Content Differences in Syntactic and Semantic Representations

Daniel Hershcovich[1,2] **Omri Abend**[2] **Ari Rappoport**[2]
[1]The Edmond and Lily Safra Center for Brain Sciences
[2]School of Computer Science and Engineering
Hebrew University of Jerusalem
{danielh,oabend,arir}@cs.huji.ac.il

Abstract

Syntactic analysis plays an important role in semantic parsing, but the nature of this role remains a topic of ongoing debate. The debate has been constrained by the scarcity of empirical comparative studies between syntactic and semantic schemes, which hinders the development of parsing methods informed by the details of target schemes and constructions. We target this gap, and take Universal Dependencies (UD) and UCCA as a test case. After abstracting away from differences of convention or formalism, we find that most content divergences can be ascribed to: (1) UCCA's distinction between a Scene and a non-Scene; (2) UCCA's distinction between primary relations, secondary ones and participants; (3) different treatment of multi-word expressions, and (4) different treatment of inter-clause linkage. We further discuss the long tail of cases where the two schemes take markedly different approaches. Finally, we show that the proposed comparison methodology can be used for fine-grained evaluation of UCCA parsing, highlighting both challenges and potential sources for improvement. The substantial differences between the schemes suggest that semantic parsers are likely to benefit downstream text understanding applications beyond their syntactic counterparts.

1 Introduction

Semantic representations hold promise due to their ability to transparently reflect distinctions relevant for text understanding applications. For example, syntactic representations are usually sensitive to distinctions based on POS (part of speech), such as between compounds and possessives. Semantic schemes are less likely to make this distinction since a possessive can often be paraphrased as a compound and vice versa (e.g., "US president"/"president of the US"), but may distinguish different senses of possessives (e.g., "some of the presidents" and "inauguration of the presidents").

Nevertheless, little empirical study has been done on what distinguishes semantic schemes from syntactic ones, which are still in many cases the backbone of text understanding systems. Such studies are essential for (1) determining whether and to what extent semantic methods should be adopted for text understanding applications; (2) defining better inductive biases for semantic parsers, and allowing better use of information encoded in syntax; (3) pointing at semantic distinctions unlikely to be resolved by syntax.

The importance of such an empirical study is emphasized by the ongoing discussion as to what role syntax should play in semantic parsing, if any (Swayamdipta et al., 2018; Strubell et al., 2018; He et al., 2018; Cai et al., 2018). See §8.

This paper aims to address this gap, focusing on *content* differences. As a test case, we compare relatively similar schemes (§2): the syntactic Universal Dependencies (UD; Nivre et al., 2016), and the semantic Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport, 2013).

We UCCA-annotate the entire web reviews section of the UD EWT corpus (§3), and develop a converter to assimilate UD and UCCA, which use formally different graphs (§4). We then align their nodes, and identify which UCCA categories match which UD relations, and which are unmatched.

Most content differences are due to (§5):

1. UCCA's distinction between words and phrases that evoke Scenes (events) and ones that do not. For example, eventive and non-eventive nouns are treated differently in UCCA, but similarly in UD.

2. UCCA's distinction between primary relations, secondary relations and Participants, in contrast to UD's core/non-core distinction.

3. Different treatment of multi-word expressions (MWEs), where UCCA has a stronger tendency to explicitly mark them.

4. UCCA's conflation of several syntactic realizations of inter-clause linkage, and disambiguation of other cases that UD treats similarly.

We show that the differences between the schemes are substantial, and suggest that UCCA parsing in particular and semantic parsing in general are likely to benefit downstream text understanding applications. For example, only 72.9% of UCCA Participants are UD syntactic arguments, i.e., many semantic participants cannot be recovered from UD.[1] Our findings are relevant to other semantic representations, given their significant overlap in content (Abend and Rappoport, 2017).

A methodology for comparing syntactic and semantic treebanks can also support fine-grained error analysis of semantic parsers, as illustrated by Szubert et al. (2018) for AMR (Banarescu et al., 2013). To demonstrate the utility of our comparison methodology, we perform fine-grained error analysis on UCCA parsing, according to UD relations (§6). Results highlight challenges for current parsing technology, and expose cases where UCCA parsers may benefit from modeling syntactic structure more directly.[2]

2 Representations

The conceptual and formal similarity between UD and UCCA can be traced back to their shared design principles: both are designed to be applicable across languages and domains, to enable rapid annotation and to support text understanding applications. This section provides a brief introduction to each of the schemes, whereas the next sections discuss their content in further detail.[3]

UCCA is a semantic annotation scheme rooted in typological and cognitive linguistic theory. It aims to represent the main semantic phenomena in text, abstracting away from syntactic forms. Shown to be preserved remarkably well across translations (Sulem et al., 2015), it has been applied to improve text simplification (Sulem et al., 2018b), and text-to-text generation evaluation (Birch et al., 2016; Choshen and Abend, 2018; Sulem et al., 2018a).

Participant	A	Linker	L
Center	C	Connector	N
Adverbial	D	Process	P
Elaborator	E	Quantifier	Q
Function	F	Relator	R
Ground	G	State	S
Parallel Scene	H	Time	T

Table 1: Legend of UCCA categories (edge labels).

