be used. These rules are somewhat arbitrary (Schwartz et al., 2011), do not support multiple heads, and often reflect syntactic rather than semantic considerations (e.g., "millions" is the head of "millions of dollars", while "dollars" is the head of "five million dollars").

Another difference is that PropBank and NomBank each annotate only a subset of predicates, while UCCA is more inclusive. This difference is most apparent in cases where a single complex predicate contains both nominal and verbal components (e.g., "limit access", "take a shower"). In addition, neither PropBank nor Nomabnk address copula clauses, despite their frequency. Finally, unlike PropBank and NomBank, UCCA's foundational layer annotates linkage relations.

In order to quantify the similarity between UCCA and PropBank, we annotated 30 sentences from the PropBank corpus with their UCCA annotations and converted the outcome to PropBank-style annotations[11]. We obtained an unlabeled F-score of 89.4% when comparing to PropBank, which indicates that PropBank-style annotations are generally derivable from UCCA's. The disagreement between the schemes reflects both annotation conventions and principle differences, some of which were discussed above.

The FrameNet project (Baker et al., 1998) proposes a comprehensive approach to semantic roles. It defines a lexical database of Frames, each containing a set of possible frame elements and their semantic roles. It bears similarity to UCCA both in its use of Frames, which are a context-independent abstraction of UCCA's Scenes, and in its emphasis on semantic rather than distributional considerations. However, despite these similarities, FrameNet focuses on constructing a lexical resource covering specific cases of interest, and does not provide a fully annotated corpus of naturally occurring text. UCCA's foundational layer can be seen as a complementary effort to FrameNet, as it focuses on high-coverage, coarse-grained annotation, while FrameNet is more fine-grained at the expense of coverage.

[11]The experiment was conducted on the first 30 sentences of section 02. The identity of the predicates was determined according to the PropBank annotation. We applied a simple conversion procedure that uses half a dozen rules that are not conditioned on any lexical item. We used a strict evaluation that requires an exact match in the argument's boundaries.

6 Conclusion

This paper presented Universal Conceptual Cognitive Annotation (UCCA), a novel framework for semantic representation. We described the foundational layer of UCCA and the compilation of a UCCA-annotated corpus. We demonstrated UCCA's relative insensitivity to paraphrases and cross-linguistic syntactic variation. We also discussed UCCA's accessibility to annotators with no background in linguistics, which can alleviate the almost prohibitive annotation costs of large syntactic annotation projects.

UCCA's representation is guided by conceptual notions and has its roots in the Cognitive Linguistics tradition and specifically in Cognitive Grammar (Langacker, 2008). These theories represent the meaning of an utterance according to the mental representations it evokes and not according to its reference in the world. Future work will explore options to further reduce manual annotation, possibly by combining texts with visual inputs during training.

We are currently attempting to construct a parser for UCCA and to apply it to several semantic tasks, notably English-French machine translation. Future work will also discuss UCCA's portability across domains. We intend to show that UCCA, which is less sensitive to the idiosyncrasies of a specific domain, can be easily adapted to highly dynamic domains such as social media.

Acknowledgements. We would like to thank Tomer Eshet for partnering in the development of the web application and Amit Beka for his help with UCCA's software and development set.

References

Omri Abend and Ari Rappoport. 2013. UCCA: A semantics-based grammatical annotation scheme. In *IWCS '13*, pages 1–12.

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley Framenet project. In *ACL-COLING '98*, pages 86–90.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (AMR) 1.0 specification. http://www.isi.edu/natural-language/people/amr-guidelines-10-31-12.pdf.

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *LREC '12*, pages 3196–3200.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank. *Treebanks*, pages 103–127.

Alistair Butler and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology*, 7(1).

Alexander Clark and Shalom Lappin. 2010. *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Robert M. W. Dixon. 2005. *A Semantic Approach To English Grammar*. Oxford University Press.

