FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

DEPARTMENT OF COMPUTER SCIENCE

Master's Degree in IT and Cognition

MSc Thesis | Academic Year 2018-2020

Refining and Parsing Implicit Argument in UCCA

Ruixiang Cui

SUPERVISOR:
Prof. Dr. Daniel Hershcovich

October, 2020

Porque se chamava moço
Também se chamava estrada
Viagem de ventania
Nem lembra se olhou pra trás
Ao primeiro passo
Porque se chamava homems
Também se chamavam sonhos
E sonhos não envelhecem
Em meio a tantos gases lacrimogênios
Ficam calmos
E lá se vai mais um dia
- Milton Nascimento e Lô Borges, Clube Da Esquina Nº 2

Acknowledgements

I remember well it was on 23rd, January 2020, that my beloved hometown, the city of Wuhan, was cut off from the entire world due to a mysterious outbreak of infectious disease, the now called COVID-19. What a drastic and unprecedented tactic in history! To say I was shocked, worried and devastated, would not do my feelings justice. As such, I had to start this thesis as the last part of my studies for a master degree.

It took me a whole two weeks to collect up my feelings, and make sure everyone I knew back home was safe, with daily necessities in their houses. There were moments I felt I could not hold up anymore. But, I was fortunate enough to have many dear friends around me, and that I had Prof. Dr. Daniel Hershcovich as my supervisor, who not only guided me to start my future journey in computational linguistics, but also offered emotion support during these hard times.

One month later, Denmark imposed a similar lockdown to contain the outbreak. Experiencing such a thing is like reading the history book twice. Now everyone is supposed to work from home. This thesis would not have been possible without the support of my advisor, friends, and family.

I would like to thank Prof. Dr. Daniel Hershcovich. Thank you for leading me to such a fantastic academic world. Moreover, thank you for always being encouraging and patient whenever I encountered problems or difficulties.

I would also like to thank Prof. Dr. Omri Abend and Dr. Dotan Dvir from Hebrew University of Jerusalem for offering insightful advice and resources when forming the theoretic proposal and conducting annotation work for the thesis.

I am grateful to always have awesome friends around me. Especially, Chang and Mengyu, whom I knew from IT and Cognition, are the best kind of friends I could imagine. We have been supportive of each other since the beginning, both in life and in academic; we even managed to travel in Cyprus and Spain where we wrote up the theses.

And of course, my parents Zhimin and Haiyin, thank you for being my greatest support and inspire me to chase my dreams freely.

The list cannot be endless. Still, I have to thank Adam, Daniel, Hector, Amelia, Xue, João, Yujie, Agustín, Yukang, Jordan, Minggang, Lujie, Abed, Fabio, Walter, Joey, Fernando and Salvo that I never regret to have you join part of my journey.

I have realized that before we reach the top of the mountain, the true happiness lies in the way we climb the slope.

Abstract

Predicate-argument structure analysis is a central component in meaning representations of text. The fact that some arguments are not explicitly mentioned in a sentence gives rise to ambiguity in language understanding, and renders it difficult for machines to interpret text correctly. However, only few resources represent implicit roles for NLU, and existing studies in NLP only make coarse distinctions between categories of arguments omitted from linguistic form.

This thesis proposes a typology for fine-grained implicit argument annotation on top of Universal Conceptual Cognitive Annotation's foundational layer. The proposed implicit argument categorisation is driven by theories of implicit role interpretation and consists of six types: Deictic, Generic, Genre-based, Type-identifiable, Non-specific, and Iterated-set. The first contribution of the thesis is to design such a typology, revisit part of the UCCA EWT corpus, provide a new dataset annotated with the refinement layer, and make a comparative analysis with other schemes both in terms of quantity and quality.

The second part of the thesis presents the first transition-based Implicit Parser that can handle implicit arguments dynamically in meaning representations. The parser differs from other implicit argument detection systems that rather than attending to a specific linguistic phenomenon, it can perform fine-grained classification of implicit arguments simultaneously when parsing primary semantic representation graphs. We show that despite the difficulties of predicting fine-grained implicit arguments, the parser manages to notice some common situation in which certain kinds of implicit roles frequently show up.

It is hoped that this work can boost studies in obtaining implicit role resolution and complement recent research on how to interpret semantics in natural language understanding elaborately.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Studies of form and meaning, the dual perspectives of language sign, can be traced back to modern linguistics since de Saussure (1916, 1978), and the topic drifts into NLP as a reflection that the current large neural language models can not intrinsically achieve human-analogous understanding of natural languages (Bender and Koller, 2020; Žabokrtský et al., 2020). Computational linguists attempt to capture syntactic and semantic features by means of constructing meaning representation frameworks. 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).

However, most efforts were put into studying linguistic complexity superficially, rather than the more latent, implicit omission of arguments in an event. For instance, in the sentence "Just take the money!", the addressee who should "take the money" is left out. Such omission cannot be recovered directly from the text in the way of gapping or ellipsis, but require a higher level of understanding and inference from the context. Traditional studies approach argument omission from different aspects, namely syntactically, semantically, or pragmatically. The interpretation of implicit roles varies to a great extent, from phonological deleted role during production (Mittwoch, 1971; Perlmutter, 1968; Pérez-Leroux et al., 2018) to timecourse reference omission from a psycholinguistic aspect (Garrod and Terras, 2000). However, few studies have explored the implicit role phenomenon in NLP.

In this paper, we propose a fine-grained cross-linguistically applicable implicit argument annotation typology as a refinement for Universal Conceptual Cognitive Annotation (Section 2.1; Abend and Rappoport, 2013) categories. The typology follows UCCA's design concept, focusing on the semantic notion of Scene rather than linguistic form phenomena. The proposed implicit argument set contains six categories: Deictic, Generic, Genre-based, Type-identifiable, Non-specific, and Iterated-set. We refine the existing UCCA relation labels and add information to them, while keeping all categories from the underlying annotation.

Based on the proposed typology, we conduct a pilot annotation study, including revisit and refinement of the UCCA EWT dataset,[1] and subsequently make a comparative analysis with the only other existing fine-grained implict role annotation scheme, Fine-grained Annotations of Referential Interpretation Types (FiGref; O'Gorman, 2019).

Thereafter, We design the first transition-based parser that has the ability to parse fine-grained implicit arguments for meaning representations and evaluate its performance on the pilot UCCA Implicit EWT dataset. At the end, we discuss the objectives of this work and the challenges to face with in the future research of implicit arguments.

A few studies have explored the possibilities to parse implicit arguments. Both Gerber and Chai (2012) and Cheng and Erk (2019) have developed parsers to recover implicit arguments for nominal predicates; Bender et al. (2011) parse against some linguistic constructions, two of which could license implicit arguments, that is, tough adjectives and verbal gerund; Elazar and Goldberg (2019) focus on resolving missing numeric fused-heads, which are implicit Centers in UCCA.

Providing the most up-to-date and fine-grained annotation of implicit arguments, our studies can potentially enhance natural language understanding with the parser. For example, when companies conduct satisfaction analyses through web reviews, customers often express themselves colloquially in these reviews. Examples include "Serves bad ice cream, Joe's is better" and "Near a nice district, bad service and expensive." If these reviews are annotated with Genre-based implicit arguments, referring to the conventional omission

[1] https://github.com/ruixiangcui/UCCA-Refined-Implicit-EWT_English

of reviewees, algorithms can study which part of the reviews really refers to the companies rather than other entities, and make better predictions, despite the omission of subjects and non-standard language.

Chapter 2

Background and Related Work

2.1 Universal Conceptual Cognitive Annotation

Universal Conceptual Cognitive Annotation is a semantic representation scheme whose design concept comes from the Basic Linguistic Theory typological framework (Dixon, 2010a,b, 2012) and Cognitive Linguistics literature (Croft and Cruse, 2004). Abstracting away from syntactic forms, it aims at representing the main semantic phenomena in text while maintaining a low learning cost and rapid annotation by non-experts (Abend et al., 2017). Already providing datasets in English, French, and German, UCCA has demonstrated its cross-linguistic applicability in several languages and has

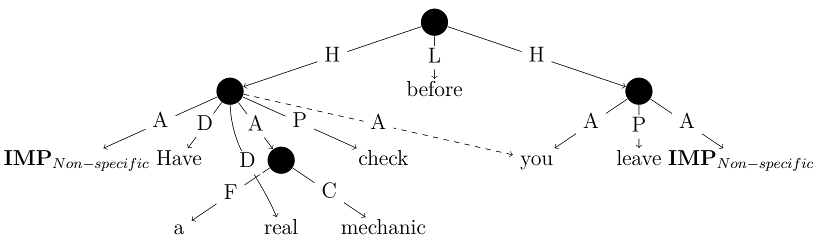

Fig. 2.1 Example of UCCA graph: “Have a real mechanic check before you leave.”. Abbreviation of UCCA edge labels is explained in Table 1. The dashed line stands for *Remote* edge. In this case it is a coreference "you". An **IMP** represents an *Implicit* argument denoting a null-instantiated core element in its corresponding Scene.

become a popular target framework in multiple parsing tasks (Hershcovich et al., 2019b; Oepen et al., 2019). Abend et al. (2017) have also developed an open-source web-based annotation system, UCCAApp, which supports fast annotation for linguistic representations. The effectiveness and efficiency in annotating and refining UCCA have been proven by several studies (Prange et al., 2019; Shalev et al., 2019).