Formally, UCCA structures are directed acyclic graphs (DAGs) whose nodes (or *units*) correspond either to words, or to elements viewed as a single entity according to some semantic or cognitive consideration. Edges are labeled, indicating the role of a child in the relation the parent represents. Figure 1 shows a legend of UCCA abbreviations. A *Scene* is UCCA's notion of an event or a frame, and is a description of a movement, an action or a state which persists in time. Every Scene contains one primary relation, which can be either a Process or a State. Scenes may contain any number of Participants, a category which also includes abstract participants and locations. They may also contain temporal relations (Time), and secondary relations (Adverbials), which cover semantic distinctions such as manner, modality and aspect.[4]

Scenes may be *linked* to one another in several ways. First, a Scene can provide information about some entity, in which case it is marked as an Elaborator. This often occurs in the case of participles or relative clauses. For example, "(child) who went to school" is an Elaborator Scene in "The child who went to school is John". A Scene may also be a Participant in another Scene. For example, "John went to school" in the sentence: "He said John went to school". In other cases, Scenes are annotated as Parallel Scenes (H), which are flat structures and may include a Linker (L), as in: "When$_L$ [he arrives]$_H$, [he will call them]$_H$".

Non-Scene units are headed by units of the category Center, denoting the type of entity or thing described by the whole unit. Elements in non-Scene units include Quantifiers (such as "*dozens* of people") and Connectors (mostly coordinating conjunctions). Other modifiers to the Center are marked as Elaborators.

UCCA distinguishes *primary* edges, corresponding to explicit relations, from *remote* edges,

[1]This excludes cases of shared argumenthood, which are partially covered by *enhanced UD*. See §4.1.

[2]Our conversion and analysis code is public available at https://github.com/danielhers/synsem.

[3]See Supplementary Material for a definition of each category in both schemes, and their abbreviations.

[4]Despite the similar terminology, UCCA Adverbials are not necessarily adverbs syntactically.

which allow for a unit to participate in several super-ordinate relations. See example in Figure 1. Primary edges form a tree, whereas remote edges (dashed) enable reentrancy, forming a DAG.

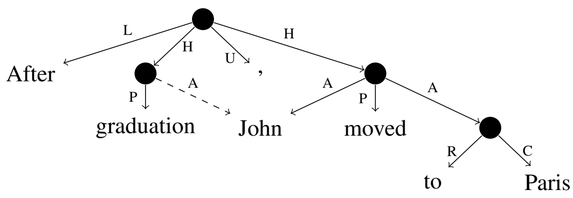


Figure 1: Example UCCA graph. Dashed: a remote edge.

UD is a syntactic dependency scheme used in many languages, aiming for cross-linguistically consistent and coarse-grained treebank annotation. Formally, UD uses bi-lexical trees, with edge labels representing syntactic relations.

One aspect of UD similar to UCCA is its preference of lexical (rather than functional) heads. For example, in auxiliary verb constructions (e.g., "is eating"), UD marks the lexical verb (*eating*) as the head, while other dependency schemes may select the auxiliary *is* instead. While the approaches are largely inter-translatable (Schwartz et al., 2012), lexical head schemes are more similar in form to semantic schemes, such as UCCA and semantic dependencies (Oepen et al., 2016).

Being a dependency representation, UD is structurally underspecified in an important way: it is not possible in UD to mark the distinction between an element modifying the head of the phrase and the same element modifying the whole phrase (de Marneffe and Nivre, 2019).

An example UD tree is given in Figure 2. UD relations will be written in `typewriter` font.

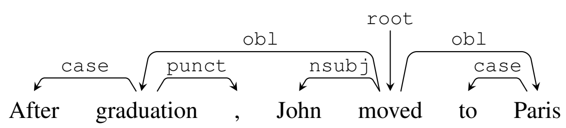


Figure 2: Example UD tree.

3 Shared Gold-standard Corpus

We annotate 723 English passages (3,813 sentences; 52,721 tokens), comprising the web reviews section of the English Web Treebank (EWT; Bies et al., 2012). Text is annotated by two UCCA annotators according to v2.0 of the UCCA guidelines[5] and cross-reviewed. As these sentences are

[5] http://bit.ly/ucca_guidelines_v2

	Train	Dev	Test
# Passages	347	192	184
# Sentences	2,723	554	535
# Tokens	44,804	5,394	5,381

Table 2: Data split for the shared gold-standard corpus.

included in the UD English_EWT treebank, this is a *shared* gold-standard UCCA and UD annotated corpus.[6] We use the standard train/development/test split, shown in Table 2.

4 Comparison Methodology

To facilitate comparison between UCCA and UD, we first assimilate the graphs by abstracting away from formalism differences, obtaining a similar graph format for both schemes. We then match pairs of nodes in the converted UD and UCCA trees if they share all terminals in their yields.

UD annotates bi-lexical dependency trees, while UCCA graphs contain non-terminal nodes. In §4.1, we outline the unified DAG converter by Hershcovich et al. (2018a,b),[7] which we use to reach a common format. In §4.2, we describe a number of extensions to the converter, which abstract away from further non-content differences.