Robert M. W. Dixon. 2010a. *Basic Linguistic Theory: Methodology*, volume 1. Oxford University Press.

Robert M. W. Dixon. 2010b. *Basic Linguistic Theory: Grammatical Topics*, volume 2. Oxford University Press.

Robert M. W. Dixon. 2012. *Basic Linguistic Theory: Further Grammatical Topics*, volume 3. Oxford University Press.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING '04*, pages 350–356.

Bonnie Dorr, Rebecca Passonneau, David Farwell, Rebecca Green, Nizar Habash, Stephen Helmreich, Edward Hovy, Lori Levin, Keith Miller, Teruko Mitamura, Owen Rambow, and Advaith Siddharthan. 2010. Interlingual annotation of parallel text corpora: A new framework for annotation and evaluation. *Natural Language Engineering*, 16(3):197–243.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In *LAW '12*, pages 2–11.

Bevan Jones, Jacob Andreas, Daniel Bauer, Karl Moritz Hermann, and Kevin Knight. 2012. Semantics-based machine translation with hyperedge replacement grammars. In *COLING '12*, pages 1359–1376.

Aravind K. Joshi and Yves Schabes. 1997. Tree-adjoining grammars. *Handbook Of Formal Languages*, 3:69–123.

Ronald M. Kaplan and Joan Bresnan. 1981. *Lexical-Functional Grammar: A Formal System For Grammatical Representation*. Massachusetts Institute Of Technology, Center For Cognitive Science.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization – step one: Sentence compression. In *AAAI '00*, pages 703–710.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *EMNLP '10*, pages 1223–1233.

R.W. Langacker. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford University Press, USA.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for Nombank. In *LREC '04*, pages 803–806.

Jason Naradowsky, Sebastian Riedel, and David Smith. 2012. Improving NLP through marginalization of hidden syntactic structure. In *EMNLP '12*, pages 810–820.

Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christopher D Manning. 2004. Lingo redwoods. *Research on Language and Computation*, 2(4):575–596.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):145–159.

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. University Of Chicago Press.

Ivan A Sag. 2010. Sign-based construction grammar: An informal synopsis. *Sign-based Construction Grammar. CSLI Publications, Stanford*, pages 39–170.

Federico Sangati and Chiara Mazza. 2009. An English dependency treebank à la Tesnière. In *TLT '09*, pages 173–184.

Roy Schwartz, Omri Abend, Roi Reichart, and Ari Rappoport. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL-HLT '11*, pages 663–672.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language & Computation*, 4(1):77–109.

Mark Steedman. 2001. *The Syntactic Process*. MIT Press.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL '08*, pages 159–177.

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An empirical study of semi-supervised structured conditional models for dependency parsing. In *EMNLP '09*, pages 551–560.

Hiroshi Uchida and Meiying Zhu. 2001. The universal networking language beyond machine translation. In *International Symposium on Language in Cyberspace*, pages 26–27.

David Vadas and James R Curran. 2011. Parsing noun phrases in the Penn Treebank. *Computational Linguistics*, 37(4):753–809.

Robert D. Van Valin Jr. 2005. *Exploring The Syntax-semantics Interface*. Cambridge University Press.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CoNLL '07*, pages 22–32.

Chapter 8

Discussion

This dissertation addresses the representation of semantic structure in NLP, the resources required to represent it and its unsupervised induction. The criteria set out by this thesis for evaluating a semantic representation are (1) domain generality and cross-linguistic applicability, (2) the ability to express a wide range of semantic distinctions, (3) the learnability of the proposed structures using statistical methods, (4) the annotation costs required for obtaining such a representation, and (5) the constraints imposed on such a scheme by other (notably syntactic) schemes. This thesis is a step towards the formulation of a semantic representation that fully complies to these criteria. It explores two complementary lines of work:

1. Completely unsupervised induction of grammatical and semantic distinctions (chapters 3, 4 and 5). This section explores the extent to which such distinctions can be induced given sufficient amounts of text, but without any manual annotation. These methods are appealing both for their applicability to a wide variety of domains and for the minimal manual effort they require.
2. Manual semantic annotation (chapters 6 and 7). In this section I present UCCA, a manual annotation scheme that addresses many of the desiderata defined above. I also demonstrate that the scheme can be consistently applied to naturally occurring text.

