SAARLAND UNIVERSITY
Faculty of Philosophy
Department of Language Science and Technology
MSc Language and Communication Technologies
MASTER THESIS

Compositional Discourse Representation Structure Parsing

Xiulin Yang

Advisors:
Prof. Dr. Alexander Koller (Saarland University)
Prof. Dr. Johan Bos (University of Groningen)

Abstract

Sequence-to-sequence (seq2seq) models have shown impressive performance in semantic parsing tasks. However, they exhibit certain limitations. They require substantial data to generalize well and often underperform when presented with Out-Of-Distribution data, for example, longer input sequences. Furthermore, the outcomes generated by these models can be challenging to interpret due to their end-to-end training scheme.

To address the problems mentioned above, we follow the Principle of Compositionality and aim to employ an algebra-based compositional approach, namely, AM (*Apply Modify*)-Algebra, to parse Discourse Representation Structure (DRS) compositionally. AM-Algebra is a linguistically motivated method that takes meaning representation graphs as tree representations of the compositional subgraphs. It assigns each meaningful token a lexical subgraph and then combines them back by building a dependency tree. It works well in simpler meaning representations like Abstract Meaning Representations, but it struggles to parse more expressive meaning representations like DRSs. Specifically, it lacks specific rules to process the reentrancies introduced by non-compositional information such as scope and coreference, which makes DRGs non-parsable by AM-Algebra. Against this backdrop, the core objective of this thesis is to navigate the intricacies of DRS parsing using AM-Algebra. We ask two questions: Firstly, how can we modify the graph formats to render DRS decomposable? Secondly, how do we effectively reintegrate the non-compositional information that is lost in the process?

To apply AM-Algebra to DRGs, we simplify DRGs into two forms: simplified DRG and scopeless DRG, achieved by assuming implicit box membership inheritance between the children nodes and the parent nodes. The simplification allows over 90% of data to be decomposable. To tackle the non-compositional aspects of DRGs, namely, anaphora and scope, we treat anaphora resolution as a lexical category tagging task and scope assignment as a dependency parsing task. In the anaphora task, we train AM-Parser to predict if a node is an antecedent/anaphor and we connect the two nodes with an `ANA` edge in the postprocessing steps. In the scope task, we leverage the accurate dependency parses from a simple biaffine dependency parser and node-token alignment generated by AM-Parser and reintroduce scope edges to scopeless DRG parses.

We evaluate our system on Parallel Meaning Bank Releases 4.0.0 and 5.0.0. The system has demonstrated impressive performance, often on par with or surpassing seq2seq models trained exclusively with gold data and even sometimes those trained on larger datasets. It also yields competitive results in anaphora resolution, scope assignment, and reentrancy structure parsing tasks compared with strong baselines. Notably, our method excels in processing longer sentences, surpassing even the fine-tuend Pretrained Language Model based on mBART with gold, silver, and bronze data. Our evaluations also underscore the strength of compositional models in processing complex structures and longer sentences.

Contents

List of Figures

List of Tables

Acronyms

AMR	Abstract Meaning Representation
DAG	Directed Acyclic Graphs
DRG	Discourse Representation Graphs
DRT	Discourse Representation Theory
DRS	Discourse Representation Structure
EDS	Elementary Dependency Structures
FFN	Feedforward Networks
LLM	Large Language Model
LSTM	Long-short Term Memory
MWE	Multiword Expression
NLP	Natural Language Processing
NLU	Natural Language Understanding
OOD	Out-Of-Distribution
PMB	Parallel Meaning Bank
POS	Part Of Speech
SBN	Simplified Box Notation
Seq2seq	Sequence-to-sequence
UCCA	Universal Conceptual Cognitive Annotation

Declaration

I hereby declare that the work presented in my thesis is entirely my own. All sources of information and data employed in this document have been duly acknowledged or cited. I understand the consequences of plagiarism and I confirm that the electronic version of my thesis is identical in content to the printed version. This thesis has not been submitted in the same or substantially similar version, not even in part, to any other authority for grading and has not been published elsewhere. I take full responsibility for the content and conclusions drawn in this document.

Signature

Acknowledgements

I would like to express my heartfelt gratitude to everyone who played a pivotal role in the completion of my thesis.

First and foremost, I want to extend my deepest appreciation to my thesis advisors, Professor Alexander Koller and Professor Johan Bos. Their unwavering support, invaluable insights, and adept guidance were instrumental in shaping this thesis. Whenever I encountered obstacles and felt unsure about how to proceed, they consistently provided helpful and inspiring suggestions that pinpointed the core problems. Their feedback often left me wondering, "How did I not think of that?" I'm truly grateful for their guidance throughout this journey.

I also want to extend a special thanks to Dr. Jonas Groschwitz for always being there to help. Even with his busy schedule, he never hesitated to carve out time for our discussions and debugging. Beyond his technical insights, his words of encouragement, detailed responses to my numerous questions, and his commitment to my project really helped me to enhance the quality of my work.

I am deeply thankful to Dr. Chunliu Wang and Dr. Xiao Zhang from Professor Bos' team for their responsiveness and willingness to answer all my questions pertaining to DRG and their research papers. My special appreciation goes to Dr. Xiao Zhang for generously providing the predicted output from the baseline models. I would also like to express my gratitude to Wessel Poelmann for his patient responses to my inquiries regarding UD-Boxer.

Additionally, I would also like to thank Ezio for answering a bunch of questions when I was learning Java. He is always very patient and uses examples to help me understand how Java works.

Last but not least, I am grateful to my family and friends, Jian, Devika, Amrita, Jingyan, and many other people, for their unconditional support, encouragement, and understanding during the challenges and demands of my academic journey.

Once again, thank you all very much!

Chapter 1

Introduction

1.1 Semantic Parsing and Discourse Representation Theory (DRT)

As one of the central tasks of Natural Language Understanding (NLU), semantic parsing aims to develop models that translate natural utterances to specific formal meaning representations (Kamath and Das, 2018). It can be applied to many downstream NLP tasks such as text generation (Liu et al., 2021; Ghazarian et al., 2022; Wang et al., 2023), machine translation (Song et al., 2019), and event extraction (Schuster et al., 2017), among others. Although Large Language Models (LLMs), without meaning representation as the intermediate step, can still achieve impressive performance in a variety of benchmark tasks (e.g., Karpinska and Iyyer, 2023; Agrawal et al., 2022), even the most advanced LLMs like ChatGPT and GPT-4, still suffer from Out-Of-Distribution (OOD) problems in reasoning (Wang et al., 2023; Zhang et al., 2023), which encourages the combination of explicit symbolic representation and latent one for more robust performance in NLU. To achieve this goal, various semantic formalisms were proposed, including but not limited to Abstract Meaning Representation (AMR; Banarescu et al., 2013), Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport, 2013), and Discourse Representation Theory (DRT; Kamp and Reyle, 2013).

In this work, we choose DRT as our research focus. DRT is a well-developed framework that cannot only model single sentences but also paragraphs and documents. It has established itself as a well-documented formal theory of meaning, covering a number of semantic phenomena (Liu et al., 2021, 2018) , ranging from pronouns, abstract anaphora

(Asher, 1993; Van der Sandt, 1992), presupposition, tense and aspect (Kamp et al., 1993), to rhetorical structures (Asher and Lascarides, 2003). Therefore, it can be naturally applied to longer text-meaning representations and more complex downstream tasks. Additionally, as DRS can be translated to first-order logic (Bos, 2008), it opens the possibility for automatic forms of inference by third parties (van Noord et al., 2020; Blackburn and Bos, 2005).

In DRT, sentences are represented by Discourse Representation Structures (DRSs) in the format of boxes for readability. DRS parsing has received more scholarly attention since the success of the first shared task (Abzianidze et al., 2019). The prevalence of sequence-to-sequence (seq2seq) models motivated researchers to convert DRSs into various sequential representations, such as character-level sequence (Liu et al., 2019), or the clausal format (Van Noord et al., 2018). Recently, Bos (2023, 2021) proposed a new DRS variant that can convert the traditional box format of DRS to simpler variable-free sequences, known as Simplified Box Notation (SBN). The sequence can be converted to graphs which are called Discourse Representation Graphs (DRGs)[1].

In terms of structure format, DRG bears resemblance to other semantic graphs, notably AMR. What sets DRG apart, however, is its inclusion of scope to represent quantification, negation, and various logical operations. The reentrancies and new nodes introduced by scope render DRGs more challenging to learn compared to AMR. Given this context, this thesis will delve deeper into the complexities of DRG parsing.

1.2 Research Questions

Seq2seq models are good at handling natural language variations. However, they encounter difficulties when addressing unseen structures (Yao and Koller, 2022). By contrast, the compositional parsers achieve more robust performance in compositional generalization (Yao and Koller, 2022; Weißenhorn et al., 2022; Shaw et al., 2021). Additionally, decomposing the complex graphs into simpler ones can help researchers interpret the decisions made by the model, thus contributing to the explainability and interpretability of the model. This is an important property that is missing in current main-stream LLMs (Tedeschi et al., 2023).

Compositional models, following the Principle of Compositionality, are good at generalization and out-of-distribution (OOD) texts, but it is obvious that the meaning of natural language is not ideally arithmetically compositional just like $1 + 1 = 2$. Aside from the most outstanding challenge in Multiword Expression (MWE), other linguistic phenomena such as anaphora and ellipsis also pose a challenge to compositional semantic parsing. Additionally, strict grammar-based compositional approaches might fail in

[1] Unless stated otherwise, the terms SBN, DRG and DRS are used interchangeably in this work, given that SBN and DRG are just variations of DRS.

broad-coverage parsing (Donatelli and Koller, 2023).

Hence, in our research, we seek methodologies that serve two key goals: **predicting DRGs for input strings compositionally** and **effectively processing non-compositional information**. As DRG shares characteristics with graph meaning representations (Bos, 2021), it opens the possibility to try graph-based semantic parsers for DRS parsing. We build our system based on AM(Apply-Modify)-algebra by Groschwitz et al. (2018, 2017) because of the flexibility of algebra-based approaches in meaning presentation combination (Donatelli and Koller, 2023). AM Algebra was originally designed to parse AMR. It considers the AMR graph as a dependency tree with leaves being a set of atomic graphs. As a result, the parsing task is divided into two subtasks: supertagging which assigns appropriate atomic graphs to individual tokens, and dependency parsing which looks for the dependency relations between those atomic graphs. Apart from AMR, AM-parser has been applied to various graph banks, for example, Elementary Dependency Structures (EDS; Oepen and Lønning, 2006), UCCA (Abend and Rappoport, 2013), and DELPH-IN MRS Bi-Lexical Dependencies (Ivanova et al., 2012), among others (Donatelli et al., 2020; Lindemann et al., 2019).

However, to parse DRGs compositionally utilizing AM-Algebra, two main obstacles emerge. First, the inherent complexity of the complete DRGs poses significant challenges to its effective decomposition via AM-Algebra. The introduction of reentrancies and nodes due to scope and discourse information renders the DRG difficult to decompose, if not entirely non-decomposable, with AM-Algebra. The second challenge stems from the fact that AM-Algebra focuses on the compositional aspects of meaning representation[2], encouraging a static representation for each token. Contrastingly, certain facets of meaning within DRT, such as coreference (Janssen and Partee, 1997) and scope, exhibit dynamic characteristics. These features are very important to DRT itself. Therefore, to further refine our research focus, we rephrase our key subquestions as follows:

- *Q1: Regarding compositionality, how can DRGs be simplified to achieve compositionality by AM-Algebra while retaining linguistic knowledge and preserving essential structural information?*
- *Q2: Regarding non-compositionality, how to recover the non-decomposable information, more specifically, anaphora and scope assignment?*

A compositional model is typically trained to acquire the alignment between input tokens and their corresponding meaning representations as well as the structural relationship between subgraphs, thus learning the underlying structures more efficiently. We therefore expect our model to exhibit superior performance, particularly under conditions of data scarcity. This advantage assumes greater significance when we are confronted with limited training data. Hence, in terms of evaluation, we want to examine:

[2] AM-Parser, the model developed based on AM-Algebra, can conduct the sense disambiguation task, which is non-compositional according to Bender et al. (2015).

- *Q3: How effectively does a compositional approach perform in comparison to its non-compositional neural or symbolic counterparts?*

The data we use in this thesis is the Parallel Meaning Bank (PMB; Abzianidze et al., 2017). It comprises gold, silver, and bronze data, with the gold subset being the smallest and the bronze subset being the largest in terms of size. In our research, we intend to explore the generalization capabilities of our compositional model when exclusively exposed to the limited gold data.

Given that DRT excels in modeling dynamic semantics, particularly in scope and anaphora resolution, our evaluation extends beyond overall performance. Specifically, we seek to assess how our system, in comparison to non-compositional models, performs in coreference resolution, scope assignment, and reentrancies.

1.3 Contributions

In this work, we introduce two novel formats, Simplified-DRG and Scopeless-DRG, which retain core information while enabling straightforward scope information recovery using simple heuristics. These formats not only simplify the parsing task, enhancing efficiency and reducing complexity, but also offer practical utility for future research.

Additionally, we leverage the advances of dependency parsing to propose two effective approaches to resolve coreference and scope assignment. These methods facilitate the mapping of non-decomposable parts back to scopeless DRGs. Our methods demonstrate competitive performance compared to strong baseline models trained exclusively on gold data.

Furthermore, our empirical findings advocate for compositional parsing over seq2seq models under limited training data conditions. Specifically, we highlight that compositional models exhibit more robust performance in generating legal and correct graphs, especially when the graphs exhibit increased complexity in both size and structure.

In summary, our contributions are expected to advance the field of compositional semantic parsing by introducing new graph formats, tackling non-composability challenges in which other neural models fail, and providing new empirical evidence for the advantages of compositional semantic parsing.

1.4 Structure of the Thesis

This thesis comprises seven chapters, beginning with Chapters 1 and 2, which introduce the research questions and background information. Chapter 3 critically reviews prior research in DRS parsing, highlighting the advantages and challenges of the existing approaches. Chapter 4 outlines the datasets and evaluation metrics used. Chapter 5 details our approach to handling compositional and non-compositional elements in DRGs. Chapter 6 reports experimental results and conducts an error analysis. Finally, Chapter 7 offers conclusions, answers research questions, and suggests future research directions. The code of the thesis is available at `https://github.com/xiulinyang/compositional_drs_parsing`.

Chapter 2

Background

In this chapter, we first introduce the principle of Compositionality, Discourse Representation Theory, and its three meaning representation variations: Discourse Representation Structure (DRS), Simplified Box Notation (SBN), and Discourse Representation Graph (DRG). These three formats can be converted to each other freely without losing any information. After that, we explain how AM-Algebra can be applied to DRG parsing and the challenges it faces.

2.1 Compositional Semantics

The principle of Compositionality holds that the meaning of a complex expression is determined by the meaning of the unit and the way that they are combined (Heim and Kratzer, 1998). It has been receiving increasing scholarly attention in NLP as more studies indicate that neural networks fail in compositional generalization tasks (e.g., Hupkes et al., 2020; Yao and Koller, 2022; Dankers et al., 2022). However, the vague expression of the principle gives a lot of space for interpretation, and the principle itself remains controversial[3].

Proving whether natural language is compositional is beyond the scope of our work. In this thesis, we adopt Bender et al. (2015)'s definition of compositionality. They assume that the compositionality of a meaning representation system should possess the following properties:

- It contains a finite number of arbitrary atomic symbol-meaning pairings;

[3] Please refer to (Pagin and Westerståhl, 2010, 2019) for a detailed comparison of different arguments.

- It can generate an infinite number of symbol-meaning pairings through a finite set of rules;
- The meaning of any non-atomic must be obtained by a function that includes it and the way it is combined;
- This function can tackle special cases but only relies on the immediate constituents and combined rules;
- and further processing does not disrupt the originally resulting symbol-meaning pairings.

According to this definition, we might conclude that the compositional part of meaning should be grammar-derived. The principle of Compositionality thus prefers a static notion of meanings even though dynamic aspects can be handled with abstract methods (Janssen and Partee, 1997). In this case, word-sense tagging, scope assignment, and anaphora resolution should be considered as non-compositional, because they all vary from context to context. Such variability fails to meet the first condition.

2.2 Discourse Representation Theory

Discourse Representation Theory (DRT; Kamp, 2013; Kamp et al., 2010; Geurts et al., 2020) is a formal semantic framework that aims to interpret meanings from the context. It can model anaphora (Kamp, 1981; Haug, 2014), tense (Kamp, 1981), and rhetorical structures (Lascarides and Asher, 2007), among others. DRT has received multiple extensions. In this paper, we refer to DRT as the theory presented in Parallel Meaning Bank (PMB; Abzianidze et al., 2017) and follow the notational convention from the same source. More specifically, apart from the standard format shown in Kamp and Reyle (2013), DRS in PMB also incorporates the neo-Davidsonian annotation with role inventories from VerbNet (Kipper et al., 2008). Word senses are expressed as WordNet synset identifiers in Princeton's American English WordNet 3.0 (Fellbaum, 1998).

Within DRT, the meaning is represented by Discourse Representation Structure (DRS), which is designed to model the evolving context of discourse and capture the implicit connections between different expressions in the text. They consist of two main components: a set of discourse referents representing the objects or events introduced in the discourse, and a set of conditions specifying the relationships and properties of these discourse referents. The referent can be either an entity or an event. Conditions can be a *Concept* (i.e., a one-place predicate), *Relation/Role* (i.e., a two-place predicate), or another DRS. The main ways in which DRT distinguishes itself from many other mainstream meaning representation frameworks are that (1) DRSs are not represented in graphs (Abzianidze et al., 2020); (2) DRSs are distant from the syntactic structures of their corresponding sequence strings (Žabokrtský et al., 2020; Abzianidze et al., 2020).

An example is given below. This DRS represents the meaning of the sentence *If Jones owns a donkey, he likes it.*

$$\begin{array}{|c|}\hline x \\ \hline \text{Jones(x)} \\ \begin{array}{|c|}\hline e_1\ y\ t_1 \\ \hline \text{donkey(y)} \\ \text{own}(e_1) \\ \text{Agent}(e_1, \text{x}) \\ \text{Theme}(e_1, \text{y}) \\ \text{Time}(e_1, t_1) \\ t_1 = \text{now} \\ \hline \end{array} \Rightarrow \begin{array}{|c|}\hline z\ t_2\ e_2\ w \\ \hline \text{z = x} \\ \text{w = y} \\ \text{like}(e_2) \\ \text{Experiencer}(e_2, z) \\ \text{Stimulus}(e_2, w) \\ t_2{=}\text{now} \\ \hline \end{array} \\ \hline \end{array}$$

Figure 2.1: The DRS with neo-Davidsonian style for the sentence *If Jones owns a donkey, he likes it.*

2.3 Simplified Box Notation

Given that many popular meaning representation frameworks, ranging from AMR to UCCA, are designed in the format of graphs, it encourages DRSs to be converted to graphs, or as referred to by Abzianidze et al. (2020) as Discourse Representation Graphs (DRGs), for cross-framework meaning representation parsing. Abzianidze et al. (2020) reviewed the existing DRG-encoding approaches (Power, 1999; Basile and Bos, 2013; Liu et al., 2018) and compared potential DRG formats resulting from the combination of four possible options for DRS-to-DRG conversion. The graph format they finally suggest in the shared task is still very complex. Fancellu et al. (2019) transformed DRSs into acyclic, single-rooted and fully-instantiated graphs. Bos (2023) later proposed a simpler linearized notion, Simplified Box Notation (SBN), which can be represented as Directed Acyclic Graphs (DAG).

An SBN is composed of a set of Concepts, Boxes, and Constants which are connected by Roles, Operators, and Separators as the edge name[4]. The connection between two nodes is encoded by the indices. An SBN has the following ingredients.

- **Concepts** refers to the node names in the graph. They are composed of three parts: lemma, word category and sense number in Wordnet (e.g., `cat.n.01`, `see.v.03`).
- **Constants** are usually numbers, names, dates, etc. (e.g., `"Mary"`, `speaker`, `20`). They are always the terminal leaves in DRGs.
- **Roles** are semantic roles that connect two concepts (e.g, `Agent`, `Theme`, `Patient`).

[4] Only the components of SBNs in the PMB release 4.0.0 are introduced for now

- **Operators** usually connect one concept and one constant, but sometimes it connects two concepts to model co-index. They indicate a non-role relation between two nodes (e.g., `EQU`, `APX`, `TIN`).
- **Indices** always follow roles or separators in SBN to connect nodes/boxes to indicate the location of the target node or box (e.g., `-2`, `-1`,`+1`, `+2`, `<2`, `>1`).
- **Boxes** are always introduced by separators. They are used to model context.
- **Separators** are discourse connectives which assign scope (context/box) to the nodes (E.g., `NEGATION`, `EXPLANATION`, `NARRATION`, . . .).

SBN supports the assumption that meaning should be interpreted in a specific context. To model context, it inherits the box notation from DRT. A separator introduces a new context and its indices encode which context the new context should be connected with.

To free DRS from variables, Bos (2023) uses bidirectional deBruijn-indices to index the arguments of the conditions. To model the scope but maintain the flat and simple structure of SBN, Bos (2023) introduces a set of *seperators* and *indices* to divide the conditions into different discourse units. Note that SBN employs logical equivalence to avoid using logical constructs such as the universal/existential quantifier and implication statement. The converted SBN from figure 2.1 is shown below where SBN is put on the left and its DRG is on the right. Here, $(p \rightarrow q \leftrightarrow \neg(p \wedge \neg q))$ is used to generate the resulting format. The indices `-1` in `own Agent -1 Theme +2 Time +1` means that the agent role of the verb `own` refers to the node two steps backward (separators do not count), namely, `male`. Similarly, the experiencer of the verb `like` refers to the same node.

male	Name "Jones"
	NEGATION -1
own	Agent -1 Theme +2 Time +1
time	TPR now
donkey	
	NEGATION -1
like	Experiencer -4 Stimulus -1 Time +1
time	TPR now
entity	= -2

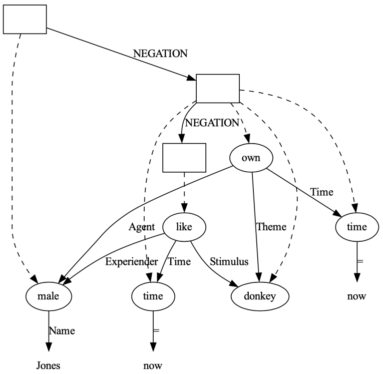


Figure 2.2: The SBN and its DRG for the sentence *If Jones owns a donkey, he likes it* without sense disambiguation

2.4 Discourse Representation Graph

As we mentioned previously, the sequential SBN format can be seamlessly converted to a DAG, coined as DRG. Consequently, just as with AMR, the DRG can be represented using the Penman notation (Matthiessen and Bateman, 1991; Kasper, 1989). The Penman notation serves as a serialization format designed for encoding DAGs. In this notation, each node in the graph is assigned a unique variable (for example, s1) which represents the node label, such as donkey. Nodes are interconnected by relations, which are denoted by edge labels, like Stimulus. In Penman notation, the / symbol specifically indicates an instance relation. The reentrancies can be freely expressed by relations between two node variables. The converted Penman for Figure 2.2 is shown below. All the dashed scope lines in the graph are instantiated with the member relation.

```
(b0 / box
:NEGATION (b1 / box
    :NEGATION (b2 / box
        :member (s0 / like
            :Stimulus (s1 / donkey
            :Experiender (s2 / male
            :Time (s3 / time
               :EQU now2))
    :member (s4 / own
        :Agent s2
        :Theme s1
        :Time (s4 / time
            :EQU now))
    :memeber s1
    :memeber s3
    :memeber s4)
:memeber s2)
```

Figure 2.3: Penman notation for the sentence *If Jones owns a donkey, he likes it.*

The converted DRGs combine the advantages of AMR and DRS. For one thing, the DRG, going beyond the meaning of predicate-argument structure, is expressive in modeling diverse semantic phenomena; for another, its graph format makes it easier to annotate and parse (Bos, 2023).