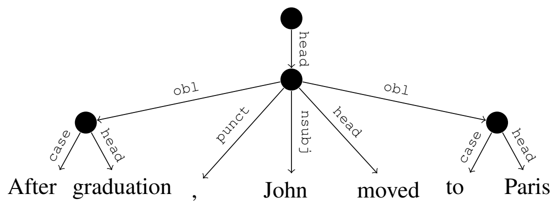


Figure 3: Converted UD tree. Intermediate non-terminals and *head* edges are introduced by the unified DAG converter.

4.1 Basic Conversion

Figure 3 presents the same tree from Figure 2 after conversion. The converter adds one pre-terminal per token, and attaches them according to the original dependency tree: traversing it from the root, for each head it creates a non-terminal parent with the edge label *head*, and adds the dependents as children of the created non-terminal. Relation subtypes are stripped, leaving only universal relations. For example, the language-specific definite article label `det:def` is replaced by the universal `det`.

[6] Our data is available at https://github.com/UniversalConceptualCognitiveAnnotation/UCCA_English-EWT.

[7] https://github.com/huji-nlp/semstr

Reentrancies. Remote edges in UCCA enable reentrancy, forming a DAG together with primary edges. UD allows reentrancy when including *enhanced dependencies* (Schuster and Manning, 2016),[8] which form (bi-lexical) graphs, representing phenomena such as predicate ellipsis (e.g., gapping), and shared arguments due to coordination, control, raising and relative clauses.

UCCA is more inclusive in its use of remote edges, and accounts for the entire class of implicit arguments termed *Constructional Null Instantiation* in FrameNet (Ruppenhofer et al., 2016). For example, in "The Pentagon is bypassing official US intelligence channels [...] in order to create strife" (from EWT), remote edges mark *Pentagon* as a shared argument of *bypassing* and *create*. Another example is "if you call for an appointment [...] so you can then make one", where a remote edge in UCCA indicates that *one* refers to *appointment*. Neither is covered by enhanced UD.

In order to facilitate comparison, we remove remote edges and enhanced dependencies in the conversion process. We thus compare basic UD and UCCA trees, deferring a comparison of UCCA and enhanced UD to future work.

4.2 Extensions to the Converter

We extend the unified DAG converter to remove further non-content differences.

Unanalyzable units. An unanalyzable phrase is represented in UCCA as a single unit covering multiple terminals. In multi-word expressions (MWEs) in UD, each word after the first is attached to the previous word, with the `flat`, `fixed` or `goeswith` relations (depending on whether the expression is grammaticalized, or split by error). We remove edges of these relations and join the corresponding pre-terminals to one unit.

Promotion of conjunctions. The basic conversion generally preserves terminal yields: the set of terminals spanned by a non-terminal is the same as the original dependency yield of its head terminal (e.g., in Figure 3, the yield of the non-terminal headed by *graduation* is "After graduation", the same as that of "graduation" in Figure 2).

Since UD attaches subordinating and coordinating conjunctions to the subsequent conjunct, this results in them being positioned in the same conjunct they relate (e.g., *After* will be included in the first conjunct in "After arriving home, John went to sleep"; *and* will be included in the second conjunct in "John and Mary"). In contrast, UCCA places conjunctions as siblings to their conjuncts (e.g., "[After] [arriving home], [John went to sleep]" and "[John] [and] [Mary]").

To abstract away from these convention differences, we place coordinating and subordinating conjunctions (i.e., `cc`-labeled units, and `mark`-labeled units with an `advcl` head such as *when*, *if*, *after*) as siblings of their conjuncts.

5 Analysis of Divergences

Using the shared format, we turn to analyzing the content differences between UCCA and UD.[9]

5.1 Confusion Matrix

Table 3 presents the confusion matrix of categories between the converted UD and UCCA, calculated over all sentences in the training and development sets of the shared EWT reviews corpus. We leave the test set out of this evaluation to avoid contamination for future parsing experiments.

In case of multiple UCCA units with the same terminal yield (i.e., units with a single non-remote child), we take the top category only, to avoid double-counting. Excluding punctuation, this results in 60,434 yields in UCCA and 58,992 in UD. Of these, 52,280 are common, meaning that a UCCA "parser" developed this way would get a very high F1 score of 87.6%, if it is provided with the gold UCCA label for every converted edge.

Some yields still have more than one UCCA category associated with them, due to edges with multiple categories (**A**|**P** and **A**|**S**). For presentation reasons, 0.15% of the UCCA units in the data are not presented here, as they belong to rare ($<0.1\%$) multiple-category combinations.

Only 82.6% of syntactic arguments (`ccomp`, `csubj`, `iobj`, `nsubj`, `obj`, `obl` and `xcomp`) are Participants, and only 72.9% of Participants are syntactic arguments—a difference stemming from the Scene/non-Scene (§5.2) and argument/adjunct (§5.3) distinctions. Moreover, if we identify predicates as words having at least one argument and Scenes as units with at least one Participant, then only 92.1% of UD predicates correspond to Scenes (many of which are secondary relations within one scene; see §5), and only 80% of

[8] https://universaldependencies.org/u/overview/enhanced-syntax.html

[9] See http://bit.ly/uccaud for a detailed explanation of each example in this section.

