Comparison by Conversion: Reverse-Engineering UCCA from Syntax and Lexical Semantics

Daniel Hershcovich[♢] **Nathan Schneider**[♠] **Dotan Dvir**[♡]
Jakob Prange[♠] **Miryam de Lhoneux**[♢] **Omri Abend**[♡]
[♢]University of Copenhagen [♠]Georgetown University [♡]Hebrew University of Jerusalem
{dh,ml}@di.ku.dk, {nathan.schneider,jp1724}@georgetown.edu,
dotan.dvir@mail.huji.ac.il, oabend@cs.huji.ac.il

Abstract

Building robust natural language understanding systems will require a clear characterization of whether and how various linguistic meaning representations complement each other. To perform a systematic comparative analysis, we evaluate the mapping between meaning representations from different frameworks using two complementary methods: (i) a rule-based converter, and (ii) a supervised *delexicalized* parser that parses to one framework using only information from the other as features. We apply these methods to convert the STREUSLE corpus (with syntactic and lexical semantic annotations) to UCCA (a graph-structured full-sentence meaning representation). Both methods yield surprisingly accurate target representations, close to fully supervised UCCA parser quality—indicating that UCCA annotations are partially redundant with STREUSLE annotations. Despite this substantial convergence between frameworks, we find several important areas of divergence.

1 Comparing Meaning Representations

Several symbolic meaning representations (MRs) support human annotation of text with broad coverage (Abend and Rappoport, 2017; Oepen et al., 2019). To date, it is still not completely clear, for all frameworks, what linguistic semantic phenomena they encode, and how it compares to the content represented by the others. It therefore behooves us to develop a firm linguistic understanding of MRs. In particular: are they merely a coarsening and rearranging of syntactic information, such as is encoded in Universal Dependencies (UD; Nivre et al., 2016, 2020)? To what extent do they take lexical semantic properties into account? What does this suggest about the potential for exploiting simpler or better-resourced linguistic representations for improved MR parsing? Intuitively, we ask whether:

$$\text{sentence-level MR} \stackrel{?}{=} \text{syntax} + \text{lexical semantics}$$

To address this question, we examine UCCA, a document-level MR often used for sentence-level semantics (see §2.1). Hershcovich et al. (2019) began to examine the relation of UCCA to syntax, contributing a corpus with gold standard UD and UCCA parses, heuristically aligning them, and quantifying the correlations between syntactic and semantic labels. Conversely, Hershcovich et al. (2018) provided some initial evidence that other MRs can be brought to bear on the UCCA parsing task via multitask learning, but left the details of the relationship between representations to latent (and opaque) parameters of neural models.

In this paper, we aim to close the gap between the two previous investigations by (1) building an interpretable rule-based system to convert from shallower representations (syntax and lexical semantic units/tags) into UCCA, forcing us to be linguistically precise about what UCCA captures and how it "decomposes"; and (2) training top-performing supervised parsers in a delexicalized setting with only syntactic and lexical semantic features, as a data-driven mapping corroborating the rule-based approach.

We perform our analysis on the Reviews section of the English Web Treebank (Bies et al., 2012), which has been manually annotated with UD and UCCA; and STREUSLE for lexical semantics (§2).

Proceedings of the 28th International Conference on Computational Linguistics, pages 2947–2966
Barcelona, Spain (Online), December 8-13, 2020

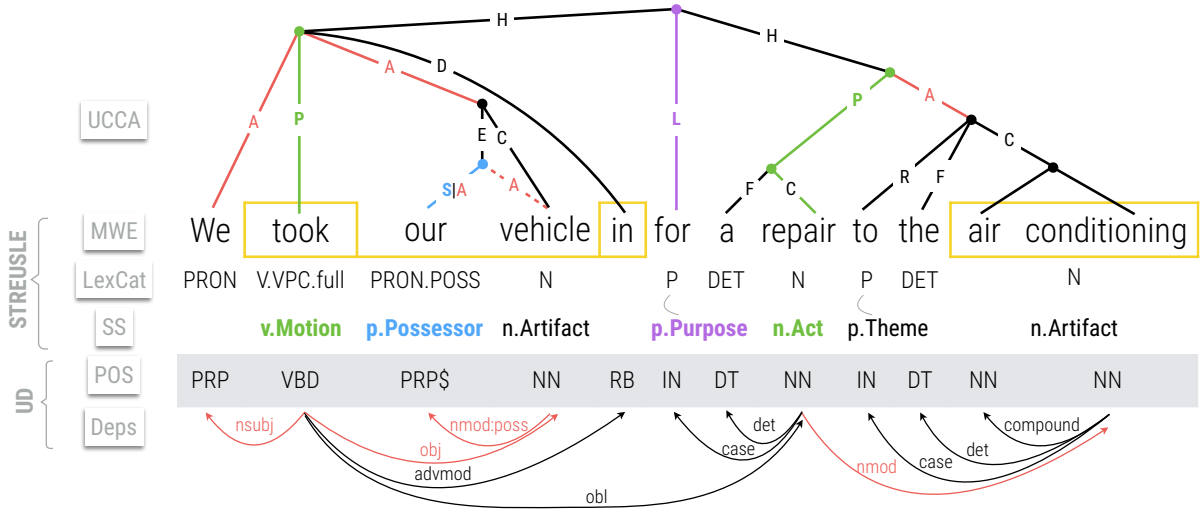

Figure 1: Example sentence from the Reviews training set (`reviews-086839-0003`, "We took our vehicle in for a repair to the air conditioning"), with UCCA, STREUSLE, and UD annotations.
◇ UCCA abbreviations: **H** = parallel scene, **L** = scene linker, **P** = process (dynamic event), **S** = state, **A** = scene participant, **D** = scene adverbial, **E** = non-scene elaborator, **C** = center (non-scene head), **R** = relator, **F** = functional element. The STREUSLE and UD part is adapted from Liu et al. (2020).

Although at present we only have the necessary evaluation data for English, the linguistic representations we examine have been applied to multiple languages (§2). Our approach can thus be applied cross-linguistically with minimal adaptation. Our contributions are:

- Delexicalized rule-based and supervised UCCA parsers, based only on syntax and lexical semantics.
- A linguistically motivated analysis of similarities and differences between the frameworks.

2 Representations under Consideration

The increasing interest in semantic representation and parsing, and the partial overlap in content between the different frameworks (Oepen et al., 2019), is a main motivation for our inquiry into content differences between UD and STREUSLE, and UCCA. We expect our inquiry to be relevant to other schemes, both in developing a general methodology, and in the insights gathered. For example, besides STREUSLE, UD also serves as the backbone of the DeComp scheme (White et al., 2016), and so information as to its semantic content is important there as well. Argument structural phenomena are at the heart of many MRs, which provide further motivation for empirical studies to the extent lexical semantics and syntax can encode them.

2.1 Universal Conceptual Cognitive Annotation

Universal Conceptual Cognitive Annotation (Abend and Rappoport, 2013) targets a level of semantic granularity that abstracts away from syntactic paraphrases in a typologically-motivated, cross-linguistic fashion (Sulem et al., 2015), building on Basic Linguistic Theory (Dixon, 2010, 2012), an influential framework for linguistic description. The scheme does not rely on language-specific resources, and sets a low threshold for annotator training. Beyond syntactic paraphrases, UCCA encodes lexical semantic properties such as the aspectual distinction between states and processes (whether an event evolves in time or not).

UCCA has been applied to text simplification (Sulem et al., 2018b) and evaluation of text-to-text generation (Birch et al., 2016; Choshen and Abend, 2018; Sulem et al., 2018a). UCCA corpora are available for English, French and German, and pilot studies have been conducted on additional languages.