2.5 SBN parsing with the AM-Algebra

AM-Algebra (Groschwitz et al., 2017, 2018) is an algebra-based method for compositional semantic parsing. It learns the meaning representations of each token and combines each of them to form a complete meaning representation graph.

As-graph, root, and source AM-Algebra aims to build graphs from a set of elementary lexical graphs. It assumes that each meaningful token should have a separate meaning representation in the format of graphs, namely, *annotated s-graph* or *as-graph*. They are directed graphs that contain type information for graph construction. The type information is encoded by their sources (Courcelle and Engelfriet, 2012) as the blue texts shown in Fig 2.4. The reference `p00/d3046` in the caption denotes the file ID as found in PMB Explorer[5]. Interested readers are encouraged to read details of the semantic annotation via this platform. How to interpret the source names will be explained later. Each as-graph also has a designated root node indicated in **bold**.

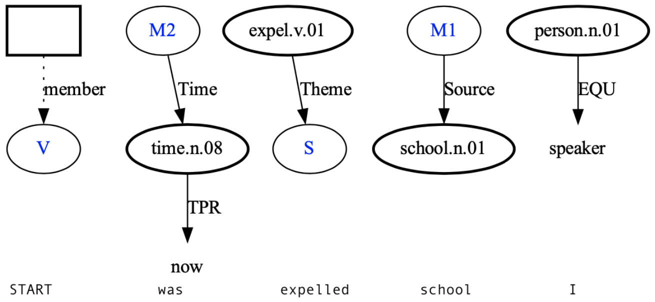


Figure 2.4: As-graphs for the sentence *START I was expelled from school.* p00/d3046

Apply and Modify AM-algebra, following the syntax rule of constituent combination, has two main operations: *Apply* and *Modify*. The former refers to the process where the as-graph of the syntactic complement is **applied** to the root node of the head as-graph, while the latter means that the as-graph as the adjunct **modifies** the root node of the head as-graph. The application rule is similar to a complement combining with its head - this combination is required by the head; the modification rule is similar to an adjunct combining with its head - this combination is not required by the head and, therefore, the type of the head is not influenced. They are explained below separately.

The *Modify* operation (MOD) works similarly to how adjunct is combined with its head, like X/X category in CCG; in other words, MOD does not change the type of original as-graph of the head nor require specific source types, but instead, the root node of the head as-graph G_{Head} is plugged into the M source of the adjunct as-graph G_{Mod}. As a result, the source in G_{Mod} is removed and the root node of the new G_{Head} remains the root. For example, we can combine G_{school} and G_{expel} with $MOD_{M1}(G_{expel}, G_{school})$ in figure 2.4. The root node of G_{expel} inserts in the M node, and it generates the graph in figure 2.5 (left). The root of the resulting graph remains `expel.v.01` and thus the graph name is still G_{expel}.

The *Apply* operation (APP) resembles how the dependent is combined with its head, like

[5] `https://pmb.let.rug.nl/explorer/explore.php`

forward/backward application in CCG. APP$_X$ represents the *Apply* operation for a source X. APP$_X$ between two as-graphs $Graph_{Predicate}$ and $Graph_{Argument}$, i.e., $APP_X(Graph_P,$ $Graph_A)$, requires the root node of $Graph_A$ to be inserted to the annotated source of $Graph_P$. As shown in 2.5, the newly generated G_{expel} after $MOD_{M1}(G_{expel}, G_{school})$ is applied with G_I. The root node of G_I plugs in the S source and the resulting new subgraph G_{expel} is shown in the middle.

After we get the new subgraph G_{expel}, we can then utilise $MOD_{M2}(G_{expel}, G_{was})$ and $APP_V(G_{START}, G_{expel})$ to generate the final graph shown in Fig 2.5 on the right.

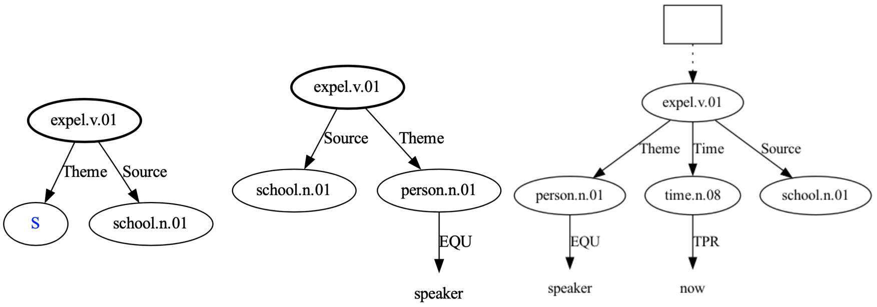


Figure 2.5: The result of $MOD_{M1}(G_{expel}, G_{school})$ (left), $APP_S(G_{expel}, G_I)$ (middle) and the complete SBN graph (right)

Types Whether two as-graphs can be combined via APP is determined by their types which are optionally represented in a bracket in the source name.

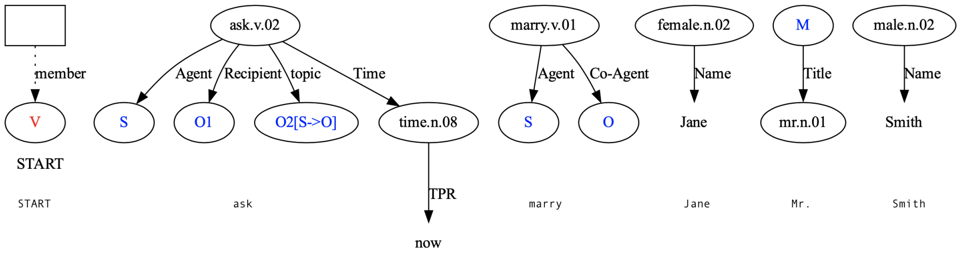


Figure 2.6: As-graphs for the sentence *START Mr. Smith asked Jane to marry him.* p24/d2048

For example, in the object control example in Fig 2.6, the `O2` source in the as-graph for *asked* G_{asked} has the type [S→O], meaning that the as-graph that undergoes the APP_{O2} should have the S source. Among all the as-graphs below, only G_{marry} suffices the condition. As a result, the root node of G_{marry} plugs in that source name. After APP_{O2}, the S source in G_{marry} should be renamed as O so that this O source can be *unified* or *merged* with the O source of G_{asked}. Therefore, the recipient of `ask.v.02` shares the same node as the Agent of *marry.v.01*. This type clearly explains how object control works. That is, the subject of the subordinate predicate should be the object of the main predicate.

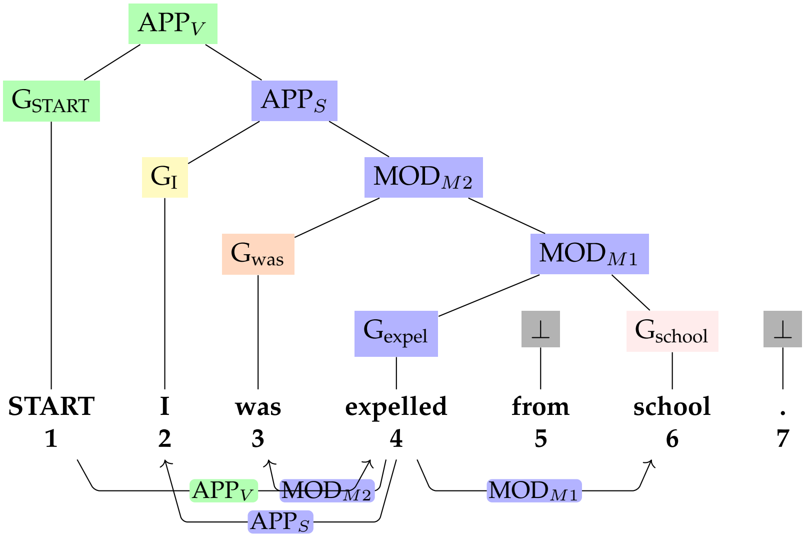


Figure 2.7: AM-term and its corresponding AM dependency tree for the sentence *I was expelled from school.* p00/d3046

Such annotations set restrictions to the combination possibilities between different as-graphs. For example, in the case of G_{asked}, only G_{marry} can be plugged into the O source of G_{asked}, because only G_{marry} has an O source. Other types can be found in Groschwitz et al. (2017); Groschwitz (2019).

AM-algebra provides very explicit operations to properly parse linguistic structures that introduce reentrancies, such as coordination, raising, control, relative clause, wh-movement, secondary predication, and parasitic gaps.

Indexed AM terms and AM dependency tree To connect the as-graphs and tokens of the input sequences, Groschwitz et al. (2018, 2017) proposed *indexed AM terms* which assume each as-graph represents the meaning of single individual tokens. The terms can be naturally converted to a dependency tree as shown in Fig 2.7. Some words such as preposition *from* do not contribute to the meaning of the meaning representation. They are assigned a $\perp$ sign and they do not receive any AM operation. The graph above the input tokens represents AM-terms. It illustrates the operations required to build a complete DRG from bottom up. The color of the operation represents the root node of the resulting subgraph. For example, after $MOD_{M1}(G_{expel}, G_{school})$, the root node remains `expel`. The operation rules between subgraphs can be equivalently converted to dependency relations between tokens that align with the corresponding subgraphs. All these dependency edges build an AM-dependency tree. It can then be evaluated as a complete meaning representation graph.

AM-Parser AM-Parser (Groschwitz et al., 2018) is a model built based on AM-Algebra. It reflects how AM-Algebra works. It contains a supertagger, a dependency parser, and a tree decoder. The supertagger is built to assign each token in the input sequence strings an elementary graph that represents the meaning of that individual token. The dependency parser is to build the optimal AM dependency tree that reveals the combination of these elementary as-graphs graphs. The symbolic decoder decodes the AM-dependency tree to a complete meaning representation graph.

2.6 The Challenges Brought by DRGs

AM-Algebra works very well with AMR which encodes less rich semantic information. However, it is very challenging (if not impossible) to parse a complete DRG because the non-compositional scope and coreference introduce a considerable number of reentrancies which AM-Algebra fails to handle.

Scope In DRG, each node is exclusively connected with one discourse box to model context information. However, the abundant reentrancies introduced by the scope information make SBN graphs *non-decomposable*.

Take a simple sentence *I was expelled from school* as an example. The as-graphs are shown in Fig 2.6. There are two paths to building the complete graph: (a) we connect the root box with other elementary graphs first and then build the rest; (b) we build the graph without the root box first and then we connect the root box with the rest. However, we will find out that neither way works. If we adopt (a), as shown in Fig 2.9, the `person.n.01` node cannot be combined with the `expel.v.01` node because it has been a part of a larger subgraph. Similarly, the `time.n.08` node and the `school.n.01` node cannot modify the verb `expel.v.01` because `expel.v.01` is not the root of this subgraph anymore and a modifier as-graph can only modify the root node which, in this case, is the root box. If we adopt the opposite composition approach, it is still impossible to apply G_{START} to the non-root node including `school.n.01`, `person.n.01` and `time.n.08` through the APP operation.

Coreference Coreference in DRGs introduces a new reentrancy edge to the antecedent node. As illustrated in Figure 4.1, the node `female.n.01` demonstrates this with two incoming edges. Similar to the failure caused by scope, the MOD and APP rules are insufficient to address reentrancies stemming from coreference, rendering AM-Algebra incapable of parsing coreference information within DRGs. This limitation underscores the need for improved methodologies in handling coreference resolution within the context of AM-Algebra and DRGs.

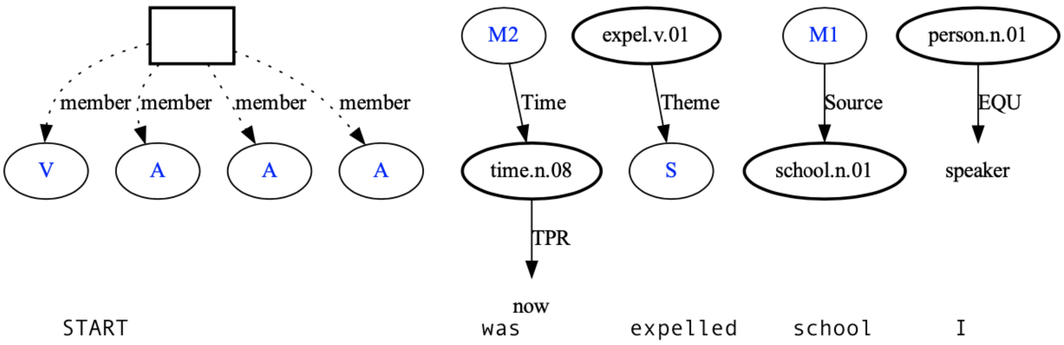


Figure 2.8: As-graphs (with scope) for the sentence *START I was expelled from school.* p00/d3046

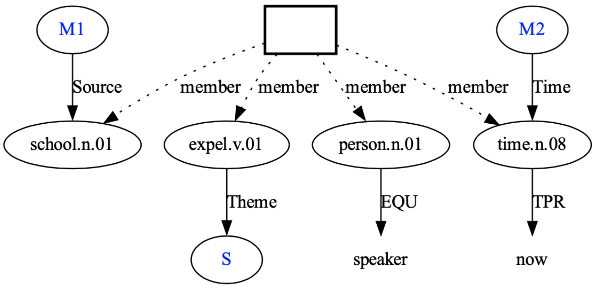


Figure 2.9: As-graphs (with scope) for the sentence *START I was expelled from school.* p00/d3046

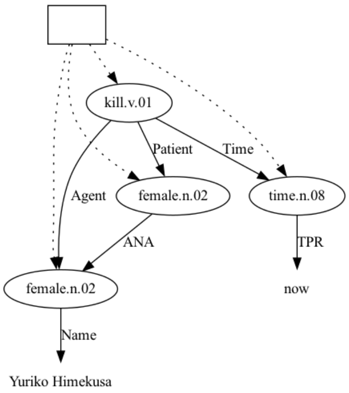
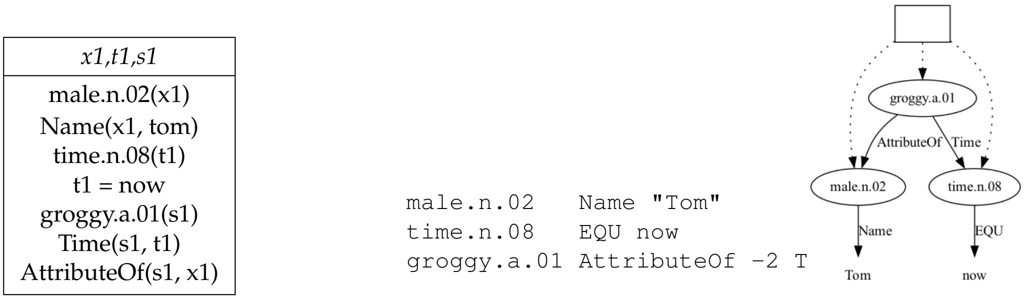


Figure 2.10: DRG for the sentence *Yuriko Himekusa killed herself.* p79/d2094

2.7 Summary

This chapter first explains the Principle of Compositionality. It then introduces two meaning representation variations for DRS, namely, the graph representation DRG and



x1,t1,s1

male.n.02(x1)
Name(x1, tom)
time.n.08(t1)
t1 = now
groggy.a.01(s1)
Time(s1, t1)
AttributeOf(s1, x1)

```
male.n.02   Name "Tom"
time.n.08   EQU now
groggy.a.01 AttributeOf -2 T
```

(a) DRS

(b) SBN

(c) DRG

Figure 2.11: Different variants of DRS *Tom's groggy.* p18/d2557

the sequential notation SBN. As shown in Figure 2.11, all three representations below express the same meaning. They can be converted to each other with ease.

We then explain how AM-Algebra combines the lexical graphs together to form a complete DRG with two simple rules, Apply and Modify. However, AM-Algebra lacks explicit rules to parse the scope and coreference information. This is also a major challenge of the thesis.

Chapter 3

Related Work

This chapter critically reviews previous research in DRS parsing. It is organized into three primary categories of focus: compositional parsing, symbolic parsing, and deep learning-based parsing. We summarize the advantages and challenges of each approach.

3.1 Compositional Approaches

Research in semantic parsing has long embraced the Principle of Compositionality. Compositional models are generally developed based on explicit grammar information (e.g., Combinatory Categorial Grammar (CCG) or Synchronous Grammars) or algebra terms (Donatelli and Koller, 2023).

Earlier traditional approaches rely on the assumption that syntactic trees provide prior knowledge for semantic composition. Consequently, they usually learn the underlying syntactic structure of the input and then construct meaning representation in a compositional manner (Van Noord and Bos, 2017).

The first data-driven compositional approach is proposed by Le and Zuidema (2012) in which they adopt a similar method to AM-Parser by converting DRS to semantic graphs and using a dependency structure to encode the relations between the elementary graphs. Different from AM-Algebra, they only leverage the direction of the dependency edges to encode the binding operation between the variable in the argument subgraph and the head subgraph. The graph they propose is also more complex than DRGs. Their parser is a probabilistic one.

In a similar vein, Boxer (Bos, 2008, 2015) is developed in a compositional way based on Combinatory Categorial Grammar (CCG; Steedman, 2001) and λ-calculus. The parser takes the CCG derivation of natural language expressions as input and generates DRSs. Additionally, it is a complex system, composed of a language-specific tokenizer, supertagger, semantic tagger, parser, and symbolizer. The complicated design of the system and its reliance on CCG makes it very hard to adapt to graph-related tasks.

These approaches usually combine certain linguistically principled heuristics or constraints with a statistical model. While the outputs generated by such systems are often more interpretable, they typically do not achieve performance metrics comparable to those of neural network-based approaches.

3.2 Rule-based Approaches

Some early work in DRS parsing also relies on rule-based systems (e.g., Johnson and Klein, 1986; Wada and Asher, 1986; Bos et al., 2001). The rule-based methodologies often encounter difficulties in handling diverse input variations, thereby posing challenges in achieving broad parsing coverage.

Recently, Poelman et al. (2022) proposed a new system named UD-Boxer. It consists of a set of rules to transform Universal Dependency trees into DRGs in four languages: English, Italian, German, and Dutch. The performance of the system relies on the dependency parser and the set of heuristics. Once the dependency parses are obtained, the DRG output is deterministic, which makes the result fully explainable.

Compared with the BERT-based Neural-Boxer (van Noord et al., 2020), the graph transformation system performs particularly better than the neural model in languages other than English and generates fewer ill-formed graphs[6]

The symbolic approach shows promising results in terms of compositionality in the syntax-semantics interface because the syntactic dependency tree input has to match the semantic meaning representation graph output. However, a syntactic tree is not informative in how the scope should be assigned to the meaning representation graphs. As a result, it does not give an explicit solution to scope and anaphora resolution. Furthermore, since the rules are specifically tailored for DRGs in PMB4, they cannot be directly applied to DRGs in PMB5 due to structural changes.

3.3 Deep Learning Approaches

The development of deep learning approaches, particularly seq2seq models, contributes to most of the recent improvements in parsing systems. Seq2seq models are one (domi-

[6] However, based on our experimental results, when we switch the evaluation format from lenient to strict (see Chapter 4.4), the performance of UD-Boxer declines more than that of Neural-Boxer. This confirms Wang et al. (2023)'s argument that the lenient format can inflate the result.

nant) type of neural model, usually containing an encoder and a decoder. The encoder encodes the representation of sequential input, while the decoder produces its corresponding meaning representation. They show impressive performance overall, but they also struggle with specific structures (Van Noord et al., 2018).

To date, the leading DRS parsers primarily utilize seq2seq models. Since it is very challenging for the neural model to generate boxes directly, researchers have explored various possible linearized representations of DRSs on top of the neural approaches. Some are graphs or trees, while some are sequences. We explain two types of models, i.e., structure-aware model and structure-unaware model, based on the target format. If a model takes the sequential input simply as a set of strings, we assume the models are structure-unaware.

Structure-unaware Models One of the earliest attempts to utilize seq2seq models for DRS parsing is by Van Noord et al. (2018) who convert the DRSs to a clausal format (for a detailed overview, see Figure 2 in (Van Noord et al., 2018)). Their optimized model employs a two-layer Bidirectional LSTM encoder-decoder setup, enhanced with global attention (Luong et al., 2015), and leverages character embedding as input.

Building on this foundation, van Noord et al. (2019) incorporated linguistic features into a multi-encoder model, significantly enhancing parsing performance. Their research underlined the pivotal role of character-level encoding and linguistic features, especially in scenarios with limited data.

Later, with the prevalence of large language models, van Noord et al. (2020) also experimented with different embeddings (e.g., `ELMO` (Peters et al., 2018), `BERT-base/large` (Devlin et al., 2019), and `ROBERTA-base/large` (Liu et al., 2019)) in combination with character embedding. They also experiment with various linguistic features, including lemma, POS tag, semantic tag (Abzianidze and Bos, 2017), dependency parses, and CCG supertags. The best model is a standard seq2seq model with attention, but they also add an extra linear layer before the initial decoder state and after each decoder state and initialize the decoder hidden state with the mean of all encoder states. In their experiments, they find that BERT-base embedding with character embedding yields the best performance. Additionally, adding linguistic features is not always beneficial.

The contemporary state-of-the-art model, the Multilingual Language Meaning framework for DRS (MLM-DRS) by Wang et al. (2023), stands out due to its multilingual capabilities. This model, rooted in mBART, was pre-trained using datasets from multiple languages with specific denoising strategies in order to force the model to learn the meaning representation structures. Then the pretrained model is fine-tuned for the parsing task, showing remarkable performance. Yet, as our results in Chapter 6 suggest, it struggles with lengthy sequence sentences.

Structure-aware Models A subset of studies have reimagined DRSs in forms of graphs or trees, mirroring the objectives of our research. One notable example is Liu et al. (2018), who viewed DRS as a tree structure. They develop a structure-aware model which takes DRS as a tree structure with conditions and referents being the leaf node. The encoder is a bi-LSTM network that takes the words as input. The decoder is a forward LSTM layer with attention mechanism. It decodes the DRS in three stages. It first predicts the DRS structure (i.e., scope information and discourse connectives) and then fills the structure with conditions and finally, it predicts the variable names based on the predicted conditions. Due to the novel format of DRS, the model is evaluated with a new metric named D-match[7] and thus it is not comparable with previous work. Their results show that a structure-aware decoder can generate accurate scope assignments (0.91 D-match F).

Fancellu et al. (2019) transformed the DRSs into DAGs which then can be linearized with PENMAN notation. The converted DAGs are largely similar to DRGs, except that the DAGs do not use logical equivalence to express logical operators (e.g., *implication* $\Longrightarrow$) by negations. In their seq2seq model, the encoder is a bi-LSTM encoding the input sequences and other linguistic features, and the decoder is composed of three models to model different actions. Their model shows competitive performance as Van Noord et al. (2018) with a lower error rate.

Seq2seq models have beaten the benchmarks, but their impressive performance largely benefits from large training datasets. It is very challenging for these neural models to generalize from limited data. They have to either rely on additional silver data or include structural features to perform well. This might explain why the structure-aware model by Fancellu et al. (2019) outperforms Van Noord et al. (2018) when only gold data is used for training and why the inclusion of silver data offers a performance boost to structure-unaware models.