	A	A|P	A|S	C	D	E	F	G	H	L	N	P	Q	R	S	T	No Match
`acl`	58			1	4	249	1		48			6			1	1	409
`advcl`	14			12	2	2		6	512	4		11					423
`advmod`	225		1	69	1778	332	27	135	14	258	2	2	15	44	9	368	273
`amod`	25			134	647	837		1	28			7	130	3	269	25	176
`appos`	21			39	2	34			18						8		33
`aux`					384	2	1335			2		1		1			17
`case`	11			31	27	25	123			213	26	11	1	2629	154	1	262
`cc`				8	4	1	4	1	1	1567	381		6	12			52
`ccomp`	345			1		1			36			2			1	1	166
`compound`	225			116	67	586	21		2			32	19	1	12	24	683
`conj`	10			449	4	5		1	1262	1		6	2		10		497
`cop`				1			1312			1		9		10	178		7
`csubj`	13								3								46
`det`	10			17	119	440	2963				1		129	16	1		124
`discourse`	1			2	1		25	29	27	16					5		19
`expl`	21			1			98								17		3
`iobj`	131			1			1										10
`list`	3			7	2	1			27						1		6
`mark`				9	7	1	531	1		654				407	1	5	143
`nmod`	844	1	1	20	9	786	8	4	12	1	1	20	2	2	11	27	488
`nsubj`	4296	7	21	25	3	2	55	1	5	61		58	1	80	14	4	247
`nummod`	2			33	12	17		4		4			334				64
`obj`	1845		1	54	21	6	11	1	4	23		52	1	23	3	11	583
`obl`	1195			19	115	41	1	17	39	34		6	6	26	7	302	611
`parataxis`	6		1	5		4		6	285						3		180
`vocative`	17							8									
`xcomp`	121			4	25				8			38			38		526
head	445	48	159	6388	717	142	564	83	2462	42	1	4163	120	52	1547	32	2235
No Match	1421	37	58	640	417	291	14	33	2291	146	6	802	94	52	369	96	

Table 3: UD-UCCA confusion matrix calculated based on EWT gold-standard annotations from the training and development sets (§3), after applying our extended converter to UD (§4), by matching UD vertices and UCCA units with the same terminal yield. The last column (row), labeled No Match, shows the number of edges of each UD (UCCA) category that do not match any UCCA (UD) unit. Zero counts are omitted.

Scenes correspond to predicates (e.g., due to eventive nouns, which are not syntactic predicates).

Examining the *head* row in Table 3 allows us to contrast the schemes' notions of a head. *head*-labeled units have at least one dependent in UD, or are single-clause sentences (technically, they are non-terminals added by the converter). Of them, 75.7% correspond to Processes, States, Parallel Scenes or Centers, which are UCCA's notions of semantic heads, and 11.6% are left unmatched, mostly due to MWEs analyzed in UD but not in UCCA (§5.4). Another source of unmatched units is inter-Scene linkage, which tends to be flatter in UCCA (§5.5). The rest are mostly due to head swap (e.g., "*all* of Dallas", where *all* is a Quantifier of *Dallas* in UCCA, but the head in UD).

In the following subsections, we review the main content differences between the schemes, as reflected in the confusion matrix, and categorize them according to the UD relations involved.

5.2 Scenes vs. Non-Scenes

UCCA distinguishes between Scenes and non-Scenes. This distinction crosses UD categories, as a Scene can be evoked by a verb, an eventive or stative noun (*negotiation*, *fatigue*), an adjective or even a preposition ("this is *for* John").

Core syntactic arguments. Subjects and objects are usually Participants (e.g., "*wine* was excellent"). However, when describing a Scene, the subject may be a Process/State (e.g., "but *service* is very poor"). Some wh-pronouns are the subjects or objects of a relative clause, but are Linkers or Relators, depending on whether they link Scenes or non-Scenes, respectively. For example, "who" in "overall, Joe is a happy camper *who* has found a great spot" is an `nsubj`, but a Linker. Other arguments are Adverbials or Time (see §5.3), and some do not match any UCCA unit, especially when they are parts of MWEs (see §5.4).

Adjectival modifiers are Adverbials when modifying Scenes ("*romantic* dinner"), States when describing non-Scenes ("*beautiful* hotel") or when semantically predicative ("such a *convenient* location"), or Elaborators where defining inherent properties of non-Scenes ("*medical* school").

Nominal and clausal modifiers. Most are Participants or Elaborators, depending on whether they modify a Scene (e.g., "discount *on services*" and "our decision *to buy when we did*" are Participants, but "*my car's* gears and brakes" and "Some of the younger kids *that work there*" are Elaborators). Unmatched `acl` are often free relative clauses (e.g., in "the prices were worth what *I got*", *what* is the `obj` of *worth* but a Participant of *I got*).

Case markers. While mostly Relators modifying non-Scenes (e.g., "the team *at* Bradley Chevron"), some case markers are Linkers linking Scenes together (e.g., "very informative website *with* a lot of good work"). Others are Elaborators (e.g., "*over* a year") or States when used as the main relation in verbless or copula clauses (e.g., "it is right *on* Wisconsin Ave").