The thesis presents unsupervised algorithms for three core NLP tasks, namely the induction of POS categories (Chapter 3), the identification of verbal arguments (Chapter 4) and their classification into core arguments and adjuncts (Chapter 5). It presents novel algorithms for the three tasks, and in the latter two chapters, presents the first work to tackle a scenario where no supervised syntactic parsers are used.

Chapter 3 presents a novel algorithm for the part of speech induction task. The contribution of Chapter 3 is four-fold. First, the algorithm obtains the best reported results for this core task at the time of publication. Second, the algorithm is novel in using a distributional representation derived from the internal representation of an unsupervised parser (Seginer, 2007). This representation has previously been applied to the task of unsupervised parsing, and we demonstrate its applicability to POS induction as well. Third, we use a non-heuristic morphological representation based on *morphological signatures* (Goldsmith, 2001) derived from an unsupervised morphological model, Morfessor (Creutz and Lagus, 2005). Fourth, we apply a novel two-step algorithm inspired by the cognitive theory of prototypes (Taylor, 2003).

Chapter 4 presents an unsupervised algorithm for identifying verbal arguments. This is the first work that tackles this task in an unsupervised setting. The algorithm works in two steps. It first detects the minimal clause that contains the verb using a novel unsupervised clause detection algorithm. It then employs a selectional preferences module that filters out spurious arguments if they are negatively correlated with the verb in question. Our algorithm outperforms a strong baseline in the two tested languages, English and Spanish.

Chapter 5 presents the first completely unsupervised algorithm for classifying verbal arguments into cores and adjuncts. This task has been tackled in the past using supervised and semi-supervised methods, but never in a completely unsupervised scenario. The work defines several measures that can be computed from plain text, and provide a quantitative measure that correlates with the distinction. As the distinction between cores and adjuncts is never a fast and hard one (Dowty, 2000), using quantitative measures can assist in defining this distinction more coherently using data-driven methods.

Despite the appeal of unsupervised methods, they are limited by the impoverished input they receive. In using only plain text, unsupervised methods focus only on the distributional aspects of language, and ignore much of its semantic and communicative aspects. However, unsupervised methods still provide valuable information and are effectively used as components in semi-supervised parsing systems (Koo et al., 2008) and in our current efforts to construct a UCCA parser (see below). They are also used in state of the art application systems (Uszkoreit and Brants, 2008; Oh et al., 2012).

In order to complement unsupervised methods, the second part of this thesis discusses manual semantic annotation. It presents UCCA, a novel semantic scheme that aims to accommodate rich semantic and abstract away from syntactic variation. UCCA is supported by extensive typological and cognitive linguistic theory. In terms of typological theory, UCCA builds on

Basic Linguistic Theory (BLT) (Dixon, 2005, 2010a,b, 2012), a descriptive framework whose principles were previously applied to a wide variety of languages (Dryer, 2006). The framework uses a combination of syntactic and semantic criteria for defining its constructions, and uses cross-linguistically motivated notions. UCCA generally adopts the semantic component of these definitions in the definition of its categories, leaving the syntactic (or distributional) categorizations to be automatically discovered in subsequent work. UCCA also shares many of the motivations discussed in the cognitive linguistics literature. First, UCCA bases many of its notions on *conceptual* rather than *extensional* semantics. Extensional semantics relates text to the entities and the relations it describes in the some reference world and is more inclined to objectivist descriptions. On the other hand, conceptual semantics focuses on the mental images and scenes a text evokes and their subjective construal (Langacker, 2008). Second, if successful, the UCCA project would support the claim that grammatical regularities are semantically motivated and that the role of syntactic bias in the acquisition of grammar is limited (Clark and Lappin, 2010). This motivation is shared by much work in cognitive linguistics (e.g., (Tomasello, 2009)).