Here we summarize the principles and main distinctions in UCCA.[1]

In UCCA, an analysis of a text passage is a DAG (directed acyclic graph) over semantic elements called **units**. A unit corresponds to (is *anchored* by) one or more tokens, labeled with one or more semantic **categories** in relation to a parent unit.[2] The principal kind of unit is a **scene** denoting a situation mentioned in the sentence, typically involving a scene-evoking **predicate**, participants, and (perhaps) modifiers. Each predicate is labeled as either State (S) or Process (P). Figure 1 contains three scenes: one anchored by the Process *took*; one anchored by the Process *a repair*; and one anchored by the possessive pronoun *our*, which indicates a stative possession relation. A Participant (A) of a scene is typically an entity or location involved. Adverbials (D) modify scenes with respect to properties like negation, modality, causativity, direction, manner, etc., which do not constitute an independent situation or entity. Temporal modifiers are labeled Time (T).

Scenes in UCCA can relate to one another in one of three ways. A Scene can serve as a Participant within a larger scene; a Scene can serve to elaborate on a Participant within a Scene (typically relative clauses); or scenes can be related by **parallel linkage** in a unit that consists of Parallel Scenes (H) and possibly Linkers (L) describing how they are related. This is seen at the top level of figure 1, where the taking and repair scenes are parallel and the purposive *for* is a linker.

Other categories only apply to units with no predicate: a semantic head—the Center (C); modifiers of Quantity (Q); and other modifiers, called Elaborators (E). An Elaborator may itself be a scene, as in *our vehicle*, where the scene of possession elaborates on the vehicle entity. Similarly, *blue vehicles* would be analyzed with a stative scene of blueness that elaborates on the vehicles.

Apart from the main semantic content of scenes and participants, UCCA provides the categories: Relator (R) for grammatical markers expressing how a unit relates to its parent unit—in English, these are mainly prepositions and the possessive *'s*; Function (F) for other grammatical markers with minimal semantic content, such as tense auxiliaries, light verbs, and articles. Other categories are used for expressing coordination and for expressions expressing speaker perspective outside the propositional structure of the sentence. Semantically opaque multi-word expressions (e.g., *air conditioning* in figure 1) are called **unanalyzable units**, and are not analyzed internally. UCCA distinguishes **primary edges** that always form a tree, and **remote edges**, which express reentrancies, such as the dotted edge from the possession scene unit to *vehicle*.

2.2 Universal Dependencies

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. An example UD tree appears at the bottom of figure 1.

2.3 STREUSLE

STREUSLE (Supersense-Tagged Repository of English with a Unified Semantics for Lexical Expressions) is a corpus annotated comprehensively for several forms of lexical semantics (Schneider and Smith, 2015; Schneider et al., 2018). All kinds of **multi-word expressions** (MWEs) are annotated, giving each sentence a lexical semantic segmentation.[3] Syntactic and semantic tags are then applied to individual units (single- and multi-word). The semantic tags are **supersenses** for noun, verb, and prepositional/possessive units. Preposition supersenses include two tiers of annotation: **scene role** labels represent the semantic role of the prepositional *phrase* marked by the preposition, and **function** labels represent the lexical contribution of the *preposition* in itself. The two labels are drawn from the same supersense inventory and are identical for many tokens.

The **lexcat** annotations (syntactic category of lexical unit) is a slight extension to the Universal POS tagset, adding categories for certain MWE subtypes, such as light verb constructions, following Walsh

[1]For further details, see the extensive UCCA annotation manual:
`https://github.com/UniversalConceptualCognitiveAnnotation/docs/blob/master/guidelines.pdf`

[2]UCCA also supports **implicit units** which do not correspond to any tokens (Cui and Hershcovich, 2020), but these are excluded from parsing evaluation and we ignore them for purposes of this paper.

[3]STREUSLE distinguishes strong MWEs, which are opaque (noncompositional) or idiosyncratic in meaning, and weak MWEs, which represent looser collocations that are nevertheless semantically compositional, like "highly recommended".

et al. (2018) and idiomatic PPs; it also distinguishes possessive pronouns, the possessive clitic *'s*, and discourse expressions.[4] Figure 1 illustrates the MWE, lexcat, and supersense layers.

STREUSLE itself is limited to English, but many of its component annotations have been applied to other languages: verbal multi-word expressions (Ramisch et al., 2018), noun and verb supersenses (Picca et al., 2008; Qiu et al., 2011; Schneider et al., 2013; Martínez Alonso et al., 2015; Hellwig, 2017), and preposition supersenses (Hwang et al., 2017; Peng et al., 2020; Hwang et al., 2020).

Liu et al. (2020) presented a comprehensive lexical semantic tagger for STREUSLE, which predicts the comprehensive lexical semantic analysis from text, and is freely available. Prange et al. (2019) proposed several procedures for integrating STREUSLE supersenses directly into UCCA, refining its coarse-grained categories with *preposition* supersenses. Enriching a supervised UCCA parser with preposition supersense features from STREUSLE—and, even more so, training a parser to predict supersenses jointly with UCCA—improved parsing performance, revealing the two frameworks to be overlapping but complementary.

2.4 Related Representations

The above annotation schemes define finite inventories of coarse-grained categories to avoid depending on language-specific lexical resources, and thus can in principle be applied to any language. This fact distinguishes UCCA and STREUSLE from finer-grained sentence-structural representations like FrameNet (Baker et al., 1998; Fillmore and Baker, 2009) and the Abstract Meaning Representation (Banarescu et al., 2013), which relies on PropBank (Palmer et al., 2005). The Prague Dependency Treebank tectogrammatical layer (Böhmová et al., 2003) uses few lexicon-free roles, but its semantics is determined by a valency lexicon.

The Parallel Meaning Bank (Abzianidze et al., 2017) uses lexicon-free[5] VerbNet (Schuler, 2005) semantic roles. The STREUSLE tagset for preposition supersenses generalizes VerbNet's role set to cover non-core arguments/adjuncts of verbs, as well as prepositional complements of nouns and adjectives. Universal Decompositional Semantics (DeComp) defines semantic roles as bundles of lexicon-free features. Cross-linguistic applicability in this case is delegated to the parser, which parses sentences in other languages to their corresponding *English* semantics (Zhang et al., 2018).

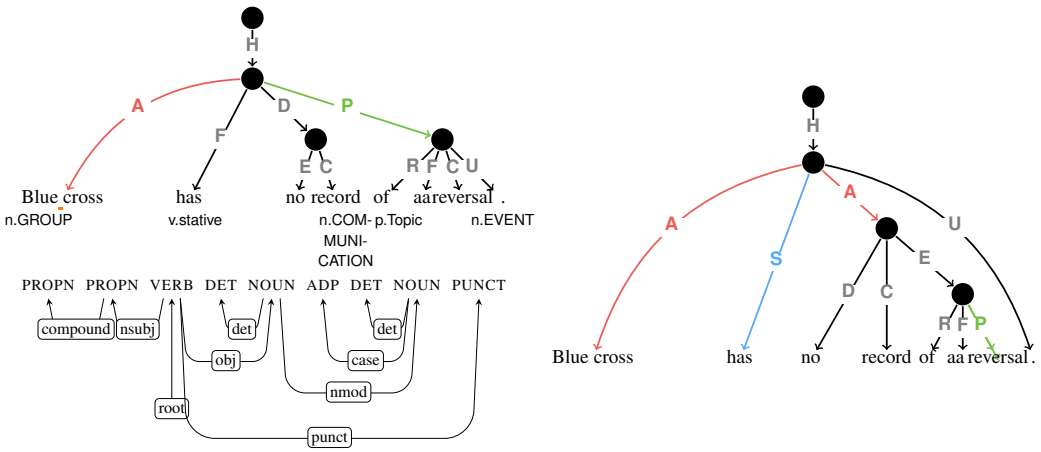

Figure 2: Example sentence (`reviews-003418-0006`, reading "Blue cross has no record of aa[*sic*] reversal") with gold-standard UCCA graph; STREUSLE MWEs and supersenses; and UD coarse-grained POS tags and relations (left); and UCCA graph output by the rule-based converter (right).

[4] STREUSLE tagset documentation: `https://github.com/nert-nlp/streusle/blob/master/CONLLULEX.md`

[5] While the lexical items comprising a linguistic utterance are naturally essential to its meaning, and therefore influence its semantic representation, by *lexicon-free* we mean the *ontology* and label set of the representation are not tied to a lexicon or a particular language.