Coordination. Coordinating conjunctions (`cc`) are Connectors where they coordinate non-Scenes (e.g., "Mercedes *and* Dan") or Linkers where they coordinate Scenes (e.g., "outdated *but* not bad"). Similarly, conjuncts and list elements (`conj`, `list`) may be Parallel Scenes (H), or Centers when they are non-Scenes.[10]

Determiners. Articles are Functions, but determiners modifying non-Scenes are Elaborators (e.g., "I will never recommend this gym to *any* woman"). Where modifying Scenes (mostly negation) they are marked as Adverbials. For example, "*no* feathers in stock", "*what* a mistake", and "the rear window had *some* leakage" are all Adverbials.

5.3 Primary and Secondary Relations

UD distinguishes core arguments, adverb modifiers, and obliques (in English UD, the latter mostly correspond to prepositional dependents of verbs). UCCA distinguishes Participants, including locations and abstract entities, from secondary relations (Adverbials), which cover manner, aspect and modality. Adverbials can be verbs (e.g., *begin*, *fail*), prepositional phrases (*with disrespect*), as well as modals, adjectives and adverbs.

Adverbs and obliques. Most UD adverb modifiers are Adverbials (e.g., "I *sometimes* go"), but they may be Participants, mostly in the case of semantic arguments describing location (e.g., *here*). Obliques may be Participants (e.g., "wait *for Nick*"), Time (e.g., "for *over 7 years*") or Adverbials—mostly manner adjuncts (*by far*).

Clausal arguments are Participant Scenes (e.g., "it was great *that they did not charge a service fee*", "did not really know *what I wanted*" or "I asked them *to change it*"). However, when serving as complements to a secondary verb, they will not match any unit in UCCA, as it places secondary verbs on the same level as their primary relation. For example, *to pay* is an `xcomp` in "they have to pay", while the UCCA structure is flat: *have to* is an Adverbial and *pay* is a Process. Single-worded clausal arguments may correspond to a Process/State, as in "this seems *great*".

Auxiliary verbs are Functions (e.g., "*do* not forget"), or Adverbials when they are modals (e.g., "you *can* graduate"). Semi-modals in UD are treated as clausal heads, which take a clausal complement. For example, in "able to do well", UD treats *able* as the head, which takes *do well* as an `xcomp`. UCCA, on the other hand, treats it as an Adverbial, creating a mismatch for `xcomp`.

5.4 Multi-Word Expressions

UD and UCCA treat MWEs differently. In UD they include names, compounds and grammaticalized fixed expressions. UCCA treats names and grammaticalized MWEs as unanalyzable units, but also a range of semantically opaque constructions (e.g., light verbs and idioms). On the other hand, compounds are not necessarily unanalyzable in UCCA, especially if compositional.

Compounds. English compounds are mostly nominal, and are a very heterogeneous category. Most compounds correspond to Elaborators (e.g., "*industry* standard"), or Elaborator Scenes (e.g., "*out-of-place* flat-screen TV"), and many are unanalyzable expressions (e.g., "*mark* up"). Where the head noun evokes a Scene, the dependent is often a Participant (e.g., "*food* craving"), but can also be an Adverbial (e.g., "*first time* buyers") depending on its semantic category. Other compounds in UD are phrasal verbs (e.g., "figure *out*", "cleaned *up*"), which UCCA treats as unanalyzable (leading to unmatched units).

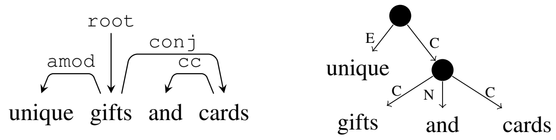
Core arguments. A significant number of subjects and objects are left unmatched as they form parts of MWEs marked in UCCA as unanalyzable. UD annotates MWEs involving a verb and its argument(s) just like any other clause, and therefore lacks this semantic content. Examples include light verbs (e.g., "give *a try*"), idioms ("bites *the dust*"), and figures of speech (e.g., "when *it* comes to", "offer *a taste* (of)"), all are UCCA units.

Complex prepositions. Some complex prepositions (e.g., *according to* or *on top of*), not encoded as MWEs in UD, are unanalyzable in UCCA.

[10] While in UD the conjunction `cc` is attached to the following conjunct, in UCCA coordination is a flat structure. This is a convention difference that we normalize (§4.2).

5.5 Linkage

Head selection. UCCA tends to flatten linkage, where UD, as a dependency scheme, selects a head and dependent per relation. This yields scope ambiguities for coordination, an inherently flat structure. For instance, "unique gifts and cards" is ambiguous in UD as to whether *unique* applies only to *gifts* or to the whole phrase—both annotated as in Figure 4a. UCCA, allowing non-terminal nodes, disambiguates this case (Figure 4b).



(a) UD (b) UCCA

Figure 4: Example for coordination in UD and UCCA.

Clausal dependents. UD categorizes clause linkage into coordination, subordination, argumenthood (complementation), and parataxis. UCCA distinguishes argumenthood but conflates the others into the Parallel Scene category. For example, "We called few companies before *we decided to hire them*" and "Check out The Willow Lounge, *you'll be happy*" are Parallel Scenes.