Chapter 6 presents the UCCA framework and discusses UCCA's rationale and the representational as well as the algorithmic approach it advocates. Concretely, it advances the approach that only semantic distinctions should be manually annotated, while distributional regularities should be automatically induced. The chapter further provides a detailed description of UCCA's foundational layer. The foundational layer covers many of the most basic semantic components conveyed through linguistic utterances, including predicate-argument structure and the linkage relations between such structures. The foundational layer is designed to be highly coarse-grained, thereby exposing similarities even between relatively distant domains. The UCCA framework is extendable and is able to accommodate a large range of semantic distinctions. The chapter provides a comparison of the UCCA formalism to the standard dependency formalism, as well as a discussion of UCCA's treatment of the core-adjunct distinction.

Chapter 7 further discusses the UCCA framework, but focuses on the compilation UCCA-annotated corpus and its potential applicability to several core semantic applications. It also discusses the structure of the corpus, its compilation process and demonstrates that unlike common syntactic annotation schemes, UCCA can be effectively annotated by annotators with no background in linguistics. It also demonstrates that UCCA can be effectively learned in a reasonable time, yielding high agreement rates between its annotators. Last, the chapter provides a detailed comparison of UCCA with other semantically annotated corpora used in NLP.

Conclusion and Future Prospects

The focus of this thesis is on highly coarse-grained distinctions. It is my view that semantic representation should rely on a rich set of features, conveying fine-grained lexical and structural information. Current efforts by my colleagues and me are focused on constructing layers for encoding semantic roles, information and focus structure phenomena, and a finer categorization of inter-scene and discourse relations. In order to allow for the extension of UCCA through community effort, we are currently developing a formal and algorithmic framework for the distributed design of the scheme. We are also planning to integrate UCCA to the extensive lexical resources available to the NLP community (e.g., FrameNet (Baker et al., 1998) and VerbNet (Schuler, 2005)), in order to exploit the rich semantic information they offer.

In addition, we are devoting efforts to the development of a statistical UCCA parser. Current work is focused on applying *conditional random fields* (Lafferty et al., 2001) to the sub-task of identifying the Scenes and their components, and we intend to further proceed to the prediction of the full hierarchical structures. We are also pursuing the development of a classifier for identifying scene-evoking elements, i.e., words or expressions that constitute the main relation of a scene.

In another strand of work, we are compiling a English-French parallel corpus, thereby examining UCCA's cross-linguistic validity. We intend to use the corpus to train a machine translation system. More generally, we intend to demonstrate the applicability of UCCA to applications that can benefit from elaborate grammatical and semantic information, such as question answering, paraphrase detection and textual entailment.

Finally, I intend to explore whether UCCA can be used to better model the information accessible to an infant during language acquisition. An appealing direction is to use UCCA to annotate the CHILDES (MacWhinney, 2000) corpus of child-directed speech, and to automatically induce grammatical and lexical information from it (cf. (Chang, 2009; Connor et al., 2010; Kwiatkowski et al., 2012)). If successful, such experiments could demonstrate that coarse-grained semantic information, accompanied by a strong statistical system, are sufficient for inducing complex grammatical regularities.

The field of semantic representation is one of the pillars of natural language processing, and is increasingly used in a wide variety of applications. The research presented in this thesis outlines an alternative to the common approach in NLP for semantic representation, both in its characterization and in its learning. I hope this work will constitute another step towards the ambitious goal of constructing a universal, cognitively-motivated, automatically learnable and easily accessible semantic representation.

