Great Service! Fine-grained Parsing of Implicit Arguments

Ruixiang Cui and **Daniel Hershcovich**
Department of Computer Science
University of Copenhagen
{rc, dh}@di.ku.dk

Abstract

Broad-coverage meaning representations in NLP mostly focus on explicitly expressed content. More importantly, the scarcity of datasets annotating diverse implicit roles limits empirical studies into their linguistic nuances. For example, in the web review "Great service!", the provider and consumer are implicit arguments of different types. We examine an annotated corpus of fine-grained implicit arguments (Cui and Hershcovich, 2020) by carefully re-annotating it, resolving several inconsistencies. Subsequently, we present the first transition-based neural parser that can handle implicit arguments dynamically, and experiment with two different transition systems on the improved dataset. We find that certain types of implicit arguments are more difficult to parse than others and that the simpler system is more accurate in recovering implicit arguments, despite having a lower overall parsing score, attesting current reasoning limitations of NLP models. This work will facilitate a better understanding of implicit and underspecified language, by incorporating it holistically into meaning representations.

1 Introduction

Studies of form and meaning, the dual perspectives of the language sign, can be traced back to modern linguistics since de Saussure (1916, 1978). They are relevant to modern NLP, as even current large neural language models cannot intrinsically achieve a human-analogous understanding of natural language (Žabokrtskỳ et al., 2020; Bender and Koller, 2020). Computational linguists attempt to capture syntactic and semantic features by means of constructing meaning representation frameworks, such as PTG (Böhmová et al., 2003), EDS (Oepen and Lønning, 2006), AMR (Banarescu et al., 2013) and UCCA (Abend and Rappoport, 2013). Through such frameworks, researchers have been exploring linguistic phenomena such as quantification (Pustejovsky et al., 2019), coreference (Prange et al., 2019), and word sense (Schneider et al., 2018).

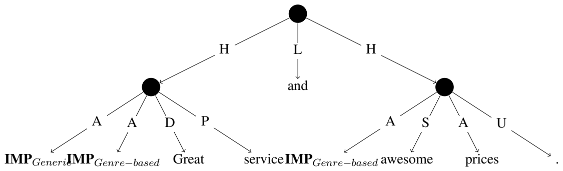


Figure 1: Example of a UCCA graph with fine-grained implicit arguments.

Participant	A	Linker	L
Center	C	Connector	N
Adverbial	D	Process	P
Elaborator	E	Quantifier	Q
Function	F	Relator	R
Ground	G	State	S
Parellel Scene	H	Time	T
Deictic		Generic	
Genre-based		Type-identifiable	
Non-specific		Iterated-set	

Table 1: UCCA Foundational Layer categories (above) and Implicit Participant Refinement Layer categories (below).

In recent years, a few studies have been dedicated to annotating and modelling implicit and underspecified language (Roesiger et al., 2018; Elazar and Goldberg, 2019; McMahan and Stone, 2020). For example, in the online review "there is no delivery", two of the omitted elements are the business agent and delivered items. However, previous works focus on specific phenomena requiring linguistic and collaborative reasoning, e.g., bridging resolution, numeric fused-heads identification and referential communication. Such datasets overexpose models to limited linguistic expressions, without leveraging the complete syntactic and semantic features of the context, regardless of the diversity of implicit roles.

O'Gorman (2019) and Cui and Hershcovich

Proceedings of the 17th International Conference on Parsing Technologies (IWPT 2021), pages 65–77
Bangkok, Thailand (online), August 6, 2021. ©2021 Association for Computational Linguistics

(2020) present works on fine-grained implicit role typology, incorporating it into the meaning representation frameworks AMR and UCCA, respectively. They lay a foundation for the interpretation of idiosyncratic behaviours of implicit arguments from a linguistic and cognitive perspective. Nevertheless, neither provided a dataset ready for computational studies of such implicit arguments.

We take the latter as a starting point, addressing several theoretical inconsistencies, and evaluate its applicability by carefully re-annotating their pilot dataset and providing inter-annotator agreement. Our categorisation set, consisting of six implicit role types, is compatible with UCCA's semantic notion of Scene rather than specific linguistic phenomena. Furthermore, as opposed to previous work, it tackles only essential implicit arguments, salient in cognitive processing.

We design the first semantic parser, with two different transition systems, that has the ability to parse fine-grained implicit arguments for meaning representations and evaluate its performance on the revisited dataset. To conclude, we reflect on this work's objectives and the challenges to face in future research on implicit arguments[1].

2 Revisiting Implicit Argument Refinement

Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport, 2013), a typologically-motivated meaning representation framework, has been targeted by several parsing shared tasks (Hershcovich et al., 2019b; Oepen et al., 2019, 2020). It uses directed acyclic graphs (DAG) anchored in surface tokens, where labelled edges represent semantic relations. In the *foundational layer*, these are based on the notion of Scenes (States and Processes), their Participants and modifiers. UCCA distinguishes *primary* edges, corresponding to explicit content, from *remote* edges, allowing cross-Scene relations. Additionally, *Implicit* units represent entities of importance to the interpretation of a Scene that are not explicitly anchored in the text. Several refinement layers have been proposed beyond the foundational layer, adding distinctions important for semantic roles and coreference (Shalev et al., 2019; Prange et al., 2019).

2.1 Fine-grained Implicit Argument Refinement for UCCA

Cui and Hershcovich (2020) proposed a fine-grained implicit argument typology implemented as a refinement layer of Participants for UCCA, centering around the semantic notion of Scene. Their proposed categorisation set consisting of six types, listed in Table 1, is argued to have low annotation complexity and ambiguity, thus requiring relatively low cognitive load for annotation than other fine-grained implicit argument typologies (O'Gorman, 2019).