That said, even with abundant training data at their disposal, structure-unaware models consistently exhibit a higher error rate compared to their counterparts. Poelman et al. (2022) reported that neural models, compared with grammar-based ones, are more likely to make mistakes. This is because DRS is essentially a graph with strict structure and binding constraints. For structure-unaware models that take the input and output simply as a sequence of strings, it is challenging for them to learn such implicit information. This is particularly true when these seq2seq models parse longer sentences.

Beyond these challenges, it is also important to note that even though the parsers can generate accurate results overall, they still face challenges when parsing specific linguistic phenomena such as scope ambiguity, coreference resolution, and discourse relations. Hence, a commendable SMATCH F score does not necessarily vouch for the model's

[7] This is a metric developed based on Smatch(Cai and Knight, 2013). Unfortunately, the reference link they provide is not valid anymore.

all-rounded robustness in parsing DRS construction. Furthermore, it is worth noting that current seq2seq models primarily concentrate on generating DRSs overall, with limited attention paid to specific linguistic nuances, such as anaphora. Given the importance of scope and anaphora in DRT theory, a targeted emphasis on these phenomena is highly recommended.

3.4 Summary

In summary, researchers have employed various strategies for the DRS parsing task. The existing compositional parsers and symbolic methods have struggled to match the competitive performance of seq2seq models. Consequently, a significant focus in the research community has shifted toward the development of more effective seq2seq parsers. New data formats are explored: DRSs have been transformed into sequential clauses, trees, and DAGs. Regarding embeddings, researchers have experimented with BERT embedding, character-level embedding, and other linguistic features. Different encoder and decoder architectures are proposed. However, seq2seq models do come with their limitations. Firstly, they are data hungry; secondly, for the structure-unaware models, they tend to generate more ill-formed graphs; thirdly, they still struggle with processing information that encodes implicit structures. Therefore, it is important to build a parser that can achieve high accuracy, while maintaining a low error rate.

Chapter 4

Data and Evaluation

In this Chapter, we present the statistics of the data we use, i.e., Parallel Meaning Bank Release 4.0.0 and 5.0.0. We then introduce the baselines we employ in our experiments, followed by the metrics, namely, the SMATCH and SMATCH++, in our work.

4.1 Data

We use the English data from Parallel Meaning Bank (PMB) release 4.0.0 and 5.0.0[8] (Abzianidze et al., 2017) for our experiments. PMB is a multilayer corpus containing rich annotation information including tokenization, semantic tagging, symbolization, lemmatization, and CCG tagging. It also includes the SBN annotation which will be used in our experiments.

4.1.1 Parallel Meaning Bank 4.0.0

The dataset is divided into three splits - Gold, Silver, and Bronze - based on the quality of annotation, with Gold being automatically annotated and manually corrected by experts and Bronze only being automatically annotated. The statistics of different splits are shown in Table 4.1.

The gold dataset contains four splits for English: train, development, evaluation, and test set.

[8] https://pmb.let.rug.nl/data.php

	# Docs	# Multi-sent	# Tokens	Tokens/ doc
Gold	10,715	79	59,444	5.5
Silver	127,303	5,581	1,256,045	9.9
Bronze	156,286	5,391	1,463,721	9.4

Table 4.1: The data distribution of English dataset in the Parallel Meaning Bank Release 4.0.0

4.1.2 Parallel Meaning Bank 5.0.0

Compared with PMB4, PMB5 removed the evaluation set. It reshuffled all the sentences, removed repetitive ones, and assigned longer and more distinct examples to the dev and test set. Additionally, 132 longer sentences are added to a separate test set. The statistics of the data are summarized in Table 4.2. As can be seen, the number of files in PMB5 increases in all splits. Also, sentences from PMB5 gold and silver splits are longer.

	# Docs	# Multi-sent	# Tokens	Tokens/ doc
Gold	11,379	89	63384	5.6
Silver	144,751	8,843	1,627,397	11.2
Bronze	145,035	3,918	1,099,710	7.6

Table 4.2: The data distribution of English dataset in the Parallel Meaning Bank Release 5.0.0

Apart from more data points, PMB5 also introduces new operators, roles, and structural changes:

- New operators: `TCT` and `ANA`;

- New roles: `Affectee`, `FeatureOf`, `Feature`;

- New Structures: The graph structure of predicate-coordination and proposition are changed. In coordination construction, rather than two verb nodes being connected by a `CONTINUATION` connective, the two predicate nodes are both connected with the root box. In proposition structure, the statement verb node introduces a new scope box.

- `SOURCE` and `ATTRIBUTION` are removed from PMB5;

One more important change made in PMB5 is that the discourse connectives can connect to further discourse chunks. In PMB4, all discourse connectives only connect the chunks preceding or following it, while in PMB5 a discourse connective can reach further than that. In SBNs with `proposition`, the `CONTINUATION` can also connect with no chunk but just serves as a discourse unit separator. These structural changes make DRGs in PMB5 more difficult to learn.

Table 4.3 provides a comparison between English PMB4 and PMB5 datasets. The number within the paratheses refers to the average count of tokens per file. We can see that the token length is basically the same in the train, dev, and test splits in both datasets, but sentences from PMB5 test_long split is considerably longer. We take it as an OOD dataset for evaluation.

PMB	Train	Dev	Test (Standard)	Eval	Test (Long)
PMB4	7,668 (5.6)	1,169 (5.2)	1,048 (5.5)	830 (5.8)	-
PMB5	9,056 (5.6)	1,137 (5.4)	1,137 (5.2)	-	132 (59.6)

Table 4.3: The data distribution of English gold dataset in PMB4 and PMB5

4.2 Data Characteristics

Anaphora PMB5 introduces a new operator labeled as ANA to facilitate coreference resolution. In total, there are 850 ANA edges distributed across all 11,360 files in the dataset. While coreference represented by ANA is not prevalent across the entire PMB dataset, it holds significant importance in DRT. Consequently, we have chosen to address this aspect, recognizing the potential for our research to inspire further exploration and advancement in DRS parsing studies.

Imperfections Both PMB4 and PMB5 have certain imperfections, including cyclic graphs, empty documents, isolated node, and index error. The first two are self-explainable. Index error refers to the situation where the indices go beyond those of the nodes. Regarding isolated node, it refers to some graphs in which the time node connects with the box rather than the event node. This does not make sense in meaning representation because the time information is associated with the event. We take them as ill-formed graphs and remove them from the evaluation data sets. PMB5 also provides an extra test dataset that contains 132 longer sentences. Among them, 17 graphs are cyclic and we remove them for further evaluation usage.

	# Cyclic Graphs	# Empty Documents	# Isolated Node	# Index Error	Sum
PMB4	35	4	6	0	45
PMB5	16	1	0	2	19

Table 4.4: The ill-formed graphs in PMB4 and PMB5 gold datasets

4.3 Baselines

While numerous studies have delved into DRS parsing, older ones primarily emphasized the generation of the sequential clausal format proposed by Van Noord et al. (2018). These earlier investigations commonly employed the COUNTER (van Noord et al., 2018) metric for evaluation and converted a DRS to a clause format, which is incompatible with the SMATCH metric we use in our work. For metric consistency, we have chosen to benchmark our work against the latest models in the field, including UD-Boxer (Poelman et al., 2022), Neural-Boxer (van Noord et al., 2020; Poelman et al., 2022), and DRS-MLM (Wang et al., 2023). The input is sentences and the output is their corresponding sequential SBN representation. The output can be converted to Penman notation and thus be evaluated with SMATCH scores. Additionally, we have fine-tuned a T5-base model (Raffel et al., 2020), named T5-Boxer, due to the prominence of T5 models in current semantic parsing endeavors. It is worth noting that UD-Boxer is a symbolic system explicitly designed for PMB4 and, therefore, cannot be considered a suitable baseline for the PMB5 dataset. All the pretrained language models used in our experiments are from Huggingface (Wolf et al., 2020).

UD-Boxer As explained in Chapter 3.2, UD-Boxer is designed to convert a dependency tree into a DRG by employing a specified set of rules. The system utilizes dependency parses derived from either Stanza (Qi et al., 2020) or Trankit (Nguyen et al., 2021). Given that Stanza has been observed to yield superior outcomes, our experiments adopt the output generated by Stanza to facilitate our analysis.

Neural-Boxer, T5-Boxer, and DRS-MLM Neural-Boxer is developed based on one of the SOTA systems in DRS parsing by van Noord et al. (2020). It employs Bert embedding and character embedding. We experiment with Neural-Boxer in two settings: (a) fine-tune Neural-Boxer with only gold data provided; (b) pretrain Neural-Boxer with the gold and silver data first and then fine-tune it solely on the gold data.

Regarding T5-Boxer and DRS-MLM, we just fine-tune them with the gold data split[9].

4.4 Evaluation

Metrics DRS parsing can be evaluated on metrics such as DSCORER (Liu et al., 2020) and COUNTER (van Noord et al., 2018). In this work, we utilize the SMATCH metric (Cai and Knight, 2013) and SMATCH++ (Opitz, 2023) as our evaluation criteria due to their compatibility with the graph-based format of the meaning representation. SMATCH and SMATCH++ are both designed to evaluate graph meaning representations. They

[9] The code can be found via `https://github.com/LastDance500/PMB5.0.0/tree/main`.

calculate the precision, recall, and f-score by finding the maximum overlap between the predicted and gold-meaning representation triples. The equations are listed below. The precision is defined as the number of matching triples between the predicted and the gold meaning representations divided by the total number of triples of the predicted meaning representations. The recall score is defined as the overlap triples divided by the total number of gold triples.

$$P = N_{match}/N_{pred} \tag{4.1}$$

$$R = N_{match}/N_{gold} \tag{4.2}$$

$$F1 = 2 * P * R/P + R \tag{4.3}$$

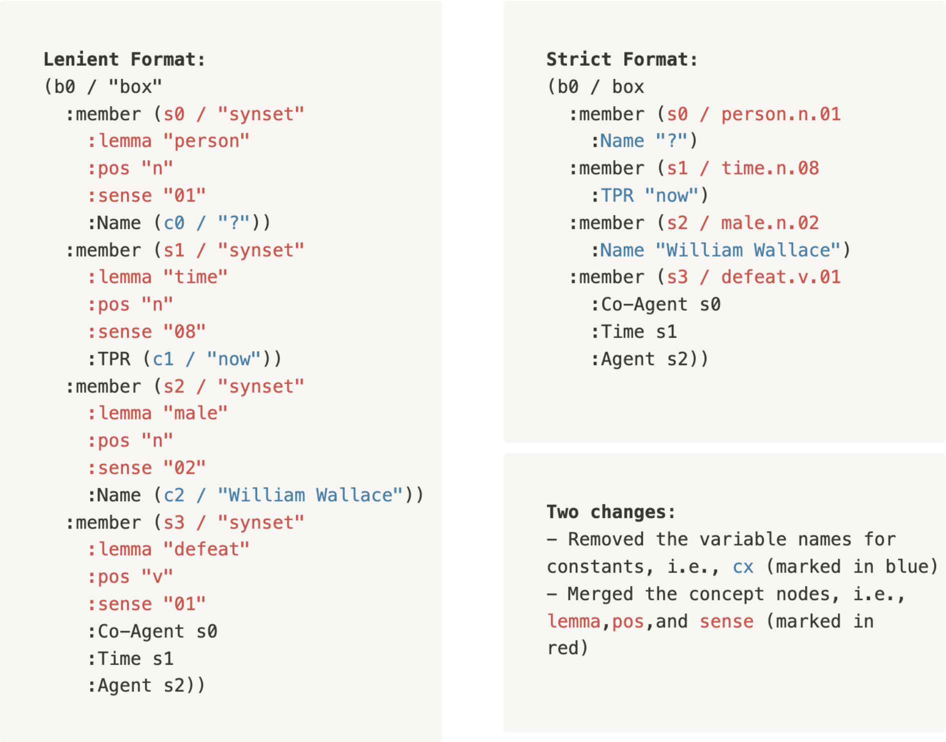
To obtain a fair evaluation, it is important to have optimal node alignment between the gold and the predicted graph pairs. However, node alignment between graph pairs is an NP-complete problem (Cai and Knight, 2013). How to tackle this problem is what distinguishes these two metrics. SMATCH employs a hill-climbing solver, but this solver cannot guarantee optimality and misses the upper-bound because, in complex graphs, there might be multiple local optima. The hill-climbing solver might result in unstable results when the graphs are essentially the same but have different internal structures[10]. To solve this problem, SMATCH++ uses Integer Linear Programming (ILP), thus generating a more reliable result.

From a practical perspective, the script for SMATCH++ removes duplicate nodes and also normalizes quotation markers during evaluation, thereby enhancing the overall efficiency of the process.

We report both metrics in order to have a fairer and more objective comparison between different models.

Evaluation Format The gold format can be either lenient (Poelman et al., 2022) or strict Wang et al. (2023). In the lenient format, the concept nodes are set apart by their lemma, lexical category, and sense number; the constant node variable is kept. By contrast, in the strict format, the concept nodes are compact and the constant node variables are removed. The two formats are shown below. We choose the strict format because the lenient one might inflate the metrics (Wang et al., 2023).

[10] Please refer to the issue mentioned in the SMATCH repository https://github.com/snowblink14/smatch/issues/43



```
Lenient Format:
(b0 / "box"
  :member (s0 / "synset"
    :lemma "person"
    :pos "n"
    :sense "01"
    :Name (c0 / "?"))
  :member (s1 / "synset"
    :lemma "time"
    :pos "n"
    :sense "08"
    :TPR (c1 / "now"))
  :member (s2 / "synset"
    :lemma "male"
    :pos "n"
    :sense "02"
    :Name (c2 / "William Wallace"))
  :member (s3 / "synset"
    :lemma "defeat"
    :pos "v"
    :sense "01"
    :Co-Agent s0
    :Time s1
    :Agent s2))
```

```
Strict Format:
(b0 / box
  :member (s0 / person.n.01
    :Name "?")
  :member (s1 / time.n.08
    :TPR "now")
  :member (s2 / male.n.02
    :Name "William Wallace")
  :member (s3 / defeat.v.01
    :Co-Agent s0
    :Time s1
    :Agent s2))
```

```
Two changes:
- Removed the variable names for
constants, i.e., cx (marked in blue)
- Merged the concept nodes, i.e.,
lemma,pos,and sense (marked in
red)
```

Figure 4.1: The lenient format and strict format for the sentence *Who did William Wallace defeat?* p10/d1983

4.5 Summary

In this chapter, we present the PMB4 and PMB5 datasets employed in our experiments. We adopt the three most recent DRS parsers and also fine-tune the popular T5 model to serve as baselines. For evaluation purposes, we utilize the SMATCH and SMATCH++ metric to provide an objective assessment of the models' performance. To ensure objectivity in evaluating models, our gold data adheres to the strict format.

Chapter 5
Method

This chapter opens with a brief discussion of the challenges in parsing both compositional and non-compositional elements of DRGs (§5.1). We then provide an overview of our approaches to address these issues (§5.2). Following this, we delve deeper into the specifics of our solutions (§5.3-5.5).

5.1 Problem Statement

Our task aims to map a natural language sequential input into a DRG represented in Penman format compositionally using AM-Algebra.

As mentioned in Chapter 2.6, the reentrancies introduced by the scope information of DRGs make the graph non-decomposable simply through the Apply and Modify operations. However, if we remove the scope information, the subgraphs might be disconnected because certain scope edges connect the isolated subgraphs. Hence, the first challenge of our task is to simplify the DRGs so that the key graph information is retained, while the graph remains decomposable by AM-Algebra.

Furthermore, AM-Algebra lacks specific rules to learn the non-compositional information of meaning representation graphs effectively. Since both scope and anaphora are important properties of DRT, it is necessary for our system to parse the scope and anaphora information accurately. Consequently, additional processing steps are necessitated atop AM-Parser to address these challenges.

5.2 Method Overview

To tackle the challenge of simplifying DRGs to achieve decompositionality, we explored various preprocessing strategies such as removing part of scope edges, reversing specific triples, and adding special token `START` for better node-token alignment.

As for the non-compositionality of anaphora, we take anaphora resolution as a lexical category classification task and train AM-Parser to learn if antecedent or anaphora nodes exist in the parses. We assign the antecedent and anaphora nodes a special category `p`, indicating they are related to pronouns.

By contrast, we take scope resolution as a dependency parsing task. Specifically, we assume that the scope edges in DRGs serve as dependency edges, linking the subgraphs that align with tokens. In this scenario, meaningful tokens are interconnected through dependency edges, forming a dependency graph. Hence, a dependency parser is trained to learn the scope edges between tokens.

Necessary postprocessing steps are required to map the anaphora and scope information back to the scopeless DRG parses generated by AM-Parser. Details can be found in the following sections.

5.3 Learning compositional DRGs

This section explains what preprocessing steps are needed to make DRGs decomposable. They include (1) DRG simplification through partial removal of scope edges; (2) elimination of coreference edges; and (3) inversion of triples that introduce reentrancies; (4) the introduction of a new special token aligned with the root box.

We provide a detailed explanation on the first preprocessing step in Chapter 5.3.1, while the subsequent steps are elaborated upon in Chapter 5.3.2.

5.3.1 Simplified DRGs and Scopeless DRGs

In this thesis, we propose two variants of DRGs, namely simplified DRG and scopeless DRG so that they can be decomposed by AM-Algebra. Below, we define them and also make a distinction between the simplified, scopeless, and complete DRGs.

Complete DRG The complete DRG refers to a complete graph with all information encoded from the SBN annotation. This is the gold graph for our evaluation.

Simplified DRG The simplified DRG assumes the implicit membership between boxes; that is, if one box is connected with one node, we assume that all of its children nodes are in the scope of that box. Therefore, we can safely remove the scope dashed lines between the box and the descendant nodes. This method keeps all necessary scope information to make sure the full picture can be easily recovered according to those necessary dashed lines.

Scopeless DRG The scopeless DRG disregards whether the scope information can be recovered or not - scope assignment will be done through another postprocessing task. In simplified DRGs, we guarantee that a scope box is connected with at least one node within the scope; in other words, a node can have reentrancies caused by the scope assignment. By contrast, in scopeless DRGs, we will not assign the scope as long as the node connects with its parent node. Therefore, scopeless DRGs have fewer reentrancies than simplified DRGs, which makes them easier to be decomposed into as-graphs. Examples of these two variants and their complete counterpart are shown in Figures 5.1 and 5.2 next page. As can be seen in Figure 5.1, the scopeless DRG graph only contains necessary dashed lines to keep its subparts connected, while simplified DRG also encodes other two dashed lines to indicate richer scope information. When DRGs only have one box, the two variants are essentially the same as shown in Figure 5.2, because only the root box contains scope information.

5.3.2 Other Preprocessing Steps

To develop a compositional parser, it is very important to align the token with its corresponding meaning representation graph. Given that in every DRG, there is always a "root" context box that does not have any token to be aligned with, we manually added a special token `START` as a signal to introduce the root box.

Additionally, we also remove reentrancies caused by coreference (see Fig 5.3 (left)) and multiple nodes connected with one box (see Fig 5.4 (left)). The preprocessed graphs are shown on the right of each figure.

For PMB4, which does not have explicit reference annotation, we detect triples in which the parent node and the child node share the same node name and have `EQU` as the edge name. Additionally, the child node has other incoming edges. For PMB5, we just remove triples that have `ANA` as the edge label.

Besides, we invert the triple if the parent node does not have any parent and does not serve as the root node of the as-graph. For example, we can invert the triple `x AttributeOf y` into `y Attribute x`. The invertible edge names include `Instance`, `Attribute`, `Colour`, `Content`, `Part`, `Sub`.

After preprocessing, only less than 10% of graphs in each graph format and dataset are non-decompsoable. Simplified DRGs are more difficult to decompose because they keep more reentrancy edges. The statistics are listed in Table 5.1 below.

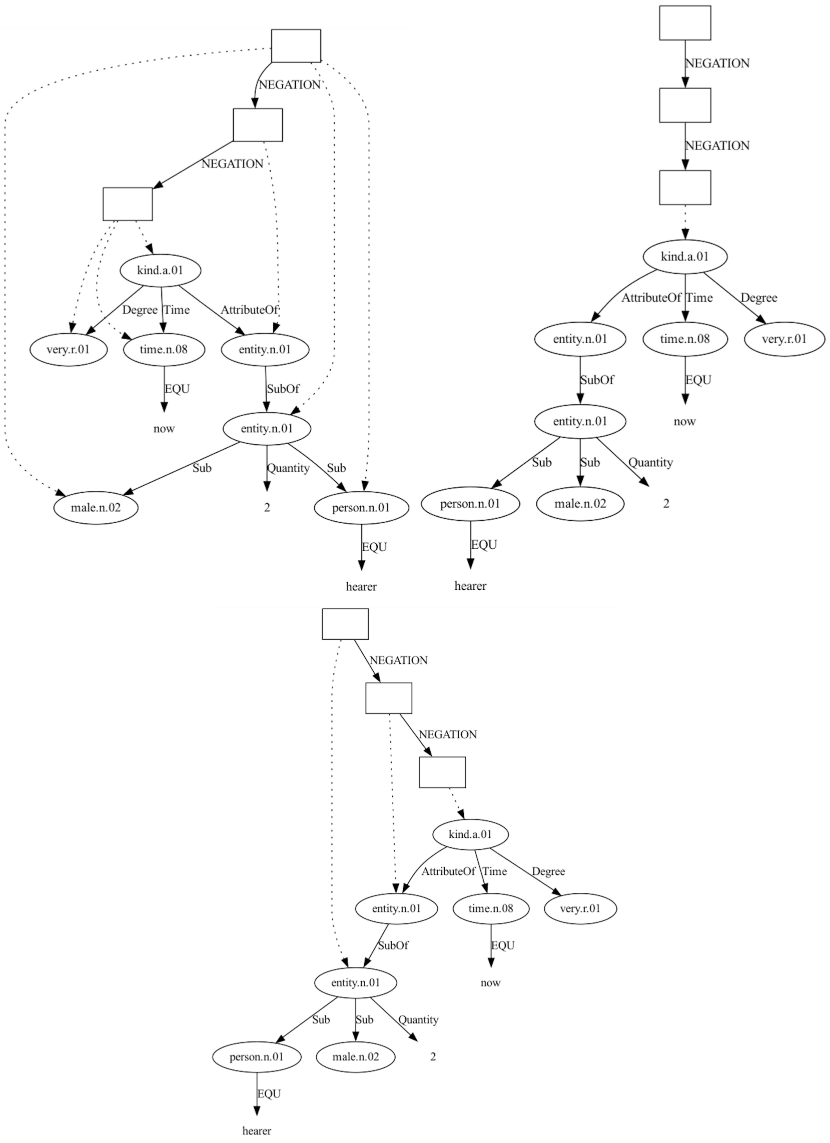


Figure 5.1: Examples of complete DRG (left), scopeless DRG(right), and simplified DRG (bottom) for the sentence *You and he both are very kind.* p04/d1630

We believe the two variants after all the preprocessing steps, particularly simplified DRG, keep the majority of the structural information. If we apply the three simple heuristic rules and the postprocessing steps for anaphora resolution over simplified DRG, which will be explained in Chapter 5.5 and 5.4 respectively, we can see the SMATCH and SMATCH++ F scores are very high (see Table 5.2). This suggests that most of the removed scope and coreference edges can be recovered.