3 First Conversion Approach: Rule-based UCCA Parsing from Syntax and Lexical Semantics

Here we describe a system to produce UCCA analyses for text, given UD syntactic graphs and STREUSLE lexical semantic annotation. This is a rule-based converter that inspects the UD structure in tandem with STREUSLE annotations to build an UCCA parse.[6] An analysis of the converter's successes and failures (§7) will, in turn, reveal the similarities and differences between the schemes. What follows is an overview of the algorithm; futher details about the rules are given in appendix A.[7]

MWEs and unanalyzable units. Based on STREUSLE MWE annotation, we group together text tokens into single unanalyzable semantic units when they are annotated as strong MWEs (we do not use weak MWEs, as UCCA does not encode them), except for light verbs and annotations that would lead to cycles. MWE-internal dependency edges are discarded so they will not be processed later. Punctuation is marked `U`. Mappings between units and dependency nodes are maintained. For example, in

> This is one of the worst places I have stayed, we **cut** out[*sic*] stay **short** and went to the Mulberry. (`reviews-023620-0001`),

cut… short is an unanalyzable unit according to the gold UCCA annotation, but the syntactic relation between *cut* and *short*, namely `xcomp`, does not indicate that; only few `xcomps`, in general, correspond to UCCA unanalyzable units (Hershcovich et al., 2019). In STREUSLE, however, *cut… short* is annotated as a strong MWE, whose lexcat is *V.VID* (idiomatic verb). Our rules create an unanalyzable unit covering this phrase, which matches the gold UCCA annotation in this case. The same is true for *Blue cross*[8] in figure 2.

STREUSLE supersenses and UCCA scene-evoking phrases. In a top-down traversal of the dependency parse, we visit each word's lexical unit and decide whether it evokes a main relation (scene-evoking phrase), using rules based on syntactic and lexical semantic features: copular *be* and stative *have*, adjectives (excluding a small list of quantity adjectives), existential *there*, non-discourse adverbs with a copula dependent, predicative prepositions and copulas introducing a predicate nominal, as well as common nouns supersense-tagged as N.ATTRIBUTE, N.FEELING, or N.STATE are labeled **S**; verbs, *thanks*, *thank you* and common nouns supersense-tagged as N.ACT, N.PHENOMENON, N.PROCESS, or N.EVENT (with the exception of nouns denoting a part of the day) are labeled **P**. For example, the noun *reversal* in figure 2 is marked as **P**, since its supersense of N.EVENT informs us that this is a scene-evoking noun. Relational nouns, supersense-tagged as N.PERSON or N.GROUP that also match kinship/occupation lists or suffixes, are labeled **P**+**A**.

STREUSLE lexical categories and UCCA edge categories. After identifying main relations and non-scene units, the exact category for some units still needs to be determined. The main relations themselves are labeled **P** or **S** depending on lexical categories and supersenses. Determiners, auxiliaries, and copulas are generally labeled **F**; vocatives and interjections, **G**. Exceptions include modal auxiliaries (**D**), and demonstrative/quantifier determiners modifying a non-scene unit (**E**/**Q** respectively). Verbal arguments and modifiers of scene-evoking non-verbal phrases are labeled **A**. Possessive clitics and prepositions are attached as **R**, unless a possessive clitic marks canonical possession, in which case it is **S**.

Secondary verb constructions. These constructions are structured differently in UCCA and UD: in "members who won't stop talking", the verb "stop" is the UD head and "talking" is a dependent. In UCCA "stop" is **D** and "talking" is the main relation, labeled **P**. To normalize the treatment of these constructions, they are marked and eventually restructured such that the syntactic head is labeled **D** and the syntactic dependent is labeled as the main relation.

[6] The new conversion and analysis code is available at `https://github.com/danielhers/streusle/tree/streusle2ucca`

[7] Before writing the new converter from the ground up, we tried modifying the Hershcovich et al. (2019) UD-only conversion code to use STREUSLE information. Details and analysis of that system appear in appendix B.

[8] Short for *Blue Cross Blue Shield*, a well-known health insurance organization in the United States.

	Training	Development
# Tokens	44,804	5,394
# Sentences	2,723	554

Table 1: EWT Reviews data statistics. We use only the training and development splits for analysis.

Coordination and lexical heads. Traversing the graph top-down again, coordination between scene units is labeled **L** (Linker), and between non-scene units, **N** (Connector). Scene units are labeled **H**, and non-scene unit heads, **C**. Lexical heads of units are labeled **C**, **P**, or **S**, and scene units as **H** where necessary; in "X of Y" constructions involving quantities/species, Y is identified as the **C**.

4 Second Conversion Approach: Delexicalized Supervised UCCA Parsing

Previous work tackled the UCCA parsing task using supervised learning. In order to complement and validate the analysis of the rule-based converter, we compare its findings to a delexicalized supervised parser, that can be seen as inducing a converter from data. By removing all word and lemma features from these parsers, and instead adding features based on gold UD and STREUSLE annotations, we obtain supervised "converters", which can be used for data-driven analysis and complement the rules.

TUPA. This UCCA parser (Hershcovich et al., 2017) is based on a transition-based algorithm with a neural network transition classifier, using a BiLSTM for encoding input representation, with word, lemma, and syntactic features embedded as real-valued features. We add the supersense and lexcat from STREUSLE as embedding inputs to the TUPA BiLSTM (concatenated with existing inputs). For prepositions, we add both the scene role and function (see §2).[9]

HIT-SCIR Parser. This is a transition-based parser for several MR frameworks, including UCCA (Che et al., 2019). It achieved the highest average score in the CoNLL 2019 shared task (Oepen et al., 2019). While Che et al. (2019) fine-tuned BERT (Devlin et al., 2019) for contextualized word representation, our delexicalized version replaces it with UD and STREUSLE features: POS tag, dependency relation, supersenses (scene role and function; see §2), the lexical category of the word or the MWE that the word is part of, and the BIO tag. These are concatenated to form word representations.[10]

5 Experiments

Data. We use the Reviews section from UD 2.6 English_EWT (Zeman et al., 2020), with lexical semantic annotations from STREUSLE 4.4 (Schneider and Smith, 2015; Schneider et al., 2018),[11] and with UCCA graphs from UCCA_English-EWT v1.0.1 (Hershcovich et al., 2019).[12] We use the standard train/development split for this dataset, and do not use the test split to avoid over-analyzing it, although all datasets contain annotations for it too. The data statistics are listed in table 1.

Rule-based converters. We evaluate the rule-based converter (§3), as well as the syntax-based converter from Hershcovich et al. (2019), which uses the UD tree and a majority-based category mapping (based on the most common UCCA category in the training set for each UD relation). This converter is oblivious to lexical semantics.

Parsers. We train TUPA v1.3 and the HIT-SCIR parser with gold-standard features from UD and STREUSLE, for equal conditions with the converters, using default hyperparameters. Categorical features are added as 20-dimensional embeddings. Scores are averaged over 3 models with different random seeds. For TUPA, we ablate UD- or STREUSLE-based features to quantify the contribution of each.

[9] Our code to enrich UCCA data with STREUSLE features is available at `https://github.com/jakpra/ucca-streusle`

[10] Our modified code for the HIT-SCIR parser is available at `https://github.com/danielhers/hit-scir-ucca-parser`

[11] `https://github.com/nert-nlp/streusle`

[12] `https://github.com/UniversalConceptualCognitiveAnnotation/UCCA_English-EWT`

	Primary F1	Remote F1
Syntax-based converter with gold UD (Hershcovich et al., 2019)	56.6	28.0
Our rule-based converter with gold UD + STREUSLE	71.7	44.2
TUPA, delexicalized with gold UD + STREUSLE features	69.5	46.4
UD features only	64.4	35.9
STREUSLE features only	62.4	27.5
HIT-SCIR, delexicalized with gold UD + STREUSLE features	67.9	41.6
TUPA with gold UD features + GloVe (Hershcovich et al., 2019)	71.7	47.0
HIT-SCIR (BERT-Large)	71.9	41.8
HIT-SCIR (GloVe)	67.0	42.4
with gold UD + STREUSLE features	72.2	46.9

Table 2: Labeled F1 (in %) for primary and remote edges on the UCCA EWT Reviews dev set, for rule-based systems (top), delexicalized supervised parsers with gold UD+STREUSLE (middle), and supervised parsers with word features (bottom).