For example, the online review "Great service and awesome price!" is annotated as follows and visualised as Figure 1:

(1) [Great$_D$ service$_P$ $IMP_{Generic}$ $IMP_{Genre-based}$]$_H$ and$_L$ [awesome$_S$ prices$_A$ $IMP_{Genre-based}$]$_H$!

There are two Scenes invoked in the sentence. "Great service" with "service" as a Process, and "awesome prices" with "awesome" as a State. Who is serving, who is being served and what is priced are distinct implicit arguments (*IMP*), which require reasoning to resolve.

Genre-based roles refer to conventional omission in the genre (Ruppenhofer and Michaelis, 2010). The corpus is based on online reviews, where reviewers typically do not mention what is under review. Therefore, an implicit argument in each Scene is marked as Genre-based referring to the reviewee. *Generic* roles denote "people in general" (Lambrecht and Lemoine, 2005). In the example, the recipient could be anyone, rather than a specific person.

2.2 Revisiting Inconsistencies

Despite the general soundness of Cui and Hershcovich (2020)'s typology, we find multiple inconsistencies in it: prominently, the treatment of Process and State Scenes and of nominalisation, and some borderline cases. We propose to revisit these cases and introduce consistent implicit argument refinement guidelines.

2.2.1 State Scenes and Process Scenes

UCCA differentiates Scenes according to their persistency in time (Abend and Rappoport, 2013): stative Scenes are temporally persistent states, while processual Scenes are evolving events. Cui and Hershcovich (2020) did not annotate implicit arguments in State Scenes, although they are essentially

[1] Revisited Implicit EWT dataset is available on https://github.com/ruixiangcui/UCCA-Refined-Implicit-EWT_English. The code for the implicit parser is on https://github.com/ruixiangcui/implicit_parser

similar. We, therefore, categorise them using the same guidelines. This phenomenon is rather pervasive in the dataset, as in Example 1 and 2:

(2) Very$_D$ friendly$_S$ [[even$_E$ at$_C$]$_R$ weekends$_C$]$_T$ *IMP*$_{Genre-based}$.

2.2.2 The Definition of Prominent Elements

While UCCA only annotates implicit units when they are "prominent in the interpretation of the Scene" (Abend and Rappoport, 2013), it is not always clear what should be regarded as such: the level of uniqueness plays an essential role in recognising the prominence of an argument. For example, even distinguishing definite and indefinite articles may pose a challenge to semantic analysis (Carlson and Sussman, 2005). There are some linguistic tests for testing whether an argument is semantically mandatory (Goldberg, 2001; Hajič et al., 2012). As for UCCA, we posit that time, location and instrument modifiers, as a rule, are ubiquitous to such an extent that no implicit argument should be annotated unless a Process or State warrants it. In Example 3 (visualised as the upper graph in Figure 2a), in the Scene "you leave," the departing action demands that the implicit source location is vital for understanding, unlike the location element in Example 1 or 2, where they are of low prominence.

(3) [(*You*)$_A$ have$_D$ [[a$_F$... mechanic$_C$] real$_D$]$_{P/A}$ check$_P$ *IMP*$_{Non-specific}$]$_H$ before$_L$ [you$_A$ leave$_P$ *IMP*$_{Non-specific}$]$_H$.

2.2.3 Nominalisation as Agent Nouns

An Agent noun derives from another word denoting an action, and that identifies an entity that perform that action. In UCCA, such an expression is annotated as a single unit with both the Process and Participant categories. As in Example 3, "a real mechanic" is marked P/A. Like time or location, it is subjective whether and how many implicit arguments should be annotated given the possible list of involved receivers, instruments, etc.

For a profession like "mechanic", whatever is being repaired, as well as the repair tools, can be omitted. While an agent noun can invoke a Scene, they could also simply serve as a title. Such cases, e.g. "doctor," "professor" and "chairman," are common. To facilitate consistent annotation, we decide never to annotate implicit Participants in Scenes licensed by agent nouns. This means that there is just one Participant, the person themselves.

2.2.4 Indefinite Deictic Participant

Deictic arguments refer to the implicit speaker or addressee. They may be confused with Generic roles when it is ambiguous whether the speaker or addressee participates in the Scene. As a rule, we posit that an implicit Participant ought to be Generic unless it evidently refers to the speaker or addressee. Example 4 shows an ambiguous case.

(4) [The$_F$ experience$_C$]$_P$ *IMP*$_{Generic}$ [with$_R$ every$_Q$ department$_C$]$_A$ has$_F$ been$_F$ great$_D$.

3 Fine-grained Implicit Argument Corpus

As a pilot annotation experiment, Cui and Hershcovich (2020) reviewed and refined 116 randomly selected passages from an UCCA-annotated dataset of web reviews (UCCA EWT; Hershcovich et al., 2019a). The dataset was annotated by only one annotator. With our revisited guidelines, we ask two annotators[2] to revisit the original dataset, adding or modifying implicit arguments, and subsequently refining their categories, using UCCAApp (Abend et al., 2017).