UCCA presents the meaning of a sentence with a directed acyclic graph (DAG) whose terminal nodes correspond either to surface lexical tokens or extra units representing implicit arguments (Figure 2.1). Non-terminal nodes correspond to semantic units that participate in some super-ordinate relation. Edges are labeled with the role of a child node related to its parent node. The basic notion in UCCA is Scene, describing a state, action, movement, or some other relation that evolves in time. Each Scene involves one main relation (Process or State), and one or more Participants, including locations, abstract entities and sub-Scenes serving as 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

Table 2.1 Legend of UCCA edge categories (Hershcovich et al., 2019a)

Furthermore, UCCA distinguishes *primary* edges, appearing explicitly in one relation, from *remote* edges, allowing a Scene to indicate its arguments by linking from another Scene. *Primary* edges form a tree while *Remote* edges allow reentrancy, forming a DAG. In some cases, an entity of importance in the interpretation of a Scene does not explicitly exist in the text. Hence, UCCA introduces the notion of *Implicit* Units to represent such kind of entity.

For instance, the sentence "Have a real mechanic check before you leave" in Figure 2.1 contains two Scenes, evoked by "check" and "leave". The individual Scenes are annotated as follows:

1. "$(You)_A$ $\text{have}_D[\text{a}_F\ \text{real}_D\ \text{mechanic}_C]_A$ check_P IMP_{1A}"

2. "You_A leave_P IMP_{2A}"

"A mechanic" is a Participant in the first Scene, while "You" is Participant in both Scenes, as *Remote* constituting reentrancy in the first one (shown in dashed line), but explicit in the latter super-ordinate relations. In the first Scene, "Check" is the main Process used in the causative construction, requiring three Participants. While "You" refers to the customer making a request and "a real mechanic" is the service provider who should check something, the object that needs to be "checked" is missing. Therefore, we introduce an IMP A node to symbolize it. In the second Scene, "leave" is the Process meaning someone moves away from a source location. Although we state our little concern for non-core elements like location in this study in §3.1, the case is different for the "leaving" Scene since the source location is vital to the understanding of the departing action. For this reason, we add an IMP A to represent the place that is being left.

2.2 Implicit Argument

In UCCA's foundational layer, only limited cases of implicit arguments have been annotated. The main focus is on omission licensed by certain grammatical structures,[1] whose notion is similar to *Constructional Null Instantiation* in FrameNet (Ruppenhofer et al., 2006). Two typical examples of such constructions are imperatives (forced omission of subjects) and passives (agent omission) in English:

1. Imperative: IMP_A Do$_F$ n't$_D$ bother$_P$.
2. Passive: [The$_F$ doctor$_C$]$_A$ has$_F$ already$_T$ been$_F$ paid$_P$ IMP_A.

Several other kinds of constructions are mentioned in UCCA foundation layer guidelines, such as infinitive clause, gerund, and *thank* construction.

3. Infinitive clause: Is$_F$ there$_F$ [no$_E$ other$_E$ Verizon$_C$]$_A$ IMP_A to$_F$ go$_P$ to$_R$ [around$_R$ downtown$_C$]$_A$?
4. Gerund: How$_D$ addicting$_S$ IMP_A going$_P$ [to$_R$ Fitness Unlimited$_C$]$_A$ can$_F D$ be$_F$!"

[1] https://github.com/UniversalConceptualCognitiveAnnotation/docs/blob/master/guidelines.pdf

5. "*Thank*" construction: IMP_A Thanks$_P$, John$_A$!

The annotation of implicit arguments in UCCA's foundation layer is restricted to certain linguistic constructions, which is coarse, language-specific, incomplete, and unlike other distinctions in UCCA, is based on criteria of form rather than meaning. We are faced with the challenge of maintaining UCCA's idiosyncrasy of differentiating *Remote* and *Implicit* while extending its boundary to include a rather refined categorisation for implicit roles.

FrameNet (Baker et al., 1998) is a source of inspiration for UCCA and Scenes can be seen as frame evocations. FrameNet developed Fillmore (1986)'s notion of null-instantiation into three types—Constructional Null Instantiation (CNI), licensed by grammatical constructions, Definite Null Instantiation (DNI), equivalent to core Frame Elements mentioned previously in text or inferrable from the discourse, and Indefinite Null Instantiation (INI), an element that is unknown and nowhere to be retrieved.

Nevertheless, this trichotomy treats unfavourably many cases where implicit roles occur, such as Free Null Instantiation (FNI; Fillmore and Kay, 1993) and Identity of Sense Null Anaphora (ISNA; Kay, 2004). FNI is neither restricted to definite nor indefinite null arguments, and ISNA is null instantiation within noun phrases. Lyngfelt (2012) even argues that the unclear definition of FNI leads to much false categorisation—some FNIs are unspecified, some are generic, and some should be considered DNI.

UCCA's foundational layer mainly focuses on CNI, that is, the current annotated datasets only include grammatically licensed implicit arguments. So far, only a few corpora for implicit role labelling have been proposed, such as SemEval-2010 Task 10 (Ruppenhofer et al., 2009), Beyond NomBank (Gerber and Chai, 2010, 2012), Stern and Dagan (2014), and Multi-sentence AMR (O'Gorman et al., 2018). But none of them is based on a more comprehensive fine grained implicit role characterisation theory aside from FiGref, refined on three corpora mentioned above for recoverability studies.

Although FiGref is not available to the public, O'Gorman (2019) has counterbalanced previous studies on implicit role description and synthesized an inventory of eleven interpretation types for implicit roles distinguished by their referential behaviours. They are Script-inferrable pragmatic, Salient/recent, Deictic, Remembered Roles of Event Reference, Implicit "Sloppy anaphora" and Bridging, Genre-based Default, Type-identifiable, Generic ("People in General"), Cataphoric, Low-information, and Iterated Events Implicit Roles.

Recoverable implicit roles fall into the category of *Remote* Participants in UCCA, which typically calls for coreference resolution (Prange et al., 2019). We discern the eleven types of implicit roles mentioned above, recoverable or not, and extract those types where only true *Implicit* arguments occur, that is, the argument cannot be explicitly recovered from text, but inference and non-specificity are allowed as they are aligned with the definition of implicit arguments of UCCA. In the following section, we will analyze these implicit role types and argue the appropriateness of the set of categories we choose.

2.3 Parsing Methods

A limited number of studies attempt to parse implicit roles in semantic representations, though not fine-grained unrecoverable ones. Parsing implicit argument as an NLP task was introduced by Gerber and Chai (2010, 2012) and Ruppenhofer et al. (2009). A few researchers lay eyes on parsing implicit arguments on a coarse-level, which appear in the company of nominal predicates. Bender et al. (2011) extracted a dataset from Penn Treebank (Marcus et al., 1993) where 10 frequent linguistic phenomena occur, and investigate the depth of linguistic analysis. Among these 10 phenomena, tough adjectives and verbal gerund, tend to induce implicit arguments in the sentences. They ran seven parser systems, namely Stanford Parser (Klein and Manning, 2003), Charniak&Johnson Reranking Parser (Charniak and Johnson, 2005), Enju (Miyao et al., 2004), C&C (Clark and Curran, 2007), RASP (Briscoe et al., 2006), MSTParser (McDonald et al., 2005), and XLE/ParGram (Riezler et al., 2002), and used linguistic-phenomena-specific regular expressions to associate parsers' output with target dependencies. However, this study is restricted to identify certain linguistic phenomena rather than aiming at implicit arguments in general; neither has it made an effort to categorise their properties.

Roth and Frank (2015) introduced a rule-based induction method for identifying implicit arguments, which depends on semantic role labelling annotations, predicate-argument alignment, and coreference resolution techniques. Nevertheless, they refrained from differentiating implicit role types. Cheng and Erk (2018) build on Granroth-Wilding and Clark (2016)'s narrative cloze model to deal with argument cloze task, with more focus on entity-specific information. Then they evaluated the model on G&C dataset (Gerber and Chai, 2010) and achieved competitive performance. However, the G&C dataset was annotated within Nombank (Meyers et al., 2004) on paragraph level, and it was limited to instances where 10 heterogeneous themantic nominal predicates occurred. Similar to Roth and Frank (2015), several other works proposed parsing methods for SemEval 2010 data (Chiarcos and Schenk, 2015; Schenk and Chiarcos, 2016; Silberer and Frank, 2012) but they also only able to parse implicit arguments on a coarse level as well.

As for UCCA parsing, Hershcovich et al. (2017) first proposed a transition-based directed acyclic parser for UCCA. Hershcovich et al. (2018) then extended the parser using multitask learning, leveraging other meaning representations' data such as Abstract Meaning Representation (AMR; Banarescu et al., 2013), Semantic Dependency Parsing (SDP; Oepen et al., 2016) and Universal Dependencies (UD; Nivre et al., 2019, 2016). SemEval-2019 Task 1: Cross-lingual Semantic Parsing with UCCA (Hershcovich et al., 2019b) and MRP 2019 Task: Cross-Framework Meaning Representation Parsing (Oepen et al., 2019) further boosted parsing technologies for UCCA. Among these tasks, several prominent works stand out. Specifically, HLT@SUDA (Jiang et al., 2019) converts UCCA graph to constituent tree and used Constituent Tree Parser and BERT word embeddings (Devlin et al., 2018) as extra features; HIT-SCIR in MRP 2019 (Che et al., 2019) proposed a transition-based neural parser for UCCA using stacked long short-term memory network and BERT as well; Zhang et al. (2019) leveraged attention mechanism to build a neural transducer which is able to generate UCCA graph via semantic-relation sequences incrementally. Unfortunately, none of these parsers, to our knowledge, is able to parse implicit argument and label them flexibly.