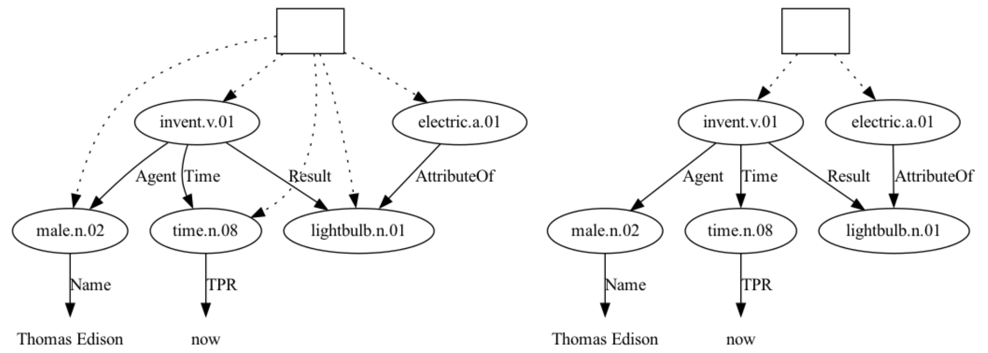


Figure 5.2: Examples of complete DRG (left), scopeless/simplified DRG (right) for the sentence *Thomas Edison invented the electric lightbulb.* p19/d1655

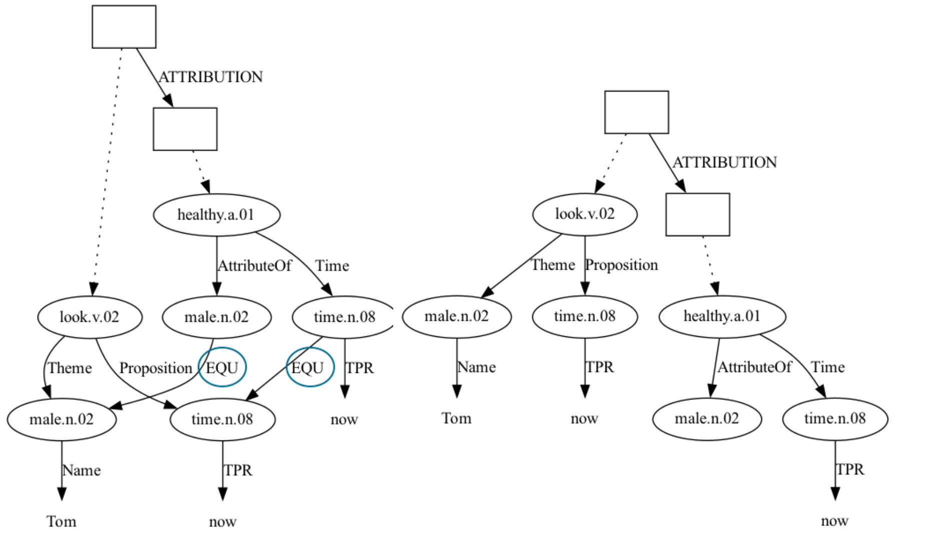


Figure 5.3: The scopeless DRG graph before (left) and after the coreference information has been removed (right) for the sentence *START Tom looked like he was healthy .* p20/d1811

5.3.3 Training AM-Parser

As-graphs are necessary in training AM-Parser but PMB does not provide as-graphs explicitly annotated with sources. To reduce the labor-intensive need, we adopt the decomposition approach proposed by Groschwitz et al. (2021) (henceforth *AM-Decomp*) that learns the globally consistent annotated as-graphs by jointly training AM-parser. With this joint learning strategy, only token-node alignment annotation is needed. AM-Decomp is essentially the AM dependency parsing the other way around. That is, it first

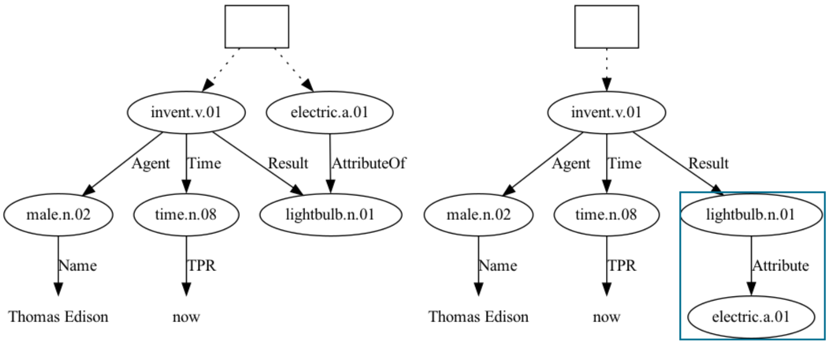


Figure 5.4: The scopeless DRG graph before (left) and after the triple is inverted (right) for the sentence *Thomas Edison invented the electric lightbulb.* p19/d1655

	Secopeless4	Simplified4	Scopeless5	Simplfied5
Success	7,007	6,663	8,646	8,346
Failure	630	974	409	710

Table 5.1: The number of successful or failed decomposition cases in PMB4 and PMB5

determines a unique AM-dependency tree T for the graph G and then assigns annotated sources to the sub-graphs of the tree through tree automata.

PMB provides loose node token alignment stated in the SBN files as shown in Figure 5.5 (left), but it is notable that not every node has a token to be aligned with and sometimes, multiple tokens might align with only one node. The node `entity.n.01` does not align with any token in the annotation while the node `kind.a.01` aligns with both the token *kind* and the period.

To address these two issues, in the process of alignment, we preserve the tree structure of subgraphs during the process. If there are nodes that do not have any token to be aligned with, we will recursively merge that node with its parent node until one parent node has an aligned token. Therefore, for the SBN annotation in Figure 5.5, the `entity.n.01` is grouped with its parent `kind.a.01` to be aligned with the token *kind*. Same applies to the `NEGATION` boxes - the two boxes inside the blue circle are aligned with the token *both*. If a node aligns with multiple tokens, we choose the token that has the most orthographic overlap with the node. Sometimes, there is no overlap between nodes and aligned tokens (e.g., `terra_incognita.n.01` vs. *of the*). In this case, we pick the longer token for the alignment.

In terms of training, we largely adhere to the setup from Groschwitz et al. (2021), with a few modifications during fine-tuning. All hyperparameters used in the experiments are reported in Appendix B.1.

DRG Type	Data	SMATCH F	SMATCH++ F
Simplified DRG	PMB4	96.4	96.4
	PMB5	97.3	97.3
Scopeless DRG	PMB4	93.5	93.5
	PMB5	94.7	94.7

Table 5.2: SMATCH and SMATCH++ F score for two types of DRGs after applying the heuristics to map scope and coreference back

```
person.n.01 EQU hearer                        % You
entity.n.01 Sub -1 Sub +1 Quantity 2          % and
male.n.02                                     % he
            NEGATION <1                       % both
entity.n.01 SubOf -2                          %
            NEGATION <1                       %
time.n.08   EQU now                           % are
very.r.01                                     % very
kind.a.01   AttributeOf -3 Time -2 Degree -1  % kind.
```

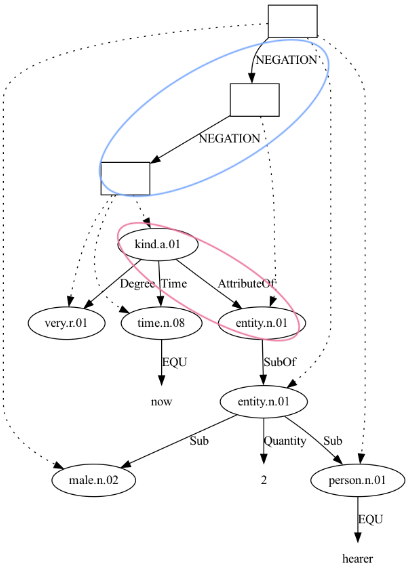


Figure 5.5: The SBN and its corresponding DRG for the sentence *You and he both are very kind* p04/d1630

5.4 Learning Anaphora Nodes and Rebuilding the Anaphora Edge

Learning coreference information is straightforward. In PMB5, coreference is explicitly indicated by an ANA edge. We take coreference resolution as a lexical category classification task. In other words, we incorporate the detection of anaphora/antecedent nodes into the training process of AM-Parser.

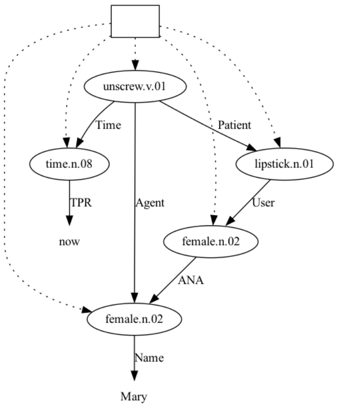


Figure 5.6: DRG for *Mary unscrewed her lipstick.* p26/d1754

To do so, in the preprocessing steps, we introduce a new lexical category `p` and replace the lexical category of the anaphor and antecedent nodes with this new category. Accordingly, the two coindexed nodes `female.n.02` in Figure 5.6 are changed to `female.p.02`

After we get the parses from AM-Parser, we extract nodes that are assigned a `p` lexical category. We find the parser only detects either one or two nodes as coreference nodes. We discard the single node. To make sure the two `p`-marked nodes corefer each other, we check if the node labels of these two nodes are the same. If yes, we add a new edge named `ANA` between two nodes marked with `p`; otherwise, we ignore them. The edge direction is from the node with a smaller node variable to the node with a greater one (e.g., `s1` → `s3`). After the edge is built, we change the `p` category back to `n`.

5.5 Learning Scope Information and Mapping it Back

In this section, we explain two approaches we propose to map scope information back. The purely symbolic approach is composed of three rules. The neuro-symbolic approach leverages the dependency parses from Dozat and Manning (2018).

5.5.1 Heurestics to Recover Scope Information

After we get the initial parses from am-parser, we add scope manually by following the three simple principles. We observe that in general, the kid nodes inherit the same scope assignment as their ancestor (parent/grandparent/grand grandparent) nodes.

After careful examination, we find that the higher the hierarchy of the box in the graph, the higher the priority it has regarding the scope assignment. In other words, if the parent box assigns scope to a node, then all the kids nodes of that node should be within the scope of the parent box and the children boxes cannot assign the scope to these nodes. In Fig 5.7, we can find that the root box assigns scope to the bottom `entity.n.01` and all of the kids of the `entity.n.01` node belong to the same box. However, the NEGATION box does not assign the scope edge to `male.n.02` node. Hence, which box a node belongs to should be determined by the highest box that its parent node belongs to.

Based on the observation, we propose three rules to map the scope back to the given

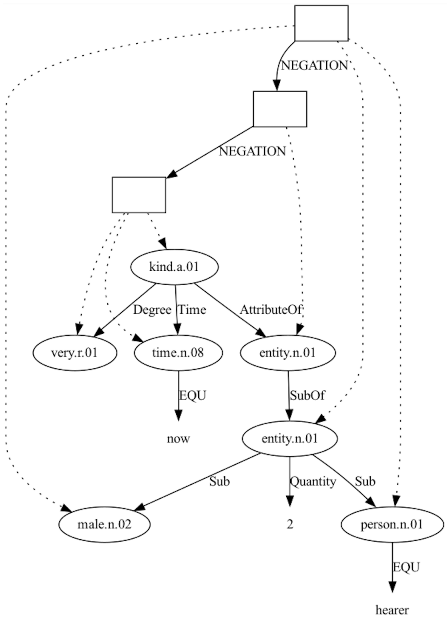


Figure 5.7: DRG for *You and he are both very kind.* p04/d1630

scopeless parses.

- *Assign the scope beginning from boxes at the higher levels of the hierarchy and moving towards the lower ones.*
- *If one node is connected with the connective box, all of its descendants are in the same scope.*
- *One node can only be exclusively assigned to scope of one box.*

For example, given Figure 5.8a, we can build the final graph as shown in 5.7 by adding scope in a top-down order. We first deal with the top root box as shown in 5.8b. `person.n.01` and `male.n.02` are the children nodes of `entity.n.01` and therefore they share the same scope. The scope edge is marked in blue. When it comes to the second `NEGATION` box, its child node `entity.n.01` does not have any child node that has not been assigned a scope. Hence, no new scope edge is added. Lastly and similarly, we add scope edge between `time.n.01/very.r.01` and the bottom box because their parent node is connected with that box. The resulting DRG 5.8c is the same as the one shown in Figure 5.7.

5.5.2 Dependency Parsing for Scope Information Recovery

Heuristic rules work well in simplified DRGs because the remaining scope edges provide sufficient information. However, in scopeless DRGs, particularly complex ones, certain scope information is missing due to the removal of edges, which makes it challenging to

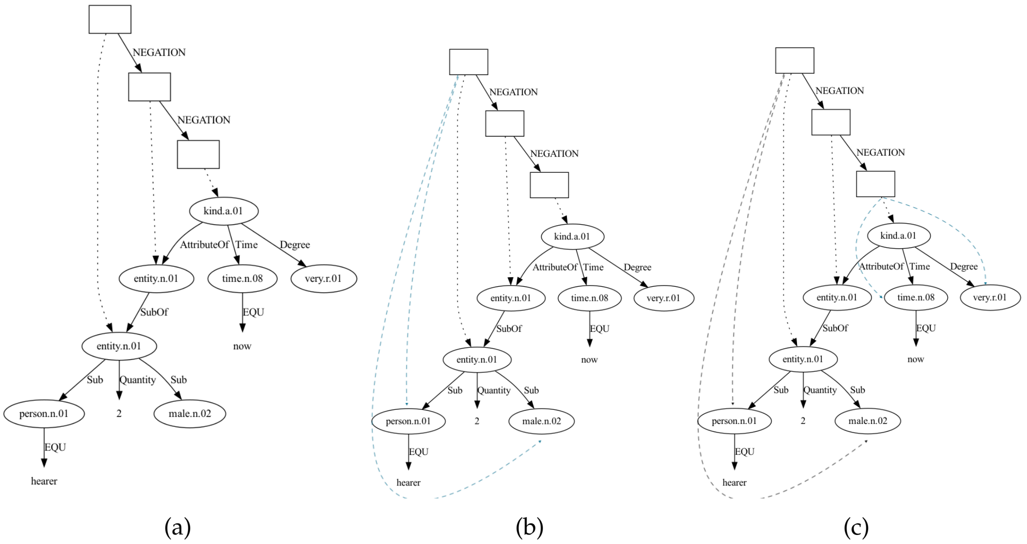


Figure 5.8: An example of mapping scope back to incomplete DRG

decide the scope assignment. In Figure 5.8a, if we remove the two leftmost dashed scope edges, we get the scopeless DRG for the same sentence. In this case, we would not be able to recover the scope - the scope connection between the box and the parent node is missing. Thus, the three heuristic rules would only assign all nodes within the scope of the bottom NEGATION box, because only the bottom box has a scope connection with the node `kind.n.01` in the DRG.

Hence, to parse more complex DRGs, we propose to take the scope assignment as a dependency parsing task. The parser is trained to learn if there exists a scope edge between any of two tokens in a given input string. As scope information is less diverse than typical dependency relations, we assume the task should be easier.

Task Description We take a DRG as a dependency graph in which all subgraphs are connected with the scope edges. By subgraphs, we refer to the as-graphs aligned with the input tokens. The DRG in Figure 5.9 can be decomposed into 6 subgraphs as shown in Figure 5.10. Each subgraph is aligned with one token. The scope edges can therefore be considered as relations between tokens, which is demonstrated at the bottom of the same figure.

According to the dependency tree at the bottom, if a node belongs to a box in terms of scope, a `scope` edge will be introduced to connect the two tokens that represent the box and the node respectively. The box token is the head, while the node token is the dependent. If a token does not contribute any meaning to the meaning representation, in the case of Figure 5.9, as illustrated in Figure 5.10, the period `.`, it will be assigned a `non_scope` edge connected with `START` aligned with the root box. Note that our task is

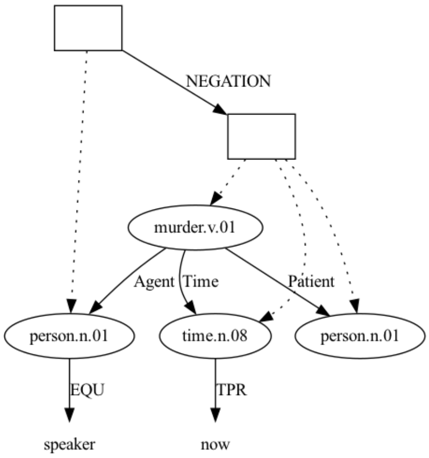


Figure 5.9: Complete DRG for *I didn't murder anyone.* p00/d3418

different from a typical dependency parsing task in that the parses are not necessarily trees. For example, in figure 5.10, the `NEGATION` box, along with the `murder` and `person` node is disconnected.

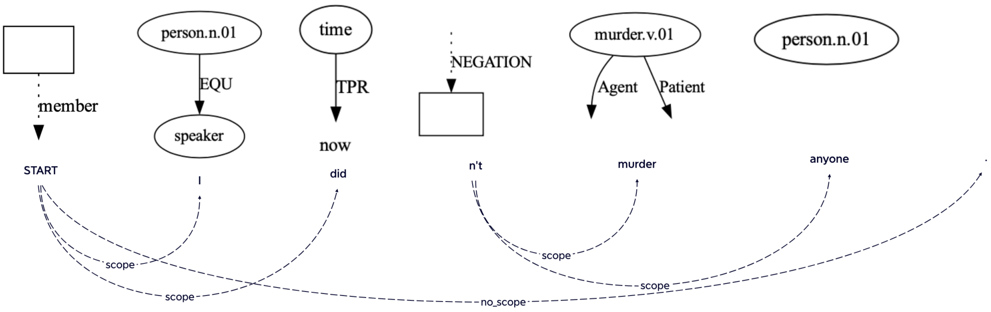


Figure 5.10: An example of scope assignment *I didn't murder anyone.* p00/d3418

Annotation To train a dependency parser, we need to define the gold edge labels. When each token exclusively aligns with one node/box, the scope mapping is easy. One just needs to assign a `scope` edge to connect the box-node pair. However, it is important to recognize that in numerous scenarios, a subgraph aligning with a single token may encompass multiple nodes or boxes, each potentially falling within the scope of different boxes. As a result, identifying which node belongs to which box specifically could be challenging. For instance, in Figure 5.11, the subgraph that represents *born* and *all* contains two nodes and two boxes respectively, but `person.n.01` is under the scope of the bottom `NEGATION` box, while the other `bear.v.02` node belongs to the top box. In this case, a binary edge, namely, `scope` and `no_scope`, is not informative anymore. Thus, adding more information to edges could be helpful.

In essence, apart from the straightforward one-to-one correspondence, where a subgraph aligning with a token, contains only one node within the scope of a single box represented by another token, we encounter three possible situations. For brevity, by subgraph, we mean the subgraph that aligns with one single token.

- Many-to-one: The node subgraph contains multiple nodes but they are all within the scope of the same box subgraph that contains only one box.
- One-to-many: The node subgraph contains only one node, and the box subgraph that assigns scope contains multiple boxes.
- Many-to-many: Both the node and the box subgraph contain multiple nodes/boxes. In this case,
  - multiple nodes can be assigned to the same box;
  - multiple nodes can be assigned to different boxes

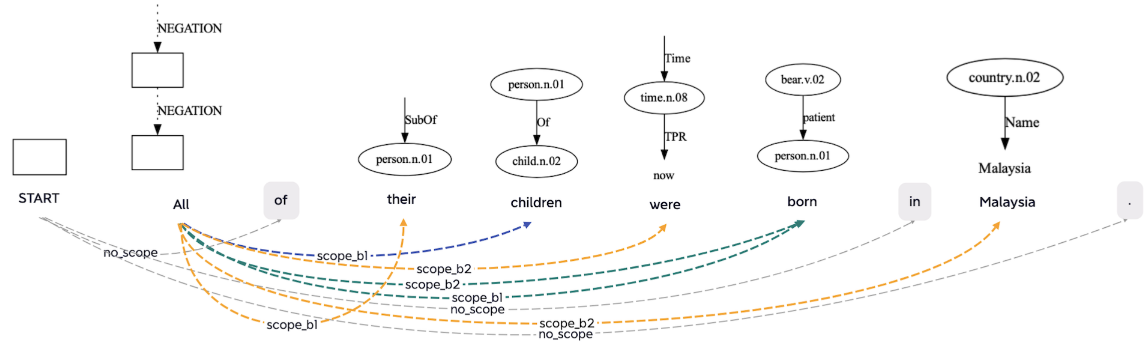


Figure 5.11: An example of complex scope assignment *All of their children were born in Malaysia.* p29/d2459

The first issue is easier to address because the node(s) can only be connected with the single existing box. By contrast, in the last two cases, which box the node should be connected with is undetermined. For example, in DRG for *All of their children were born in Malaysia* in fig 5.11, the graph that represents the token *born* contains two nodes, `bear.v.02` and `person.n.01`. Each node, however, is connected with different boxes. The edges are marked in green. A less tricky situation is *one-to-many* as shown by the relation between the subgraph `person.n.01` and two `NEGATION` boxes where the person node only connects with the bottom box. Such type of edges is marked in yellow.

Hence, we need to make the edge annotation more informative. In the case of Figure 5.11, the scope edge between *all* and *children* can be `scope_s2b1_s3b2` where the `sx` refers to the node variable and the `bx` refers to the box variable. The annotation indicates that the node `s2` should be connected with `b1`. However, during training, we find that adding node variables to scope edges makes the task more challenging as it makes the edge name more complex. Additionally, we notice that the node variables generated by AM-Parser largely follow the hierarchy of the graph in an ascending order, which means the preceding nodes should be on top of the following nodes. Therefore, we only make the box IDs explicit in the annotation because of the limited number of boxes in DRGs. We only rely on the node label (i.e., concept) to track the nodes.

The annotated example can be found in Table 5.3. As we can see, the token `were` is annotated with `scope_b2`, which indicates the box graph that it is connected with is a multibox one and the node should be within the scope of the `b2` box. Similarly, `born` is annotated with `scope_b2_b1`, which means the node graph contains two nodes and the top node is connected with `b2` and the bottom one is connected with `b1`.

#	text	lemma	upos	xpos	feats	head	deprel
0	START	START	START	START	_	0	root
1	All	all	PRON	DT	_	0	no_scope
2	of	of	ADP	IN	_	0	no_scope
3	their	their	PRON	PRP$	Number=Plur | Person=3 | Poss=Yes	1	scope_b1
4	children	child	NOUN	NNS	Number=Plur	1	scope_b1_b1
5	were	be	AUX	VBD	Mood=Ind | Tense=Past	1	scope_b2
6	born	bear	VERB	VBN	Aspect=Perf | Tense=Past	1	scope_b2_b1
7	in	in	ADP	IN	_	0	no_scope
8	Malaysia	Malaysia	PROPN	NNP	Number=Sing	1	scope_b2
9	.	.	PUNCT	.	PunctType=Peri	0	no_scope

Table 5.3: Edge annotation for the sentence *All of their children were born in Malaysia.*

Training a dependency parser We adopt Dozat and Manning (2018)'s Biaffine dependency parser for our task due to its simplicity and high accuracy.

The parser is adapted from Dozat and Manning (2017), an LSTM-based syntactic parser, to generate graph-structured representations for semantic dependency parsing. It takes POS tags, lemma- and character-level word embeddings as input and through a multilayer BiLSTM as well as a single layer Forward Network (FNN) to learn to predict if there is an edge between two tokens as well as the corresponding edge label. Then a biaffian classifier is used to predict the existence of an edge and the edge label.