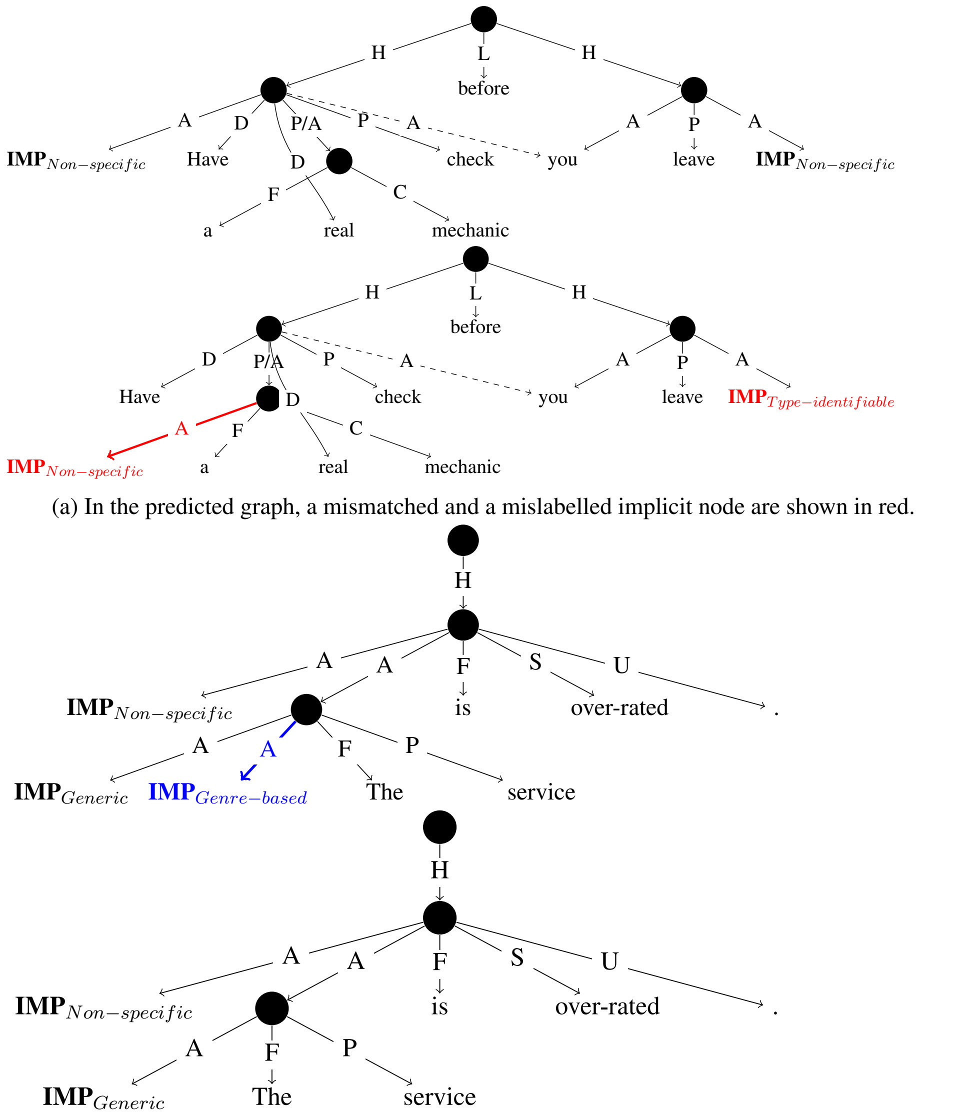
3.1 Evaluation of Implicit Argument Annotation

Standard UCCA evaluation compares two graphs (e.g., created by different annotators, or one being the gold annotation and the other predicted by a parser), providing an F1 score by matching the edges by their terminal span and category (Hershcovich et al., 2019b). However, the standard evaluation completely ignores implicit argument annotation. To quantify inter-annotator agreement and later parser performance (§6), we provide an evaluation metric taking these units into account.

To compare a graph G_1 with implicit units I_1 to a graph G_2 with implicit units I_2 over the same sequence of terminals $W = w_1, \ldots, w_n$, for each implicit node i, we identify its parent node $p(i)$, denoting the set of terminals spanned by it as the yield $y(p(i)) \subseteq W$, and its category as the label $\ell(p(i), i)$. Define the set of mutual implicit units between G_1 and G_2:

$$M(G_1, I_1, G_2, I_2) = \left\{ (i,j) \in I_1 \times I_2 \;\middle|\; \begin{array}{l} y_1(p_1(i)) = y_2(p_2(j)) \wedge \\ \ell_1(p_1(i), i) = \ell_2(p_2(j), j) \end{array} \right\}$$

[2] The annotators have a background in linguistics and cognitive science. One has experience in annotating UCCA's Foundational Layer. Training for the other took ~10 hours.



(a) In the predicted graph, a mismatched and a mislabelled implicit node are shown in red.

(b) In the gold graph, one implicit node is marked in blue, indicating it is not matched in the predicted graph.

Figure 2: Evaluation examples. Above: gold graphs. Below: predicted graphs.

The F-score is the harmonic mean of labelled precision and recall, defined by dividing $|M(G_1, I_1, G_2, I_2)|$ by $|I_1|$ and $|I_2|$, respectively.

In labelled evaluation, it is worth noting that we require a full match of the two sets of labels/categories rather than just intersection, which suffices in standard UCCA evaluation (of non-implicit units). That is, if there are two or more implicit nodes under one parent, it is only considered a correct match when both the numbers of implicit nodes and their labels are equal. We also introduce unlabelled evaluation, which only requires that parents' spans match.

For example, in Figure 2a, the reference graph has two implicit arguments. The first implicit unit's parent spans {have, a, real, mechanic, check} and the second's {you, leave}. In the predicted graph, two implicit units are predicted. The first implicit unit's parent spans {a, real, mechanic} while the second one spans the same terminals as the reference graph. We can see that the spans of the first implicit unit do not match—the second matches but with the wrong label. Therefore, in labelled and unlabelled implicit unit evaluation, the precision, recall, and F1 scores are all 0 and 0.5, respectively.

In Figure 2b, the reference graph has three implicit units, labelled Non-specific, Generic and Genre-based. One spans {The, service, is, over-rated} while two span {The, service}. Although the predicted graph has two implicit units with correct labels (Non-specific and Generic), it misses both implicit units under the second parent span,

	Deictic	Generic	Genre-based	Type-identifiable	Non-specific	Iterated-set	*Total Implicit Arguments*
Cui and Hershcovich (2020)	66	65	103	39	100	12	*385*
Our dataset	107	86	147	6	36	9	*391*

Table 2: Statistics of Revisited Implicit Corpus compared to the pilot annotation.

viz. the two sets merely intersect but are not equal. Therefore, the triplet precision, recall and F1 score for labelled evaluation are all 0.5 and the triplet for unlabelled evaluation are 1.