Chapter 3

Refining Implicit Argument Annotation

3.1 Forming UCCA Implicit Argument Typology

For the sake of operability and consistency, we only focus on core arguments in Scenes where these arguments are essential to the meaning of corresponding predicates Goldberg (1992); Grimshaw (1993); Jackendoff (1992, 1997). Elements such as location, time, and manner are of little interest in this study whilst they are able to appear as foundational units or *Remote* Participant in UCCA.

As UCCA distinguishes *Remote* edges from *Implicit* units, it is natural to take advantage of this property to account for argument recoverability. The definition of implicit arguments in UCCA, particularly for its strong emphasis on the inability of explicit recovering from text, is not strictly corresponding with the eleven implicit role types.

Table 3.1 shows the comparison of the primary eleven implicit role types and UCCA's implicit argument set. Among these types of implicit roles in his inventory, four are definite implicit role constructions, viz. Salient/recent, Remember Roles, Script-inferrable, and Deictic. Only the last one of four, Deictic, we would consider a candidate category for UCCA's *Implicit* arguments. Salient/recent roles, which can be directly found in the recent prior discourse,

O'Gorman's Set	Implicit	Indefinite	Edge Cases	**UCCA's Typology**
Salient/recent	✓			✗
Remember Roles	✓			✗
Script-inferrable	✓			✗
Deictic	✓			Deictic
Cataphoric		✓		✗
Low-information		✓		Non-specific
Iterated Events		✓		Iterated-set
Bridging			✓	✗
Genre-based			✓	Genre-based
Generic			✓	Generic
Type-Identifiable			✓	Type-Identifiable

Table 3.1 The primary eleven implicit role types and the set for UCCA's implicit argument typology

is the quintessential type of DNI, and they can be easily replaced by pronouns. Remember Roles and Script-inferrable roles, however, require a cognitive and reasoning process, as the referents can be understood through a common ground in the text shared between the speaker and the addressee, or are reflecting a different facet of the same or a subordinate event. Deictic roles, albeit explicit reference to the speaker or addressee, is an extra-linguistic and cannot be annotated as *Remote* Participant, since we are unable to retrieve them explicitly in the text. Therefore, we incorporate it in our set of *Implicit* arguments categories.

Three out of eleven implicit roles are marked as clearly indefinite arguments, namely Cataphoric, Low-information Arbitrary role, and Iterated Events Implicit Roles. Cataphoric, which Bhatia et al. (2014) define as "pragmatically specific indefinite", is the only type we do not include in UCCA's typology since it relies heavily on the interpretation of the discourse whether it will be referred to again. We would annotate it as *Remote* Participant if the role is mentioned in a later text, or Non-specific type if not so as not to complicate the reasoning process.

The other four, Bridging implicit roles, Genre-based Default, Generic and Type Identifiable, are regarded as edge cases. Once again, we will only admit the latter three in our typology. As far as we are concerned, bridging in ellipsis situations might not refer to the same referent conceptually. Nonetheless, it can be clearly resolved in the text. Therefore, it will be annotated as a *Remote* edge in UCCA.

We will focus on referents that do not appear anywhere in the text. Therefore, we follow the philosophy of UCCA and propose six categories of *implicit argument*, that is, Deictic, Generic, Genre-based, Type-identifiable, Non-specific and Iterated-set. In the next section, we will present and exemplify each one of them.

3.1.1 Categorisation Set for UCCA Implicit Arguments

Deictic

Deictic implicit arguments specifically refer to the speaker or the addressee in a sentence. In example 1, the second-person subject is exhorted to take a certain action, and such imperative construction allows the subject not to appear in the text explicitly. Shown in example 2, Deictic can also occur with certain interjections, where the subject as the speaker is habitually implicit.

(1) Just$_D$ ask$_P$ them$_A$ exactly$_E$ what$_C$ [you$_A$ want$_S$ $(what)_A]_E$ $IMP_{Deictic}$.

(2) [Thank you]$_P$ guys$_{G/A}$ $IMP_{Deictic}$.

It should be mentioned that only in certain languages is imperative likely to induce implicit arguments.

As an execption, in morphologically rich languages, deictic information tends to be encoded morphologically due to person agreement Ingram (1971). For example, in Spanish "Estoy caminando.", the verb inflectionalready encodes Participants, indicating it is " I" that performed the action. Therefore, it would be annotated as "Estoy caminando$_{A+P}$." rather than "Estoy caminando$_P$ $IMP_{Deictic}$".

Generic

Generic implicit arguments denote "people in general" (Lambrecht and Lemoine, 2005). In example 3, the agent who "understands how this place has survived the earthquake" is not explicitly mentioned in the text, but it can be understood as it is the set of people in general. Example 4 can be construed as a gerund construction. "I" recommend taking a certain action. While the patient would not be specific, it conveys the message that "people in general" should follow such advice.

(3) It$_F$'s$_F$ impossible$_D$ to$_F$ understand$_P$ [how$_C$ [[this$_E$ place$_C$]$_A$ has$_F$ survived$_P$ [the$_F$ earthquake$_C$]$_A$]$_E$]$_A$ $IMP_{Generic}$.

(4) I$_A$ would$_F$ recommend$_P$ [not$_D$ using$_P$ [this$_E$ company$_C$]$_A$ $IMP_{Generic}$]$_A$.

Genre-based

Ruppenhofer and Michaelis (2010) found certain text genres, namely instructional imperatives, *labelese*, diary style, match report, and judgment-expressing quotative verbs, are closely linked with conventional omission. UCCA EWT corpus is based on online reviews of businesses and services by individuals. The review genre is so prominent acoss all dataset that it forms a pattern where the reviewers do not bother to mention the reviewees explicitly. In example 5 and 6, the review genre licenses the omission of the deliverer of the action "deliver" and server of the action "serve", as they refer to the reviewees by default, because in both contexts it is the restaurants that are being reviewed.

(5) Delivery$_P$ is$_F$ [lightning$_E$ fast$_C$]$_D$ $IMP_{Genre-based}$ $IMP_{Non-specific}$!

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

Type-identifiable

There exist some predicates allowing listeners to naturally think they "know" the omitted referents because of their high predictability. In example 7, the vague referent of "eat" can be understood from an inherent understanding of the listeners as "some kind of food". In example 8, the thing that "I drive" is not mentioned. Instead, it comes from common sense that the referent should be a kind of vehicle. Whatever kind it is, the lack of explicit mention barely affects the understanding of the text.

(7) It_A is_F my_A favourite_S [place_C [$(\textit{my})_A$ to_F eat_P $\textit{IMP}_{Type-identifiable}]_E]_A$.

(8) I_A 'll_F drive_P [an_Q $\text{hour}_C]_T$ [just_E for_R [their_S $(\text{volcano})_A]_E$ $\text{volcano}_C]_A$ $\textit{IMP}_{Type-identifiable}$.

Non-specific

Non-specific implicit arguments refer to the kind of referents that cannot be inferred or understood at all. Such required information absent from the context attributes to the low interpretability of the implicit arguments, leaving them non-specific. As in example 9 and 10, it is impossible to infer what is "delivered" or who "charged me" neither from common knowledge nor given context. Such kind of implicit arguments are commonly found in nominalization and passive because there are high possibilities that not all agents/patients are always mentioned despite the fact that they might be core frame elements.

(9) There_F is_F no_D delivery_P $\textit{IMP}_{Genre-based}$ $\textit{IMP}_{Non-specific}$.

(10) I_A don_F 't_D think_P [I_A have_F ever_T been_F charged_P before_T $\textit{IMP}_{Non-specific}]_A$.

Iterated-set

Iterated-set implicit arguments refer to a heterogeneous set of entities when the predicates are often an action that happens repeatedly, either iteratively or generically (Goldberg, 2001). For example, in sentence 11, the predicate "wait" implies high repetition, and the set of patients of "what/who I am waiting for" is so general that it does not hold any meaning beyond the context. As in example 12, the action " steal" designates a Scene where anything could be stolen, but "I" do not and will never steal. Unlike §3.1.1 Type-identifiable referring to a specific type of referents, the set of "things" in Iterated-set points to a vague set of entities to fill a role in a more functional way.

(11) I_A never_T wait_P [in_R the_F waiting_E $\text{room}_C]_A$ [[more_C [than_R $\text{two}_C]_C]_Q$ $\text{minutes}_C]_T$ $\textit{IMP}_{Iterated-set}$.

(12) I_A don_F 't_D steal_P $\textit{IMP}_{Iterated-set}$.

3.1.2 Inherent Ambiguity and "Continuum" of Coreness

Category Priority

There are a few cases when it is difficult to choose between two categories. Since UCCA does not aspire to annotate all possible interpretations (Abend and Rappoport, 2013), the annotator should make a best guess and choose one option. The first one is between Deictic and Generic, shown in example 13. The second one is between Genre-based and Non-specific, as in example 14. To keep the annotation consistent and maintain as much information as possible, we always choose Deictic over Generic, and Genre-based over Non-specific if available.

