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.

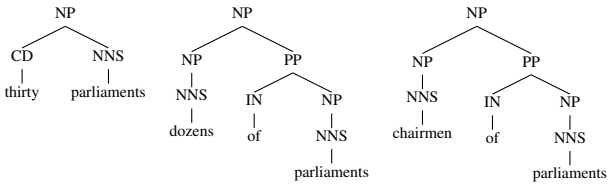

Figure 1: Demonstrating the difference between distributional and semantic representations. The central example is formally more similar to the example on the right, but semantically more similar to the example on the left.

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

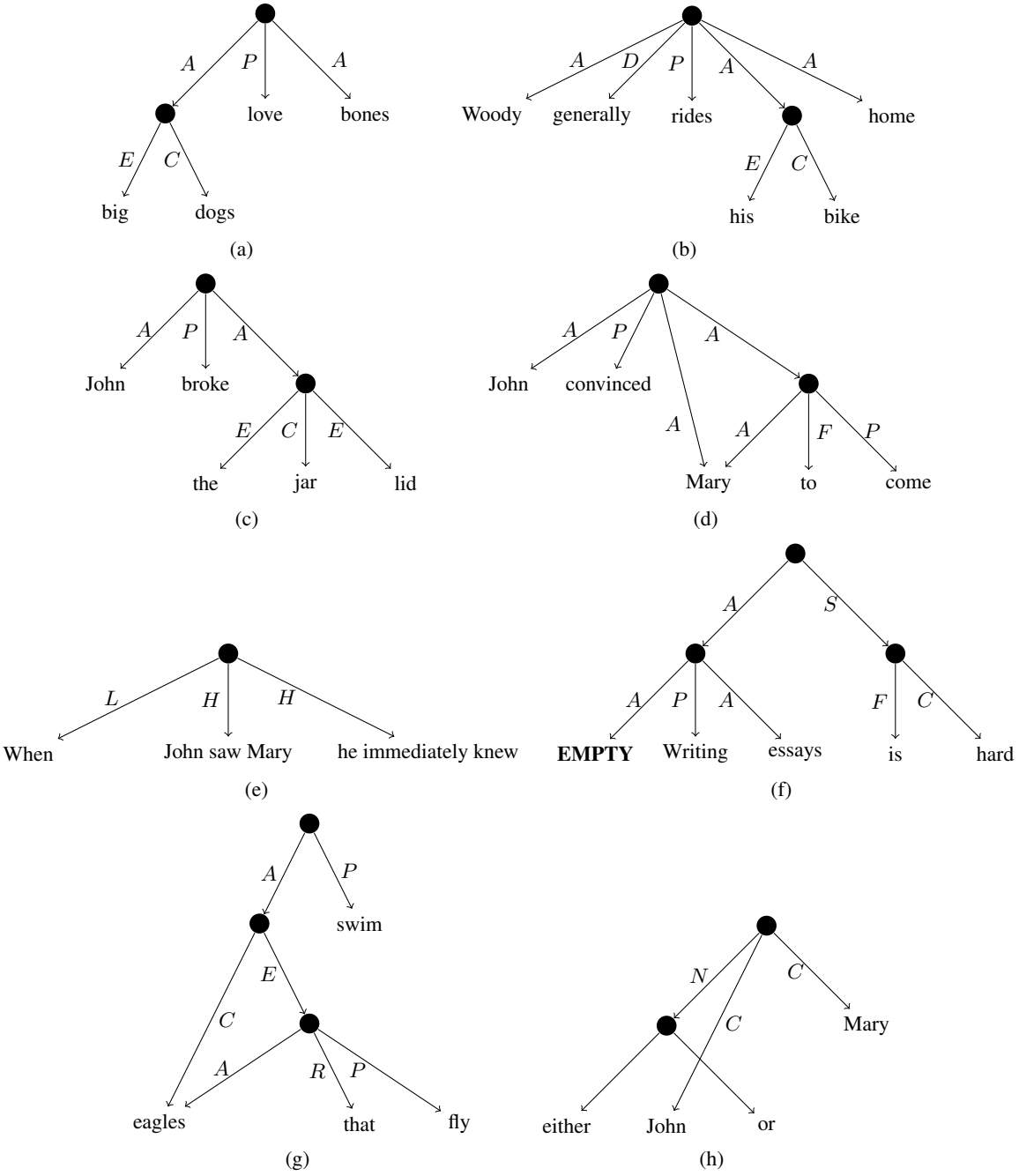

Figure 2: Examples of UCCA annotations.

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.