Evaluation. We use standard UCCA parsing evaluation, matching edges by the terminal yields of their endpoint units.[13] Labeled precision, recall and F1-score consider the edge categories when matching edges. Where an edge has multiple categories, each of them is considered separately.

6 Results

Table 2 shows the EWT Reviews dev scores. For comparison with parsers that have access to words, we also show the TUPA dev results from Hershcovich et al. (2019), who used syntactic features from the gold UD annotation and GloVe (Pennington et al., 2014);[14] and the HIT-SCIR parser with BERT/GloVe, and with UD+STREUSLE features.

Rules with gold UD and STREUSLE close the gap between the syntax-based converter and parsers with word information, reaching the same primary labeled F1 as TUPA with word features. This is surprising (since supervised parsers are known to usually outperform rule-based ones), and suggests that the training data (see table 1) was insufficient for the parser to learn a mapping as accurate as the complex conversion rules (described in §3). Enhancing GloVe-based HIT-SCIR with UD and STREUSLE yields similar results. However, many errors remain in both approaches, indicating that UCCA and STREUSLE are *far from equivalent*. We analyze these errors in §7 to investigate the frameworks and the relationship between them.

Ablations. Noticeable drops in the ablations (TUPA with UD/STREUSLE only) show that both UD-provided structure and relation/entity types from STREUSLE supersenses are needed to make up for the missing lexical information, but also that lacking UD hurts more. This is expected, as the parser resorts to guessing when it lacks sufficiently informative input, and the chance of errors when guessing the UCCA *structure* (for which UD is informative) is much larger than for assigning edge labels (for which STREUSLE provides more fine-grained cues). The ablated TUPA models still outperform the syntax-based converter, indicating that there are indeed structural signals in STREUSLE and semantic signals in UD, which TUPA can salvage.

[13] The terminal yield of a unit is defined based on the graph's primary edges only, as standard in UCCA evaluation.

[14] Parsing from gold features is by no means a realistic scenario, but we give the scores as a reference for the converters.

Predicted Category \ Gold Category	A	A|G	A|P	A|S	C	D	D|T	E	F	G	H	L	N	P	Q	R	S	T	∅
A	758	4	7	12	17	11		9	4	1	6	1		14	1	1	19		150
A|P				1	1														
A|S				8	2														
C	50		7	12	457	27		11	1	1	12	3		31	2	5	12	1	48
D	10				12	280		40	8	12	2	2		6	4	1	7	18	20
E	48	1			20	42	1	294	3	1	17			3	7	1	24	4	49
F	3								613					1	1		3		1
G		2							2	6	2						2		4
H	40	2		1	29	6		13	1		450	4		22		2	8		265
L						7		1	19	1		221	14	1		27			5
N					1	1		1				10	31		1				2
P	3				16	15	1	2	13	12	1	1		345		2	29		32
Q					8	5		1							40				1
R	3				6							13		1		211	14		3
S	6				48	49		4	26		6			10		1	251		5
T	2				4	2	3								1			45	5
∅	148	1	3	6	136	60		100	32	1	124	9	2	65	12	34	23	6	

Table 3: dev set confusion matrix for the **rule-based converter**. The last column (row) shows the number of predicted (gold-standard) edges of each category that do not match any gold-standard (predicted) unit.

7 Analysis

Table 3 presents the EWT Reviews dev confusion matrix for the converter's output and gold UCCA. The delexicalized parsers' confusion matrix (in appendix C) is similar.[15] Note that we consulted the *training* set iteratively while developing the rules, addressing many recurring issues that would show up as prominent confusions.[16]

We proceed with an extensive error analysis of the converter, to point out similarities and delineate remaining divergences, which we stipulate constitute content differences between UCCA and the combination of syntax and lexical semantics from UD and STREUSLE. Figure 3 shows gold annotation and the converter's predictions.

7.1 High Match—Converging Analyses

Participants. As are recovered with high precision and recall. This is generally expected as most syntactic subjects and objects, as well as some obliques and even clauses, signify scene participants. Where syntax and semantics diverge, STREUSLE supersenses can rule out unlikely candidates. The most common sources of missed As are structural errors, i.e., incorrect scene structures, overly flat units containing more than the referential words, or misinterpreted noun compounds (see §7.3 below).

Function words. As evident in table 3, Function words (F) are accurately predicted. The distinction between words that contribute to the semantic meaning and those that do not is preserved between STREUSLE and UCCA, except for some cases—mainly infinitive "to".

[15] For 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.

[16] A full report of dev set outputs is included in `https://github.com/danielhers/streusle/blob/streusle2ucca/uccareport.dev.tsv`.

STREUSLE Annotation	Predicted UCCA Annotation	Gold UCCA Annotation	
Noun compounds			
tap_water N.SUBSTANCE	tap water (*unanalyzable*)	[E tap] [C water]	✗
road_construction N.EVENT	[P road construction]	[A road] [P construction]	✗
Adverbs and linkage			
Gets_busy so come early V.VID V.MOTION	[H [P Gets busy]] [L so] [H [P come] [T early]]	[H [D Gets] [S busy]] [L so] [H [P come] [T early]]	✓
so easy to load V.MOTION	[L so] [H [S easy] [A [F to] [P load]]]	[D [E so] [C easy]] [F to] [P load]	✗
Scene-evoking nouns			
a meal on the menu N.FOOD P.LOCUS N.COMMUNICATION	[F a] [C meal] [E [R on] [F the] [C menu]]]	[F a] [C meal] [E [R on] [F the] [C menu]]]	✓
answered all my questions V.COMMUNICATION P.ORIGINATOR P.GESTALT N.COMMUNICATION	[P answered] [A [Q all] [A my] [C questions]]	[P answered] [A [D all] [A my] [P questions]]	✗

Figure 3: Examples for cases where the rule-based converter produced the correct UCCA annotation due to converging analyses, as well as cases where it produced a wrong annotation due to a divergence.

Linkers. Linkers (L) are relatively easy: they are prototypically instantiated by syntactic co- and subordinators. To the extent that these are considered adpositional by STREUSLE, their supersense helps disambiguate between inter-scene linkage and Connectors (N) of non-scenes.

7.2 Partial Match—Inferrable by Combining Syntax and Lexical Semantics

Time (T) and Quantifier (Q) expressions frequently coincide with certain syntactic categories such as adverbs and prepositions, and can typically be identified from corresponding supersenses, if available. The converter tends to err on the conservative side, falling back to Adverbials (D) and Elaborators (E) when it cannot find sufficient explicit semantic evidence.

7.3 Low Match—Divergences or Insufficient Information

Noun compound interpretation. Lexical composition in noun compounds evokes various forms of event structures, which are underspecified by the meaning of the constituent words (Shwartz and Dagan, 2019). While often compounding is used for Elaboration, as in [E tap] [C water], it is not necessarily always the case. For example, in [C sea] [C bottom] both "sea" and "bottom" are Center, since they reflect part-whole relations. The modifier may also be a Participant in the scene evoked by the head, as in [A road] [P construction]. This is partially encoded in STREUSLE, as the fact that the MWE "road construction" has the N.EVENT supersense indicates that it is scene-evoking, but it still does not reveal the relationship between the constituent words.

Adverbs and linkage. While many syntactic adverbs are semantically Linkers ("well", "though"), neither UD nor STREUSLE distinguish them from Adverbials ("really", "possibly"). Some adverbs, like "so", can serve either role (see figure 3), a distinction that is only made in UCCA.

Centers. C is often unaligned due to the different notions of multi-word expressions in STREUSLE and UCCA: "tap water" is considered a strong MWE in STREUSLE, but is internally analyzed (with "water" being the Center) in UCCA (see figure 3), leading to an unmatched C.