(13) [The$_F$ experience$_C$]$_A$ [with$_R$ every$_Q$ department$_C$]$_A$ has$_F$ been$_F$ great$_D$ $IMP_{Deictic}$.

(14) I$_A$ will$_F$ definitely$_D$ refer$_P$ [[my$_A$ friends$_{A/S}$]$_C$ and$_N$ [(my)$_A$ family$_{A/S}$]$_C$]$_A$ $IMP_{Genre-Based}$.

Nominalization As Occupation

It is a judgment call whether the patient of Process instantiated by a profession should be annotated at all. Even so it remains debatable which category such kind of implicit argument belongs to. In the current version of corpus, we will always annotate it as Type-identifiable. As in example 15, the patients of whom has been taught is unclear but neither require clarification. The Scene of teacher/teaching is annotated with a Type-identifiable implicit argument denoting a type of people recieving education.

(15) They$_A$ are$_F$ [very$_E$ good$_C$]$_D$ teachers$_{A/P}$ $IMP_{Type-identifiable}$.

3.2 Refined Implicit Corpus

In furtherance of investigating the characteristics of UCCA's implicit arguments, we piloted a study to revisit and refine part of English Web Treebank[1]

[1]https://catalog.ldc.upenn.edu/LDC2012T13

	# Passages	# Passages w/ Implicit	# Sentences	# Sentences w/ Implicit	# Implicit (Valid)
Original	200 (out of 723)	103	306	111	153 (98)
Refined	200	116	393	221	**415 (385)**

Table 3.2 Statistics of the UCCA EWT dataset sampled passages before and after reviewing. The additional implicit arguments result from both refining UCCA EWT and conducting new annotation according to out fine-grained typology. Implicit (Valid) denotes implicit argument whose role is Participant in UCCA.

annotated with the UCCA foundational layer.[2] 200 passages were randomly selected for experiment from the total 723 passages comprising the UCCA EWT dataset.

We use UCCAApp to carry out annotation. The process is divided into two stages. Firstly, we create passage-level review tasks to check the existing annotation whether they contain implicit arguments, and add missing arguments if necessary. Secondly, we split passages into sentences, create tasks with refinement layer and then annotate with corresponding fine-grained implicit categories. Since all the annotation works were undertaken by one single annotator, the dataset preferably serves as a demonstration of concept, and thus further measurement of inter-annotator agreement would be desired to establish a more sound dataset.

3.2.1 Revisiting Original EWT UCCA Dataset

The original implicit argument annotation in EWT UCCA corpus is restricted only to put concern on constructional null instantiations, and when a unit lacks a Center or a Process/State, which is out of the scope of this study. We only regard implicit argument whose category is Participant in UCCA as valid implicit in this research. Therefore, it is necessary to check and refine the dataset. Considering the original corpus was annotated on passage-level, whereas our new dataset will be done on sentence-level, *Remote* edges across sentences will be treated by adding a new *Implicit* node under its origin Scene.

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

Table 3.2 shows the statistics before and after reviewing according to the new UCCA Implicit Argument Typology. It can be seen that in the refined dataset, 116 out of 200 passages contain implicit arguments, 13 more passages than the original dataset, in which only 103 passages contain implicit arguments. Yielding an increase of 255%, the review process added 250 more valid implicit arguments in the corpus.

3.2.2 Statistics

Tokenized and split according to the Universal Dependencies English Web Treebank (Hershcovich et al., 2019a; Silveira et al., 2014), this pilot corpus consists of 3702 tokens, 1411 nodes, and 4759 edges over 393 sentences. In total, 385 valid implicit arguments are found and annotated on 221 sentences. Figure 3.1 demonstrates each implicit argument category with its corresponding number in the pilot refined implicit corpus, and illustrates the percentage of each implicit category in the dataset. One can see that Genre-based and Non-specific are the two most frequent categories, both of which have more than 100 instances in the dataset, making up approximately 52% combined. They are followed by Generic and Deictic, and each occupies about 17%. Type-identifiable comes penultimate with 39 instances, while Iterated-set is the least frequent type, which merely has 12 instances, accounting for 3.12% of the whole corpus.

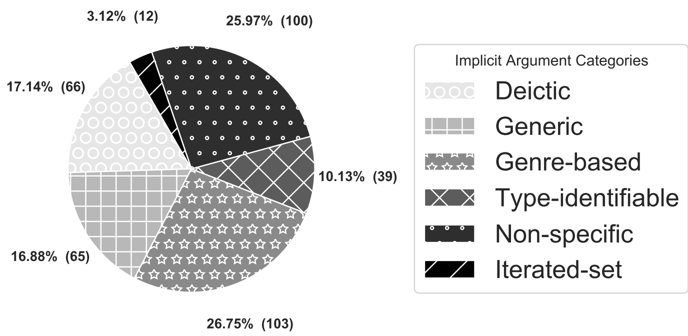

Fig. 3.1 Statistics of pilot UCCA implicit dataset

	Type-Identifiable	Deictic	Generic	Non-specific	Genre-based	Iterated-set	Script-inferrable	Other	Invalid
Ours	9.79%	17.29%	17.29%	25.93%	25.93%	2.59%	\	\	\
FiGref	7%	5%	10%	4%	\	\	9%	12%	53%

Table 3.3 Relative frequency of annotated implicit types in UCCA's refinement layer and FiGref's annoation for non-recoverable roles in Multi-sentence AMR. The first four types are shared by both annotation corpora. The following two are exclusive to UCCA's refinement layer. The last three are additonally introduced into FiGref's set of interpretation types.

3.2.3 Comparisons to FiGref Annotation

FiGref is annotated over Multi-sentence AMR training data, SemEval-2010 Task 10 training data, and Beyond NomBank. It contains 856 implicit roles classified into 14 types, which includes all 11 proposed interpretation types except Genre-based, and four more kinds of invalid roles to account for those implicit roles of low importance or deal with tricky occasions. However, FiGref has a relatively low Cohen's κ score (Cohen, 1960) of 55.2 due to its ambiguity and relatively high annotation complexity.

Comparatively, we do not annotate genuinely invalid implicit role in the implicit argument refinement layer for UCCA since we consider it is not a core element of the event Scene. As UCCA differs *Implicit* units from *Remote* edges, it naturally reduces the annotation complexity as we only need to annotate essential and unrecoverable implicit arguments within the set of six types we propose in §3.1.1.

Owing to the distinct annotation design and lack of statistics provided by FiGref, it is difficult to perform a comparative quantitative study between UCCA's implicit refinement layer and FiGref. However, we are able to look into the relative frequency of annotated implicit types in UCCA and FiGref's annotation for non-recoverable roles in Multi-sentence AMR shown in Table 3.3.

The distribution distinction can be possibly explained by the different domains of the corpora and their annotation methodology. We keep Genre-based type to account for the particular "review" genre in EWT dataset. Among the three types FiGref has introduced, We can see that invalid roles dominate the FiGref annotation with 53%. This is because a large amount of non-important interpretations of null-instantiation are taken into account in FiGref, whereas the implicit refinement annotation designed for UCCA is limited to essential important implicit arguments so as to lower annotation complexity and ambiguity.

Chapter 4

Parsing Implicit Arguments in UCCA

4.1 Transition-Based Dependency Parser

In recent years, the area of computational semantics has seen remarkable advances in parsing technologies, where research's target shifts from parsing tree-structured representations to the more expressive beyond-syntax graph-based representations, with the aim of representing sentence-level semantic structure more adequately (Oepen et al., 2019). Along with factorization-based parsing, composition-based parsing and translation-based parsing (Zhang et al., 2019), transition-based architecture is one of the widest adopted methods for meaning representation parsing, which has its origin in syntactic dependency tree parsing (Nivre, 2003; Yamada and Matsumoto, 2003). What differs transition-based meaning representation parsing from dependency parsing is that the semantic representation is more elaborated with higher computation complexity due to the existence of attributes, properties, labels of nodes or edges, non-terminal nodes, various types of node anchoring, edge reentrancies and discontinuous constituency.

A transition-based parser will take the representation of parsing state as input and predict a most probable transition action. The parser will then be used to reconstitute the representation graph in a deterministic way. The set of transition actions can vary out of different consideration. Typically,

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	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	$-$	$\mathrm{i}(x) < \mathrm{i}(y)$
[root]	$\emptyset$	V	E	FINISH	$\emptyset$	$\emptyset$	V	E	$+$	

Table 4.1 The transition set of Refined Implicit Parser. We write the **stack** with its top to the right and the **buffer** with its head to the left. $(\cdot,\cdot)_X$ denotes a primary 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 can not have a primary parent. The newly generated node by IMPLICIT action is prohibited to have any descendent. This table is adapted from Hershcovich et al. (2017).

SHIFT, REDUCE, LEFTARC, RIGHTARC are employed as basic actions for transition parsers (Nivre, 2003; Yamada and Matsumoto, 2003). Among others, NODE action is designed to generate concept node when it is not anchored to word tokens from the input (Hershcovich et al., 2017). Their work has also proposed LEFTREMOTE and RIGHTREMOTE to deal with edge reentrencies.