Note that while in UD, `mark` (e.g., *before*) is attached to the dependent adverbial clause, a UCCA Linker lies outside the linked Scenes. To reduce unmatched `advcl` instances, this convention difference is fixed by the converter (§4.2). Many remaining unmatched units are due to conjunctions we could not reliably raise. For instance, the marker *to* introducing an `xcomp` is ambiguous between Linker (purposive *to*) and Function (infinitive marker). Similarly, wh-pronouns may be Linkers ("he was willing to budge a little on the price *which* means a lot to me"), but have other uses in questions and free relative clauses. Other mismatches result from the long tail of differences in how UD and UCCA construe linkage. Consider the sentence in Figure 5. While *moment* is an oblique argument of *know* in UD, *From the moment* is analyzed as a Linker in UCCA.

5.6 Other Differences

Appositions in UD always follow the modified noun, but named entities in them are UCCA Centers, regardless of position (e.g., in "its sister store Peking Garden", the UD head *its sister store* is an Elaborator, while *Peking Garden* is the Center).

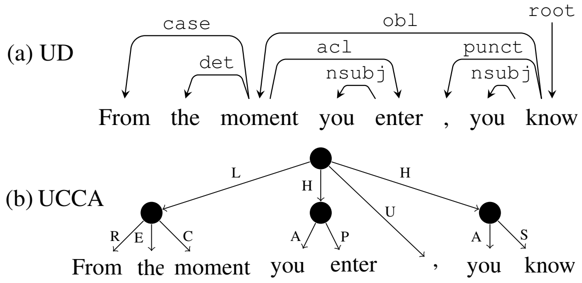


Figure 5: Example for clause linkage in UD and UCCA.

Copulas. UCCA distinguishes copular constructions expressing identity (e.g., "This *is* the original Ham's restaurant") where the copula is annotated as State, and cases of attribution (e.g., "Mercedes and Dan *are* very thorough") or location (e.g., "Excellent chefs *are* in the kitchen"), where the copula is a Function.

Discourse markers and interjections. Units relating a Scene to the speech event or to the speaker's opinion are Ground (e.g., "*no*, Warwick in New Jersey" and "*Please* visit my website"). On the other hand, discourse elements that relate one Scene to another are Linkers (e.g., *anyway*).

Vocatives are both Ground and Participants if they participate in the Scene *and* are the party addressed. For example, *Mark* in "Thanks *Mark*" is both the person addressed and the one thanked.[11]

Expletives and subjects. Expletives are generally Functions, but some instances of *it* and *that* are analyzed as `nsubj` in UD and as Function in UCCA (e.g., "*it*'s like driving a new car").

Excluded relations. We exclude the following UD labels, as they are irrelevant to our evaluation: `root` (always matches the entire sentence); `punct` (punctuation is ignored in UCCA evaluation); `dep` (unspecified dependency), `orphan` (used for gapping, which is represented using remote edges in UCCA—see §4.1); `fixed`, `flat` and `goeswith` (correspond to parts of unanalyzable units in UCCA, and so do not represent units on their own—see §4.2); `reparandum` and `dislocated` (too rare in EWT).

6 Fine-Grained UCCA Parsing Evaluation

In §5 we used our comparison methodology, consisting of the conversion to a shared format and

[11]The **A**|**G** column is omitted from Table 3 as this category combination occurs in only 0.02% of edges in the corpus.

matching units by terminal yield, to compare gold-standard UD and UCCA. In this section we apply the same methodology to parser outputs, using gold-standard UD for fine-grained evaluation.

6.1 Experimental Setup

Data. In addition to the UCCA EWT data (§3), we use the reviews section of the UD v2.3 English_EWT treebank (Nivre et al., 2018),[12] annotated over the exact same sentences. We additionally use UDPipe v1.2 (Straka et al., 2016; Straka and Straková, 2017), trained on English_EWT,[13] for feature extraction. We apply the extended converter to UD as before (§4.2).

Parser. We train TUPA v1.3 (Hershcovich et al., 2017, 2018a) on the UCCA EWT data, with the standard train/development/test split. TUPA uses POS tags and syntactic dependencies as features. We experiment both with using gold UD for feature extraction, and with using UDPipe outputs.

Evaluation by gold-standard UD. UCCA evaluation is generally carried out by considering a predicted unit as correct if there is a gold unit that matches it in terminal yield and labels. Precision, Recall and F-score (F1) are computed accordingly. For the fine-grained analysis, we split the gold-standard, predicted and matched UCCA units according to the labels of the UD relations whose dependents have the same terminal yield (if any).

6.2 Results

Table 4 presents TUPA's scores on the UCCA EWT development and test sets. Surprisingly, using UDPipe for feature extraction results in better scores than gold syntactic tags and dependencies.

	Primary			Remote		
Features	**LP**	**LR**	**LF**	**LP**	**LR**	**LF**
Development						
Gold UD	72.1	71.2	71.7	61.2	38.1	47.0
UDPipe	73.0	72.1	72.5	53.7	40.8	46.4
Test						
Gold UD	72.2	71.2	71.7	60.9	36.8	45.9
UDPipe	72.4	71.7	72.1	60.3	38.5	47.0

Table 4: Labeled precision, recall and F1 (in %) for primary and remote edges output by TUPA on the UCCA EWT development (top) and test (bottom) sets, using either gold-standard UD or UDPipe for TUPA features.

Table 5 shows fine-grained evaluation by UD relations. TUPA does best on auxiliaries and determiners, despite the heterogeneity of corresponding UCCA categories (see Table 3), possibly by making lexical distinctions (e.g., modals and auxiliary verbs are both UD auxiliaries, but are annotated as Adverbials and Functions, respectively).