7.3.1 Scene-evokers

While the concept of *scenes* is central to UCCA, correctly identifying scene-evoking words is one of the more difficult tasks for our converter. "Scene-ness" clearly goes beyond syntax (not all verbs evoke scenes and scenes can be evoked by a wide range of POS) and STREUSLE supersenses in isolation are often too coarse to resolve the question whether a given word evokes a scene and, if so, whether it is a Process (P) or a State (S). The former decision is generally somewhat easier for the converter (Recall of scene-evokers: 71.3%) than distinguishing between P (Recall: 69.1%) and S (64.0%). Below we examine a few recurring phenomena involving scenes.

Scene-evoking nouns. STREUSLE underspecifies whether nouns evoke scenes. For example, "menu" and "question" both have the N.COMMUNICATION supersense in STREUSLE, but "menu" does not

evoke a scene, while "question" might. Other similarly broad supersenses include N.COGNITION ("decision": potential scene, "fact": non-scene) and N.POSSESSION ("purchase": potential scene, "money": non-scene).

Relational nouns. This is a special case of scene-evoking nouns (Newell and Cheung, 2018; Meyers et al., 2004), both refering to an entity and evoking a scene in which the entity generally or habitually participates.[17] These units have two categories in UCCA, either **A|P** or **A|S**. The converter relies here on a combination of N.PERSON or N.GROUP supersenses and lexical lists. However, these nouns' scene-ness is often not recognized and they are confused with regular **A** or **C**.

Scene-evoking adjectives. Inspecting the high-frequency confusions, **adjectives** stand out as persistent error inducers. Different classes of adjectives are handled differently in UCCA: e.g., while most adjectives are scene-evoking, pertainyms (*academic*), inherent-composition modifiers (*sugary*), and quantity modifiers (*many*) are not. Some adjectives are ambiguous: a *legal practice* may refer to a behavior that is legal as opposed to illegal, in which case it should be scene-evoking, or to a law office, in which case it should not. Enriching STREUSLE with supersenses for adjectives (Tsvetkov et al., 2014) might be fruitful for such distinctions. Even with lexical disambiguation, the scene attachment of the adjective may be ambiguous: e.g. *a good chef* probably means a chef who cooks well, so *good* should be an Adverbial in the scene evoked by *chef*—in contrast with *a tall chef*, where *tall* is not part of the cooking scene and instead should evoke a State. Predicative adjectives, and adjective modifiers in predicative NPs, are another source of difficulty, especially when they occur in fragments: sometimes the adjective is annotated as evoking the main scene, and sometimes not. Determining this requires making various semantic distinctions, which are not fully represented in STREUSLE.

8 Conclusion

We have presented an extensive analysis of the similarities and differences between STREUSLE and UCCA on the EWT Reviews corpus, assisted by two complementary methods: manual rule-based conversion, and delexicalized parsing. Both approaches arrived at similar results, showing that the conversion between the frameworks can be moderately accurate, while also revealing important divergences, namely distinctions made in UCCA but not in STREUSLE: semantic relation between nouns in compounds, adverbial and linkage usage of adverbs, and the scene-evoking status of nouns, possessives and adjectives, among others.

Enriching supervised parsers with lexical semantic features improves parsing performance when using gold input. While this paper focuses on analysis, future work will investigate using predicted features with a parser/tagger (Liu et al., 2020). This approach is expected to improve parsing performance and robustness, demonstrating the utility of linguistically-informed approaches in complementing general supervised semantic parsers.

Acknowledgements

This research was supported in part by grant 2016375 from the United States–Israel Binational Science Foundation (BSF), Jerusalem, Israel. ML is funded by a Google Focused Research Award. We acknowledge the computational resources provided by CSC in Helsinki and Sigma2 in Oslo through NeIC-NLPL (www.nlpl.eu).

References

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

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

[17] E.g., a *teacher* is a person who teaches, and a *friend* is a person in a friendship relation with another person.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *EACL*, pages 242–247.

Daniel Zeman et al. 2020. Universal Dependencies 2.6. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

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

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 *Linguistic Annotation Workshop*.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English Web Treebank. Technical Report LDC2012T13, Linguistic Data Consortium, Philadelphia, PA.

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

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, Text, Speech and Language Technology, pages 103–127. Springer, Dordrecht.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *CoNLL*, pages 76–85.

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

Ruixiang Cui and Daniel Hershcovich. 2020. Refining implicit argument annotation for UCCA. In *DMR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Robert M. W. Dixon. 2010, 2012. *Basic Linguistic Theory*. Oxford University Press.

Charles J. Fillmore and Collin Baker. 2009. A frames approach to semantic analysis. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, pages 791–816. Oxford University Press.

Oliver Hellwig. 2017. Coarse semantic classification of rare nouns using cross-lingual data and recurrent neural networks. In *IWCS*.

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

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

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2019. Content differences in syntactic and semantic representation. In *NAACL-HLT*, pages 478–488.

Jena D. Hwang, Archna Bhatia, Na-Rae Han, Tim O'Gorman, Vivek Srikumar, and Nathan Schneider. 2017. Double trouble: the problem of construal in semantic annotation of adpositions. In **SEM*, pages 178–188.

Jena D. Hwang, Hanwool Choe, Na-Rae Han, and Nathan Schneider. 2020. K-SNACS: Annotating Korean adposition semantics. In *DMR*.

Nelson F. Liu, Daniel Hershcovich, Michael Kranzlein, and Nathan Schneider. 2020. Lexical semantic recognition. *arXiv:2004.15008 [cs]*.

Héctor Martínez Alonso, Anders Johannsen, Sussi Olsen, Sanni Nimb, Nicolai Hartvig Sørensen, Anna Braasch, Anders Søgaard, and Bolette Sandford Pedersen. 2015. Supersense tagging for Danish. In *NODALIDA*, pages 21–29.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: an interim report. In *Frontiers in Corpus Annotation Workshop*, pages 24–31.

Edward Newell and Jackie Chi Kit Cheung. 2018. Constructing a lexicon of relational nouns. In *LREC*, pages 3405–3410.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *LREC*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, 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 *LREC*, pages 1659–1666.

Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdeňka Urešová. 2019. MRP 2019: Cross-framework Meaning Representation Parsing. In *CoNLL*, pages 1 – 27.

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

Siyao Peng, Yang Liu, Yilun Zhu, Austin Blodgett, Yushi Zhao, and Nathan Schneider. 2020. A corpus of adpositional supersenses for Mandarin Chinese. In *LREC*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Davide Picca, Alfio Massimiliano Gliozzo, and Massimiliano Ciaramita. 2008. Supersense Tagger for Italian. In *LREC*, pages 2386–2390.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019. Made for each other: Broad-coverage semantic structures meet preposition supersenses. In *CoNLL*.

Likun Qiu, Yunfang Wu, Yanqiu Shao, and Alexander Gelbukh. 2011. Combining contextual and structural information for supersense tagging of Chinese unknown words. In *Computational Linguistics and Intelligent Text Processing: Proceedings of the 12th International Conference (CICLing'11)*, volume 6608 of *Lecture Notes in Computer Science*, pages 15–28. Springer, Berlin.

Carlos Ramisch, Silvio Ricardo Cordeiro, Agata Savary, Veronika Vincze, Verginica Barbu Mititelu, Archna Bhatia, Maja Buljan, Marie Candito, Polona Gantar, Voula Giouli, Tunga Güngör, Abdelati Hawwari, Uxoa Iñurrieta, Jolanta Kovalevskaitė, Simon Krek, Timm Lichte, Chaya Liebeskind, Johanna Monti, Carla Parra Escartín, Behrang QasemiZadeh, Renata Ramisch, Nathan Schneider, Ivelina Stoyanova, Ashwini Vaidya, and Abigail Walsh. 2018. Edition 1.1 of the PARSEME Shared Task on Automatic Identification of Verbal Multiword Expressions. In *LAW-MWE-CxG-2018*.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *ACL*, pages 185–196.