HIT-SCIR (Che et al., 2019) achieved best performance for UCCA parsing in MRP 2019 (Oepen et al., 2019). In this thesis, we will look HIT-SCIR 2019 as the prototype parsing system, inheriting two advantages; we use stack LSTM to stabilize gradient descent process and speed up training; we enrich contextual information by employing pre-trained language model BERT word embeddings as a feature input. We extend the parser so that it can read implicit node properly; then we add an IMPLICIT transition action so that implicit node can be dynamically generated and labelled. In the next subsection, we describe the set of transition actions that we include in our parsing system.

4.1.1 Transition System for Parsing Implicit Arguments

Our parsing state design is based on that of Nivre (2003). A stack S holds processed words. B is a buffer containing tokens or nodes to be processed. V is a set of nodes, and E is a set of 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 action based on its current parsing state. Oracle action sequences will be generated for training on gold-standard annotations.

Based on HIT-SCIR 2019 parser, our transition system incorporate all of their 9 transition actions, that is, LEFT-EDGE, RIGHT-EDGE, SHIFT, REDUCE, NODE, SWAP, LEFT-REMOTE, RIGHT-REMOTE and FINISH. We improve the parser by introducing a new transition action IMPLICIT so that in spite of no existing element on buffer corresponding to implcit arguemnt, when the parser encounters one, an implcit node can be generated and add labelled dependency arc accordingly.

Table 4.1 shows the transition set. We describe in detail the ten transition actions below.

- SHIFT: Together with REDUCE below, these two are the most standard actions in transition process. SHIFT is performed when there does not exist arc between b_0 and any element in S other than s_0. It will push all elements in list and s_0 into S.
- REDUCE: Pop s_0 from the S when it is neither parent nor child of any element in the B.
- NODE$_X$: Modelling after transition-based constituent parsing (Sagae and Lavie, 2005), NODE transition creates a new non-terminal node. Such node will be created on the buffer by NODE$_X$ transition, which is a parent of s_0 on the S with X-labelled edge.
- SWAP: To deal with non-planar graph (generalization of non-projective trees), in another word, discontinuous constituents, SWAP pops the second node on the S and add it to the top of the B.

- LEFT-EDGE$_X$ and RIGHT-EDGE$_X$: These two actions add an X-labelled primary edge between the first two elements on the S. When the first element is the parent of the second element on the 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 where the arc points to.
- LEFT-REMOTE$_X$, RIGHT-REMOTE$_X$: Similar to LEFT-EDGE$_X$ and RIGHT-EDGE$_X$, yet these two transitions create remote edges aiming at solving reentrencies. The X-labelled edge will be assigned a Remote attribute.
- FINISH: The terminal transition. FINISH pops the Root node and marks the transition state as Terminal.
- IMPLICIT: See below.

Besides the nine transition actions described above, we introduce IMPLICIT$_X$ action. Such action creates a new concept node on the B as the child of the first element on the S, with an X-labelled edge. The IMPLICIT action is different from NODE action in the sense that the one-step generated dependency arc is the child of the element on the S rather than its parents like what NODE does. Equally importantly, the new node is prohibited have any child in contrast to the kind of nodes NODE action generates.

4.2 Stacked Long Short-Term Memory Networks

LSTMs are developed to cope with the vanishing gradient problems in recurrent neural networks (RNNs) (Graves, 2013; Hochreiter and Schmidhuber, 1997). At each timestep, RNNs take a vector $\mathbf{x}_t$ and compute a new hidden state $\mathbf{h}_t$ by concatenating the hidden state of previous timestep $\mathbf{h}_{t-1}$ and the input, applying a linear map to it, and passing the output to a logistic sigmoid function. RNNs is designed to model long-range dependencies. Nonetheless, it suffers from an exponential decay in the error signal with time due to the constant application of suqashing function.

LSTMs improved on RNNs by using a memory cell $\mathbf{c}_t$ to represent the combination of state $\mathbf{h}_{t-1}$ and vector input. LSTMs also use three gates: input gate $\mathbf{i}_t$ controls the proportion of the current input to flow into the cell $\mathbf{c}_t$; forget gate $\mathbf{f}_t$, controls the proportion of the previous memory to remain in the cell $\mathbf{c}_t$;

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{W}_{ic}\mathbf{c}_{t-1} + \mathbf{b}_i\right) \tag{4.1}$$

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{W}_{fc}\mathbf{c}_{t-1} + \mathbf{b}_f\right) \tag{4.2}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh\left(\mathbf{W}_{cx}\mathbf{x}_t + \mathbf{W}_{ch}\mathbf{h}_{t-1} + \mathbf{b}_c\right), \tag{4.3}$$

$$\tag{4.4}$$

output gate $\mathbf{o}_t$ controls the proportion of the value in the current cell $\mathbf{c}_t$ to be used for computing the output activation of the LSTM. σ is the element-wise sigmoid function and $\odot$ is the element-wise Hadamard product.

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{W}_{oc}\mathbf{c}_t + \mathbf{b}_o\right) \tag{4.5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh\left(\mathbf{c}_t\right). \tag{4.6}$$

Dyer et al. (2015) proposed a parsing model that uses stack Long Short Term Memories (stack LSTM). The model introduces an additional stack pointer to conventional left-to-right LSTM so that the location of the stack pointer can determine which cell in the LSTM provides $\mathbf{c}_{t-1}$ and $\mathbf{h}_{t-1}$ for computing the new memory cell. Shown as Figure 4.1, The stack LSTM has three operations: POP operation can move the stack pointer one position back pointing to the previous element; INSERT operation adds a new element at the end of the list. The element has a back pointer to the previous TOP; QUERY function returns the output vector that the stack pointer points to.

Therefore, the transition parser represents the stack S, buffer B and action history with stack LSTM, and will output a transition action a at state s that maximizes the score, which is computed as follows (Che et al., 2019):

$$p(a|s) = \frac{\exp\left\{g_a \cdot \textsc{stack}\,\mathrm{LSTM}(s) + b_a\right\}}{\sum_{a'} \exp\left\{g_{a'} \cdot \textsc{stack}\,\mathrm{LSTM}(s) + b_{a'}\right\}}. \tag{4.7}$$

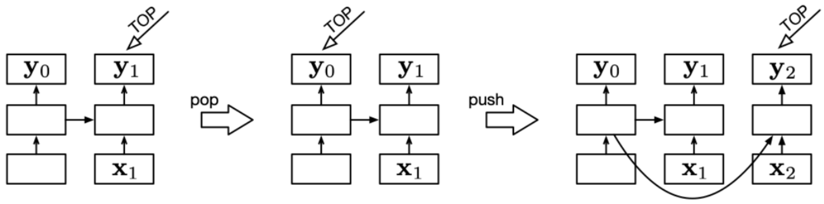

Fig. 4.1 Visualization of stack LSTM (Dyer et al., 2015). TOP indicates the stack pointer. This figure schematizes three steps: a stack with a single element, the result after POP and the result after INSERT operation.

STACK LSTM encodes the state s into a vector. g_a and b_a represent the embedding vector and bias vector of action a respectively. For stack LSTM, POP and QUERY functions do not require heavy computing to the exclusion of the INSERT operation. Therefore, Che et al. (2019) propose to use batch training for speeding up training and increase gradient stability (Kiperwasser and Goldberg, 2016). The parser constructs batch data on operation level instead of data level by forming a batch with under-computed operations between different pieces of data.

4.3 BERT

Recent techniques in language models benefit from pre-training on large amount unsupervised text, showing the ability to transferring contextual sentence-level information to improve the parsing accuracy (Devlin et al., 2018; Howard and Ruder, 2018; Peters et al., 2018).

Previous language models mainly adapt either unidirectional (Radford et al., 2018) or semi-biderectional architecture (Peters et al., 2018), Deep Bidirectional Encoder Representations from Transformers (BERT; Devlin et al., 2018) is the first language model based on multi-layer bidirectional Transformer (Vaswani et al., 2017) in a way that it joins both left and right context during training on unsupervised data. BERT is pre-trained over two tasks, viz. masked language model (MLM) inspired by the Cloze task (Taylor, 1953) and Next Sentence Prediction (NSP).

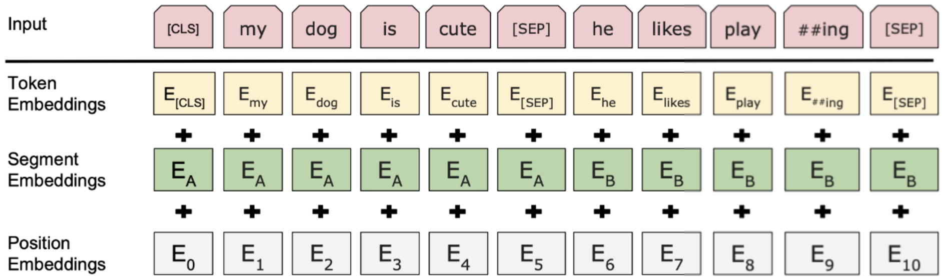

Fig. 4.2 BERT input representation (Devlin et al., 2018), which includes token embeddings, segment embeddings and position embeddings.

In MLM, 15% of tokens from input are randomly masked, and the goal is to predict the masked tokens based on its context. Unlike traditional left-to-right pre-training, MLM allows the language model to considerate context in both left and right direction. Besides, to mitigate the disadvantage of mismatching pre-traning and fine-tuning, the training data gernerator would treat the supposed-to-be masked token by (1) replacing it with `[MASK]` 80% of the time (2) replacing it with a random token 10% of the time (3) keep it unmasked 10% of the time. Therefore, the Transformer encoder is forced to keep a distributional contextual representation of every input token because it would not know which token it should predict or whether a token has been masked with another token.