In our experiment, we fine-tune `roberta-large` (Liu et al., 2019) and take POS tags and characters as feature embeddings[11]. The result can be found in Table 5.4[12]. We use Labeled Attachment Score (LAS) and Unlabeled Attachment Score (UAS) as the metrics. They both are very common metrics for dependency parsing evaluation. The former measures the percentage of correctly predicted head-dependency relationship between words without considering the edge label, while the latter is more stringent in that it also

[11] The detailed hyperparameters can be found in Appendix B.2.

[12] Note that the dev/test sets in PMB4 and PMB5 are not comparable. We put it together to save space. Additionally, the test_long split in PMB5 has not been manually corrected yet, so the gold dependency annotation is not available.

takes the correctness of edge label into account.

Models	Dev		Test		Eval	
	UAS	LAS	UAS	LAS	UAS	LAS
PMB4	98.8	95.7	98.7	95.4	98.7	94.4
PMB5	98.4	94.5	98.1	93.4	-	-

Table 5.4: UAS and LAS for dependency parsing in our data in PMB4 and PMB5.

As can be seen from the table, our task is easier than other semantic dependency parsing tasks. The high accuracy achieved by the model can enhance the accuracy of our mapping procedure.

Mapping Precedure After obtaining the parsed scopes and the predicted token-node alignment, our next step is to extract scope details and determine the assignment of the predicted scope to the anticipated nodes. Given that neither piece of information is gold, mismatches are expected. The mismatch is more common when it comes to processing multibox graphs. For instance, the DRG for the sentence *Not everybody wins!* contains 4 boxes and the dependency parse also indicates there are at least 4 boxes, but the predicted scopeless DRG only contains 3.

To address this, we designed a decision tree to map the scope information back. We first check how many boxes the predicted scopeless graph has. If it only has one box, we connect all nodes to that box; otherwise, we process the scope assignment based on the edge annotation and the number of nodes that a token-aligned subgraph has. If the subgraph has one or multiple nodes that belong to the scope of the same box, or in other words, the scope edge looks like either `scope` or `scope_bx`**n* (n≥1), we connect each node of that subgraph with the target box. Otherwise, if the dependency parse shows that multiple nodes in a subgraph are connected with different boxes that are aligned with the same token such as two NEGATION boxes introduced by universal quantifiers, we then check if the number of boxes in the scopeless graph parses is consistent or not. If the number of boxes in the am-parses and the dependency parses are the same, we connect the nodes with their corresponding box index. However, if the number of boxes is not the same, we connect all the nodes with the bottom box because in most cases, the bottom box is the most likely one to assign the scope. If the number of nodes between AM-parser and the biaffine dependency parser is inconsistent, we skip them. When all other nodes are assigned scopes, we then check the scope of their parent nodes and assign the same scope as their parent node.

We elaborate on the decision tree with the sentence *All my cakes are delicious!* In Table 5.5, it shows the parsed scope edge labels, the head of each edge, and the aligned subgraphs predicted by AM-Parser.

Since the graph contains multiple boxes, we first need to find out if there is any subgraph

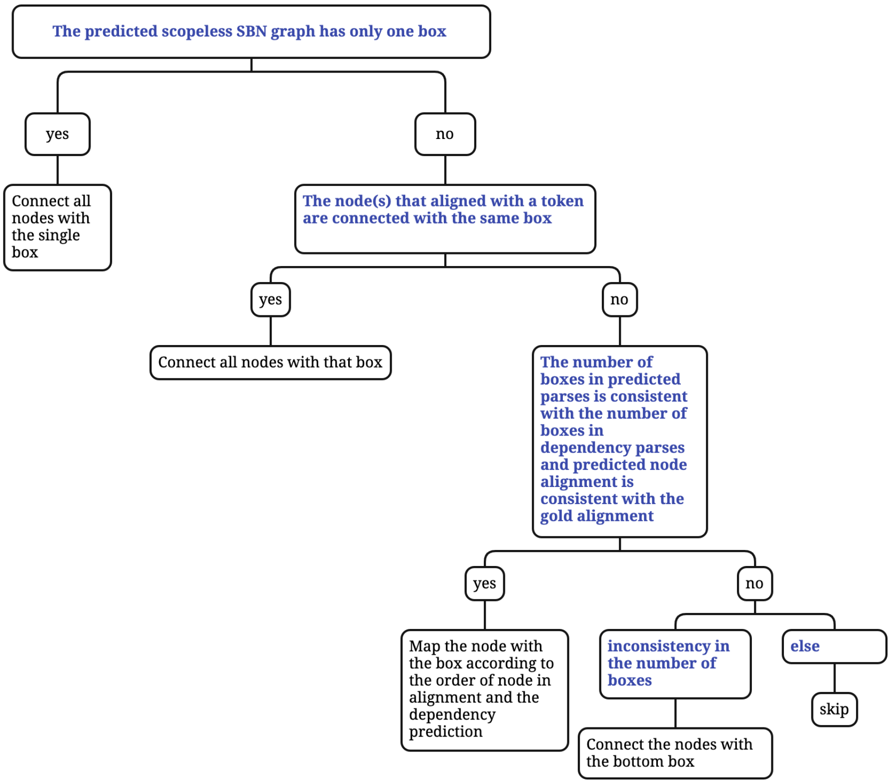

The predicted scopeless SBN graph has only one box
yes
no
Connect all nodes with the single box
The node(s) that aligned with a token are connected with the same box
yes
no
Connect all nodes with that box
The number of boxes in predicted parses is consistent with the number of boxes in dependency parses and predicted node alignment is consistent with the gold alignment
yes
no
Map the node with the box according to the order of node in alignment and the dependency prediction
inconsistency in the number of boxes
else
Connect the nodes with the bottom box
skip

Figure 5.12: The decision tree for mapping the scope edges back to the scopeless DRGs

the nodes of which are all connected with one box. The scope edge label for token *cake* is `scope_b1_b1` meets the condition. Apparently, the dependency parser predicts that there are two nodes, but AM-Parser only generates one node. No matter how many nodes that subgraph contains, all of the nodes are connected with the `b1`, namely, the second box, in the whole graph. In this case, we disregard the consistency between the scope edge and the aligned subgraph. We also find the single-node subgraphs aligned with *my* and *are* connected with a single box. This is also in line with the condition that all nodes connect with the same box. Therefore, we connect the `time.n.08` node with the `b2` or the third box in the graph, and `person.n.01` node is connected with box `b1`. Next, if the nodes in subgraphs connect with different boxes, as shown in the edge prediction `scope_b2_b1`, we first check if the number of nodes is consistent with the prediction of edge label. In this case, the number is inconsistent, because AM-Parser predicts that *delicious* contains one node, while the dependency edge predicts that the subgraph should have two nodes: one connects with box `b1` and the other connects with box `b2`. This inconsistency leads us to connect the node `delicious.a.03` with the bottom box, which is box `b2`. The resulting DRG is shown in Figure 5.13.

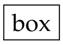
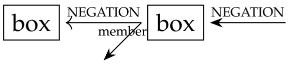
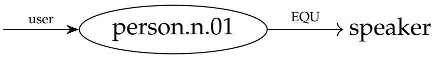
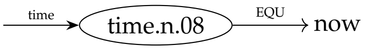
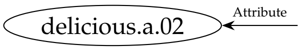
ID	Token	Lemma	Scope Edge	Scope Head	Aligned Subgraphs
0	START	START	root	0	
1	All	all	no_scope	1	
2	my	my	scope_b1	1	
3	cakes	cake	scope_b1_b1	1	
4	are	be	scope_b2	1	
5	delicious	delicious	scope_b2_b1	1	
6	!	!	no_scope	0	-

Table 5.5: Predicted scope edges, scope head, and aligned nodes for the sentence *All my cakes are delicious!*

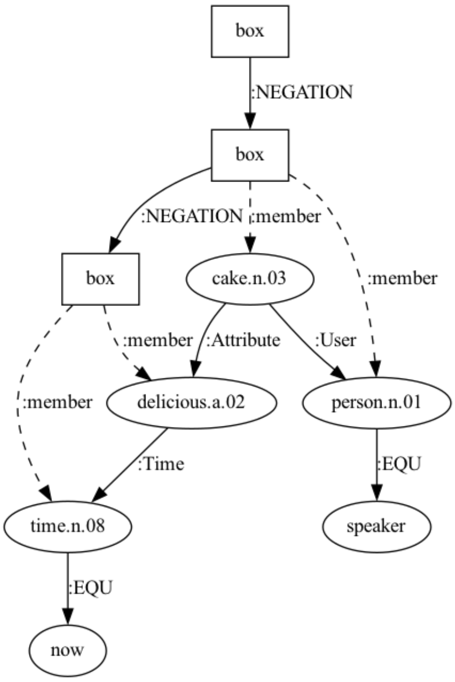


Figure 5.13: The resulting DRG after scope recovery

5.6 Summary

To summarize, this chapter explains how our system learns the compositional and non-compositional information of DRGs. In order to decompose DRGs, we propose two simplified DRG variants, i.e., simplified DRG and scopeless DRG. We also remove coreference triples and reverse certain invertible edges to allow more DRGs to be decomposed. We adopt AM-Parser, developed based on AM-Algebra, to learn the decomposed subgraphs and then map them back.

To resolve coreference, we first train AM-Parser to identify if a node is an anaphor/antecedent node or not. We then connect the two nodes in postprocessing steps. As for scope resolution, we propose two approaches. The symbolic one consists of three simple heuristic rules, relying on the assumption that the children nodes inherit the scope information from their parent nodes, while the neural-symbolic approach utilizes the edge parses from an accurate dependency parser (Dozat and Manning, 2018). We design a decision tree to map the scope edge back to the scopeless DRGs predicted by AM-Parser.

Chapter 6

Results & Discussion

This chapter reports the results of the experiments we have conducted. We compare the performance of our system with UDBoxer (Poelman et al., 2022) and other seq2seq neural models including NeuralBoxer (van Noord et al., 2020), T5-Boxer, and DRS-MLM (Wang et al., 2023). Additionally, we investigate the factors that hinder the performance of a compositional parser, specifically focusing on reentrancies and graph scopes. Finally, we perform an error analysis on graph parses with very low SMATCH scores.

6.1 General Results

Overall, we experimented with two simplified variants of DRG to make the graphs decomposable. To map the non-decomposable parts, i.e., anaphora and scope, back to the parsed meaning representation graphs, we proposed two solutions: (1) one symbolic approach that purely relies on the assumption that children nodes inherit the scope information from their parents; (2) one neuro-symbolic method that leverages the result of a biaffine dependency parser and token-node alignment predicted by the AM-Parser.

Since the datasets are reconstructed and reshuffled in PMB5, we report the results evaluated on PMB4 and PMB5 separately. Tables 6.5 and 6.2 present the outcomes of four experiments conducted on the two releases. Only the predictions of test_long set have ill-formed graphs, so we only report the error rate for this specific split.

As evident from our evaluation on PMB4 and PMB5 datasets, it is clear that AM-Parser, trained with scopeless DRGs, demonstrates better performance in both the development and test splits. Furthermore, the utilization of dependency edge information for scope retrieval exhibits a slight advantage over the heuristics-based approach. More fine-grained results are reported in Appendix A. It is also worth noting that there is no big difference between the SMATCH and SMATCH++ scores except in the test_long split in PMB5.

Unsurprisingly, our system's performance exhibited a substantial decline when evaluated on the extended test_long split because the longer the sentence, the more possible combinations of supertags and am-dependency trees, and therefore, more mistakes might be made. Additionally, the sentences contain substantial proper nouns which AM-Parer is not good at (see Error Analysis in Section 6.3).

ScopeMapping	Experiments(Metrics)	Dev			Test			Eval		
		P	R	F1	P	R	F1	P	R	F1
Heuristics	Scopeless	86.5	85.0	85.7	85.9	85.7	85.8	84.1	83.7	83.9
	Scopeless(++)	86.4	84.9	85.7	85.8	85.5	85.6	84.1	83.7	83.9
	Simplified	85.5	83.9	84.7	84.5	83.5	84.0	85.2	83.7	84.4
	Simplified(++)	85.4	83.9	84.6	85.6	84.5	85.1	85.1	83.6	84.3
Dependency	Scopeless	86.9	85.4	86.2	86.4	86.2	86.3	84.5	84.1	84.3
	Scopeless(++)	86.9	85.4	86.1	86.4	86.1	86.2	84.4	84.0	84.2
	Simplified	85.6	84.1	84.8	85.7	84.7	85.2	85.0	83.6	84.3
	Simplified(++)	85.5	84.0	84.7	85.6	84.5	85.1	85.1	83.6	84.4

Table 6.1: Performance Results of AM-parser on PMB 4.0.0: SMATCH++ (++) and SMATCH.

ScopeMapping	Experiments(Metrics)	Dev			Test			Test Long			
		P	R	F1	P	R	F1	P	R	F1	Err
Heuristics	Scopeless	86.7	86.4	86.6	85.6	85.4	85.5	48.2	42.0	44.9	1.7%
	Scopeless(++)	86.6	86.3	86.4	85.4	85.2	85.3	50.0	42.5	46.0	1.7%
	Simplified	86.9	86.3	86.6	85.7	84.7	85.2	44.7	36.5	40.2	6.0%
	Simplified(++)	86.8	86.2	86.5	84.5	85.5	85.0	43.6	36.1	39.2	6.0%
Dependency	Scopeless	87.3	87.0	87.2	85.8	85.7	85.7	48.2	40.2	43.9	3.4%
	Scopeless(++)	87.1	86.9	87.0	85.7	85.5	85.6	50.9	42.5	46.3	3.4%
	Simplified	86.9	86.3	86.6	85.8	84.7	85.3	45.1	30.9	36.7	14.7%
	Simplified(++)	86.7	86.1	86.4	85.6	84.5	85.1	37.9	25.6	30.6	14.7%

Table 6.2: Performance Results of AM-parser on PMB 5.0.0: SMATCH++ (++) and SMATCH

6.2 Comparison with baselines

Given that the *dependency+scopeless* configuration yields the most favorable results among all the experiments, we employ its associated metrics for comparative analysis with other symbolic or neural parsers.

As mentioned in 4.3, the output of seq2seq models is sequential SBNs, e.g., `entity.n.01 NEGATION -1 time.n.08 EQU now right.a.01 AttributeOf -2 Time -1`. To have a fair comparison, we transform the SBNs to DRGs. Sometimes the generated pen-

TrainingData	Models	Dev				Test				Eval			
		P	R	F1	Err	P	R	F1	Err	P	R	F1	Err
Gold only	UD-Boxer	75.1||75.0	71.8||71.6	73.4||73.3	.2%	75.6||75.4	72.1||71.9	73.8||74.0	**0%**	74.5||74.4	70.6||70.4	72.5||72.4	.4%
	Neural-Boxer	83.6||83.5	73.9||73.6	78.4||78.4	8%	83.9||84.0	75.2||75.2	79.3||79.3	6%	81.0||80.5	70.7||70.8	75.5||75.3	8%
	T5 Boxer	**92.3**||**92.3**	**87.1**||**87.1**	**89.6**||**89.6**	4%	**92.6**||**92.6**	**88.3**||**88.3**	**90.4**||**90.4**	4%	**91.3**||**91.3**	**86.0**||**86.0**	**88.6**||**88.6**	4%
	AM-Paser	86.9||86.9	85.4||85.4	86.2||86.1	**0%**	86.4||86.4	86.1||86.1	86.3||86.2	**0%**	84.5||84.4	84.5||84.0	84.3||84.2	**0%**
Gold+Silver(EN)	Neural-Boxer	92.3 ||89.1	88.8||87.0	90.7||89.0	3%	92.6||92.5	88.8||88.8	90.6||90.6	3%	91.6||91.6	86.9||86.9	89.2||89.2	4%
Gold+Silver(MUL)	DRS-MLM	-	-	-	-	-	-	94.0	.2% -	-	-	-	

TrainingData	Models	Dev				Test				TestLong			
		P	R	F1	Err	P	R	F1	Err	P	R	F1	Err
Gold only	Neural-Boxer	83.6 ||83.6	72.6||72.7	77.8||77.8	7%	82.1||82.1	70.6||70.6	75.9||76.0	9%	8.8||9.27	60.1||62.9	15.4||16.2	79%
	T5 Boxer	**92.9**||**92.9**	71.2||71.2	80.6||80.6	21%	**91.9**||**91.9**	72.6||72.6	81.1||81.1	18%	**75.6**||**76.5**	2.5||2.5	4.8||4.8	92%
	AM-Paser	87.3||87.1	**87.0**||**86.9**	**87.2**||**87.0**	**0%**	85.8||85.7	**85.7**||**85.5**	**85.7**||**85.6**	**0%**	48.2||50.9	**40.2**||**42.5**	**43.9**||**46.3**	**3.4%**
Gold+Silver(EN)	Neural-Boxer	89.1||89.1	81.9||82.1	85.4||85.5	6%	91.1||91.0	79.1||79.3	84.7||84.7	12%	60.0||62.8	8.8||9.3	15.4||16.1	79%
Gold+Silver(MUL)	DRS-MLM	94.7||94.7	90.5||90.5	92.5||92.5	3%	94.4||94.4	88.7||88.7	91.5||91.5	4%	82.0||81.9	5.5||5.7	10.2||10.6	82%

Table 6.5: SMATCH (left) and SMATCH++ (right) of all models on PMB4 (Top) and PMB5 (Bottom) Datasets - the best model trained exclusively on gold data is denoted in bold, while the overall best-performing model is indicated with underlining. 94.0 by MLM-DRS in PMB4 is copied from (Wang et al., 2023).

man output is legal but the node concept is not accepted by the SMATCH evaluation script (Cai and Knight, 2013). For example, AM-Parser might generate `).n.01`; T5 model might generate `" William"`. In this case, we replace all nonacceptable nodes with `entity.n.01` or remove the inappropriate space. For the ill-formed graphs, we set the output as `(b0 / box)` by default.

The results are presented in Table 6.5. Unsurprisingly, DRS-MLM pretrained with gold, silver, and bronze data performs the best in general. However, despite the utilization of a slightly larger training dataset in PMB5, the performance of all models, except AM-parser, exhibits a decline. In particular, the T5 model experiences a substantial decrease of approximately 10%, resulting from an error rate exceeding 20%. We postulate that this decline can be attributed to the increased complexity of discourse connective structures within DRGs in PMB5—a facet that poses a significant challenge for seq2seq models. In contrast, our compositional model excels due to its capacity to learn structural information during supertagging and dependency parsing training.

We can also see that AM-parser outperforms UD-Boxer and Neural-Boxer in both PMB4 and PMB5 if only the gold data is used in training. In PMB5, our parser even outperforms Neural-Boxer trained with both silver and gold data, which indicates that implicit linguistic knowledge does help the model learn and generalize structures more effectively.

More importantly, because of the symbolic decoding approach used by am-parser, the output is always legal, guaranteeing the well-formedness of DRGs. In PMB4, the fine-tuned T5 model yields the best results in most of the metrics. The better performance might be caused by the simplicity of the data and the high efficiency of the transformer-based architecture. By contrast, in PMB5, we notice that T5 model has a very high error rate (around 20%)[13]. Upon closer examination of the parsing results, we find that, in most error cases, the model does not generate legal separators (e.g., `<1`, `>1`) but just generates a single number. As a result, the script cannot assign the scope of connectives to the nodes and a legal penman cannot be exported. Such a big discrepancy made between two datasets might be caused by the increased complexity of the graph structures in PMB5. The huge drop in parsing longer sentences is echoed with (Opitz and Frank, 2022) who pointed out that T5 outperforms BART in short sentence AMR parsing.

In PMB5, all seq2seq models failed in longer sentences with a very high error rate. We conducted a manual examination of the output generated by the largest model, DRS-MLM. Within the test_long dataset, 4 out of 5 generated graphs were ill-formed. Among these, over 10% contain cyclic subgraphs, while 50% of the errors were attributed to issues with ill-formed sequential notations. These issues often manifested as nodes lacking sense numbers or operators missing required arguments. Seq2seq models taking input sentences as plain strings do not inherently encode structural information and thus

[13] We also fine-tuned a T5-Large model to check if results get better but the error rate remains 21% and F1 score only increases by 1%.

Models	Dev			Test			TestLong		
	P	R	F1	P	R	F1	P	R	F1
Neural-Boxer(G)	58.9	70.5	64.1	53.4	48.4	50.8	0	0	0
T5 Boxer(G)	90.9	60.6	72.7	67.2	67.2	67.2	0	0	0
Neural-Boxer(G+S)	**94.2**	74.2	83.1	79.7	79.7	79.7	0	0	0
DRS-MLM(G+S+B)	83.8	93.9	88.6	80.0	**100.0**	88.9	100	10.3	18.6
AM-Paser(Gold)	93.9	**93.9**	**93.9**	**93.8**	93.8	**93.8**	**100**	**28.8**	**44.8**

Table 6.6: SMATCH score on anaphora resolution of all models evaluated on PMB 5.0.0. G, S, or B in parentheses refers to the dataset (Gold, Silver, Bronze) used for training.

cannot guarantee the grammaticality of the output. Especially when the input becomes longer and graph complexity increases, the likelihood of errors in seq2seq model outputs also rises.

Lastly, we find that if we adopt the more strict evaluation format (Wang et al., 2023), the advantage of UD-Boxer in accuracy disappears. In PMB4, the SMATCH F score for the test set drops from 82.0 to 73.8, far behind Neural-Boxer trained with only gold data (F = 79.3).

6.2.1 Anaphora

In PMB5, explicit anaphora information is available, and we are particularly interested in assessing how effectively our system handles this specific linguistic phenomenon. For the sake of readability, we will only present the SMATCH score in the text, and the detailed results are listed in Table 6.6. The SMATCH score only calculates the precision, recall, and F1 score between the predicted and gold anaphora triple pairs. Notably, our system outperforms DRS-MLM by a margin of over 5%, despite DRS-MLM being the top-performing model in overall DRS parsing.

In the test_long set, none of the models exhibit satisfactory performance. No `ANA` edge is found in the transformed penman output of T5-Boxer or Neural-Boxer because those sequential SBNs that contain `ANA` are filtered out due to their ill-formedness. Although AM-Parser achieves a relatively low error rate, the recall and F-score for anaphora triples remain very low. These results underscore the considerable challenge associated with coreference resolution in longer texts.

6.2.2 Scope

Since scope is an important feature in DRT, we would also like to examine how well our dependency parsing system performs regarding scope information. To evaluate the performance, we calculate the SMATCH score between predicted and gold triples that take `member` as the edge label. The results for PMB4 and PMB5 can be found in Table 6.7

and 6.8 respectively. We include the scope results obtained through AM-Parser+heuristic rules (labeled as *AM-Parser-H*) for comparison with the dependency parsing strategy (labeled as *AM-Parser-D*).

In both tables, our AM-Parser+Dependency system outperforms all models trained with only gold data except for T5-Boxer evaluated on the eval split. Notably, even the three heuristic rules, despite their simplicity, still show competitive performance with those more complex seq2seq models.

Models	Dev			Test			Eval		
	P	R	F1	P	R	F1	P	R	F1
UD-Boxer(G)	81.4	78.4	79.9	81.9	79.3	80.6	81.1	77.8	79.4
Neural-Boxer(G)	84.7	74.6	79.3	85.5	76.4	80.7	83.0	72.2	77.2
T5 Boxer(G)	**93.6**	88.1	90.8	93.2	86.5	89.7	**94.0**	**90.5**	**92.2**
Neural-Boxer(G+S)	93.3	89.7	**91.5**	**94.0**	90.1	**92.0**	92.9	88.1	90.4
AM-Paser-D(G)	91.8	**90.7**	91.2	91.4	**91.8**	91.6	89.7	89.8	89.8
AM-Parser-H(G)	90.6	88.8	89.7	90.4	90.0	90.2	88.7	88.1	88.4

Table 6.7: SMATCH score on scope resolution of all models evaluated on PMB 4.0.0.