Bibliography

Abend, O., R. Reichart, and A. Rappoport (2010). Improved unsupervised pos induction through prototype discovery. In *ACL 2010*, pp. 1298–1307.

Agirre, E., K. Gojenola, K. Sarasola, and A. Voutilainen (1998). Towards a single proposal in spelling correction. In *COLING 1998*, pp. 22–28.

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley Framenet project. In *ACL-COLING '98*, pp. 86–90.

Banarescu, L., C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider (2013). Abstract meaning representation for sembanking. In *LAW 2013*.

Banko, M., M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2007). Open information extraction from the web. In *IJCAI*, Volume 7, pp. 2670–2676.

Barzilay, R., M. Elhadad, et al. (1997). Using lexical chains for text summarization. In *The ACL workshop on intelligent scalable text summarization*, Volume 17, pp. 10–17.

Basile, V., J. Bos, K. Evang, and N. Venhuizen (2012). Developing a large semantically annotated corpus. In *LREC 2012*, pp. 3196–3200.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *the Journal of machine Learning research 3*, 993–1022.

Blitzer, J., R. McDonald, and F. Pereira (2006). Domain adaptation with structural correspondence learning. In *EMNLP 2006*, pp. 120–128.

Boguraev, B. and E. Briscoe (1989). Introduction to computational lexicography for natural language processing. In *Computational lexicography for natural language processing*, pp. 1–40.

Böhmová, A., J. Hajič, E. Hajičová, and B. Hladká (2003). The Prague dependency treebank. *Treebanks*, 103–127.

Brants, T. (1997). The negra export format. *CLAUS Report, Saarland University*.

Buchholz, S. and E. Marsi (2006). Conll-x shared task on multilingual dependency parsing. In *CoNLL 2006*, pp. 149–164.

Chang, M.-W., D. Goldwasser, D. Roth, and V. Srikumar (2010). Discriminative learning over constrained latent representations. In *NAACL 2010*, pp. 429–437.

Chang, N. C.-L. (2009). *Constructing grammar: A computational model of the emergence of early constructions*. Ph. D. thesis, EECS Department, University of California, Berkeley.

Christodoulopoulos, C., S. Goldwater, and M. Steedman (2011). A Bayesian mixture model for PoS induction using multiple features. In *EMNLP 2011*, pp. 638–647.

Clark, A. (2003). Combining distributional and morphological information for part of speech induction. In *EACL 2003*, pp. 59–66.

Clark, A. and S. Lappin (2010). *Linguistic Nativism and the Poverty of the Stimulus*. Wiley-Blackwell.

Cohen, S. B., K. Stratos, M. Collins, D. P. Foster, and L. Ungar (2013). Experiments with spectral learning of latent-variable PCFGs. In *NAACL 2013*.

Connor, M., Y. Gertner, C. Fisher, and D. Roth (2010). Starting from scratch in semantic role labeling. In *ACL '10*, pp. 989–998. Association for Computational Linguistics.

Copestake, A., D. Flickinger, C. Pollard, and I. A. Sag (2005). Minimal recursion semantics: An introduction. *Research on Language and Computation 3*(2-3), 281–332.

Creutz, M. and K. Lagus (2005). Inducing the morphological lexicon of a natural language from unannotated text. In *AKRR*.

Croft, W. (2001). *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press.

Croft, W. and D. A. Cruse (2004). *Cognitive linguistics*. Cambridge University Press.

Croft, W. B., D. Metzler, and T. Strohman (2010). *Search engines: Information retrieval in practice*. Addison-Wesley Reading.

Dasgupta, S. and V. Ng (2007). Unsupervised part-of-speech acquisition for resource-scarce languages. In *EMNLP-CoNLL 2007*, pp. 218–227.

Davidov, D., A. Rappoport, and M. Koppel (2007). Fully unsupervised discovery of concept-specific relationships by web mining. In *ACL 2007*, pp. 232.