In order to capture the relationship between sentences, the Transformer encoder is pre-trained for a binary Next Sentence Prediction (NSP) task in which 50% of the time a sentence A is followed by the actual next sentence B, labelled as `IsNext`, while the rest 50% of the time it is followed by a random sentence, labelled as `NotNext`. The task has been proven to be benificial for Question Answering (QA) and Natural Language Inference (NLI) (Devlin et al., 2018). The BERT model segments text into sub-word units using a wordpiece tokenizer (Wu et al., 2016) with a 30,0000 token vocabulary. Sentence pairs are packed in a sequence. Every sequence is started with a special classification token `[CLS]`, whose corresponded representation in the final hidden state would be used for sequence classification. Between each sentence, a separator token `[SEP]` is inserted. In addition, as is shown in

Figure 4.2, segment embeddings are also learned to tell which sentence every token belongs to.

We adopt the BERT large cased model (`wwm_cased_L-24_H1024_A16`) [1] with a self-attention network of 24 layers, 16 attention head per layer, and hidden dimension of 1024 to encode the sentence. We represent each token with the first wordpiece of it after applying a scalar mix on all layers of transformer.

4.4 Data Preprocessing

Since our prototype parser HIT-SCIR 2019 is designed for MRP 2019 task, it is naturally only able to take MRP format data. Therefore, we need to convert UCCA XML data to MRP foormat using the open-source `mtool` software (the Swiss Army Knife of Meaning Representation) [2]. As the UCCA data provided in MRP 2019 share task did not contains implicit information, HIT-SCIR 2019 is not able to read our new dataset UUCA Refined EWT. we modified and enabled the parser to read node properties, and to convert UCCA data from and to MRP format. The updated version of `mtool` is available at Github [3].

With the UCCA data ready in MRP format, HIT-SCIR system is designed to read input as a set of triple of incoming arc, outgoing arc and arc label. Therefore, we have to reconstruct the input data further. For node anchoring, we link the layer 0 nodes with surface tokens with the edge labelled "Terminal". In post-processing, we collapse "Terminal" edge to combine surface tokens and layer 0 nodes, and extract alignment information from how the MRP 2019 share task companion data alignment segmented anchors.

4.5 Evaluation

To evaluate implicit argument detection in UCCA, we compare predicted graph G_p with implicit arguments I_p to reference graph G_r with implicit

[1] https://github.com/google-research/bert
[2] https://github.com/cfmrp/mtool
[3] https://github.com/ruixiangcui/mtool

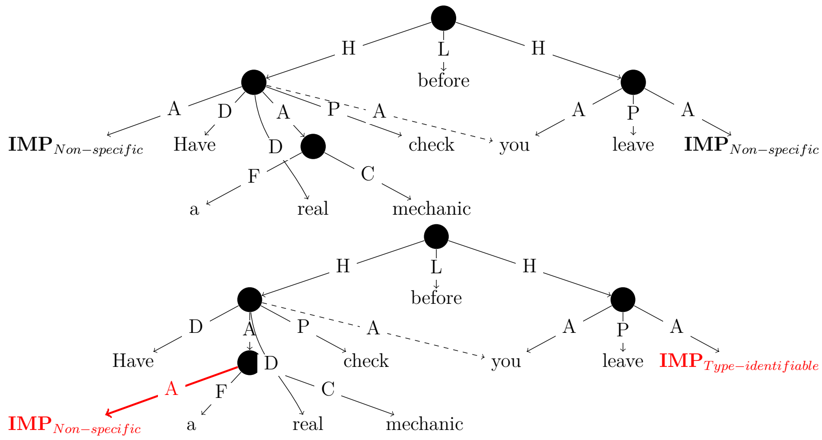

Fig. 4.3 Evaluation example 1: Upper is the golden graph. In the lower graph, there are a mis-matched implicit node and a wrong-labelled implicit node in red. The labelled implicit evaluation precision, recall and F1 scores are 0, 0 and 0; The unlabelled implicit evalutation precision, recall and F1 scores are 0.5, 0.5 and 0.5.

arguments I_r over the same sequence of terminals $W = \{w_1, \ldots, w_n\}$. For an implicit node i in each graph, we look up to its parent node j, marking the set of terminals spanned by it as the yield $y(j) \subseteq W$, with edge label $\ell = (j, i)$. Given an implicit node in reference graph i_r, if an implicit node in predicted graph i_p 's parent j_p yields the same terminals as these of i_r 's parent j_r, and edge label l_p matches with l_r, we regard it as a correct implicit match. Define the set of mutual implicit arguments between G_p and G_r:

$$M(G_p, G_r) = \{(i_1, i_2) \in I_p \times I_r | y(j_1) = y(j_2) \wedge \ell_p(i_1) = \ell_r(i_2)\}$$

F-score is taken the harmonic mean of labelled precision and recall, defined by dividing $|M(G_p, G_g)|$ by $|G_p|$ and $|G_r|$, respectively. Apart from labelled implicit argument evaluation, we also introduce unlabelled evaluation, which only requires parents' spans match.

We demonstrate two examples here. In figure 4.3, reference graph has two implicit arguments. The parent of the first implicit argument spans {have, a, real, mechanic, check} and the parent of the second implicit argument spans {you, leave}. In the predicted graph, there are also two implicit arguments predicted. The first implicit node's parent spans {a, mechanic} while the second one spans the same terminals as the reference graph. We can see that

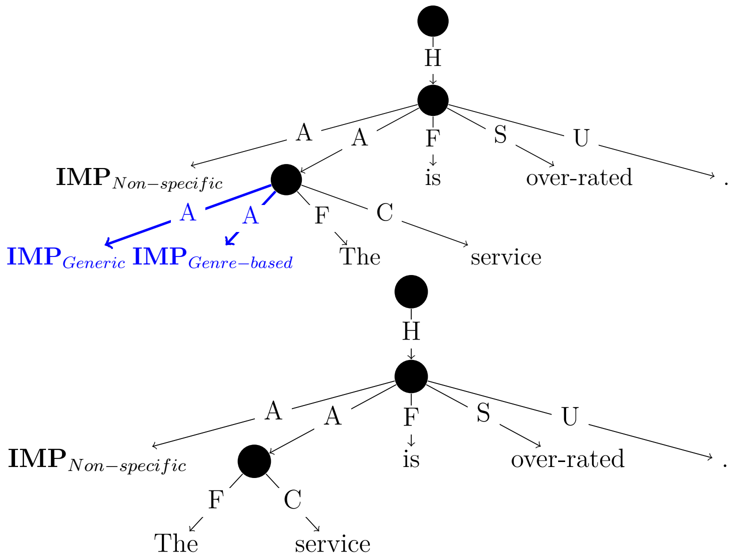

Fig. 4.4 Evaluation example 2: Upper is the golden graph with two implicit nodes marked in blue, indicating they are not detected comparing to the partial matched graph below. Both their labelled and unlabelled implicit evaluation precision, recall and F1 scores are 1, 1/3 and 0.5.

for the first implicit argument, their spans do not match—the second matches but with the wrong label. Therefore, in labelled implicit argument evaluation, the precision, recall, and F1 scores are 0, 0 and 0. The unlabelled precision, recall and F1 scores would be 0.5, 0.5 and 0.5. In figure 4.4, the reference graph has three implicit arguments, Non-specific, Generic and Genre-based for each. They span {The, service, is, overrated, .}, {The, service} and {The, service} respectively. The predict graph only manages to predict one implicit argument with the correct label Non-specific. So in both labelled and unlabelled evaluation, the precision, recall, and F1 scores are 1, 1/3 and 0.5.

4.6 Experiment Setup

The model is implemented using the open-resource NLP library AllenNLP built on Pytorch (Gardner et al., 2017). We evaluate the original HIT-SCIR 2019 parser on both the Original UCCA EWT and Refined Implicit EWT and use it as our baseline for comparison. Default settings are used in both cases. We use the same hyperparameters as Che et al. (2019) except batch size,

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 4.2 Implicit Parser hyperparameters.

adjusted from 8 to 4. The hyperparameter setting for our model is listed in Table 4.2. We do not hyper-tune on either original EWT or Refined Implicit EWT.

We randomly split train, validation and test sets according to their passage number with the ratio of 0.75, 0.125 and 0.125. After splitting these 116 passages composed of 393 sentences, we obtained 285, 59, 49 sentences for train, dev and eval set respectively. Table 4.3 shows detailed statistics of train, dev and eval set of both original EWT and Refined Implicit EWT, including numbers of sentences, tokens, nodes, instances of each implicit category and the sum of implicit arguments.

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 Test	535	5381	7431	11907							
Overall					not refined	not refined	not refined	not refined	not refined	not refined	153
IMP Train	285	2671	1019	3443	46	39	65	17	70	8	293
IMP Dev	59	540	211	705	7	14	10	9	16	0	64
IMP Eval	49	489	184	620	7	11	15	8	4	1	58
Overall					60	64	90	34	90	9	415

Table 4.3 Statistics of train, dev and eval dataset in Original EWT and Refined Implicit EWT. For each dataset, number of sentences, number of tokens, number of nodes, number of instances of 6 implicit categories and their sum are listed.