Copulas and coordinating conjunctions pose a more difficult distinction, since the same lexical items may have different categories depending on the context: State/Function for copulas, due to the distinction between identity and attribution, and Connector/Linker for conjunctions, due to the distinction between Scenes and non-Scenes. However, the reviews domain imposes a strong prior for both (Function and Linker, respectively), which TUPA learns successfully.

Inter-clause linkage (`conj`, `advcl`, `xcomp`, `ccomp`, `parataxis`, `acl` and `csubj`) is a common source of error for TUPA. Although the match between UCCA and UD is not perfect in these cases, it is overall better than TUPA's unlabeled performance, despite using gold-standard syntactic features. Our results thus suggest that encoding syntax more directly, perhaps using syntactic scaffolding (Swayamdipta et al., 2018) or guided attention (Strubell et al., 2018), may assist in predicting unit boundaries. However, TUPA often succeeds at making distinctions that are not even encoded in UD. For example, it does reasonably well (71%) on distinguishing between noun modifiers of Scene-evoking nouns (Participants) and modifiers of other nouns (Elaborators), surpassing a majority baseline based on the UD relation (51%). Lexical resources that distinguish eventive and relational nouns from concrete nouns may allow improving it even further. In the similar case of compounds, lexical resources for light verbs and idioms may increase performance.

7 Discussion

NLP tasks often require semantic distinctions that are difficult to extract from syntactic representations. Consider the example "after graduation, John moved to Paris" again. While *graduation* evokes a Scene (Figure 1), in UD it is an oblique modifier of *moved*, just like *Paris* is (Figure 2). The Scene/non-Scene distinction (§5.2) would assist structural text simplification systems to paraphrase this to "John graduated. (Then,) John moved to Paris", such that each sentence contains

[12] https://hdl.handle.net/11234/1-2895
[13] https://hdl.handle.net/11234/1-2898

		aux	det	cop	cc	expl	iobj	nsubj	case	list	advmod	amod	nummod	mark	vocative	compound	obj	nmod	conj	advcl	obl	xcomp	discourse	ccomp	parataxis	appos	acl	csubj
(a)	**Labeled F1 %**	94	93	89	86	83	83	80	76	76	72	71	71	70	62	59	57	55	50	49	48	41	38	29	23	21	20	0
	Unlabeled F1 %	99	99	100	99	100	83	84	95	76	95	95	86	97	92	84	65	77	61	51	61	63	95	29	36	48	37	33
(b)	**Total in UD #**	156	392	187	212	12	8	463	335	15	378	374	38	116	1	219	222	231	244	52	208	1	16	29	52	22	81	5
	Match Gold #	156	385	187	206	12	6	468	305	12	359	361	33	111	7	146	187	198	210	40	162	28	10	20	48	17	56	4
	Match Predicted #	154	388	187	203	12	6	446	313	9	345	339	32	113	6	136	163	183	177	30	147	26	11	15	30	12	36	2
	Labeled Correct #	145	361	166	175	10	5	365	236	8	253	248	23	78	4	83	99	104	96	17	74	11	4	5	9	3	9	0
	Unlabeled Correct #	154	381	187	203	12	5	386	293	8	336	334	28	109	6	118	113	147	119	18	94	17	10	5	14	7	17	1
(c)	**Labeled/Unlabeled %**	94	95	89	86	83	100	95	81	100	75	74	82	72	67	70	88	71	81	94	79	65	40	100	64	43	53	0
	Mode/Match Gold %	79	82	86	75	58	100	91	79	83	51	35	85	45	71	54	91	51	70	92	68	44	30	94	98	41	72	100
(d)	**Average Words #**	1.0	1.0	1.0	1.0	1.0	1.1	1.6	1.0	2.2	1.2	1.2	1.1	1.0	1.6	1.2	3.0	2.4	5.8	6.6	3.8	6.0	1.1	9.0	6.7	4.0	5.6	7.5

Table 5: Fine-grained evaluation of TUPA (with gold-standard UD features) on the EWT development set. (a) Columns are sorted by labeled F1, measuring performance on each subset of edges. Unlabeled F1 ignores edge categories, evaluating unit boundaries only. (b) Total number of instances of each UD relation; of them, matching UCCA units in gold-standard and in TUPA's predictions; their intersection, with/without regard to categories. (c) Percentage of correctly categorized edges; for comparison, percentage of most frequent category (see Table 3). (d) Average number of words in corresponding terminal yields.

one Scene (Sulem et al., 2018a).

Another example is machine translation—translating the same sentence into Hebrew, which does not have a word for *graduation*, would require a clause to convey the same meaning. The mapping would therefore be more direct using a semantic representation, and we would benefit from breaking the utterance into two Scenes.