Deemter, K. v. and R. Kibble (2000). On coreferring: Coreference in muc and related annotation schemes. *Computational linguistics 26*(4), 629–637.

Dixon, R. M. W. (2005). *A Semantic Approach To English Grammar*. Oxford University Press.

Dixon, R. M. W. (2010a). *Basic Linguistic Theory: Grammatical Topics*, Volume 2. Oxford University Press.

Dixon, R. M. W. (2010b). *Basic Linguistic Theory: Methodology*, Volume 1. Oxford University Press.

Dixon, R. M. W. (2012). *Basic Linguistic Theory: Further Grammatical Topics*, Volume 3. Oxford University Press.

Dorr, B., R. Passonneau, D. Farwell, R. Green, N. Habash, S. Helmreich, E. Hovy, L. Levin, K. Miller, T. Mitamura, O. Rambow, and A. Siddharthan (2010). Interlingual annotation of parallel text corpora: A new framework for annotation and evaluation. *Natural Language Engineering 16*, 197.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 547–619.

Dowty, D. (2000). The dual analysis of adjuncts and complements in categorial grammar. In E. Lang, C. Maienborn, and C. Fabricius-Hansen (Eds.), *Modifying Adjuncts*. Walter de Gruyter.

Dryer, M. S. (2006). Descriptive theories, explanatory theories, and basic linguistic theory. *Trends in Linguistics Studies and Monographs 167*, 207.

Finkel, J. R. and C. D. Manning (2009). Nested named entity recognition. In *EMNLP 2009*, pp. 141–150.

Gao, J. and M. Johnson (2008). A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *EMNLP 2008*, pp. 344–352.

Goldberg, A. E. (1995). *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational linguistics 27*(2), 153–198.

Goldwater, S. and T. Griffiths (2007). A fully Bayesian approach to unsupervised part-of-speech tagging. In *ACL 2007*, pp. 744.

Greene, B. B. and G. M. Rubin (1971). *Automatic grammatical tagging of English*. Department of Linguistics, Brown University.

Grenager, T. and C. D. Manning (2006). Unsupervised discovery of a statistical verb lexicon. In *EMNLP 2006*, pp. 1–8.

Grishman, R., C. Macleod, and A. Meyers (1994). Comlex syntax: Building a computational lexicon. In *COLING 1994*, pp. 268–272.

Haghighi, A. and D. Klein (2006). Prototype-driven learning for sequence models. In *NAACL 2006*, pp. 320–327.

Jackendoff, R. (1994). *Patterns in the Mind. Language and Human Nature*. New York: Basic Books.

Johnson, M. (2007). Why doesn't EM find good HMM pos-taggers. In *EMNLP-CoNLL 2007*, pp. 296–305.

Jones, B., J. Andreas, D. Bauer, K. M. Hermann, and K. Knight (2012). Semantics-based machine translation with hyperedge replacement grammars. In *COLING 2012*, pp. 1359–1376.

Joshi, A. K. and Y. Schabes (1997). Tree-adjoining grammars. *Handbook Of Formal Languages 3*, 69–123.

Kaplan, R. M. and J. Bresnan (1981). *Lexical-Functional Grammar: A Formal System For Grammatical Representation*. Massachusetts Institute Of Technology, Center For Cognitive Science.

Koo, T., X. Carreras, and M. Collins (2008). Simple semi-supervised dependency parsing. In *ACL 2008*, pp. 595–603.

Korhonen, A. (2002). Semantically motivated subcategorization acquisition. In *The ACL workshop on Unsupervised lexical acquisition*, Volume 9, pp. 51–58.

Kwiatkowski, T., S. Goldwater, L. Zettlemoyer, and M. Steedman (2012). A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *EACL 2012*, pp. 234–244.

Lafferty, J., A. McCallum, and F. C. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pp. 380–393.