4.6.1 Training

We trained the baseline parser (original HIT-SCIR 2019) and Implicit Parser on both Original UCCA EWT and Refined Implicit EWT. Shown as Table 4.4, the training time is 2 days 22 hours for original HIT-SCIR 2019 on Original UCCA EWT (50 epochs). Best epoch is 3rd; 3 hours for original HIT-SCIR 2019 on Refined Implicit EWT (30 epochs). Best epoch is 22nd; 1 day 8 hours for Implicit Parser on Original UCCA EWT (10 epochs), with the best epoch being the 3rd; And finally, 8 hours for Implicit Parser on Refined Implicit EWT (50 epochs). The best epoch is 11th. One can see that both parsers achieved the best performance at early stage on Original UCCA EWT. However, Implicit Parser took longer time to train on Original UCCA EWT while the training time was less lengthy on Refined EWT.

	Orignial UCCA EWT			Refined EWT		
	Training time	# Epochs	# Best Epoch	Training time	# Epochs	# Best Epoch
orignial HIT-SCIR 2019	2 days 22 hours	50	3	3 hours	30	22
Implicit Parser	1 day 8 hours	10	3	8 hours	50	11

Table 4.4 Training details of orignial HIT-SCIR 2019 and Implicit Parser on orignial UCCA EWT and Refined EWT, including training times, the number of best epoch and total epochs.

4.7 Results

Table 4.5 presents experiment results of four experiments by two parsers, the original HIT-SCIR 2019 parser as the baseline and our Implicit Parser on Original UCCA EWT data and Refined Implicit Data respectively. Regarding performance on Refined Implicit EWT, the baseline is not able to predict implicit argument as expected. The Implicit Parser managed to reach 0.162 on labelled implicit F-score, and 0.297 on unlabelled implicit F-score.

Based on the evaluation method mentioned in section 4.5, the labelled precision and labelled recall for Refined EWT are 0.214 and 0.13; the unlabelled precision and unlabelled recall are 0.393 and 0.239. For primary edge and remote edge evaluation, noticeably, the Implicit Parser also outperforms the baseline on primary edges by 0.007 in F-score on Refined EWT. However, the baseline produced better results for Refined EWT in remote edges

	Primary			Remote			**Implicit**					
	LP	**LR**	**LF**	**LP**	**LR**	**LF**	**LP**	**LR**	**LF**	**UP**	**UR**	**UF**
Baseline on Original EWT	0.710	0.701	0.706	0.547	0.365	0.438						
Baseline on Refined EWT	0.495	0.467	0.480	0.538	0.304	0.389	1	0	0	1	0	0
Implicit Parser on Original EWT	0.675	0.597	0.634	0.527	0.344	0.416						
Implicit Parser on Refined EWT	0.503	0.472	0.487	0.333	0.1	0.154	**0.214** (6/28)	**0.13** (6/46)	**0.162**	**0.393** (11/28)	**0.239** (11/28)	**0.297**

Table 4.5 Experiment results on Original EWT and Refined 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 prediction.

substantially, and Original EWT both in terms of primary edges and remote edges.

4.7.1 Error Analysis

As is indicated in Table 4.5, the Implicit Parser successfully predicted 6 implicit arguments with the correct implicit category labelled. In the unlabelled evaluation, 11 implicit arguments are predicted, without taking into consideration their implicit labels. It is worth noting that the current evaluation metric follows a strict matching rule, meaning that if 2 implicit arguments are under the same parent span, they would not be considered correctly matched unless both arguments are predicted with the correct labels.

Table 4.6 shows the confusion matrix of Implicit Parser's performance on the set of test data of Refined EWT. The Implicit Parser has emitted predictions for all implicit categories. However, only six of these predictions are correctly labelled, whose distribution are 2, 1, 3 for Deictic, Generic and Genre-based respectively. Genre-based is the most predicted categories with a total of 11, while Iterated-set has only been predicted once, which was also incorrectly labelled. One can see that Generic has the best prediction recall, which is 1/(2+1), with the runner-up being Deictic, which is 2/(4+2+1). Unfortunately, the parser failed to predict Non-specific, Iterated-set and Type-identifiable despite its 3, 3 and 1 try for each category. We assume that due to the few instances of iterated-set and Type-identifiable in all train, dev and eval set, it is difficult for the parser to see such instances enough times so as to learn anything from them. Distinctively, despite the fair number of

Non-specific instances, their internal difference might be too great that the parser could not capture traits of such type of implicit arguments. As a result, it is only realistic for the parser to predict Deictic, Generic and Genre-based with the current conditions provided so far.

<table>
<tr><th></th><th><UNMATCHED></th><th>A|Non-specific</th><th>A|Deictic</th><th>A|Generic</th><th>A|Genre-based</th><th>A|Iterated-set</th><th>A|Type-identifiable</th><th>A|Generic&A|Genre-based</th><th>A</th><th>P</th></tr>
<tr><td><UNMATCHED></td><td>0</td><td>5</td><td>4</td><td>2</td><td>9</td><td>1</td><td>8</td><td>4</td><td>1</td><td>1</td></tr>
<tr><td>A|Non-specific</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>A|Deictic</td><td>3</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>A|Generic</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>A|Genre-based</td><td>7</td><td>0</td><td>1</td><td>0</td><td>3</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>A|Iterated-set</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>A|Type-identifiable</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>A|Deictic&A|Genre-based</td><td>0</td><td>0</td><td>0</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Table 4.6 Confusion matrix on test set of Refined EWT: The column is the predicted labels while the row is the actual labels.

Chapter 5

Discussion

5.1 Objectives

This thesis proposed a fine-grained typology for implicit argument annotation. Based on the typology design, we constructed an open dataset with a rich conceptually inspired annotation of implicit arguments on Universal Conceptual Cognitive Annotation (UCCA). The corpus comprises 200 passages. We also provide descriptive statistics of the data with comparison to FiGref. Subsequently, we developed the first parser that can deal with and predict implicit arguments for meaning representation schemes and evaluated its performance on the new dataset.

5.2 Discussion and Challenges

Our work focuses on implicit arguments in meaning reprensentations, specifically for UCCA. Although we argue that the proposed six-type implicit argument typology requires a relatively low cognitive load and should be easy for annotators to learn, the current dataset retains its pilot status due to the fact that it only involved one annotator; therefore no inter-annotator agreement measurement could be provided and used to conduct evaluation of annotation ambiguity and complexity. However, we expect to invite a new annotator in the future to perform the experiment mentioned above, and formally re-assess both the soundness of the annotation scheme we opt for and the reproducibility of annotation work.

In addition, since UCCA has already been annotated in multiple languages and has been proven that it largely maintains semantic structures across sentences expressing same meanings across languages. The second future work direction is to annotate more data on languages other than English and to investigate to what extent implicit arguments remain in different languages, and look into substantial convergence and divergence of fine-grained implicit type distribution across languages. Such work is important for deep natural language understanding that no existing NLP technologies can address.

Given the small data size of Refined EWT, to train an effective parser ought to be considered a tough challenge - and the same goes for conducting a proper evaluation. As has been discussed about previous works on parsing implicit roles in Section 2.3 , much as they did not target parsing fine-grained implicit annotation, it would be interesting to see how well linguistic-phenomena-specific regular expression parser the like of work by Bender et al. (2011) can capture implicit arguments in Refined UCCA EWT; Since our Refined parser is the first transition-based parser that is able to handle implicit arguments, it is also worth evaluating its performance on other implicit datasets such as G&C (Gerber and Chai, 2010), SemEval 2010 (Ruppenhofer et al., 2009) and especially on parsing implicit arguments licensed by nominal predicates (Roth and Frank, 2015).

5.3 Conclusion

We proposed a novel typology for different implicit arguments in UCCA, which allows annotation with a relatively low cognitive load. We refined and reviewed part of the existing UCCA English Web Treebank dataset and piloted annotation of our guidelines with a refinement layer of fine-grained implicit arguments. Moreover, it is currently the only published dataset with this kind of information. Then we introduced the first parser that is able to handle and predict implicit nodes dynamically, and label them with fair accuracy in meaning representations. Finally, we evaluated it on the new dataset.

Our work addresses a deep linguistic problem, impossible to deal with current machine learning approaches, in implicit role interpretation and provides an example for conducting research about it. Our study has the potential to benefit natural language understanding, especially in genre-specific context, e.g., social media and customer reviews. It is anticipated that our work will inspire tailored design of implicit role annotation in other meaning representation frameworks and improve current semantic parsing work. Downstream tasks such as coreference resolution and human-robot interaction are also likely to benefit from reducing ambiguity and increase contextual understanding by explicitly modelling implicit arguments.

While the pilot work is promising, it is crucial to evaluate the annotation quality further, expand the corpus to all UCCA EWT and possibly extend it to multiple languages. To predict and label implicit argument with higher accuracy would be an important but challenging task in the future.

References

Abend, O. and Rappoport, A. (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.

Abend, O., Yerushalmi, S., and Rappoport, A. (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.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, Montreal, Quebec, Canada. Association for Computational Linguistics.

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. (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.

Bender, E. M., Flickinger, D., Oepen, S., and Zhang, Y. (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.

Bender, E. M. and Koller, A. (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.