Nathan Schneider, Behrang Mohit, Chris Dyer, Kemal Oflazer, and Noah A. Smith. 2013. Supersense tagging for Arabic: the MT-in-the-middle attack. In *NAACL-HLT*, pages 661–667.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *NAACL-HLT*, pages 1537–1547.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

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

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. Semantic structural annotation for text simplification. In *NAACL*, pages 685–696.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. Simple and effective text simplification using semantic and neural methods. In *ACL*, pages 162–173.

Yulia Tsvetkov, Nathan Schneider, Dirk Hovy, Archna Bhatia, Manaal Faruqui, and Chris Dyer. 2014. Augmenting English adjective senses with supersenses. In *LREC*, pages 4359–4365.

Abigail Walsh, Claire Bonial, Kristina Geeraert, John P. McCrae, Nathan Schneider, and Clarissa Somers. 2018. Constructing an annotated corpus of verbal MWEs for English. In *LAW-MWE-CxG-2018*, pages 193–200.

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 *EMNLP*, pages 1713–1723.

Sheng Zhang, Xutai Ma, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2018. Cross-lingual decompositional semantic parsing. In *EMNLP*, pages 1664–1675.

A Details of Rule-based Converter

The following is a detailed description of the rules used in the rule-based parser (§3). This gives a step-by-step overview of the algorithm for constructing an UCCA semantic graph using STREUSLE/UD annotations. It is not a full specification and omits many details of the criteria and operations, but should be helpful for understanding the full code, available in `https://github.com/danielhers/streusle/blob/streusle2ucca/conllulex2ucca.py`. A running example is given at each step for the sentence:

> There's plenty of parking, and I've never had an issue with audience members who won't stop talking or answering their cellphones.

It has the following gold annotation:

> [H [**F** There] [F 's] [**D** plenty] [**P** [R of] [C parking]]] [U ,] [L and] [H [A I] [F 've] [**D|T** never] [F had] [P [F an] [C issue]] [A [R with] [C [**A** audience] [**P** members]] [E [R who] [**H** [A* members] [F wo] [D n't] [D stop] [P talking]] [L or] [**H** [A* members] [P answering] [A [E [S|A their] [A* cellphones]] [C cellphones] [U .]]]]]]

Step 0.1: Transform the UD dependency parse

Split the final preposition off from *V.IAV* MWEs like *take care of*, as it is usually not treated as part of the verbal unit in UCCA. If the remaining part is still an MWE, it is labeled *V.LVC.full* or *V.VPC.full* depending on its syntax.

Step 0.2: Initialize lexical units under the UCCA root

- Each strong lexical expression (single-word or MWE) in STREUSLE is treated as an UCCA unit, with the following exceptions:
  - An MWE with lexcat *V.LVC.cause* is broken into two units: `D` for the light verb modifies the main predicate in a `+` unit.
  - An MWE with lexcat *V.LVC.full* is broken into two units: `F` for the light verb modifies the main predicate in a `+` unit.
  - An MWE annotation is discarded if it would lead to a cycle in the dependencies such that the highest token of the MWE is dependent on a token outside of the MWE, which is dependent on another token of the MWE.
- MWE-internal dependency edges are discarded so they will not be processed later.
- Punctuation is marked `U`.
- Mappings between units and dependency nodes are maintained.

Step 0.3: Identify which lexical units are main relations (scene-evoking)

In a top-down traversal of the dependency parse, visit each word's lexical unit and decide whether it evokes a state (`S`), process (`P`), undetermined between state or process (`+`), or does not evoke a scene (`-`):

- If already labeled `D`, `F`, or `+`, do nothing.
- If an adjective not from a small list of quantity adjectives, label `S`.
- If existential *there*, label `S`. If a *be* verb in an existential construction, label `-` and swap the positions of *be* and *there* in the dependency parse so *there* is the head.
- If an adverb not attached as discourse and it has a copula, label `S`.
- *thanks* and *thank you* are `P`.

- In most cases, predicative prepositions are `S`.
- A copula introducing a predicate nominal (non-PP) is labeled `S` and promoted to the head of the dependency parse, unless the nominal is scene-evoking. (The top-down traversal order ensures the nominal is reached first.)
- If a common noun, mark as
  - `S` if supersense-tagged as ATTRIBUTE, FEELING, or STATE
  - `P` if ACT, PHENOMENON, PROCESS, or EVENT (with the exception of nouns denoting a part of the day)
  - a relational noun if PERSON or GROUP and matching kinship/occupation lists or suffixes
  - `-` otherwise
- If a verb or copula not handled above, label `+`
- Else label `-`

In the notation, "UNA" means "lexical" (it originally meant "unanalyzable").

> [DUMMYROOT [S [UNA There]] [- [UNA 's]] [- [UNA plenty]] [- [UNA of]] [S [UNA parking]] [U ,] [- [UNA and]] [- [UNA I]] [- [UNA 've]] [- [UNA never]] [+ [F had] … [UNA issue]] [- [UNA an]] … [- [UNA with]] [- [UNA audience]] [- [UNA|A|P [UNA members]]] [- [UNA who]] [- [UNA wo]] [- [UNA n't]] [+ [UNA stop]] [+ [UNA talking]] [- [UNA or]] [+ [UNA answering]] [- [UNA their]] [- [UNA cellphones]] [U .]]

Step 1: Attach functional and discourse modifier words

Determiners, auxiliaries, copulas are generally `F`; vocatives and interjections, `G`. Exceptions include modal auxiliaries (`D`), demonstrative determiners modifying a non-scene unit (`E`), quantifier determiners modifying a non-scene unit (`Q`).

Omitting the root:

> [S [UNA There] [**F** [UNA 's]]] [- [UNA plenty]] [- [UNA of]] [S [UNA parking]] [U ,] [- [UNA and]] [- [UNA I]] [+ [**F** [UNA 've]] … [**F** had] [**F** [UNA an]] [UNA issue]] [- [UNA never]] … [- [UNA with]] [- [UNA audience]] [- [UNA|A|P [UNA members]]] [- [UNA who]] [+ [F [UNA wo]] … [UNA stop]] [- [UNA n't]] … [+ [UNA talking]] [- [UNA or]] [+ [UNA answering]] [- [UNA their]] [- [UNA cellphones]] [U .]

Step 2: Attach other modifiers: adverbial, adjectival, numeric, compound, possessive, predicative-PP, adnominal-PP; as well as possessive clitic and preposition (as `R`, unless possessive clitic marks canonical possession in which case it is `S`)

> [S [UNA There] [F [UNA 's]]] [- [UNA plenty] [**E** [S [**R** [UNA of]] [UNA parking]]]] [U ,] [- [UNA and]] [- [UNA I]] [+ [F [UNA 've]] [**T** [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [**R** [UNA with]] [**E** [UNA audience]] [UNA|A|P [UNA members]] … [**E** [+ **[A* members]** [**F** [UNA wo]] [**D** [UNA n't]] [UNA stop]]]]] [- [UNA who]] … [+ [UNA talking]] [- [UNA or]] [+ [UNA answering]] [- [**E** [**A|S** [UNA their]] **[A* cellphones]**] [UNA cellphones]] [U .]

Step 3: Process verbal argument structure relations: subjects, objects, obliques, clausal complements; flag secondary (non-auxiliary) verb constructions

> [S [UNA There] [F [UNA 's]] [**A** [UNA plenty] [E [S [R [UNA of]] [UNA parking]]]]] [U ,] [- [UNA and]] [+ [**A** [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [R [UNA with]] [E [UNA audience]] [UNA|A|P [UNA members]] [E [+ [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [UNA stop] [^ [+ [UNA talking]]]]]]] [- [UNA or]] [+ [UNA answering] [**A** [E [A|S [UNA their]] [A* cellphones]] [UNA cellphones]]] [U .]

Step 4: Coordination

Traversing the graph top-down: for each coordinate construction with conjuncts' units' categories X and Y, create a ternary-branching structure `[X(COORD) X L Y]` if X is scene-evoking (`+`, `P`, or `S`) and `[X(COORD) X N Y]` otherwise.