Lamar, M., Y. Maron, M. Johnson, and E. Bienenstock (2010). SVD and clustering for unsupervised pos tagging. In *ACL 2010*, pp. 215–219.

Lang, J. and M. Lapata (2010). Unsupervised induction of semantic roles. In *NAACL 2010*, pp. 939–947.

Langacker, R. W. (1987). *Foundations of Cognitive Grammar: Theoretical prerequisites*, Volume 1. Stanford University Press.

Langacker, R. W. (1991). *Foundations of Cognitive Grammar: Descriptive application*, Volume 2. Stanford University Press.

Langacker, R. W. (2008). *Cognitive grammar: A basic introduction*. Oxford University Press, USA.

Levin, B. and M. R. Hovav (2005). *Argument realization*. Cambridge University Press.

Litkowski, K. C. and O. Hargraves (2005). The preposition project. In *Proceedings of the Second ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, pp. 171–179.

MacWhinney, B. (2000). *The CHILDES Project: Tools for Analyzing Talk. Transcription, format and programs*, Volume 1. Lawrence Erlbaum.

Marcus, M. P., M. A. Marcinkiewicz, and B. Santorini (1993). Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics 19*(2), 313–330.

Màrquez, L., X. Carreras, K. C. Litkowski, and S. Stevenson (2008). Semantic role labeling: an introduction to the special issue. *Computational linguistics 34*(2), 145–159.

Màrquez, L., L. Villarejo, M. Martí, and M. Taulé (2007). Semeval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *Semeval 2007*, pp. 42–47.

McDonald, R., K. Crammer, and F. Pereira (2005). Online large-margin training of dependency parsers. In *ACL 2005*, pp. 91–98. Association for Computational Linguistics.

Melli, G., Y. Wang, Y. Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar, and F. Popowich (2004). Description of Squash, the SFU question answering summary handler for the DUC-2005 summarization task. In *HLT/EMNLP Document Understanding Workshop*.

Meyers, A., R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman (2004). The nombank project: An interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*, pp. 24–31.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM 38*(11), 39–41.

Moon, T., K. Erk, and J. Baldridge (2010). Crouching Dirichlet, hidden Markov model: Unsupervised pos tagging with context local tag generation. In *EMNLP 2010*, pp. 196–206. Association for Computational Linguistics.

Naradowsky, J., S. Riedel, and D. A. Smith (2012). Improving nlp through marginalization of hidden syntactic structure. In *EMNLP-CoNLL 2012*, pp. 810–820.

Narayanan, S. and S. Harabagiu (2004). Question answering based on semantic structures. In *COLING 2004*, pp. 693.

Oepen, S., D. Flickinger, K. Toutanova, and C. D. Manning (2004). Lingo redwoods. *Research on Language and Computation 2*(4), 575–596.

Oh, J.-H., K. Torisawa, C. Hashimoto, T. Kawada, S. De Saeger, J. Kazama, and Y. Wang (2012). Why question answering using sentiment analysis and word classes. In *EMNLP-CoNLL 2012*, pp. 368–378.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics 31*(1), 145–159.

Pantel, P. and M. Pennacchiotti (2006). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *ACL-COLING 2006*, pp. 113–120.

Pollard, C. and I. A. Sag (1994). *Head-driven Phrase Structure Grammar*. University Of Chicago Press.

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

Prasad, R., N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, and B. L. Webber (2008). The penn discourse treebank 2.0. In *LREC*, pp. 2961–2968. Citeseer.

Price, P. (1990). Evaluation of spoken language systems: The atis domain. In *The Third DARPA Speech and Natural Language Workshop*, pp. 91–95.

Reichart, R. and A. Rappoport (2007). Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL 2007*, pp. 616.

Riloff, E. and R. Jones (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI 1999*, pp. 474–479.

Sagae, K. and J. Tsujii (2008). Shift-reduce dependency dag parsing. In *COLING 2008*, pp. 753–760.

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph. D. thesis, University of Pennsylvania.