Models	Dev			Test			TestLong		
	P	R	F1	P	R	F1	P	R	F1
Neural-Boxer(G)	86.3	74.6	80.0	85.6	73.1	78.8	9.2	66.0	16.1
T5 Boxer(G)	94.1	70.9	80.9	93.7	72.8	81.9	79.2	1.8	3.4
Neural-Boxer(G+S)	90.5	83.1	86.6	92.8	79.7	85.7	66.2	9.1	16.1
DRS-MLM(G+S+B)	**95.6**	**91.3**	**93.4**	**95.5**	89.8	**92.5**	**86.9**	5.1	9.6
AM-Paser-D(G)	91.3	91.2	91.2	90.9	**90.9**	90.9	60.4	50.1	54.8
AM-Parser-H(G)	90.8	90.0	90.4	90.6	90.0	90.3	59.8	**51.7**	**55.3**

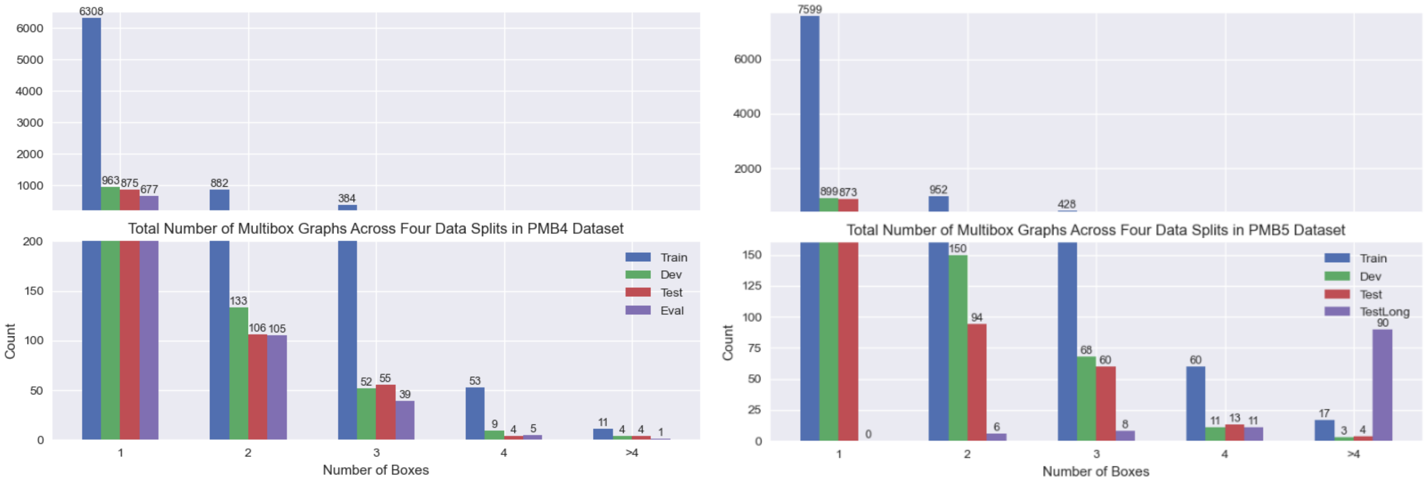
Table 6.8: SMATCH score on scope resolution of all models evaluated on PMB 5.0.0.

As demonstrated in Figure 6.1, the majority of DRGs in the datasets consist of single-box structures that have straightforward scope recovery. It would be valuable to assess model performance specifically on multi-box graphs to check if the models can assign nodes to specific scopes. Hence, we extract all multi-box graphs from the dev and test splits (and eval split in PMB4). There are 517/3,540 multibox graphs in PMB4 evaluation datasets and 403/2,175 in PMB5. Note that we exclude the test_long data from evaluation due to its inclusion of examples with a high number of boxes, resulting in significantly lower scope recovery scores. The high error rates of baselines might introduce bias into the averaged scope score as well.

Table 6.9 and 6.10 show the SMATCH score only on multibox graphs overall (left) and specifically the SMATCH score on scope information (right). To calculate the SMATCH scope score, we only consider the triples that have `member` as edges. To gain a higher score, the model must predict the child node, the edge name, and the parent box correctly.

The SMATCH$_{SCOPE}$ in both table does show a similar trend as Table 6.7 and 6.8: our dependency-based approach can effectively map the lost scope back to the scopeless DRG parses. AM-parser with the dependency parses beats all baselines that are trained exclusively with gold data in both PMB4 and PMB5. In PMB5, the result is even better than Neural-Boxer trained with both silver and gold data.

Additionally, it is also observed that the general SMATCH score for multi-box DRGs is lower than DRGs overall, indicating that scope assignment does add more difficulty in DRG parsing.



(a) PMB4

(b) PMB5

Figure 6.1: Distribution of Multibox Graphs in PMB4 and PMB5

Models	SMATCH			SMATCH_{Scope}		
	P	R	F1	P	R	F1
UD-Boxer	58.2	46.9	51.9	60.3	50.0	54.6
Neural-Boxer(Gold)	63.7	46.8	54.0	68.3	51.5	58.7
T5 Boxer(Gold)	66.6	53.6	59.4	71.5	59.2	64.8
Neural-Boxer(+Silver)	**70.0**	**59.8**	**64.5**	**74.0**	**64.8**	**69.1**
AM-Paser-D(Gold)	64.4	55.8	59.8	69.0	62.1	65.4
AM-Parser-H(Gold)	63.1	54.6	58.5	65.2	57.1	60.9

Table 6.9: SMATCH score on scope resolution of all models evaluated on PMB 4.0.0 **multibox graphs**.

6.2.3 Reentrancies

Similar to AMR, DRG is graphs rather than trees because it allows reentrancies. In Table 6.11, we present the SMATCH reentrancy scores achieved by various models in all of the evaluation datasets in PMB4 and PMB5 respectively.

As noted by Damonte et al. (2017), reentrancies pose a significant challenge for seq2seq

Models	SMATCH			SMATCH_{Scope}		
	P	R	F1	P	R	F1
Neural-Boxer(Gold)	81.4	62.3	70.6	83.8	63.6	72.3
T5 Boxer(Gold)	90.6	12.2	21.5	75.7	3.9	7.4
Neural-Boxer(+Silver)	91.2	75.9	82.8	92.0	76.2	83.3
DRS-MLM(E)	**94.8**	**81.5**	**87.7**	**95.7**	81.7	**88.2**
AM-Paser-D(Gold)	84.0	81.9	83.0	85.2	**82.9**	84.0
AM-Parser-H(Gold)	81.4	79.4	80.4	80.9	76.1	78.4

Table 6.10: SMATCH score on scope resolution of all models evaluated on PMB 5.0.0 **multibox graphs**.

models. Given that our compositional parser incorporates linguistic knowledge, our research aims to empirically investigate its performance in handling reentrancies in DRG parsing.

By reentrancies in DRG, we do not mean the reentrancies introduced by scope because otherwise, every concept node can be the case. We also ignore the reentrancies caused by invertible edges such as `PartOf`, `Part`, `FeatureOf`, and `Feature`, because the reentrancy can disappear if we invert the triple. In other words, we only consider the reentrancies caused by non-scope and non-invertible edges. In PMB4, out of a total of 10,670 examples, 1,273 exhibit reentrancies, with 361 of them originating from the dev, test, and eval splits. In PMB5, among the 11,377 examples, 1,257 contain reentrancies, and 158 of these instances are drawn from the dev and test splits.

The evaluation script employed is based on Damonte et al. (2017), with minor modifications to exclude scope reentrancies from consideration.

Models	PMB4			PMB5		
	P	R	F1	P	R	F1
UD-Boxer(G)	69.6 (48.5)	59.1 (9.5)	64.0 (15.9)	-	-	-
Neural-Boxer(G)	76.3 (51.0)	76.0 (26.9)	65.6 (35.2)	79.3 (58.5)	61.3 (4.6)	69.2 (43.5)
T5 Boxer(G)	89.1 (69.8)	76.4 (52.6)	82.3 (60.0)	90.5 (77.8)	63.3 (43.2)	74.5 (55.6)
Neural-Boxer(G+S)	**90.2 (78.9)**	**82.3 (65.9)**	**86.1 (71.8)**	89.7 (83.6)	78.2 (57.5)	83.6 (68.2)
DRS-MLM (G+S+B)	-	-	-	**93.6 (86.0)**	**84.9 (70.9)**	**89.0 (77.7)**
AM-Parser-D(G)	78.2 (70.5)	74.9 (26.6)	76.5 (38.6)	82.4 (70.6)	81.6 (53.4)	82.0 (60.8)

Table 6.11: SMATCH Overall and Reentrancy Scores (in parentheses) for Graphs Containing Reentrancies in all Evaluation datasets of PMB4 and PMB5.

As depicted in Table 6.11, when trained solely on gold data and evaluated on PMB4, AM-Parser achieves the second-highest ranking, following T5 Boxer. We attribute this outcome partly to AM-Parser's potential limitation in predicting coreference relations, which are known to be pivotal triggers for reentrancies. In PMB4, coreference is not explicitly annotated. To make more graphs decomposable for training data, we just remove most of the reentrancy edges. In contrast, within the context of PMB5, AM-Parser exhibits superior performance compared to its gold-based baselines. Nonetheless, it

Reentrancies	AM-Parser(G)	Neural-Boxer(G)	T5-Boxer(G)	Neural-Boxer(G+S)	DRS-MLM(G+S+B)
Coref	84.7 (64.4)	74.1 (55.2)	72.1 (68.6)	85.5 (79.3)	90.7 (82.9)
Coordination	73.8 (46.9)	53.9 (22.1)	75.3 (38.9)	75.3 (58.8)	86.5 (70.4)
Control	84.1 (70.9)	71.7 (57.5)	70.3 (69.1)	86.5 (88.0)	94.8 (90.0)
Relative clause	82.5 (40.0)	57.6 (40.0)	64.8 (21.1)	60.3 (17.1)	92.4 (73.5)
Time	81.7 (67.0)	65.6 (67.0)	77.6 (59.2)	80.0 (59.3)	80.7 (70.5)
Verbalization	86.1 (60.6)	74.3 (31.4)	89.8 (50.0)	80.0 (56.5)	91.8 (78.7)
Miscellaneous	80.1 (30.3)	65.5 (13.3)	54.3 (0.0)	67.1 (13.3)	83.4 (37.0)

Table 6.12: SMATCH score of all models for different types of reentrancies in PMB5.

is noteworthy that a performance gap persists between AM-Parser and the other two models trained on larger datasets in predicting reentrancies (60.8 vs. 68.2/77.7).

We then did a more fine-grained analysis by categorizing reentrancies into different linguistic structures. Our primary focus lies on 158 examples from PMB5 because our model trained on PMB5 incorporates coreference resolution and can provide a more objective comparison with other baselines. We subsequently categorize these 158 instances into 7 groups based on Szubert et al. (2020) but it has been tailored to suit the data in PMB. We automatically categorize the examples based on POS and NER tags from spaCy pipeline followed by manual correction. The categories include coreference, control, coordination, relative clauses, verbalization, time, and miscellaneous. The first four categories are common linguistic constructions. Verbalization refers to the phenomenon that a non-predicate element, such as an adjective or a gerund, does not have explicit arguments in sentences but requires node arguments in meaning representation graphs. Time is a special category in PMB where a complex time expression introduces reentrancies. Miscellaneous includes 7 examples: they are adjunct control (2), presupposition (2), and compound noun phrase (3). Because of their limited number, we group them together. Examples can be found in Table 6.14 at the end of the chapter.

Overall, AM-Parser still outperforms other gold-only neural models in all reentrancy types. It performs particularly better in predicting the time reentrancy and relative clause than Neural-Box trained with both silver and gold data. In Table 6.12, **Miscellaneous** and **Coordination** seem hard to parse. After having a close look at the parse errors, we find that these two groups are typically multi-box graphs. For example, for the coordination sentence *Is this baby a he or a she?*, there are three NEGATION boxes introduced by *or*. However, all models fail to predict the correct number of boxes. Apart from structural complexity, because there are only 7 examples in the Miscellaneous group, one tiny error can make a big difference in the result.

Yao and Koller (2022) and Kim and Linzen (2020) mentioned that seq2seq models fail in structural generalization which means the model cannot derive meaningful representations from a new combination of learned structures (e.g., PP recursion). However, according to Table 6.12, more data used in training does lead to better structural prediction. This hypothesis gains credibility owing to the substantial presence of overlapping vocabulary and sentence structures observed across different partitions of the PMB datasets. Whether more data forces the seq2seq models to be more structure-aware or generalize structurally will be explored in future work.

6.3 Error Analysis

In this section, we analyze errors from two levels: macro-level and micro-level, where the former refers to the analysis based on various important aspects of DRGs and the latter focuses on the analysis of individual cases.

6.3.1 Macro Level

Table 6.13 presents a detailed overview of the AM-Parser's performance on the PMB5 test set. Notably, our findings on other splits and datasets, including PMB4, exhibit consistent trends. For the sake of brevity, we include a single table here, while additional results for other datasets can be referenced in Appendix A.

In the table, **No Discourse**, **No Operators**, **No Senses** evaluates the DRGs without considering discourse edges, operator edges, and sense number respectively. **Names, Negations, Discourses, Constants, Roles, Members,** and **Concepts** only evaluate the performance of the model in predicting named entities, negation edge, discourse edge, constant node, roles, box memberships, concept nodes respectively. More specifically, the evaluation counts how many target nodes/edges are correctly predicted. **Con_** metrics specify the performance of the model in predicting specific POS, i.e., noun, adjective, adverb, and verb, in concepts. The **_triple** metrics evaluate the prediction of specific types of triples. In other words, it evaluates the combination of nodes and edges.

As can be found from the table, our parser performs relatively worse in the `Names` category and `Discourse` category. For sense disambiguation, it is challenging for AM-Parser to disambiguate the senses of adjectives and verbs. This also explains why if we ignore the sense number of the node and reevaluate the output, the F score increases by 2%.

6.3.2 Micro-Level

We filter to 60 graph parses that have a SMATCH F score below 0.6. What seems very challenging to AM-Parser is proper nouns. 11/16 parses that have an F score below 0.5 contain at least one proper noun. Another challenge is the ellipsis. Different from AMR where ellipsis introduces reentrancies, in DRG, the omitted node is made explicit in the graph, which makes the node-token alignment challenging.

Some errors are caused by imperfection of the data, such as node-token alignment errors

Category	P	R	F
Smatch	85.8	85.7	85.7
No Roles	87.3	87.1	87.2
No Discourse	85.8	85.7	85.7
No Operators	85.8	85.6	85.7
No Senses	79.0	78.9	79.0
Names	80.1	77.9	79.0
Negation	98.8	82.5	89.9
Discourse	88.9	72.7	80.0
Roles	89.5	89.9	89.7
Ana	93.8	93.8	93.8
Members	97.6	98.0	97.8
Concepts	84.7	84.9	84.8
Con_noun	87.5	88.0	87.7
Con_adj	76.1	76.8	76.4
Con_adv	83.0	77.2	80.0
Con_verb	76.4	75.6	76.0
Roles_triple	81.7	81.6	81.7
Ana_triple	93.8	93.8	93.8
Names_triple	72.9	73.1	73.0
Members_triple	90.8	90.8	90.8
Operators_triple	89.9	89.2	89.5
Discourses_triple	86.2	71.9	78.4

Table 6.13: Evaluation results of AM-Parser+Dependency on PMB5 test split

and examples themselves. For example, in Figure 6.2, three NEGATION boxes are aligned with a meaningless comma. All rhetorical questions are annotated in this way. Given that in most cases, punctuations do not represent meaning, this special case of the comma is probably not learned by AM-Parser. There are also examples that are purely names or website addresses such as *Yedinstvo* or *The El Aqsa Intifadah* which stand as a single DRG.

```
              NEGATION <1            % ,
               NEGATION <1            %
flower.n.01                           % This flower
time.n.08      EQU now                % is
beautiful.a.01 AttributeOf -2 Time -1 % beautiful
               NEGATION <2            %
               NEGATION <1            % n’t
beautiful.a.01 Time +1 AttributeOf +2 %
time.n.08      EQU now                % is
entity.n.01                           % it?
```

Figure 6.2: SBN for the sentence *The flower is beautiful, isn’t it?*

6.4 Summary

In summary, in this chapter, we evaluate the compositional AM-Parser and its symbolic or seq2seq baseline models. We find that

- When data is limited, our AM-Parser+Dependency system outperforms most of the purely symbolic or purely neural seq2seq models in PMB dataset particularly in PMB5. It also shows robust performance in scope assignment, coreference resolution, and the parsing of reentrancies.
- Both our heuristics-based and dependency-based approach can effectively map the scope back and achieve competitive performance or even surpass the seq2seq models with larger training data.
- A compositional model shows more stable and better performance as well as generates more grammatical graphs when the graphs become more complex in size and structure.
- However, AM-Parser still faces challenges in parsing proper nouns, ellipsis, and complex multibox graphs.

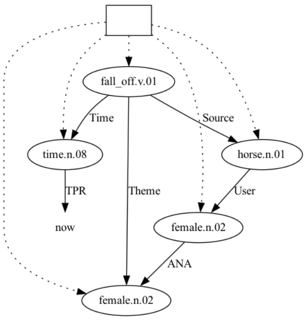
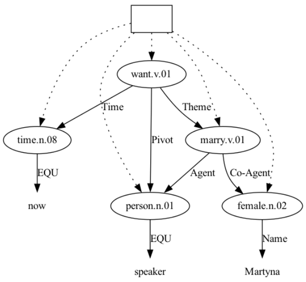
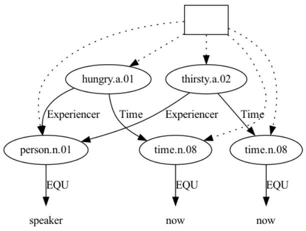
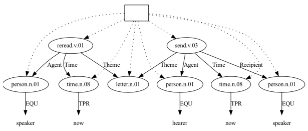
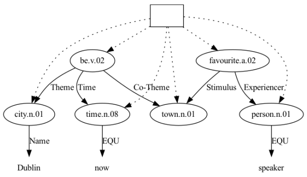
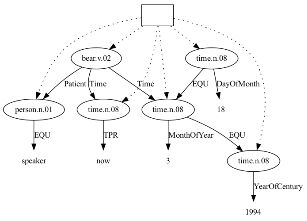
Category	Count	Example	DRG
Coreference	44	*She fell off her horse.*	 fall_off.v.01 Time Source time.n.08 horse.n.01 TPR Theme User now female.n.02 ANA female.n.02
Control	21	*I want to marry Martyna.*	 want.v.01 Time Theme time.n.08 Pivot marry.v.01 EQU Agent Co-Agent now person.n.01 female.n.02 EQU Name speaker Martyna
Coordination	21	*I'm hungry and thirsty.*	 hungry.a.01 thirsty.a.02 Experiencer Time Experiencer Time person.n.01 time.n.08 time.n.08 EQU EQU EQU speaker now now
Relative clause	8	*I was rereading the letters you sent to me.*	 reread.v.01 send.v.03 Agent Time Theme Theme Agent Time Recipient person.n.01 time.n.08 letter.n.01 person.n.01 time.n.08 person.n.01 EQU TPR EQU TPR EQU speaker now hearer now speaker
Verbalization	31	*Dublin is my favourite town.*	 be.v.02 favourite.a.02 Theme Time Co-Theme Stimulus Experiencer city.n.01 time.n.08 town.n.01 person.n.01 Name EQU EQU Dublin now speaker
Time	29	*I was born on 18th March 1994.*	 bear.v.02 time.n.08 Patient Time Time EQU DayOfMonth person.n.01 time.n.08 time.n.08 18 EQU TPR MonthOfYear EQU speaker now 3 time.n.08 YearOfCentury 1994

Table 6.14: Examples of different reentrancy types and their count in PMB5 dev and test set.

Chapter 7
Conclusion

In this chapter, we answer the research questions we raised and point out future directions for further research.

7.1 Overall Assessment

This work is the first attempt to parse DRGs compositionally with performance as competitive as other neural baselines. Before addressing our research questions, we summarize the strengths and weaknesses of our system. Our system demonstrates several notable advantages.

- It effectively handles both compositional and non-compositional aspects (i.e., scope and anaphora resolution) of DRS.
- When training data is limited, it consistently achieves competitive performance, often outperforming strong baselines in both the PMB4 and PMB5 datasets.
- It exhibits a remarkably low error rate.
- Furthermore, it showcases robust performance when parsing more complex, longer, or OOD inputs.

However, we must acknowledge certain limitations:

- It might not perform optimally with sentences containing proper nouns or ellipses.
- It relies on token-node alignment provided by PMB. In cases where SemBank lacks such annotations, they must be acquired before training.
- The system comprises two models: AM-Parser and Dozat and Manning (2018)'s biaffine dependency parser, which makes it somewhat cumbersome.

We list some possible solutions to some of the drawbacks mentioned above in Section 7.3.

7.2 Overall Conclusion

We now turn to answering the research questions we proposed in Chapter 1.

> ***Q1**: Regarding compositionality, how can DRGs be simplified to achieve compositionality by AM-Algebra while retaining linguistic knowledge and preserving essential structural information?*

To tackle the first question, we propose new simplified DRG formats, which we call Simplified DRG and Scopeless DRG. The former format keeps the necessary scope edges by assuming that children nodes inherit the scope information from their parents. The latter format only keeps the scope edges that connect isolate subgraphs. In other words, the second format keeps as few reentrancy edges introduced by scope as possible. The scope information of simplified DRG can mostly recovered by three simple rules.

To allow more graphs to be decomposable, we add a special token `START` to be aligned with the top box. We also reverse invertible edges and remove coreference edges to reduce reentrancy edges.

After the simplification of graphs, over 90% of the DRGs can be decomposed for training. They also contain the key semantic information of the graph. The removed edges can also be recovered after postprocessing.

> ***Q2**: Regarding non-compositionality, how to recover the non-decomposable information, more specifically, anaphora and scope assignment?*

Regarding coreference resolution, we incorporate this task into the training of AM-Parser. We introduce a new POS tag `p` to mark the antecedent and anaphor nodes. The F1 score for detecting them achieves over 93% in PMB5.

To resolve the scope assignment, we propose two approaches. The symbolic one follows the same assumption of DRG simplification. It only contains three rules, but it performs well overall. By contrast, the other approach relies on the parse result from a dependency parser. Since AM-Parser generates token-node alignment, the dependency parser predicts the relationship between tokens in terms of scope. We then map the scope back to DRGs by leveraging the two outputs. This approach works particularly well in multi-box DRGs. In both PMB data, the Smatch F score of scope of our system shows a competitive result as Neural-Boxer trained with silver and gold data (PMB4 65.4 vs. 69.1; PMB5 84.0 vs. 83.3).

> ***Q3**: How effectively does a compositional approach perform in comparison to its non-compositional neural or symbolic counterparts?*

We find that with limited training data, our parser performs well in both PMB4 and PMB5. It ranks second among all models trained with only gold data in PMB4 and beats all other parsers in PMB5. An interesting observation is that, despite PMB5 having a larger training set, most seq2seq models, with the exception of our compositional parser, exhibit comparatively weaker performance in PMB5. This discrepancy can be attributed to the structural complexity of PMB5. Our parser, however, demonstrates remarkable robustness and maintains high performance levels in the face of this increased structural complexity presented by PMB5. It even shows superior performance than Neural-Boxer trained with both silver and gold data.