> There's plenty... and I've never had an issue with...
>
> [**S(COORD)** [S [UNA There] [F [UNA 's]] [A [UNA plenty] [E [S [R [UNA of]] [UNA parking]]]]] ... [**L** [UNA and]] [+ [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [R [UNA with]] [E [UNA audience]] [UNA|A|P [UNA members]] [E [+ [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [UNA stop] [^ [+ [UNA talking]]]]]]]] [U ,] ... [- [UNA or]] [+ [UNA answering] [A [E [A|S [UNA their]] [A* cellphones]] [UNA cellphones]]] [U .]
>
> ... won't stop talking or answering...
>
> [S(COORD) [S [UNA There] [F [UNA 's]] [A [UNA plenty] [E [S [R [UNA of]] [UNA parking]]]]] ... [L [UNA and]] [**P** [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [R [UNA with]] [E [UNA audience]] [UNA|A|P [UNA members]] [E [**P** [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [UNA stop] [^ [+(COORD) [**P** [UNA talking]] [L [UNA or]] [**P** [UNA answering] [A [E [A|S [UNA their]] [A* cellphones]] [UNA cellphones]]]]]]]]]] [U ,] ... [U .]

Step 5: Decide `S` or `P` for remaining `+` scenes

Copula *be* and stative *have* are `S`; other verbs, as well as nouns tagged as ACT, PHENOMENON, PROCESS, or EVENT, are `P`.

> [S(COORD) [S [UNA There] [F [UNA 's]] [A [UNA plenty] [E [S [R [UNA of]] [UNA parking]]]]] ... [L [UNA and]] [**P** [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [R [UNA with]] [E [UNA audience]] [UNA|A|P [UNA members]] [E [**P** [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [UNA stop] [^ [+(COORD) [**P** [UNA talking]] [L [UNA or]] [**P** [UNA answering] [A [E [A|S [UNA their]] [A* cellphones]] [UNA cellphones]]]]]]]]]] [U ,] ... [U .]

Step 6.1: Restructure for secondary verbs

> [S(COORD) [S [UNA There] [F [UNA 's]] [A [UNA plenty] [E [S [R [UNA of]] [UNA parking]]]]] ... [L [UNA and]] [P [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [UNA issue] [A [R [UNA with]] [E [UNA audience]] [UNA|A|P [UNA members]] [E [P [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [**D** [UNA stop]] [**+(COORD)** [P [UNA talking]] [L [UNA or]] [P [UNA answering] [A [E [A|S [UNA their]] [A* cellphones]] [UNA cellphones]]]]]]]]] [U ,] ... [U .]

Step 6.2: Articulation—marking lexical heads of units as `C`, `P`, or `S`, and renaming scene units as `H` where necessary; determination of `C` involves "X of Y" constructions involving quantities/Species

> [S(COORD) [**H(S)** [S [UNA There]] [F [UNA 's]] [A [**Q** [UNA plenty]] [E [**H(S)** [R [UNA of]] [**S** [UNA parking]]]]]] ... [L [UNA and]] [**H(P)** [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [**P** [UNA issue]] [A [R [UNA with]] [E [UNA audience]] [**H(A|P)|A|P** [**A|P** [UNA members]]] [E [**H** [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [D [UNA stop]] [+(COORD) [**H(P)** [P [UNA talking]]] [L [UNA or]] [**H(P)** [P [UNA answering]] [A [E [**H(A|S)|S** [A|S [UNA their]]] [A* cellphones]] [**C** [UNA cellphones]]]]]]]]]] [U ,] ... [U .]

Steps 7+: Cleanup

Remove temporary decorations on `H` units from articulation; move `U` units for punctuation to more convenient attachment points; convert remaining `-` and `+` labels; wrap stray `P` and `S` units with `H` scenes; remove `UNA` and other temporary designations in the graph

> [S(COORD) [**H** [S [UNA There]] [F [UNA 's]] [A [Q [UNA plenty]] [E [**H** [R [UNA of]] [S [UNA parking]]]]]] . . . [L [UNA and]] [**H** [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [P [UNA issue]] [A [R [UNA with]] [E [UNA audience]] [**C** [A|P [UNA members]]] [E [H [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [D [UNA stop]] [+(COORD) [**H** [P [UNA talking]]] [L [UNA or]] [**H** [P [UNA answering]] [A [E [**C** [A|S [UNA their]]] [A* cellphones]] [C [UNA cellphones]]]]]]]]] [U ,] . . . [U .]
>
> [S(COORD) [H [S [UNA There]] [F [UNA 's]] [A [Q [UNA plenty]] [E [H [R [UNA of]] [S [UNA parking]]]]]] **[U ,]** [L [UNA and]] [H [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [P [UNA issue]] [A [R [UNA with]] [E [UNA audience]] [C [A|P [UNA members]]] [E [H [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [D [UNA stop]] [+(COORD) [H [P [UNA talking]]] [L [UNA or]] [H [P [UNA answering]] [A [E [C [A|S [UNA their]]] [A* cellphones]] [C [UNA cellphones]]]]]]]]] **[U .]**]
>
> [**H** [H [S [UNA There]] [F [UNA 's]] [A [Q [UNA plenty]] [E [H [R [UNA of]] [S [UNA parking]]]]]] [U ,] [L [UNA and]] [H [A [UNA I]] [F [UNA 've]] [T [UNA never]] [F had] [F [UNA an]] [P [UNA issue]] [A [R [UNA with]] [E [UNA audience]] [C [A|P [UNA members]]] [E [H [A* members] [R [UNA who]] [F [UNA wo]] [D [UNA n't]] [D [UNA stop]] [**H** [H [P [UNA talking]]] [L [UNA or]] [H [P [UNA answering]] [A [E [C [A|S [UNA their]]] [A* cellphones]] [C [UNA cellphones]]]]]]]]] [U .]]
>
> [H [H [S There] [F 's] [A [Q plenty] [E [R of] [S parking]]]] [U ,] [L and] [H [A I] [F 've] [T never] [F had] [F an] [P issue] [A [R with] [E audience] [C [A|P members]] [E [A* members] [R who] [F wo] [D n't] [D stop] [H [H [P talking]] [L or] [H [P answering] [A [E [C [A|S their]] [A* cellphones]] [C cellphones]]]]]]] [U .]]

B An Alternative Converter

Prior to the converter described in this paper, we had developed a converter based on the UD-to-UCCA converter of Hershcovich et al. (2019). However, after rewriting the converter from scratch in a more modular fashion, yielding the current converter (§3), we found that it obtains better performance, and therefore abandoned the previous, alternative converter. For completeness, we describe the previous converter here. The source code is available online.[18]

B.1 Dependency Transformations

Hershcovich et al. (2019) apply several pre-conversion dependency transformation, to match UCCA's flat linkage structure: `cc` dependents and `mark` dependents of `advcl` are promoted to be siblings of their heads, as in UCCA they are often **L** between scenes. `advcl`, `appos`, `conj` and `parataxis` are also promoted. We implement the same transformations using DepEdit.[19]

B.2 Label Mapping

We assign UCCA categories to graph edges, translating each UD relation to the most commonly co-occurring UCCA category in the training set.

We augment this simple majority-based UD-to-UCCA category mapping with rules based on lexical semantics, whose synopsis follows. The rules are based on the concept of scene-evoking words and phrases, which is not directly encoded in either UD or STREUSLE. However, by defining a set of

[18] `https://github.com/danielhers/streusle/blob/streusle2ucca-2019/conllulex2ucca.py`

[19] `https://corpling.uis.georgetown.edu/depedit`

heuristics on the lexical semantic categories, we obtain a good approximation of the UCCA scene/non-scene distinction. These rules first apply heuristics based on STREUSLE noun and verb supersenses—determined by examining the training data—and then take into account POS tags and word lists for the more complex cases.