3.2 Inter-annotator Agreement

The two annotators separately reviewed and refined 15 out of the 116 passages (taken from the original test split). Using the evaluation metric proposed in §3.1, the labelled and unlabelled F1 scores are 73.8% and 91.3% respectively (see Appendix A). Annotators have a Cohen's κ (Cohen, 1960) of 69.3% on the six-type fine-grained classification of the implicit arguments whose parents' spans match. For comparison, on the FiGref scheme, O'Gorman (2019) report a Cohen's κ of 55.2% on a 14-way classification and of 58.1% on a four-way classification. Gerber and Chai (2010) proposed a relevant task to annotate implicit arguments for instances of nominal predicates in sentences, which has a Cohen's κ of 67%. While we maintain a comprehensive fine-grained typology, we still see an improvement in agreement over other corpora.

3.3 Statistics of Revisited Implicit Corpus

Finally, one annotator reviewed and refined all 116 passages. The second reviewed their annotation after completion. The full revisited dataset contains 393 passages, 3700 tokens and 5475 nodes. Table 2 compares our revisited dataset to the unreviewed dataset of Cui and Hershcovich (2020).

We see a major decline in Non-specific and Type-identifiable implicit arguments, because of the clearer definition of prominent elements and cases of agent nouns. Deictic, Generic and Genre-based increase their amount thanks to incorporating implicit arguments in State Scenes rather than only Process Scenes. The number of Iterated-set remains small due to the rareness of aspectual morphology and habitual/iterative constructions in English and in the corpus. However, it is still necessary to keep the category and separate out its instances rather than lump into another category or even ignore them. Since implicit arguments of such kind could be more common in morphologically rich languages, we want to keep a clean mapping of habitual/iterative constructions so as to facilitate the studies of implicit roles' diverse behaviours in languages other than English.

4 Two Transition Systems for Parsing Implicit Arguments

We build the first neural parser that supports parsing implicit arguments dynamically in meaning representations, with two different transition systems. We design a transition-based parser, modelled upon Nivre (2003): a stack $S = (\ldots, s_1, s_0)$ holds processed words. $B = (b_0, b_1, \ldots)$ is a buffer containing tokens or nodes to be processed. V is a set of nodes, and E is a set of labelled edges. We denote s_0 as the first element on S and b_0 as the first element on B. Given a sentence composed by a sequence of tokens t_1, t_2, ..., t_n, the parser is initialized to have a Root node on S, and all surface tokens in B. The parser will at each step deterministically choose the most probable transition based on its current parsing state. Oracle action sequences are generated for training on gold-standard annotations.

We propose two transition systems, IMPLICIT-EAGER and IMPLICIT-STANDARD, to deal with implicit arguments over the architecture of HIT-SCIR 2019 (Che et al., 2019), which ranked first in UCCA parsing in the MRP 2019 shared task (Oepen et al., 2019). The transition system incorporates all nine transitions, namely, LEFT-EDGE, RIGHT-EDGE, SHIFT, REDUCE, NODE, SWAP, LEFT-REMOTE, RIGHT-REMOTE and FINISH.

SHIFT, together with REDUCE, are standard transitions. SHIFT moves b_0 to S, while REDUCE pops s_0 from S (when it should not be attached to any element in B).

Following transition-based constituent parsing, NODE$_X$ creates a new non-terminal node (Sagae and Lavie, 2005). Such node will be created on the buffer, as a parent of s_0 with an X-labelled edge.

LEFT-EDGE$_X$ and RIGHT-EDGE$_X$ add an X-labelled primary edge between the first two elements on S. When the first element is the parent of the second element on S, LEFT-EDGE$_X$ is executed; in reverse, RIGHT-EDGE$_X$ will be chosen when the second element has the first element as its child. The left/right direction is the same as

Before Transition				Transition	After Transition					Condition
Stack	Buffer	Nodes	Edges		Stack	Buffer	Nodes	Edges	Terminal?	
S	$x \mid B$	V	E	SHIFT	$S \mid x$	B	V	E	−	
$S \mid x$	B	V	E	REDUCE	S	B	V	E	−	
$S \mid x$	B	V	E	NODE$_X$	$S \mid x$	$y \mid B$	$V \cup \{y\}$	$E \mid (y,x)_X$	−	$x \neq$ root
$S \mid x$	B	V	E	NODE	$S \mid x$	$y \mid B$	$V \cup \{y\}$	E	−	
$S \mid x$	B	V	E	IMPLICIT$_X$	$S \mid x$	$y \mid B$	$V \cup \{y\}$	$E \mid (x,y)_X^{\#}$	−	
$S \mid y,x$	B	V	E	LEFT-EDGE$_X$	$S \mid y,x$	B	V	$E \mid (x,y)_X$	−	
$S \mid x,y$	B	V	E	RIGHT-EDGE$_X$	$S \mid x,y$	B	V	$E \mid (x,y)_X$	−	$\begin{cases} x \notin w_{1:n}, \\ y \neq \text{ root }, \\ y \wedge_G x \end{cases}$
$S \mid y,x$	B	V	E	LEFT-REMOTE$_X$	$S \mid y,x$	B	V	$E \mid (x,y)_X^*$	−	
$S \mid x,y$	B	V	E	RIGHT-REMOTE$_X$	$S \mid x,y$	B	V	$E \mid (x,y)_X^*$	−	
$S \mid x,y$	B	V	E	SWAP	$S \mid y$	$x \mid B$	V	E	−	i$(x) <$ i(y)
[root]	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	V	E	+	

Table 3: The transition sets of two implicit transition systems. Actions marked in red are for IMPLICIT-EAGER, blue for IMPLICIT-STANDARD. We write the **stack** with its top to the right and the **buffer** with its head to the left. $(\cdot,\cdot)_X$ denotes a X-labelled edge, $(\cdot,\cdot)_X^*$ a remote X-labelled edge, and $(\cdot,\cdot)_X^{\#}$ an X-labelled edge to an implicit node. $i(x)$ is a running index for the created nodes. The prospective child of the EDGE action cannot have a primary parent. The newly generated node by IMPLICIT$_X$ action is prohibited from having any descendant. NODE generates a concept node on the buffer, but deos not produce an arc. This table is adapted from Hershcovich et al. (2017).

where the arc points to. LEFT-REMOTE$_X$ and RIGHT-REMOTE$_X$ are similar to LEFT-EDGE$_X$ and RIGHT-EDGE$_X$, yet these two transitions create remote edges, creating reentrencies. The X-labelled edge will be assigned a Remote attribute.

