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).

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-

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

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 228–238,
Sofia, Bulgaria, August 4-9 2013.

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).

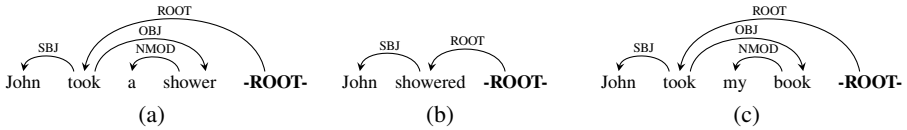

Figure 1: CoNLL-style dependency annotations. Note that (a) and (c), which have different semantics but superficially similar syntax, have the same annotation.

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).

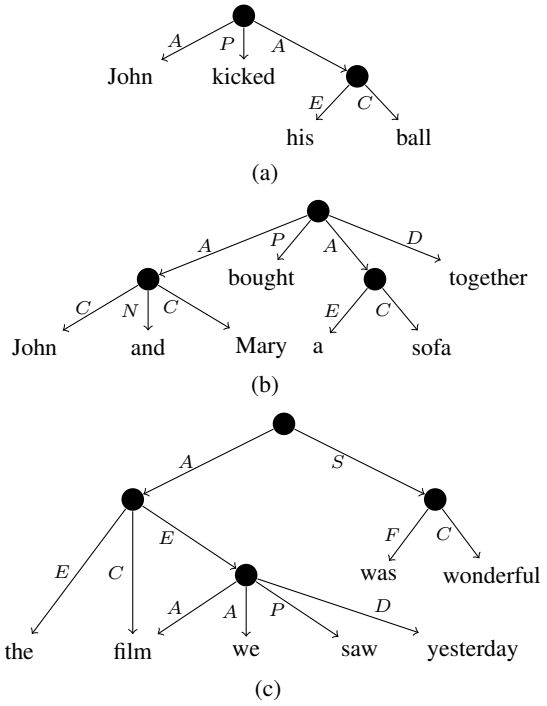

Figure 2: Examples of UCCA annotation graphs.

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 to 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.