Additionally, only our compositional parser performs relatively well when parsing longer input sequences. All seq2seq models, even including the SOTA model DRS-MLM, fail in parsing long sentences with a low error rate.

7.3 Directions for future work

There is still much space for improvement in our system. As we look ahead to future research endeavors, it is worth mentioning the following interesting directions.

Multilingual compositional DRS parsing PMB is a parallel multilingual corpus. Both PMB4 and PMB5 contain DRS representations in German, Dutch, and Italian. PMB5 also contains some examples in Chinese and Japanese. If our parser works well in English, it is also expected to work in other languages.

Joint learning Since AM-Parser contains a dependency parser by , it is possible to combine AM-tree dependency parsing and scope edge dependency parsing by joint learning. The dependency parser is required to generate both the dependency tree and the scope-related dependency graph. It would also be helpful to take advantage of an off-the-shelf NER tagger to help the model parse proper nouns.

Exploring Compositional Generalization of our system and its neural counterparts Dankers et al. (2022) find that neural networks exhibit increased compositional abilities with larger training datasets in machine translation tasks. However, the study also highlights the instability of model performance under these conditions. Building upon these findings, we find it interesting to investigate how the incorporation of silver and bronze data into the training process affects the performance and stability of seq2seq models in DRG parsing. To answer this question, a dataset to test compositional generalization is needed. Since PMB has a relatively small vocabulary, it is common to observe a significant overlap in terms of words across different sentences (e.g., *She is eight months pregnant* in the train split vs. *Mary is two months pregnant.* in the test split). The lexical similarity provides a favorable condition for designing a corpus to test structural complexity.

Bibliography

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

Abzianidze, L., J. Bjerva, K. Evang, H. Haagsma, R. van Noord, P. Ludmann, D.-D. Nguyen, and J. Bos (2017, April). The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain, pp. 242–247. Association for Computational Linguistics.

Abzianidze, L. and J. Bos (2017). Towards universal semantic tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Abzianidze, L., J. Bos, and S. Oepen (2020, November). DRS at MRP 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, Online, pp. 23–32. Association for Computational Linguistics.

Abzianidze, L., R. van Noord, H. Haagsma, and J. Bos (2019, May). The first shared task on discourse representation structure parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Agrawal, M., S. Hegselmann, H. Lang, Y. Kim, and D. Sontag (2022, December). Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, pp. 1998–2022. Association for Computational Linguistics.

Asher, N. (1993). *Reference to abstract objects in discourse*, Volume 50. Springer Science & Business Media.

Asher, N. and A. Lascarides (2003). *Logics of conversation*. Cambridge University Press.

Banarescu, L., C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider (2013). Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pp. 178–186.

Basile, V. and J. Bos (2013). Aligning formal meaning representations with surface strings for wide-coverage text generation. In *ENLG 2013*.

Bender, E. M., D. Flickinger, S. Oepen, W. Packard, and A. Copestake (2015, April). Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, London, UK, pp. 239–249. Association for Computational Linguistics.

Blackburn, P. and J. Bos (2005). Representation and inference for natural language. *A first course in computational semantics. CSLI*.

Bos, J. (2008). Wide-coverage semantic analysis with Boxer. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pp. 277–286. College Publications.

Bos, J. (2015). Open-domain semantic parsing with boxer. In *Proceedings of the 20th nordic conference of computational linguistics (NODALIDA 2015)*, pp. 301–304.

Bos, J. (2021). Variable-free discourse representation structures. *Semantics Archive*.

Bos, J. (2023). The sequence notation: Catching complex meanings in simple graphs. In *Proceedings of the 15th International Conference on Computational Semantics (IWCS 2023)*, Nancy, France, pp. 1–14.

Bos, J. et al. (2001). Doris 2001: Underspecification, resolution and inference for discourse representation structures. *ICoS-3, Inference in Computational Semantics*, 117–124.

Cai, S. and K. Knight (2013, August). Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Sofia, Bulgaria, pp. 748–752. Association for Computational Linguistics.

Courcelle, B. and J. Engelfriet (2012). *Graph structure and monadic second-order logic: a language-theoretic approach*, Volume 138. Cambridge University Press.

Damonte, M., S. B. Cohen, and G. Satta (2017, April). An incremental parser for Abstract Meaning Representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain, pp. 536–546. Association for Computational Linguistics.

Dankers, V., E. Bruni, and D. Hupkes (2022, May). The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, pp. 4154–4175. Association for Computational Linguistics.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.

Donatelli, L., J. Groschwitz, M. Lindemann, A. Koller, and P. Weißenhorn (2020, December). Normalizing compositional structures across graphbanks. In *Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain (Online), pp. 2991–3006. International Committee on Computational Linguistics.

Donatelli, L. and A. Koller (2023). Compositionality in computational linguistics. *Annual Review of Linguistics 9*, 463–481.

Dozat, T. and C. D. Manning (2017). Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Dozat, T. and C. D. Manning (2018, July). Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia, pp. 484–490. Association for Computational Linguistics.

Fancellu, F., S. Gilroy, A. Lopez, and M. Lapata (2019, November). Semantic graph parsing with recurrent neural network DAG grammars. 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)*, Hong Kong, China, pp. 2769–2778. Association for Computational Linguistics.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. MIT press.

Geurts, B., D. I. Beaver, and E. Maier (2020). Discourse Representation Theory. In E. N. Zalta (Ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2020 ed.). Metaphysics Research Lab, Stanford University.

Ghazarian, S., N. Wen, A. Galstyan, and N. Peng (2022, May). DEAM: Dialogue coherence evaluation using AMR-based semantic manipulations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, pp. 771–785. Association for Computational Linguistics.

Groschwitz, J. (2019). *Methods for taking semantic graphs apart and putting them back together again*. Ph. D. thesis, Saarländische Universitäts-und Landesbibliothek.

Groschwitz, J., M. Fowlie, M. Johnson, and A. Koller (2017). A constrained graph algebra for semantic parsing with amrs. In *IWCS 2017-12th International Conference on Computational Semantics-Long papers*.

Groschwitz, J., M. Fowlie, and A. Koller (2021, August). Learning compositional structures for semantic graph parsing. In *Proceedings of the 5th Workshop on Structured Prediction for NLP (SPNLP 2021)*, Online, pp. 22–36. Association for Computational Linguistics.

Groschwitz, J., M. Lindemann, M. Fowlie, M. Johnson, and A. Koller (2018, July). AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 1831–1841. Association for Computational Linguistics.

Haug, D. T. T. (2014). Partial dynamic semantics for anaphora: Compositionality without syntactic coindexation. *Journal of Semantics 31*(4), 457–511.

Heim, I. and A. Kratzer (1998). *Semantics in Generative Grammar*. Number 13 in Blackwell Textbooks in Linguistics. Oxford: Blackwell.

Hupkes, D., V. Dankers, M. Mul, and E. Bruni (2020). Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research 67*, 757–795.

Ivanova, A., S. Oepen, L. Øvrelid, and D. Flickinger (2012, July). Who did what to whom? a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, Jeju, Republic of Korea, pp. 2–11. Association for Computational Linguistics.

Janssen, T. M. and B. H. Partee (1997). Compositionality. In *Handbook of logic and language*, pp. 417–473. Elsevier.

Johnson, M. and E. Klein (1986). Discourse, anaphora and parsing. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

Kamath, A. and R. Das (2018). A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Kamp, H. (1981). Evénements, représentations discursives et référence temporelle. *Langages* (64), 39–64.

Kamp, H. (1981/2013). A theory of truth and semantic representation. In *Meaning and the Dynamics of Interpretation*, pp. 329–369. Brill.

Kamp, H. and U. Reyle (2013). *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, Volume 42. Springer Science & Business Media.

Kamp, H., U. Reyle, H. Kamp, and U. Reyle (1993). Tense and aspect. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, 483–689.

Kamp, H., J. Van Genabith, and U. Reyle (2010). Discourse representation theory. In *Handbook of Philosophical Logic: Volume 15*, pp. 125–394. Springer.

Karpinska, M. and M. Iyyer (2023). Large language models effectively leverage document-level context for literary translation, but critical errors persist. *preprint arXiv: 2304.03245*.

Kasper, R. T. (1989). A flexible interface for linking applications to Penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Kim, N. and T. Linzen (2020, November). COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp. 9087–9105. Association for Computational Linguistics.

Kipper, K., A. Korhonen, N. Ryant, and M. Palmer (2008). A large-scale classification of english verbs. *Language Resources and Evaluation 42*, 21–40.

Lascarides, A. and N. Asher (2007). Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pp. 87–124. Springer.

Le, P. and W. Zuidema (2012, December). Learning compositional semantics for open domain semantic parsing. In *Proceedings of COLING 2012*, Mumbai, India, pp. 1535–1552. The COLING 2012 Organizing Committee.

Lindemann, M., J. Groschwitz, and A. Koller (2019, July). Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 4576–4585. Association for Computational Linguistics.

Liu, J., S. B. Cohen, and M. Lapata (2018, July). Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 429–439. Association for Computational Linguistics.

Liu, J., S. B. Cohen, and M. Lapata (2019, May). Discourse representation structure parsing with recurrent neural networks and the transformer model. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Liu, J., S. B. Cohen, and M. Lapata (2020, July). Dscorer: A fast evaluation metric for discourse representation structure parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 4547–4554. Association for Computational Linguistics.

Liu, J., S. B. Cohen, and M. Lapata (2021, June). Text generation from discourse representation structures. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, pp. 397–415. Association for Computational Linguistics.

Liu, J., S. B. Cohen, M. Lapata, and J. Bos (2021). Universal discourse representation structure parsing. *Computational Linguistics 47*(2), 445–476.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR abs/1907.11692*.

Luong, T., H. Pham, and C. D. Manning (2015, September). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, pp. 1412–1421. Association for Computational Linguistics.

Matthiessen, C. M. and J. A. Bateman (1991). *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publisher.

Nguyen, M. V., V. D. Lai, A. Pouran Ben Veyseh, and T. H. Nguyen (2021, April). Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Online, pp. 80–90. Association for Computational Linguistics.

Oepen, S. and J. T. Lønning (2006, May). Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Opitz, J. (2023, May). SMATCH++: Standardized and extended evaluation of semantic graphs. In *Findings of the Association for Computational Linguistics: EACL 2023*, Dubrovnik, Croatia, pp. 1595–1607. Association for Computational Linguistics.

Opitz, J. and A. Frank (2022, November). Better Smatch = better parser? AMR evaluation is not so simple anymore. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, Online, pp. 32–43. Association for Computational Linguistics.

Pagin, P. and D. Westerståhl (2010). Compositionality i: Definitions and variants. *Philosophy Compass 5*(3), 250–264.

Pagin, P. and D. Westerståhl (2019). Compositionality. *Semantics: foundations, history and methods. De Gruyter Mouton, Berlin*, 122–155.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018, June). 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)*, New Orleans, Louisiana, pp. 2227–2237. Association for Computational Linguistics.

Poelman, W., R. van Noord, and J. Bos (2022). Transparent semantic parsing with universal dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 4186–4192.

Power, R. (1999). Controlling logical scope in text generation. In *Proceedings of the European Workshop on Natural Language Generation, Toulouse, France*, Volume 1.

Qi, P., Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning (2020, July). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online, pp. 101–108. Association for Computational Linguistics.

Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21*(140), 1–67.

Schuster, S., É. V. de La Clergerie, M. D. Candito, B. Sagot, C. D. Manning, and D. Seddah (2017). Paris and stanford at epe 2017: Downstream evaluation of graph-based dependency representations. In *EPE 2017-The First Shared Task on Extrinsic Parser Evaluation*, pp. 47–59.

Shaw, P., M.-W. Chang, P. Pasupat, and K. Toutanova (2021, August). Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Online, pp. 922–938. Association for Computational Linguistics.

Song, L., D. Gildea, Y. Zhang, Z. Wang, and J. Su (2019). Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics 7*, 19–31.

Steedman, M. (2001). *The syntactic process*. MIT press.

Szubert, I., M. Damonte, S. B. Cohen, and M. Steedman (2020, November). The role of reentrancies in Abstract Meaning Representation parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online, pp. 2198–2207. Association for Computational Linguistics.

Tedeschi, S., J. Bos, T. Declerck, J. Hajič, D. Hershcovich, E. Hovy, A. Koller, S. Krek, S. Schockaert, R. Sennrich, E. Shutova, and R. Navigli (2023, July). What's the meaning of superhuman performance in today's NLU? In *Proceedings of the 61st Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), Toronto, Canada, pp. 12471–12491. Association for Computational Linguistics.

Van der Sandt, R. A. (1992). Presupposition projection as anaphora resolution. *Journal of semantics 9*(4), 333–377.

van Noord, R., L. Abzianidze, H. Haagsma, and J. Bos (2018, May). Evaluating scoped meaning representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Van Noord, R., L. Abzianidze, A. Toral, and J. Bos (2018). Exploring neural methods for parsing discourse representation structures. *Transactions of the Association for Computational Linguistics 6*, 619–633.

Van Noord, R. and J. Bos (2017). Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *arXiv preprint arXiv:1705.09980*.

van Noord, R., A. Toral, and J. Bos (2019, May). Linguistic information in neural semantic parsing with multiple encoders. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, Gothenburg, Sweden, pp. 24–31. Association for Computational Linguistics.

van Noord, R., A. Toral, and J. Bos (2020, November). Character-level representations improve DRS-based semantic parsing even in the age of BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp. 4587–4603. Association for Computational Linguistics.

Wada, H. and N. Asher (1986). Buildrs: An implementation of DR theory and LFG. In *Coling 1986 Volume 1: The 11th International Conference on Computational Linguistics*.

Wang, C., H. Lai, M. Nissim, and J. Bos (2023, July). Pre-trained language-meaning models for multilingual parsing and generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada, pp. 5586–5600. Association for Computational Linguistics.

Wang, C., X. Zhang, and J. Bos (2023). Discourse representation structure parsing for Chinese. *arXiv preprint arXiv:2306.09725*.

Wang, Y., Y. Zhao, and L. Petzold (2023). Are large language models ready for healthcare? a comparative study on clinical language understanding. *preprint arXiv: 2304.05368*.

Weißenhorn, P., L. Donatelli, and A. Koller (2022, July). Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, Seattle, Washington, pp. 44–54. Association for Computational Linguistics.

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45.

Yao, Y. and A. Koller (2022, December). Structural generalization is hard for sequence-to-sequence models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Abu Dhabi, United Arab Emirates, pp. 5048–5062. Association for Computational Linguistics.

Žabokrtský, Z., D. Zeman, and M. Ševčíková (2020, September). Sentence meaning representations across languages: What can we learn from existing frameworks? *Computational Linguistics 46*(3), 605–665.

Zhang, C., S. Bauer, P. Bennett, J. Gao, W. Gong, A. Hilmkil, J. Jennings, C. Ma, T. Minka, N. Pawlowski, and J. Vaughan (2023). Understanding causality with large language models: Feasibility and opportunities. *preprint arXiv: 2304.05524*.

Appendices

Appendix A
Evaluation Results of All Experiments

A.1 Results of PMB4

This section reports the result details of experiments on PMB4.

Metric	Dev			Test			Eval		
	P	R	F	P	R	F	P	R	F
Smatch	86.9	85.4	86.2	86.4	86.2	86.3	84.5	84.1	84.3
No Roles	88.1	86.6	87.4	87.7	87.6	87.7	86.1	85.7	85.9
No Discourse	86.9	85.4	86.2	86.4	86.2	86.3	84.5	84.1	84.3
No Operators	86.9	85.5	86.2	86.5	86.3	86.4	84.6	84.2	84.4
No Senses	79.3	77.9	78.6	79.1	78.9	79.0	77.7	77.3	77.5
Names	90.7	89.2	89.9	90.8	89.5	90.1	88.0	88.2	88.1
Negation	97.1	89.8	93.3	94.2	85.3	89.5	89.6	92.6	91.1
Discourse	92.2	74.0	82.1	85.3	77.3	81.1	84.7	62.5	71.9
Roles	90.6	89.8	90.2	90.1	89.9	90.0	87.9	88.2	88.0
Members	98.3	97.4	97.8	97.4	98.4	97.9	97.2	97.9	97.5
Concepts	86.0	84.9	85.5	86.0	86.5	86.2	83.3	83.4	83.3
Con_noun	90.4	89.4	89.9	89.5	90.5	90.0	87.1	87.5	87.3
Con_adj	78.8	77.7	78.3	80.1	80.4	80.3	74.3	74.6	74.4
Con_adv	90.5	90.5	90.5	80.6	79.4	80.0	83.8	83.8	83.8
Con_verb	71.6	70.3	70.9	74.8	73.6	74.2	71.6	70.8	71.2
Roles_triple	82.3	81.4	81.9	82.8	82.8	82.8	79.3	79.7	79.5
Names_triple	88.0	88.1	88.0	85.9	86.3	86.1	82.8	84.5	83.6
Members_triple	91.8	90.7	91.2	91.4	91.8	91.6	89.7	89.9	89.8
Operators_triple	92.0	87.3	89.6	90.7	88.2	89.4	90.7	86.0	88.3
Discourses_triple	90.0	72.7	80.4	78.9	71.7	75.1	80.4	60.7	69.2

Table A.1: SMATCH Scores for AM-Parser+Dependency trained on scopeless DRGs on PMB4

Metric	Dev			Test			Eval		
	P	R	F	P	R	F	P	R	F
Smatch	86.5	85.0	85.7	85.9	85.7	85.8	84.1	83.7	83.9
No Roles	87.7	86.2	87.0	87.3	87.1	87.2	85.7	85.3	85.5
No Discourse	86.5	85.0	85.7	85.9	85.7	85.8	84.1	83.7	83.9
No Operators	86.5	85.0	85.8	86.0	85.8	85.9	84.1	83.8	83.9
No Senses	78.8	77.5	78.1	78.6	78.5	78.5	77.5	77.1	77.3
Names	90.7	89.2	89.9	90.8	89.5	90.1	88.0	88.2	88.1
Negation	97.1	89.8	93.3	94.2	85.3	89.5	89.6	92.6	91.1
Discourse	92.2	74.0	82.1	85.3	77.3	81.1	84.7	62.5	71.9
Roles	90.6	89.8	90.2	90.1	89.9	90.0	87.9	88.2	88.0
Members	98.3	97.4	97.8	97.4	98.4	97.9	97.2	97.9	97.5
Concepts	86.0	84.9	85.5	86.0	86.5	86.2	83.3	83.4	83.3
Con_noun	90.4	89.4	89.9	89.5	90.5	90.0	87.1	87.5	87.3
Con_adj	78.8	77.7	78.3	80.1	80.4	80.3	74.3	74.6	74.4
Con_adv	90.5	90.5	90.5	80.6	79.4	80.0	83.8	83.8	83.8
Con_verb	71.6	70.3	70.9	74.8	73.6	74.2	71.6	70.8	71.2
Roles_triple	82.4	81.6	82.0	82.7	82.6	82.6	79.3	79.7	79.5
Names_triple	87.5	87.5	87.5	86.0	86.4	86.2	81.7	83.5	82.6
Members_triple	90.6	88.8	89.7	90.4	90.0	90.2	88.7	88.1	88.4
Operators_triple	92.0	87.3	89.6	91.0	88.5	89.8	90.7	86.0	88.3
Discourses_triple	90.0	72.7	80.4	78.9	71.7	75.1	80.9	61.0	69.5

Table A.2: SMATCH Scores for AM-Parser+Heurestics trained on scopeless DRGs on PMB4

Metric	Dev			Test			Eval		
	P	R	F	P	R	F	P	R	F
Smatch	85.5	83.9	84.7	85.7	84.7	85.2	85.2	83.7	84.4
No Roles	86.9	85.3	86.1	87.0	86.0	86.5	86.7	85.2	85.9
No Discourse	85.5	83.9	84.7	85.7	84.7	85.2	85.1	83.7	84.4
No Operators	85.5	84.0	84.8	85.8	84.8	85.3	85.2	83.8	84.5
No Senses	78.2	76.9	77.5	78.5	77.5	78.0	78.2	76.9	77.5
Names	88.4	88.8	88.6	89.8	89.0	89.4	92.8	89.7	91.2
Negation	97.4	59.4	73.8	92.4	57.1	70.5	92.4	70.2	79.8
Discourse	84.3	44.8	58.5	84.1	49.3	62.2	78.7	46.3	58.3
Roles	88.8	89.6	89.2	89.4	89.1	89.3	87.5	87.9	87.7
Members	97.7	97.6	97.6	97.2	97.8	97.5	97.7	97.4	97.5
Concepts	84.8	84.3	84.5	85.4	85.5	85.4	84.8	84.0	84.4
Con_noun	89.0	88.7	88.9	89.2	89.8	89.5	89.1	88.0	88.6
Con_adj	79.6	77.5	78.5	79.4	79.4	79.4	76.0	76.3	76.2
Con_adv	84.7	85.7	85.2	80.3	83.8	82.0	81.9	86.8	84.3
Con_verb	70.6	69.6	70.1	72.8	71.4	72.1	71.7	70.8	71.3
Roles_triple	80.9	80.8	80.9	81.8	81.8	81.8	80.7	81.0	80.8
Names_triple	87.2	87.0	87.1	88.0	87.5	87.8	88.0	87.9	88.0
Members_triple	90.3	88.8	89.5	90.6	90.3	90.5	90.5	89.9	90.2
Operators_triple	91.7	86.6	89.1	90.9	89.2	90.0	91.4	86.3	88.8
Discourses_triple	89.2	72.7	80.1	89.6	72.0	80.2	89.3	71.9	80.3

Table A.3: SMATCH Scores for AM-Parser+Heurestics trained on simplified DRGs on PMB4

Metric	Dev			Test			Eval		
	P	R	F	P	R	F	P	R	F
Smatch	85.6	84.1	84.8	85.7	84.7	85.2	85.0	83.6	84.3
No Roles	87.0	85.5	86.3	87.0	85.9	86.5	86.6	85.1	85.8
No Discourse	85.6	84.1	84.8	85.7	84.7	85.2	85.0	83.6	84.3
No Operators	85.7	84.2	84.9	85.8	84.8	85.3	85.1	83.7	84.4
No Senses	78.4	77.0	77.7	78.5	77.5	78.0	78.1	76.8	77.4
Names	88.4	88.8	88.6	89.8	89.0	89.4	92.8	89.7	91.2
Negation	97.4	59.4	73.8	92.4	57.1	70.5	92.4	70.2	79.8
Discourse	84.3	44.8	58.5	84.1	49.3	62.2	78.7	46.3	58.3
Roles	88.8	89.6	89.2	89.4	89.1	89.3	87.5	87.9	87.7
Members	97.7	97.6	97.6	97.2	97.8	97.5	97.7	97.4	97.5
Concepts	84.8	84.3	84.5	85.4	85.5	85.4	84.8	84.0	84.4
Con_noun	89.0	88.7	88.9	89.2	89.8	89.5	89.1	88.0	88.6
Con_adj	79.6	77.5	78.5	79.4	79.4	79.4	76.0	76.3	76.2
Con_adv	84.7	85.7	85.2	80.3	83.8	82.0	81.9	86.8	84.3
Con_verb	70.6	69.6	70.1	72.8	71.4	72.1	71.7	70.8	71.3
Roles_triple	80.9	80.8	80.9	81.8	81.8	81.8	80.7	81.0	80.8
Names_triple	85.8	87.4	86.6	85.2	86.5	85.8	87.6	86.2	86.9
Members_triple	90.7	90.0	90.4	90.7	90.7	90.7	90.3	89.6	90.0
Operators_triple	90.9	86.4	88.6	90.8	87.8	89.3	92.0	86.3	89.1
Discourses_triple	83.0	44.9	58.2	78.8	45.7	57.9	76.9	46.9	58.2