SWAP deals with non-planar graphs (a generalisation of non-projective trees), in other words, discontinuous constituents. It pops the second node on S and adds it to the top of B. FINISH is the terminal transition, which pops the Root node and marks the transition state as terminal.

In IMPLICIT-EAGER, we introduce a new transition IMPLICIT$_X$ adding an implicit node to the buffer and attaching it with a labelled edge in one step. In IMPLICIT-STANDARD, we simplify the existing NODE transition to only create a node without attaching it, with the purpose of treating implicit units like primary ones and generating them dynamically. We elaborate their designs in Section 4.1 and Section 4.2. Table 3 shows the transition set.

4.1 IMPLICIT-EAGER

Besides the nine transitions described above, IMPLICIT-EAGER introduces the IMPLICIT$_X$ transition, which creates a new unit on B as the child of s_0, with an X-labelled edge. The IMPLICIT action is different from the NODE action of IMPLICIT-STANDARD in the sense that the integrally generated edge makes the new node a child of s_0 rather than its parent, as in NODE$_X$. Equally importantly, the new node is prohibited to have any child in contrast to the *primary* nodes that the NODE$_X$ action generates.

4.2 IMPLICIT-STANDARD

IMPLICIT-STANDARD adopts a more modular approach. Rather than complicating the transition systems, it treats primary non-terminal nodes and implicit nodes equally by simplifying the NODE$_X$ action, making it generate a new unit on the buffer without attaching it with any (labelled) edge. We assume primary non-terminal nodes and implicit nodes are identical in essence, thus handling them without discrimination. Whenever an ungenerated child or parent of s_0 is found, NODE is executed so that a concept node will be created on B. This action does not cope with edge generation; the work is left to LEFT-EDGE$_X$ or RIGHT-EDGE$_X$. In the oracle, we can tell whether the node is primary or implicit by observing its relations. If the newly created node is the child of s_0 and does not have any descendants, it is an implicit node; otherwise, a primary node.

5 Experiments

5.1 Data Preprocessing

We convert UCCA XML data to MRP format using the open-source `mtool` software.[3] As the UCCA data provided in MRP 2019 shared task did not contain implicit information, HIT-SCIR 2019 is not designed to read this information in our dataset. We modify the parser to read node properties, and to

[3] https://github.com/cfmrp/mtool

Data	# Sentences	# Tokens	# Nodes	# Edges	# Deictic	# Generic	# Genre-based	# Type-i	# Non-s	# Iterated-s	# Implicit sum
EWT Train	2723	44751	59654	97561							
EWT Dev	554	5394	7534	11987							
EWT Eval	535	5381	7431	11907							
Overall					not refined	not refined	not refined	not refined	not refined	not refined	153
IMP Train	285	2671	3936	6146	87	59	103	3	18	4	274
IMP Dev	59	540	781	1217	11	15	19	1	10	0	56
IMP Eval	49	489	709	1106	9	12	25	2	8	5	61
Overall					107	86	147	6	36	9	391

Table 4: Statistics of train, dev and evaluation set in Original EWT and Revisited Implicit EWT. For each set, number of sentences, number of tokens, number of nodes, number of instances of 6 implicit categories and their sum are listed.

	Primary			Remote			**Implicit**					
	LP	**LR**	**LF**	**LP**	**LR**	**LF**	**LP**	**LR**	**LF**	**UP**	**UR**	**UF**
Baseline	0.495	0.467	0.480	0.538	0.304	0.389	1	0	0	1	0	0
Implicit-Eager	0.503	0.472	0.487	0.333	0.100	0.154	0.333 (7/21)	0.140 (7/50)	0.197	0.428 (9/21)	0.180 (9/50)	0.254
Implicit-Standard	0.474	0.431	0.451	0.438	0.280	0.341	**0.409** (9/22)	**0.180** (9/50)	**0.250**	**0.500** (11/22)	**0.220** (11/50)	**0.306**

Table 5: Experiment results on Revisited Implicit EWT in percents. For primary edges, remote edges, and implicit prediction, listed are Labelled Precision(LP), Labelled Recall (LR) and Labelled F-score (LF). In addition, Unlabelled precision (UL), Unlabelled Recall (UR) and Unlabelled F-score are also listed for implicit evaluation.

convert UCCA data from and to MRP format. The updated version of `mtool` is available on GitHub.[4]

5.2 Experimental Setup

Our parsers use stack LSTM to stabilize gradient descent process and speed up training; we enrich contextual information by employing the pretrained language model BERT as a feature input (Graves, 2013; Devlin et al., 2018). The model is implemented in AllenNLP (Gardner et al., 2018). We use the HIT-SCIR 2019 parser as the baseline for comparison. We keep the same hyperparameters as Che et al. (2019) except batch size, adjusted from 8 to 4 due to resource constraints. We do not tune hyperparameters on either the original or revisited dataset.[5]

We use the train, validation and evaluation split from Hershcovich et al. (2019b), which was originally from UD EWT, with the ratio of 0.75, 0.125 and 0.125. The evaluation set has been validated as the gold standard. Table 4 shows detailed statistics of train, dev and eval set of both the original and revisited dataset, on which we trained the baseline parser, Implicit-Eager and Implicit-Standard.[6]

6 Results

Table 5 presents experimental results on Revisited Implicit EWT by three parsers, the baseline HIT-SCIR 2019 parser, Implicit-Eager and Implicit-Standard on Revisited Implicit EWT. Regarding performance on the dataset, the baseline is not able to predict implicit argument as expected. However, both Implicit-Eager and Implicit-Standard managed to predict implicit arguments.