Scene-evoking or not? If a noun has one of the supersenses N.ACT, N.EVENT, N.PHENOMENON, N.PROCESS, N.STATE, N.ATTRIBUTE, or N.FEELING, it is identified as scene-evoking. Proper nouns (UPOS=PROPN) are classified as non-scene-evoking. Other nouns with the supersense N.PERSON are first matched against a hand-crafted list of relational suffixes[20] and then against a list of relational nouns based on AMR heuristic lists and WordNet synsets.[21] Copulas with UPOS=AUX and supersense V.STATIVE are classified as non–scene-evoking. V.CHANGE verbs are matched against a hand-crafted list of aspectual verbs,[22] which are non–scene-evoking (D) in UCCA. Nouns and verbs that do not satisfy any of these criteria are canonically classified as non–scene-evoking and scene-evoking, respectively.

Categories. Scene-evoking verbs and nouns are labeled P instead of C by default (and their modifiers D instead of E). However, they are S under the following conditions: nouns with the supersenses N.STATE, N.ATTRIBUTE, and N.FEELING, and adjectives modifying non-scene-evoking nouns. Predicative nouns with no subject are labeled as A, and their modifiers as S (e.g. "Great food!"). Possessives with P.SOCIALREL or P.ORGROLE supersenses are E scenes with the possessive labeled both S and A, and their head as a A in the scene they evoke (see example in Figure 1).

Special cases of verbs. Verbs with the V.STATIVE supersense are labeled F if they have scene-evoking objects (e.g., "they *have* great customer service"), as S if their lemma is "be"/"have", and as P otherwise. Verbs in full light verb constructions or verb idiomatic expressions (*V.LVC.full*, *V.VID*) are labeled F (e.g., "*pay* attention").

Further lexical decisions. Expressions with the N.TIME supersense are labeled T. Locative pro-adverbs from a pre-defined lexicon ("here", "there", etc.) are labeled as A. Expressions with the *NUM* lexcat are labeled Q. If they are not labeled as R or D by the majority-based mapping, words from a pre-defined lexicon or having the *DISC* lexcat or the P.PURPOSE supersense are labeled L. Adverbs with the P.APPROXIMATOR supersense are E instead of D (e.g. "about 30%").

Further structural decisions. While `conj` (conjunct, e.g., in coordination) most often correspond to H, if the head is a non-scene then the dependent is labeled as C instead, as its corresponding unit is likely non-scene too. Compound or possessive modifiers of scene nouns are labeled as A. `vocative` dependents are labeled as both A and G.

B.3 UCCA Postprocessing

UCCA enforces a number of well-formedness restrictions, in terms of which categories may be siblings or children of which. These are sometimes violated by applying the rules described so far. We apply the category replacements listed in Table 4 to enforce meeting them. Additionally, H or L inside a scene are promoted to be a sibling of the scene, and remote H, N, L are removed.

B.4 Comparison with the Primary Converter

In the following head-to-head comparison, we refer to the primary system presented in the main part of the paper as "system A" and the secondary version presented here as "system B".

[20] *-er*, *-ess*, *-or*, *-ant*, *-ent*, *-ee*, *-ian*, *-ist*

[21] From `http://amr.isi.edu/download.html` we obtained `have-org-role-91-roles-v1.06`, which lists 39 types of government officials; and `have-rel-role-91-roles-v1.06`, which includes 85 kinship terms and a handful of person-to-person relations like *boss*, *client*, and *roommate* (the 'MAYBE' entries in these lists, which contain ambiguous words, are disregarded). The WordNet list consists of 1,487 occupations given by the single-word lemmas in the synsets `leader.n.01`, `professional.n.01`, `worker.n.01`, and their hyponyms, minus the words *man* and *woman*. The heuristics assign the State category to kinship terms like 'father' and Process to occupation nouns.

[22] *start*, *stop*, *begin*, *end*, *finish*, *complete*, *continue*, *resume*, *get*, *become*, *quit*, *keep*

C of unit with A	→	P
P, S or D in unit with C and N	→	C
P, S or D in unit with C and without N	→	E
N in unit with H	→	L
L in unit without H, starting a scene	→	R
L in unit without H, not starting a scene	→	N
Top-level, category not in {L,H,F,G,U}	→	H

Table 4: UCCA postprocessing category replacements in the alternative converter.

Easy for both:

- both systems perform well on As
- both systems are good at recalling Fs (system A: 84.9, system B: 89.9), but system A (in contrast to system B) has almost perfect precision (98.6 vs 82.5)

Difficult for both:

- both systems perform okay on Cs; system B tends to confuse Cs for As and Ps more than system A, which tends to fail at predicting units matching gold Cs entirely
- Ds are difficult for both systems; system A underpredicts Ds more than system B, but it is also more precise
- Es are difficult for both systems; Es often get confused (by both systems and in both directions) with Ds, As and Ss
- relational nouns (A|S, A|P) are very difficult for both systems; system B doesn't predict them at all, and system A predicts a few A|Ss which are mostly correct, but still misses 3/4 of them (and all A|Ps)
- Gs are very difficult for both systems; system B doesn't predict them at all, and system A predicts a few but with low precision and recall

Differences:

- system A is better at recalling Qs
- system A is better at recall (64.0 vs 53.6) and precision (61.2 vs 48.4) on Ss, but also confuses some gold Fs for Ss
- system A is better at recalling Ts; both systems tend to confuse Ts for Ds, but system B does it more than half of the time whereas system A only a quarter
- system A is more eager to predict Ls, thus has higher recall (83.7 vs 61.0) but lower precision (74.7 vs 87.5) here than system B; system A confuses some gold Fs and Rs for Ls, system B confuses some gold Ls for Cs, Ns and Rs
- system B is more eager to predict Ns, thus has higher recall (76.6 vs 66.0) but lower precision (37.1 vs 66.0) here than system A; system B confuses some gold Ls for Ns
- system B is more eager to predict Ps, thus has higher recall (78.6 vs 69.1) but lower precision (56.3 vs 73.1) here than system A; system B confuses some gold Cs, Ds and Fs for Ps, system A confuses some gold Ps for Hs

- system B is more eager to predict **R**s, thus has higher recall (88.8 vs 74.0) but lower precision (65.5 vs 84.1) here than system A; system B confuses some gold **F**s and **L**s for **R**s, system A confuses some gold **R**s for **L**s

Given the small differences between the converters in terms of performance, we decided to use system A for the main analysis in the paper, as it is more modular and interpretable.

C Delexicalized Parser Confusion Matrix

Table 5 shows the confusion matrix for the delexicalized HIT-SCIR parser on the EWT reviews development set.

Predicted Category \ Gold Category	A	A|G	A|P	A|S	C	D	D|T	E	F	G	H	L	N	P	Q	R	S	T	∅
A	741	8	3	14	52	14		40	3	1	15	5		16	1	5	6	4	199
A|C	14	1		2	77			1		1				4	1		1		6
A|P			4	1	1									2					
A|Q											1								
A|S	1			6	6									1					1
C	31		5	4	406	5		17	4	1	18		1	39	2	1	23	1	69
D	18				18	275	2	43	15	8	7	6		5	2	1	28	24	25
D|S																	1		
D|T					1	2	2											6	
E	49		2		24	36		169	20	1	9	3		3	8	2	9	4	147
F	2				3	42		16	626		1	2		7	12	6	19		13
G					1			2		1	1	1			1				
H	11	1			10			6	1	1	578	1		4			3		264
H|R																			1
L	3				5	6	1	2	10	6	1	187	16		1	16	2	5	6
N						1						14	29			5	1		1
P	12		1	5	37	20		9	10	5	13	1		357	2	1	53		43
Q					6	5		4							31				1
R	10				8	1		1	15			34		1		225	11		12
S	7		1		33	37		67	10	5	20			17	1	4	224		14
T	1				1	6		1		1	1	2					1	21	7
∅	173		1	7	65	55		98	8	4	273	8	1	38	7	19	11	9	

Table 5: Development set confusion matrix for the **delexicalized HIT-SCIR parser**. The last column (row), labeled ∅, shows the number of predicted (gold-standard) edges of each category that do not match any gold-standard (predicted) unit.