Schulte Im Walde, S. (2006). Experiments on the automatic induction of german semantic verb classes. *Computational Linguistics 32*(2), 159–194.

Schütze, H. (1995). Distributional part-of-speech tagging. In *EACL 1995*, pp. 141–148.

Schwartz, R., O. Abend, and A. Rappoport (2012). Learnability-based syntactic annotation design. In *COLING 2012*, pp. 2405–2422.

Schwartz, R., O. Abend, R. Reichart, and A. Rappoport (2011). Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *ACL 2011*, pp. 663–672.

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *ACL 2007*, pp. 384.

Smith, N. A. and J. Eisner (2006). Annealing structural bias in multilingual weighted grammar induction. In *ACL-COLING 2006*, pp. 569–576.

Spitkovsky, V. I., H. Alshawi, A. X. Chang, and D. Jurafsky (2011). Unsupervised dependency parsing without gold part-of-speech tags. In *EMNLP 2011*, pp. 1281–1290.

Srikumar, V., R. Reichart, M. Sammons, A. Rappoport, and D. Roth (2008). Extraction of entailed semantic relations through syntax-based comma resolution. In *ACL*, pp. 1030–1038.

Srikumar, V. and D. Roth (2013). Modeling semantic relations expressed by prepositions. *TACL 1*, 231–242.

Steedman, M. (2001). *The Syntactic Process*. MIT Press.

Sun, L. and A. Korhonen (2009). Improving verb clustering with automatically acquired selectional preferences. In *EMNLP 2009*, pp. 638–647.

Surdeanu, M., S. Harabagiu, J. Williams, and P. Aarseth (2003). Using predicate-argument structures for information extraction. In *ACL 2003*, pp. 8–15.

Swier, R. and S. Stevenson (2004). Unsupervised semantic role labelling. In *EMNLP 2004*, pp. 102.

Talmy, L. (2000a). *Toward a cognitive semantics: Concept structuring systems*, Volume 1. MIT Press.

Talmy, L. (2000b). *Toward a cognitive semantics: Typology and process in concept structuring*, Volume 2. MIT Press.

Taylor, J. R. (2003). *Linguistic Categorization*. Oxford University Press.

Titov, I. and A. Klementiev (2012). A Bayesian approach to unsupervised semantic role induction. In *EACL 2012*, pp. 12–22.

Tomasello, M. (2009). *Constructing a language: A usage-based theory of language acquisition*. Harvard University Press.

Toutanova, K., D. Klein, C. D. Manning, and Y. Singer (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL 2003*, pp. 173–180.

Uszkoreit, J. and T. Brants (2008). Distributed word clustering for large scale class-based language modeling in machine translation. *ACL 2008*, 755–762.

Verhagen, M., R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky (2007). Semeval-2007 task 15: Tempeval temporal relation identification. In *Semeval 2007*, pp. 75–80.

Vossen, P. (1998). *EuroWordNet: a multilingual database with lexical semantic networks*. Kluwer Academic Boston.

Vossen, P. (2003). Ontologies. In R. Mitkov (Ed.), *The Oxford handbook of computational linguistics*. Oxford University Press.

Wang, M., N. A. Smith, and T. Mitamura (2007). What is the Jeopardy model? A quasi-synchronous grammar for QA. In *EMNLP-CoNLL 2007*, pp. 22–32.

Wong, Y. W. and R. Mooney (2007). Learning synchronous grammars for semantic parsing with lambda calculus. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, Volume 45, pp. 960.

Yamada, K. and K. Knight (2001). A syntax-based statistical translation model. In *ACL 2001*, pp. 523–530.

Zelle, J. M. and R. J. Mooney (1996). Learning to parse database queries using inductive logic programming. In *AAAI 1996*, pp. 1050–1055.

Zettlemoyer, L. S. and M. Collins (2005). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI 2005*, pp. 658–666.