Based on the evaluation method mentioned in section 3.1, Implicit-Eager's labelled precision and labelled recall on Revisited Implicit EWT are 0.333 and 0.14; the unlabelled precision and unlabelled recall are 0.428 and 0.18. For the primary edge and remote edge evaluation, noticeably, Implicit-Eager also outperforms the baseline on primary edges by 0.007 in F-score on the revisited dataset.

Even though Implicit-Standard has the worse results in terms of primary parsing, it gains boosted performance on all targets in implicit evaluation. Its unlabelled implicit precision, recall and F-score are 0.5, 0.22 and 0.306, defeating Implicit-Eager by 0.072, 0.04 and 0.052, respectively.

For labelled evaluation, Implicit-Standard surpasses Implicit-Eager even further; the labelled precion, recall and F-score are 0.419, 0.180 and 0.250, respectively exceeding Implicit-Eager by 0.076, 0.04 and 0.053.

Table 6 presents the three parsers' performances on Original EWT. The baseline produced better results on primary edges and remote edges on Original EWT.

The reason why Implicit-Standard outper-

[4] `https://github.com/ruixiangcui/mtool`

[5] Hyperparameter settings are listed in Appendix B.

[6] See training details in Appendix C.

	Primary			Remote		
	LP	**LR**	**LF**	**LP**	**LR**	**LF**
Baseline	0.710	0.701	0.706	0.547	0.365	0.438
Implicit-Eager	0.675	0.597	0.634	0.527	0.344	0.416
Implicit-Standard	0.656	0.571	0.610	0.458	0.225	0.302

Table 6: Experiment results on Original EWT in percents. As there is no implicit argument in the dataset, only performances on primary edges and remote edges are listed.

forms Implicit-Eager on implicit evaluation but decrease in accuracy on primary evaluation might be attributed to its equal treatment of primary nodes and implicit nodes.

7 Discussion

As is indicated in Table 5, Implicit-Eager and Implicit-Standard successfully predicted respectively seven and nine implicit arguments with the correct fine-grained implicit labels. In the unlabelled evaluation, nine and 11 implicit arguments were predicted each.

Table 7 shows the confusion matrix of the performances of Implicit-Eager and Implicit-Standard on the evaluation set of Revisited Implicit EWT. Both parsers have predicted roughly the same amount of implicit arguments, 22 and 21, respectively.

Noticeably, Implicit-Eager has emitted 12 Deictic implicit arguments, accounting for 57.1% of all predictions, 14 and 66.7% if including partial matches as Deictic & Generic and Deictic & Genre-based. While Implicit-Standard has a more uniform distribution over prediction categories. Deictic, Generic and Genre-based are the most predicted ones. Moreover, one Non-specific implicit argument, despite being wrongly labelled, is also predicted by Implicit-Standard, while Implicit-Eager has never predicted labels other than Deictic, Generic and Genre-based, which have significantly more instances in the training set.

Both implicit parsers have predicted correctly four Deictic and three Generic & Genre-based implicit arguments. Besides, Implicit-Standard managed to predict two more correct Genre-based while Implicit-Eager has never predicted successfully stand-alone Genre-based implicit argument. Implicit-Standard has a remarkably higher labelled precision of 66.7% in Deictic than Implicit-Eager of 33.3%, while the latter has a higher precision of 75% in Generic & Genre-based than Implicit-Standard of 50%. Lamentably, neither of the implicit parsers has emitted prediction for Type-identifiable nor Iterated-set. It is necessarily expected as both categories have less than five instances in the training set.

Implicit-Eager	UNMATCHED	Non-specific	Non-specific & Generic	Non-specific & Type-identifiable	Deictic	Deictic & Iterated-set	Generic	Generic & Genre-based	Genre-based	Iterated-set	Type-identifiable	P	*Total*
UNMATCHED	0	5	1	1	4	1	1	7	14	4	1	2	*41*
Deictic	6	1	0	0	**4**	0	0	0	1	0	0	0	*12*
Deictic & Generic	1	0	0	0	0	0	0	0	0	0	0	0	*1*
Deictic & Genre-based	1	0	0	0	0	0	0	0	0	0	0	0	*1*
Generic & Genre-based	1	0	0	0	0	0	0	**3**	0	0	0	0	*4*
Genre-based	3	0	0	0	0	0	0	0	0	0	0	0	*3*
Implicit-Standard													
UNMATCHED	0	6	1	1	4	1	1	6	12	4	1	2	*39*
Non-specific	1	0	0	0	0	0	0	0	0	0	0	0	*1*
Deictic	2	0	0	0	**4**	0	0	0	0	0	0	0	*6*
Deictic & Genre-based	0	0	0	0	0	0	0	1	0	0	0	0	*1*
Generic	2	0	0	0	0	0	0	0	0	0	0	0	*2*
Generic & Genre-based	3	0	0	0	0	0	0	**3**	0	0	0	0	*6*
Genre-based	3	0	0	0	0	0	0	0	**2**	0	0	0	*5*
Genre-based & P	0	0	0	0	0	0	0	0	1	0	0	0	*1*

Table 7: Confusion matrix on the Revisited Implicit EWT evaluation set: The column is the predicted labels while the row is the actual labels. Noticeably, the parsers are able to predict implicit elements of other categories in theory, such as Process (P). If not clarified otherwise, the fine-grained implicit categories are Participant by default.