8 Related Work

The use of syntactic parsing as a proxy for semantic structure has a long tradition in NLP. Indeed, semantic parsers have leveraged syntax for output space pruning (Xue and Palmer, 2004), syntactic features (Gildea and Jurafsky, 2002; Hershcovich et al., 2017), joint modeling (Surdeanu et al., 2008; Hajič et al., 2009), and multi-task learning (Swayamdipta et al., 2016, 2018; Hershcovich et al., 2018a). Empirical comparison between syntactic and semantic schemes, however, is still scarce. Rudinger and Van Durme (2014) mapped Stanford Dependencies (precursor to UD) to Hobbsian Logical Form, identifying semantic gaps in the former. PredPatt (White et al., 2016), a framework for extracting predicate-argument structures from UD, was evaluated by Zhang et al. (2017) on a large set of converted PropBank annotations. Szubert et al. (2018) proposed a method for aligning AMR and UD subgraphs, finding that 97% of AMR edges are evoked by one or more words or syntactic relations. Damonte et al. (2017) refined AMR evaluation by UD labels, similar to our fine-grained evaluation of UCCA parsing.

Some syntactic representation approaches, notably CCG (Steedman, 2000), directly reflect the underlying semantics, and have been used to transduce semantic forms using rule-based systems (Basile et al., 2012). A related line of work tackles the transduction of syntactic structures into semantic ones. Reddy et al. (2016) proposed a rule-based method for converting UD to logical forms. Stanovsky et al. (2016) converted Stanford dependency trees into proposition structures (PROPS), abstracting away from some syntactic detail.

9 Conclusion

We evaluated the similarities and divergences in the content encoded by UD and UCCA. After annotating the reviews section of the English Web Treebank with UCCA, we used an automated methodology to evaluate how well the two schemes align, abstracting away from differences of mere convention. We provided a detailed picture of the content differences between the schemes. Notably, we quantified the differences between the notions of syntactic and semantic heads and arguments, finding substantial divergence between them. Our findings highlight the potential utility of using semantic parsers for text understanding applications (over their syntactic counterparts), but also expose challenges semantic parsers must address, and potential approaches for addressing them.

Acknowledgments

This work was supported by the Israel Science Foundation (grant No. 929/17), and by the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office. We thank Jakob Prange, Nathan Schneider and the anonymous reviewers for their helpful comments.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proc. of ACL*, pages 228–238.

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proc. of ACL*, pages 77–89.

Joakim Nivre et al. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proc. of the Linguistic Annotation Workshop*.

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

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank. *Linguistic Data Consortium, Philadelphia, PA*.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. HUME: Human UCCA-based evaluation of machine translation. In *Proc. of EMNLP*, pages 1264–1274.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proc. of COLING*, pages 2753–2765.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. In *Proc. of NAACL-HLT*.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for Abstract Meaning Representation. In *Proc. of EACL*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3).

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štepánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proc. of CoNLL*, pages 1–18.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proc. of ACL*, pages 2061–2071.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL*, pages 1127–1138.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018a. Multitask parsing across semantic representations. In *Proc. of ACL*, pages 373–385.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2018b. Universal dependency parsing with a general transition-based DAG parser. In *Proc. of CoNLL UD Shared Task*, pages 103–112.

Marie-Catherine de Marneffe and Joakim Nivre. 2019. Dependency grammar. *Annual Review of Linguistics*, 5(1):197–218.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proc. of LREC*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Zdenka Uresova. 2016. Towards comparability of linguistic graph banks for semantic parsing. In *Proc. of LREC*.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming dependency structures to logical forms for semantic parsing. *TACL*, 4:127–141.

Corentin Ribeyre, Eric Villemonte de la Clergerie, and Djamé Seddah. 2015. Because syntax does matter: Improving predicate-argument structures parsing with syntactic features. In *Proc. of NAACL-HLT*, pages 64–74.

Rachel Rudinger and Benjamin Van Durme. 2014. Is the Stanford dependency representation semantic? In *Proc. of EVENTS*, pages 54–58.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L Petruck, Christopher R. Johnson, Collin F. Baker, and Jan Scheffczyk. 2016. *FrameNet II: Extended Theory and Practice*.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proc. of LREC*. ELRA.

Roy Schwartz, Omri Abend, and Ari Rappoport. 2012. Learnability-based syntactic annotation design. In *Proc. of COLING*, pages 2405–2422.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting more out of syntax with PropS. *arXiv preprint arXiv:1603.01648*.

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

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proc. of LREC*, Portoro, Slovenia. European Language Resources Association.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proc. of CoNLL UD Shared Task*, pages 88–99, Vancouver, Canada.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proc. of EMNLP*, pages 5027–5038.

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations: A French-English case study. In *Proc. of S2MT*, pages 11–22.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. Semantic structural annotation for text simplification. In *Proc. of NAACL*.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. Simple and effective text simplification using semantic and neural methods. In *Proc. of ACL*.

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 *Proc. of CoNLL*, pages 159–177.

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2016. Greedy, joint syntactic-semantic parsing with stack LSTMs. In *Proc. of CoNLL*, pages 187–197.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proc. of EMNLP*, pages 3772–3782.

Ida Szubert, Adam Lopez, and Nathan Schneider. 2018. A structured syntax-semantics interface for English-AMR alignment. In *Proc. of NAACL-HLT*, pages 1169–1180.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proc. of EMNLP*, pages 1713–1723.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proc. of EMNLP*.

Sheng Zhang, Rachel Rudinger, and Benjamin Van Durme. 2017. An evaluation of PredPatt and open IE via stage 1 semantic role labeling. In *Proc. of IWCS*.