Bhatia, A., Simons, M., Levin, L., Tsvetkov, Y., Dyer, C., and Bender, J. (2014). A unified annotation scheme for the semantic/pragmatic components of definiteness. In *Proceedings of the Ninth International Conference on*

Language Resources and Evaluation (LREC'14), pages 910–916, Reykjavik, Iceland. European Language Resources Association (ELRA).

Briscoe, T., Carroll, J., and Watson, R. (2006). The second release of the RASP system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 77–80, Sydney, Australia. Association for Computational Linguistics.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, Michigan. Association for Computational Linguistics.

Che, W., Dou, L., Xu, Y., Wang, Y., Liu, Y., and Liu, T. (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.

Cheng, P. and Erk, K. (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.

Cheng, P. and Erk, K. (2019). Implicit argument prediction as reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6284–6291.

Chiarcos, C. and Schenk, N. (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.

Clark, S. and Curran, J. R. (2007). Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

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

Croft, W. and Cruse, D. A. (2004). *Cognitive linguistics*. Cambridge University Press.

de Saussure, F. (1916). Cours de linguistique générale. *Paris: Payot. edited posthumously by C. Bally and A. Riedlinger.*

de Saussure, F. (1978). Cours de linguistique générale. *Paris: Payot. edited posthumously by C. Bally and A. Riedlinger.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Dixon, R. M. (2010a). *Basic linguistic theory Volume 1: Methodology*, volume 1. Oxford University Press.

Dixon, R. M. (2010b). *Basic linguistic theory Volume 2: Grammatical topics.* Oxford-New York, Oxford University Press.

Dixon, R. M. (2012). *Basic linguistic theory: Further grammatical topics, Vol. 3.* Oxford: Oxford University Press.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A., and Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Elazar, Y. and Goldberg, Y. (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.

Fillmore, C. J. (1986). Pragmatically controlled zero anaphora. In *Annual Meeting of the Berkeley Linguistics Society*, volume 12, pages 95–107.

Fillmore, C. J. and Kay, P. (1993). *Construction grammar coursebook: Chapters 1 thru 11.* University of California.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N. F., Peters, M., Schmitz, M., and Zettlemoyer, L. S. (2017). Allennlp: A deep semantic natural language processing platform.

Garrod, S. and Terras, M. (2000). The contribution of lexical and situational knowledge to resolving discourse roles: Bonding and resolution. *Journal of memory and language*, 42(4):526–544.

Gerber, M. and Chai, J. (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.

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

Goldberg, A. (1992). The inherent semantics of argument structure: The case of the english ditransitive construction. *Cognitive Linguistics*, 3(1):37–74.

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

Granroth-Wilding, M. and Clark, S. (2016). What happens next? event prediction using a compositional neural network model. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2727–2733. AAAI Press.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Grimshaw, J. B. (1993). *Semantic structure and semantic content in lexical representation*. Rutgers University.

Hershcovich, D., Abend, O., and Rappoport, A. (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.

Hershcovich, D., Abend, O., and Rappoport, A. (2018). Multitask parsing across semantic representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 373–385, Melbourne, Australia. Association for Computational Linguistics.

Hershcovich, D., Abend, O., and Rappoport, A. (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.

Hershcovich, D., Choshen, L., Sulem, E., Aizenbud, Z., Rappoport, A., and Abend, O. (2019b). SemEval 2019 task 1: Cross-lingual semantic parsing with UCCA. In *Proc. of SemEval*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Ingram, D. (1971). Toward a theory of person deixis. *Research on Language & Social Interaction*, 4(1):37–53.

Jackendoff, R. (1992). *Semantic structures*, volume 18. MIT press.

Jackendoff, R. (1997). Twistin'the night away. *Language*, pages 534–559.

Jiang, W., Li, Z., Zhang, Y., and Zhang, M. (2019). HLT@SUDA at SemEval-2019 task 1: UCCA graph parsing as constituent tree parsing. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 11–15, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Kay, P. (2004). Null instantiation of nominal complements. *Unpublished ms., University of California, Department of Linguistics, Berkeley.*

Kiperwasser, E. and Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

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

Lyngfelt, B. (2012). Re-thinking fni: On null instantiation and control in construction grammar. *Constructions and frames*, 4(1):1–23.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (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.

Mittwoch, A. (1971). Idioms and unspecified np deletion. *Linguistic Inquiry*, 2(2):255–259.

Miyao, Y., Ninomiya, T., and Tsujii, J. (2004). Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *International Conference on Natural Language Processing*, pages 684–693. Springer.

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

Nivre, J., Abrams, M., Agić, Ž., Ahrenberg, L., Aleksandravičiūtė, G., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Basmov, V., Bauer, J., Bellato, S., Bengoetxea, K., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Bielinskienė, A., Blokland, R., Bobicev, V., Boizou, L., Borges Völker, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Brokaitė, K., Burchardt, A., Candito, M., Caron, B., Caron, G., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., Diaz de Ilarraza, A., Dickerson, C., Dione, B., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eckhoff, H., Eli, M., Elkahky, A., Ephrem, B., Erjavec, T., Etienne, A., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Fujita, K., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., González Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Heinecke, J., Hennig, F., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ikeda, T., Ion, R., Irimia, E., Ishola, Ọ., Jelínek, T., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kaasen, A., Kahane, S., Kanayama, H., Kanerva, J., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Köhn, A., Kopacewicz, K., Kotsyba, N., Kovalevskaitė, J., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lam, L., Lando, T., Larasati, S. D., Lavrentiev, A., Lee, J., Lê Hồng, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Li, Y., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., McGuinness, S., Mendonça, G., Miekka, N., Misirpashayeva, M., Missilä, A., Mititelu, C., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, K. S., Morioka, T., Mori, S., Moro, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikaido, Y., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peljak-Łapińska, A., Peng, S., Perez, C.-A., Perrier, G., Petrova, D., Petrov, S., Piitulainen, J., Pirinen, T. A., Pitler, E., Plank, B., Poibeau, T., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Rießler, M., Rimkutė, E., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O., Rueter, J., Sadde, S., Sagot, B., Saleh, S., Salomoni, A., Samardžić, T., Samson, S., Sanguinetti, M., Särg, D., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shirasu, H., Shohibussirri, M., Sichinava, D., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Spadine, C., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Suzuki, S., Szántó, Z., Taji, D., Takahashi, Y., Tamburini, F., Tanaka, T., Tellier, I., Thomas, G., Torga, L., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Wallin, L., Walsh, A., Wang, J. X., Washington, J. N., Wendt, M., Williams, S., Wirén, M., Wittern, C., Woldemariam, T., Wong, T.-s., Wróblewska, A., Yako, M., Yamazaki, N., Yan, C., Yasuoka, K., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zeman, D., Zhang, M., and Zhu, H. (2019). Universal dependencies 2.4. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Oepen, S., Abend, O., Hajic, J., Hershcovich, D., Kuhlmann, M., O'Gorman, T., Xue, N., Chun, J., Straka, M., and Uresova, Z. (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.

Oepen, S., Kuhlmann, M., Miyao, Y., Zeman, D., Cinková, S., Flickinger, D., Hajič, J., Ivanova, A., and Urešová, Z. (2016). Towards comparability of linguistic graph Banks for semantic parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3991–3995, Portorož, Slovenia. European Language Resources Association (ELRA).

O'Gorman, T., Regan, M., Griffitt, K., Hermjakob, U., Knight, K., and Palmer, M. (2018). AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

O'Gorman, T. J. (2019). *Bringing Together Computational and Linguistic Models of Implicit Role Interpretation*. PhD thesis, University of Colorado at Boulder.

Perlmutter, D. M. (1968). *Deep and surface structure constraints in syntax.* PhD thesis, Massachusetts Institute of Technology.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Prange, J., Schneider, N., and Abend, O. (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.

Pustejovsky, J., Lai, K., and Xue, N. (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.

Pérez-Leroux, A. T., Pirvulescu, M., and Roberge, Y. (2018). *Direct Objects and Language Acquisition.* Cambridge Studies in Linguistics. Cambridge University Press.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Riezler, S., King, T. H., Kaplan, R. M., Crouch, R., Maxwell III, J. T., and Johnson, M. (2002). Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 271–278, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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

Ruppenhofer, J., Ellsworth, M., Schwarzer-Petruck, M., Johnson, C. R., and Scheffczyk, J. (2006). Framenet ii: Extended theory and practice.

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

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., and Palmer, M. (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.

Sagae, K. and Lavie, A. (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.

Schenk, N. and Chiarcos, C. (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.

Schneider, N., Hwang, J. D., Srikumar, V., Prange, J., Blodgett, A., Moeller, S. R., Stern, A., Bitan, A., and Abend, O. (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.

Shalev, A., Hwang, J. D., Schneider, N., Srikumar, V., Abend, O., and Rappoport, A. (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.

Silberer, C. and Frank, A. (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.

Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Stern, A. and Dagan, I. (2014). Recognizing implied predicate-argument relationships in textual inference. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 739–744, Baltimore, Maryland. Association for Computational Linguistics.

Taylor, W. L. (1953). “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Yamada, H. and Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France.

Žabokrtskỳ, Z., Zeman, D., and Ševčíková, M. (2020). Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics*, pages 1–61.

Zhang, S., Ma, X., Duh, K., and Van Durme, B. (2019). Broad-coverage semantic parsing as transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.