Although this paper focuses on fine-grained implicit Participants, there are already some implicit arguments of other categories annotated in the foundational layer, especially Process and Center. Interestingly, in the sentence “Fresh and excellent quality,” Implicit-Standard generated two implicit arguments, Genre-based and Process in the Scene “Fresh.” It means not only that it infers the Scene misses a Genre-based Participant, but also a Process, i.e., the main relation that evolves in time.

8 Related Work

Parsing implicit argument was introduced into NLP by Ruppenhofer et al. (2009); Gerber and Chai (2010, 2012), but has been coarse-grained and annotated within Nombank (Meyers et al., 2004), limited to ten nominal predicates.

Bender et al. (2011) identified ten relevant linguistic phenomena, ran several parsers and associated their output with target dependencies. Roth and Frank (2015) used a rule-based method for identifying implicit arguments, which depends on semantic role labelling and coreference resolution. Similarly, Silberer and Frank (2012); Chiarcos and

Schenk (2015); Schenk and Chiarcos (2016) proposed parsing methods for SemEval 2010 data, but they are only able to parse implicit arguments on a coarse level as well. Cheng and Erk (2018) built a narrative cloze model and evaluated it on Gerber and Chai (2010)'s dataset.

9 Conclusion

Implicit arguments are pervasive in the text but have not been well studied from a general perspective in NLP. In this work, we revisited a recently proposed fine-grained implicit argument typology by addressing its current deficiencies. We annotated a corpus based on the revised guidelines and designed an evaluation metric for measuring implicit argument parsing performance, demonstrating the annotation's reliability with a superior inter-annotator agreement comparing to other fine-grained implicit argument studies. The dataset will be available to facilitate relevant research.

We introduced the first semantic parser, with two different transition systems, that can handle and predict implicit nodes dynamically, and label them with promising accuracy as part of the meaning representations. We evaluated it on the new dataset and found that some types of implicit arguments are harder to parse than others and that a simpler transition system performs better on parsing implicit arguments at the cost of primary parsing. The fine-grained implicit argument task is challenging and calls for further research.

In future work, we plan to create a large resource of implicit arguments by automatically extracting them from various linguistic constructions in unlabelled text, use it for pre-training of language models, and evaluate them on our and other datasets to gain more insights into this linguistic phenomenon. A post-processing baseline to find implicit arguments after parsing the whole graph would also be interesting for future investigation.

10 Acknowledgments

The authors thank Miryam de Lhoneux and the anonymous reviewers for their helpful feedback.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Omri Abend, Shai Yerushalmi, and Ari Rappoport. 2017. UCCAApp: Web-application for syntactic and semantic phrase-based annotation. In *Proceedings of ACL 2017, System Demonstrations*, pages 109–114, Vancouver, Canada. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Emily M. Bender, Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 397–408, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague dependency treebank. In *Treebanks*, pages 103–127. Springer.

Greg Carlson and Rachel Shirley Sussman. 2005. *Seemingly Indefinite Definites*, pages 71 – 86. De Gruyter Mouton, Berlin, Boston.

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 *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76–85, Hong Kong. Association for Computational Linguistics.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 831–840, New Orleans, Louisiana. Association for Computational Linguistics.

Christian Chiarcos and Niko Schenk. 2015. Memory-based acquisition of argument structures and its application to implicit role detection. In *Proceedings of the 16th Annual Meeting of the Special Interest*

Group on Discourse and Dialogue, pages 178–187, Prague, Czech Republic. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Ruixiang Cui and Daniel Hershcovich. 2020. Refining implicit argument annotation for UCCA. In *Proceedings of the Second International Workshop on Designing Meaning Representations*, pages 41–52, Barcelona Spain (online). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yanai Elazar and Yoav Goldberg. 2019. Where's my head? Definition, data set, and models for numeric fused-head identification and resolution. *Transactions of the Association for Computational Linguistics*, 7:519–535.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Matthew Gerber and Joyce Chai. 2010. Beyond NomBank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592, Uppsala, Sweden. Association for Computational Linguistics.

Matthew Gerber and Joyce Y. Chai. 2012. Semantic role labeling of implicit arguments for nominal predicates. *Computational Linguistics*, 38(4):755–798.

Adele E Goldberg. 2001. Patient arguments of causative verbs can be omitted: The role of information structure in argument distribution. *Language sciences*, 23(4-5):503–524.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2019a. Content differences in syntactic and semantic representation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 478–488, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Hershcovich, Zohar Aizenbud, Leshem Choshen, Elior Sulem, Ari Rappoport, and Omri Abend. 2019b. SemEval-2019 task 1: Cross-lingual semantic parsing with UCCA. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1–10, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Knud Lambrecht and Kevin Lemoine. 2005. Definite null objects in (spoken) french. *Grammatical constructions: Back to the roots*, 4:13.

Brian McMahan and Matthew Stone. 2020. Analyzing speaker strategy in referential communication. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 175–185, 1st virtual meeting. Association for Computational Linguistics.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The NomBank project: An interim report. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 24–31, Boston, Massachusetts, USA. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 149–160.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural*

Language Learning, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Timothy J O'Gorman. 2019. *Bringing Together Computational and Linguistic Models of Implicit Role Interpretation*. Ph.D. thesis, University of Colorado at Boulder.

Jakob Prange, Nathan Schneider, and Omri Abend. 2019. Semantically constrained multilayer annotation: The case of coreference. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 164–176, Florence, Italy. Association for Computational Linguistics.

James Pustejovsky, Ken Lai, and Nianwen Xue. 2019. Modeling quantification and scope in Abstract Meaning Representations. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 28–33, Florence, Italy. Association for Computational Linguistics.

Ina Roesiger, Arndt Riester, and Jonas Kuhn. 2018. Bridging resolution: Task definition, corpus resources and rule-based experiments. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3516–3528, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Michael Roth and Anette Frank. 2015. Inducing implicit arguments from comparable texts: A framework and its applications. *Computational Linguistics*, 41(4):625–664.

Josef Ruppenhofer and Laura A Michaelis. 2010. A constructional account of genre-based argument omissions. *Constructions and Frames*, 2(2):158–184.

Josef Ruppenhofer, Caroline Sporleder, Roser Morante, Collin Baker, and Martha Palmer. 2009. SemEval-2010 task 10: Linking events and their participants in discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 106–111, Boulder, Colorado. Association for Computational Linguistics.

Kenji Sagae and Alon Lavie. 2005. A classifier-based parser with linear run-time complexity. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 125–132, Vancouver, British Columbia. Association for Computational Linguistics.

Ferdinand de Saussure. 1916. Cours de linguistique générale. *Paris: Payot. edited posthumously by C. Bally and A. Riedlinger*.

Ferdinand de Saussure. 1978. Cours de linguistique générale. *Paris: Payot. edited posthumously by C. Bally and A. Riedlinger*.

Niko Schenk and Christian Chiarcos. 2016. Unsupervised learning of prototypical fillers for implicit semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1473–1479, San Diego, California. Association for Computational Linguistics.

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 *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 185–196, Melbourne, Australia. Association for Computational Linguistics.

Adi Shalev, Jena D. Hwang, Nathan Schneider, Vivek Srikumar, Omri Abend, and Ari Rappoport. 2019. Preparing SNACS for subjects and objects. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 141–147, Florence, Italy. Association for Computational Linguistics.

Carina Silberer and Anette Frank. 2012. Casting implicit role linking as an anaphora resolution task. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 1–10, Montréal, Canada. Association for Computational Linguistics.

Zdeněk Žabokrtskỳ, Daniel Zeman, and Magda Ševčíková. 2020. Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics*, pages 1–61.

Appendices

A Inter-annotator Confusion Matrix

Tabel 8 shows the confusion matrix for measuring inter-annotator agreement on the evaluation set. The unlabelled F-score is 91.3%, and the labelled F1-score is 73.8%. The Cohen's κ between two annotators is 69.3%.

	UNMATCHED	Nonspecific	Nonspecific|Generic	Nonspecific|Type-identifiable	Deictic	Deictic|Iterated-set	Generic	Generic|Genre-based	Genre-based	Iterated/repeated/set	Type-identifiable	P
UNMATCHED	0	1	0	0	0	0	1	0	1	1	1	0
Nonspecific	0	4	0	0	0	0	0	0	4	0	0	0
Nonspecific|Deictic	0	0	0	0	0	1	0	0	0	0	0	0
Nonspecific|Generic	0	0	1	0	0	0	0	0	0	0	0	0
Nonspecific|Genre-based	0	0	0	0	0	0	0	1	0	0	0	0
Nonspecific|Type-identifiable	0	0	0	1	0	0	0	0	0	0	0	0
Deictic	2	0	0	0	8	0	0	0	0	0	0	0
Deictic|Genre-based	0	0	0	0	0	0	0	1	0	0	0	0
Generic|Genre-based	0	0	0	0	0	0	0	8	0	0	0	0
Genre-based	0	2	0	0	0	0	0	0	11	0	0	0
Type-identifiable	1	0	0	0	0	0	0	0	0	0	0	0
Iterated-set	1	0	0	0	0	0	0	0	0	3	0	0
P	0	0	0	0	0	0	0	0	0	0	0	2

Table 8: Confusion matrix of the evaluation set for measuring inter-annotator agreement

B Hyperparameter Settings

Our parsers use stack LSTM to stabilize gradient descent process and speed up training; we enrich contextual information by employing the pre-trained language model BERT as a feature input. We keep the same hyperparameter setting for all three parsers, the baseline parser HIT-SCIR 2019, Implicit-Eager and Implicit-Standard. The setting is shown as the Table 9.

Hyperparameter	**Value**
Hidden dimension	20
Action dimension	50
Optimizer	Adam
β_1, β_2	0.9, 0.99
Dropout	0.5
Layer dropout	0.2
Recurrent dropout	0.2
Input dropout	0.2
Batch size	4
Epochs	50
Base learning rate	1×10^{-3}
BERT learning rate	5×10^{-5}
Gradient clipping	5.0
Gradient norm	5.0
Learning rate scheduler	slanted triangular
Gradual Unfreezing	True
Cut Frac	0.1
Ratio	32

Table 9: Implicit Parser hyperparameters.

C Training Details

As Table 10 shows, the training time is 2 days 22 hours for the baseline on Original UCCA EWT (50 epochs). Best epoch is 3rd; 3 hours for the baseline on Revisted Implicit EWT (30 epochs). Best epoch is 22nd; 1 day 8 hours and 1 day 19 hours for Implicit-Eager and Implicit-Standard on Original UCCA EWT (10 epochs, 13 epochs), respectively, with the best epoch being the 3rd and 2nd; And finally, 6 hours and 8 hours for Implicit-Eager and Implicit-Standard on Revisited Implicit EWT (50 epochs). The best epoch is 21st and 37th, respectively. One can see that all parsers achieved the best performance at an early stage on Original UCCA EWT. However, both implicit parsers took longer time to train on Original UCCA EWT than the baseline.

	Original Full UCCA EWT			Revisited Implicit Dataset		
	Training time	# Epochs	# Best Epoch	Training time	# Epochs	# Best Epoch
Baseline	2 days 22 h	50	3	3 h	30	22
IMPLICIT-EAGER	1 day 8 h	10	3	6 h	50	21
IMPLICIT-STANDARD	1 day 19 h	13	2	8 h	50	37

Table 10: Training details of the baseline, IMPLICIT-EAGER and IMPLICIT-STANDARD on orignial UCCA EWT and Revisited Implicit EWT., including training times, the number of best epoch and total epochs.