Table A.4: SMATCH Scores for AM-Parser+Dependency trained on simplified DRGs on PMB4

	dev			test			eval		
	P	R	F	P	R	F	P	R	F
Smatch	83.6	73.9	78.4	84.0	75.2	79.3	81.0	70.7	75.5
No Roles	85.6	75.7	80.4	86.0	77.0	81.3	83.2	72.6	77.5
No Discourse	83.6	73.9	78.4	83.9	75.2	79.3	81.0	70.7	75.5
No Operators	83.7	74.0	78.6	84.2	75.4	79.6	81.3	71.0	75.8
No Senses	83.2	73.5	78.0	83.6	74.9	79.0	80.4	70.2	74.9
Names	86.8	73.8	79.8	82.6	75.2	78.7	87.1	71.3	78.4
Negation	88.0	66.8	76.0	86.9	70.0	77.5	75.0	69.4	72.1
Discourse	87.3	57.3	69.2	77.2	58.7	66.7	72.7	50.0	59.3
Roles	85.8	74.5	79.8	87.5	76.7	81.8	83.6	70.7	76.6
Members	98.2	86.1	91.8	98.9	87.7	93.0	98.0	84.4	90.7
Concepts	65.9	57.8	61.6	67.4	59.8	63.4	63.7	54.9	59.0
Con_noun	75.6	66.7	70.9	76.2	67.8	71.8	72.6	62.9	67.4
Con_adj	26.5	23.9	25.1	34.9	29.9	32.2	28.0	24.1	25.9
Con_adv	68.1	56.0	61.4	69.1	55.9	61.8	56.1	47.1	51.2
Con_verb	43.2	36.5	39.6	44.5	39.5	41.9	43.0	36.4	39.4
Roles_triple	68.8	60.4	64.3	70.5	62.4	66.2	66.6	57.3	61.6
Names_triple	81.1	69.3	74.8	74.0	67.4	70.5	77.4	64.2	70.2
Members_triple	84.7	74.6	79.3	85.5	76.4	80.7	83.0	72.2	77.2
Operators_triple	93.7	83.2	88.1	94.3	85.8	89.9	91.4	81.2	86.0
Discourses_triple	84.5	55.9	67.3	75.0	57.3	65.0	70.0	50.5	58.7

Table A.5: SMATCH Scores for Neural-Boxer trained on gold SBNs in PMB4

	dev			test			eval		
	P	R	F	P	R	F	P	R	F
Smatch	75.1	71.8	73.4	75.6	72.1	73.8	74.5	70.6	72.5
No Roles	80.1	76.6	78.3	80.8	77.1	78.9	79.7	75.5	77.5
No Discourse	75.1	71.8	73.4	75.6	72.1	73.8	74.5	70.6	72.5
No Operators	76.0	72.6	74.2	76.3	72.8	74.5	75.3	71.3	73.2
No Senses	74.3	71.0	72.6	74.7	71.2	72.9	73.8	69.9	71.8
Names	14.1	12.5	13.2	16.7	14.9	15.8	14.7	12.4	13.4
Negation	90.9	85.6	88.2	94.9	75.9	84.3	88.4	81.8	85.0
Discourse	75.0	6.2	11.5	100.0	6.7	12.5	91.7	13.8	23.9
Roles	63.0	58.1	60.5	63.5	58.1	60.7	62.6	57.0	59.7
Members	94.5	92.2	93.4	94.5	92.4	93.4	94.7	91.9	93.3
Concepts	67.4	65.8	66.6	68.1	66.6	67.4	67.1	65.1	66.1
Con_noun	71.7	68.3	70.0	72.4	69.7	71.0	72.0	67.8	69.9
Con_adj	58.8	67.0	62.7	58.9	69.4	63.7	56.9	65.3	60.8
Con_adv	34.8	64.3	45.2	26.7	51.5	35.2	31.6	52.9	39.6
Con_verb	60.3	55.8	58.0	61.2	55.3	58.1	58.7	55.5	57.1
Roles_triple	59.1	57.4	58.3	59.4	57.4	58.4	58.3	56.1	57.2
Names_triple	37.5	33.7	35.5	38.6	35.2	36.8	37.6	32.5	34.9
Members_triple	81.4	78.4	79.9	81.9	79.3	80.6	81.1	77.8	79.4
Operators_triple	78.0	77.4	77.7	80.0	79.7	79.9	79.1	77.6	78.3
Discourses_triple	68.8	5.9	10.9	100.0	6.8	12.8	89.6	14.1	24.4

Table A.6: SMATCH Scores for UD-Boxer in PMB4

	dev			test			eval		
	P	R	F	P	R	F	P	R	F
Smatch	92.3	87.1	89.6	91.3	86.0	88.6	92.6	88.3	90.4
No Roles	93.4	88.0	90.6	92.7	87.3	89.9	93.6	89.3	91.4
No Discourse	92.3	87.1	89.6	91.3	86.0	88.6	92.5	88.3	90.4
No Operators	92.4	87.1	89.7	91.4	86.0	88.6	92.6	88.4	90.5
No Senses	92.6	87.3	89.8	91.5	86.2	88.8	92.7	88.5	90.5
Names	91.7	87.5	89.6	95.0	86.4	90.5	92.5	89.0	90.7
Negation	98.6	74.9	85.1	94.8	90.9	92.8	98.6	82.9	90.1
Discourse	95.5	87.5	91.3	92.5	77.5	84.4	93.2	90.7	91.9
Roles	92.5	86.8	89.6	91.8	85.3	88.5	93.7	88.1	90.8
Members	98.9	93.1	95.9	98.7	92.5	95.5	99.1	94.5	96.8
Concepts	86.4	81.3	83.8	85.2	79.8	82.4	86.8	82.8	84.7
Con_noun	92.4	86.8	89.5	90.4	84.6	87.4	92.8	88.6	90.6
Con_adj	77.0	73.7	75.3	77.4	74.2	75.8	73.8	73.1	73.5
Con_adv	86.4	83.3	84.8	89.8	77.9	83.5	91.2	76.5	83.2
Con_verb	67.2	62.9	65.0	67.7	63.4	65.5	68.2	64.6	66.4
Roles_triple	85.1	79.9	82.4	84.0	78.5	81.2	85.9	81.2	83.5
Names_triple	90.4	86.6	88.5	91.7	84.0	87.7	88.4	85.1	86.7
Members_triple	93.7	88.2	90.8	93.0	87.4	90.1	94.0	89.7	91.8
Operators_triple	96.4	90.5	93.3	95.7	90.6	93.0	95.9	91.9	93.9
Discourses_triple	95.3	88.6	91.9	92.0	78.7	84.8	92.7	90.8	91.7

Table A.7: SMATCH Scores for T5 trained on gold SBNs on PMB4

	dev			test			eval		
	P	R	F	P	R	F	P	R	F
Smatch	92.3	89.1	90.7	92.6	88.8	90.6	91.6	86.9	89.2
No Roles	93.4	90.1	91.7	93.7	89.9	91.8	93.0	88.2	90.6
No Discourse	92.3	89.1	90.7	92.6	88.8	90.6	91.6	86.9	89.2
No Operators	92.4	89.1	90.7	92.6	88.9	90.7	91.7	87.0	89.3
No Senses	92.4	89.1	90.7	92.6	88.8	90.6	91.6	86.9	89.2
Names	90.9	85.0	87.9	91.0	82.4	86.5	93.3	84.6	88.7
Negation	98.3	92.5	95.3	97.5	90.0	93.6	96.4	88.4	92.2
Discourse	95.6	89.6	92.5	91.7	88.0	89.8	93.1	83.8	88.2
Roles	92.5	88.7	90.5	93.0	88.0	90.5	91.5	85.6	88.5
Members	99.0	95.1	97.0	99.2	94.9	97.0	98.9	93.6	96.2
Concepts	84.9	81.6	83.3	86.5	82.8	84.6	84.4	79.8	82.0
Con_noun	90.3	87.1	88.7	91.3	87.3	89.3	89.8	84.9	87.3
Con_adj	72.3	67.3	69.7	76.4	74.1	75.2	70.4	66.3	68.3
Con_adv	87.5	75.0	80.8	85.2	76.5	80.6	93.0	77.9	84.8
Con_verb	68.9	66.9	67.9	72.0	69.0	70.5	68.2	65.3	66.7
Roles_triple	85.4	81.9	83.6	86.3	82.3	84.2	84.2	79.5	81.8
Names_triple	89.1	84.0	86.5	87.0	79.3	83.0	89.9	82.4	86.0
Members_triple	93.3	89.7	91.5	94.0	90.1	92.0	92.9	88.1	90.4
Operators_triple	96.4	93.2	94.8	96.3	92.7	94.5	96.8	91.5	94.1
Discourses_triple	96.3	90.5	93.3	91.8	88.1	89.9	94.1	87.2	90.5

Table A.8: SMATCH Scores for Neural-Boxer trained on gold and silver SBNs on PMB4

A.2 Results of PMB5

This section reports the result details of experiments on PMB5. The tables here are larger than the ones in the previous section because they also contain evaluation of the anaphora pairs.

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	86.7	86.4	86.6	85.6	85.4	85.5	48.2	42.0	44.9
No Roles	87.9	87.6	87.8	87.0	86.8	86.9	50.7	44.2	47.2
No Discourse	86.7	86.4	86.6	85.6	85.4	85.5	48.3	42.0	44.9
No Operators	86.7	86.4	86.5	85.5	85.3	85.4	48.3	42.0	45.0
No Senses	80.1	79.8	79.9	78.7	78.5	78.6	47.6	41.4	44.3
Names	85.7	82.8	84.2	80.1	77.9	79.0	-	-	-
Negation	98.1	92.0	95.0	98.8	82.5	89.9	-	-	-
Discourse	90.9	82.4	86.4	88.9	72.7	80.0	-	-	-
Roles	91.6	90.8	91.2	89.5	89.9	89.7	-	-	-
Ana	93.9	93.9	93.9	93.8	93.8	93.8	-	-	-
Members	97.8	98.0	97.9	97.6	98.0	97.8	-	-	-
Concepts	86.2	86.0	86.1	84.7	84.9	84.8	-	-	-
Con_noun	89.4	89.5	89.5	87.5	88.0	87.7	-	-	-
Con_adj	76.1	75.2	75.6	76.1	76.8	76.4	-	-	-
Con_adv	84.7	76.9	80.6	83.0	77.2	80.0	-	-	-
Con_verb	76.2	75.8	76.0	76.4	75.6	76.0	-	-	-
Roles_triple	84.0	83.6	83.8	81.7	81.5	81.6	48.4	42.5	45.2
Ana_triple	93.9	93.9	93.9	93.8	93.8	93.8	100.0	28.8	44.8
Names_triple	78.8	78.4	78.6	72.9	73.2	73.0	35.3	28.2	31.4
Members_triple	90.3	89.5	89.9	90.3	89.7	90.0	59.6	51.6	55.3
Operators_triple	90.3	90.1	90.2	89.4	88.7	89.0	72.8	62.5	67.3
Discourses_triple	90.6	83.9	87.1	86.2	71.9	78.4	66.1	32.7	43.7

Table A.9: SMATCH Scores for AM-Parser+Heurestics trained on scopeless DRGs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	86.9	86.3	86.6	85.7	84.7	85.2	44.7	36.5	40.2
No Roles	88.1	87.5	87.8	87.0	85.9	86.5	48.0	39.1	43.1
No Discourse	86.9	86.3	86.6	85.7	84.7	85.2	44.8	36.6	40.3
No Operators	86.9	86.3	86.6	85.7	84.6	85.1	45.4	37.0	40.8
No Senses	80.1	79.6	79.9	78.9	78.0	78.4	44.1	36.0	39.7
Names	82.6	79.2	80.8	82.6	79.2	80.8	-	-	-
Negation	98.3	57.0	72.2	98.3	57.0	72.2	-	-	-
Discourse	78.1	64.9	70.9	78.1	64.9	70.9	-	-	-
Roles	90.4	89.6	90.0	90.4	89.6	90.0	-	-	-
Ana	100.0	93.8	96.8	100.0	93.8	96.8	-	-	-
Members	96.9	97.0	96.9	96.9	97.0	96.9	-	-	-
Concepts	85.2	84.8	85.0	85.2	84.8	85.0	-	-	-
Con_noun	87.9	88.1	88.0	87.9	88.1	88.0	-	-	-
Con_adj	76.2	76.5	76.4	76.2	76.5	76.4	-	-	-
Con_adv	80.7	80.7	80.7	80.7	80.7	80.7	-	-	-
Con_verb	77.8	75.1	76.4	77.8	75.1	76.4	-	-	-
Con_pron	0.0	0.0	0.0	0.0	0.0	0.0	-	-	-
Roles_triple	82.2	81.0	81.6	82.2	81.0	81.6	45.4	38.3	41.6
Ana_triple	100.0	93.8	96.8	100.0	93.8	96.8	93.2	39.0	55.0
Names_triple	73.7	73.8	73.7	73.7	73.8	73.7	34.2	30.0	32.0
Members_triple	90.6	90.3	90.4	90.6	90.3	90.5	58.1	46.3	51.5
Operators_triple	89.7	88.9	89.3	89.7	88.9	89.3	64.0	53.5	58.3
Discourses_triple	79.1	66.8	72.4	79.1	66.8	72.4	55.1	15.6	24.3

Table A.10: SMATCH Scores for AM-Parser+Heurestics trained on simplified DRGs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	Dev			Test			Test Long		
Category	P	R	F	P	R	F	P	R	F
Smatch	87.3	87.0	87.2	85.8	85.7	85.7	48.6	40.5	44.1
No Roles	88.5	88.2	88.3	87.3	87.1	87.2	-	-	-
No Discourse	87.3	87.0	87.2	85.8	85.7	85.7	-	-	-
No Operators	87.2	86.9	87.1	85.8	85.6	85.7	-	-	-
No Senses	80.5	80.2	80.4	79.0	78.9	79.0	-	-	-
Names	85.7	82.8	84.2	80.1	77.9	79.0	-	-	-
Negation	98.1	92.0	95.0	98.8	82.5	89.9	-	-	-
Discourse	90.9	82.4	86.4	88.9	72.7	80.0	-	-	-
Roles	91.6	90.8	91.2	89.5	89.9	89.7	-	-	-
Ana	93.9	93.9	93.9	93.8	93.8	93.8	-	-	-
Members	97.8	98.0	97.9	97.6	98.0	97.8	-	-	-
Concepts	86.2	86.0	86.1	84.7	84.9	84.8	-	-	-
Con_noun	89.4	89.5	89.5	87.5	88.0	87.7	-	-	-
Con_adj	76.1	75.2	75.6	76.1	76.8	76.4	-	-	-
Con_adv	84.7	76.9	80.6	83.0	77.2	80.0	-	-	-
Con_verb	76.2	75.8	76.0	76.4	75.6	76.0	-	-	-
Roles_triple	83.8	83.4	83.6	81.7	81.6	81.7	48.5	40.7	44.3
Ana_triple	93.9	93.9	93.9	93.8	93.8	93.8	100.0	27.8	43.5
Names_triple	79.2	78.8	79.0	72.9	73.1	73.0	35.1	26.8	30.4
Members_triple	91.3	91.2	91.3	90.8	90.8	90.8	60.6	50.2	54.9
Operators_triple	90.2	90.0	90.1	89.9	89.2	89.5	73.0	60.3	66.0
Discourses_triple	90.6	83.9	87.1	86.2	71.9	78.4	65.9	31.3	42.4

Table A.11: SMATCH Scores for AM-Parser+Dependency trained on scopeless DRGs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	86.9	86.3	86.6	85.8	84.7	85.3	45.1	30.9	36.7
No Roles	88.1	87.4	87.7	84.0	87.2	85.6	-	-	-
No Discourse	86.9	86.3	86.6	85.8	84.8	85.3	-	-	-
No Operators	86.9	86.3	86.6	86.3	85.3	85.8	-	-	-
No Senses	80.2	79.7	80.0	78.9	77.9	78.4	-	-	-
Names	85.5	82.2	83.8	82.6	79.2	80.8	-	-	-
Negation	98.2	72.9	83.7	98.3	57.0	72.2	-	-	-
Discourse	86.5	75.3	80.5	78.1	64.9	70.9	-	-	-
Roles	91.2	90.6	90.9	90.4	89.6	90.0	-	-	-
Ana	94.3	100.0	97.1	100.0	93.8	96.8	-	-	-
Members	97.5	97.8	97.7	96.9	97.0	96.9	-	-	-
Concepts	86.1	86.0	86.1	85.2	84.8	85.0	-	-	-
Con_noun	88.8	88.9	88.9	87.9	88.1	88.0	-	-	-
Con_adj	76.6	76.1	76.3	76.2	76.5	76.4	-	-	-
Con_adv	83.1	75.4	79.0	80.7	80.7	80.7	-	-	-
Con_verb	78.4	77.8	78.1	77.8	75.1	76.4	-	-	-
Roles_triple	83.5	83.0	83.3	82.2	81.0	81.6	82.2	81.0	81.6
Ana_triple	94.3	100.0	97.1	100.0	93.8	96.8	100.0	27.8	30.4
Names_triple	79.4	78.4	78.9	74.2	74.4	74.3	35.1	26.8	30.4
Members_triple	91.3	91.2	91.2	90.7	90.4	90.5	60.6	50.2	54.9
Operators_triple	89.3	89.5	89.4	89.8	89.0	89.4	73.0	60.3	66.0
Discourses_triple	85.7	77.7	81.5	78.7	66.4	72.1	65.9	31.3	42.4

Table A.12: SMATCH Scores for AM-Parser+Dependency trained on simplified DRGs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	71.2	92.9	80.6	72.6	91.9	81.1	2.4	74.8	4.7
No Roles	72.0	93.9	81.5	73.5	93.0	82.1	-	-	-
No Discourse	71.2	92.9	80.6	72.6	91.9	81.1	-	-	-
No Operators	71.2	92.9	80.6	72.6	91.9	81.1	-	-	-
No Senses	73.0	95.3	82.7	74.3	94.0	83.0	-	-	-
Names	73.6	88.9	80.6	75.4	83.0	79.0	-	-	-
Negation	0.0	0.0	0.0	0.0	0.0	0.0	-	-	-
Discourse	0.0	0.0	0.0	0.0	0.0	0.0	-	-	-
Roles	72.2	93.3	81.4	74.2	92.2	82.3	-	-	-
Ana	63.6	95.5	76.4	68.8	68.8	68.8	-	-	-
Members	75.5	99.1	85.7	77.2	98.4	86.5	-	-	-
Concepts	66.4	87.2	75.4	67.9	86.6	76.1	-	-	-
Con_noun	70.9	92.5	80.3	71.8	90.7	80.2	-	-	-
Con_adj	56.0	72.1	63.0	63.9	80.4	71.2	-	-	-
Con_adv	60.0	79.6	68.4	61.4	87.5	72.2	-	-	-
Con_verb	52.0	70.0	59.7	54.4	71.9	61.9	-	-	-
Roles_triple	65.8	86.6	74.8	66.5	85.4	74.8	1.2	57.3	2.4
Ana_triple	60.6	90.9	72.7	67.2	67.2	67.2	0.0	0.0	0.0
Names_triple	74.4	89.7	81.4	76.9	84.6	80.5	1.7	46.6	3.3
Members_triple	70.9	94.1	80.9	72.8	93.7	81.9	1.8	79.2	3.4
Operators_triple	73.7	96.8	83.7	74.1	94.8	83.2	2.3	85.1	4.4
Discourses_triple	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table A.13: SMATCH Scores for T5-Boxer trained on gold SBNs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	72.6	83.6	77.7	70.6	82.1	75.9	8.8	59.7	15.4
No Roles	74.4	85.7	79.6	72.3	84.2	77.8	-	-	-
No Discourse	72.6	83.6	77.7	70.6	82.1	75.9	-	-	-
No Operators	72.8	83.8	77.9	70.7	82.3	76.1	-	-	-
No Senses	73.0	84.1	78.2	70.9	82.5	76.3	-	-	-
Names	62.7	77.5	69.3	59.5	74.7	66.2	-	-	-
Negation	80.4	93.3	86.4	73.0	91.8	81.3	-	-	-
Discourse	49.4	82.4	61.8	39.0	85.7	53.6	-	-	-
Roles	73.0	86.5	79.2	71.6	85.6	78.0	-	-	-
Ana	72.7	57.1	64.0	50.0	53.3	51.6	-	-	-
Members	84.5	98.7	91.0	83.0	98.1	89.9	-	-	-
Concepts	59.4	69.4	64.0	57.9	68.4	62.7	-	-	-
Con_noun	66.4	77.0	71.3	65.2	76.7	70.5	-	-	-
Con_adj	28.6	37.6	32.5	28.1	33.9	30.7	-	-	-
Con_adv	41.5	58.7	48.6	35.1	57.1	43.5	-	-	-
Con_verb	41.9	47.9	44.7	39.8	46.7	43.0	-	-	-
Roles_triple	60.8	71.2	65.6	58.2	68.9	63.1	7.0	50.5	12.3
Ana_triple	70.5	58.9	64.1	48.4	53.4	50.8	0.0	0.0	0.0
Names_triple	56.2	68.7	61.8	49.8	61.6	55.1	5.3	42.2	9.4
Members_triple	74.5	86.3	80.0	73.0	85.5	78.8	9.2	66.0	16.1
Operators_triple	81.8	91.8	86.5	80.1	90.1	84.8	9.8	72.4	17.2
Discourses_triple	49.8	79.3	61.2	39.3	82.9	53.3	15.2	69.2	24.9

Table A.14: SMATCH Scores for Neural-Boxer trained on gold SBNs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	81.9	89.1	85.4	79.1	91.1	84.7	5.5	60.0	15.4
No Roles	83.0	90.3	86.5	80.1	92.2	85.7	-	-	-
No Discourse	81.9	89.1	85.4	79.1	91.1	84.7	-	-	-
No Operators	82.0	89.2	85.5	79.2	91.2	84.7	-	-	-
No Senses	84.2	91.6	87.7	80.2	92.3	85.8	-	-	-
Names	53.9	66.9	59.7	61.7	85.6	71.7	-	-	-
Negation	86.2	97.5	91.5	84.0	91.8	87.7	-	-	-
Discourse	77.6	97.1	86.3	80.5	98.4	88.6	-	-	-
Roles	80.9	90.6	85.5	76.5	91.3	83.2	-	-	-
Ana	78.8	100	88.1	81.2	81.2	81.2	-	-	-
Members	89.9	98.2	93.9	84.0	98.3	90.6	-	-	-
Concepts	72.8	79.4	76.0	72.7	84.9	78.3	-	-	-
Con_noun	78.9	86.2	82.4	76.6	89.6	82.5	-	-	-
Con_adj	59.7	64.4	62.0	59.6	72.0	65.2	-	-	-
Con_adv	61.5	75.5	67.8	50.9	70.7	59.2	-	-	-
Con_verb	53.2	57.2	55.2	62.7	71.5	66.8	-	-	-
Roles_triple	72.4	79.9	76.0	72.1	84.4	77.8	3.4	33.0	6.2
Ana_triple	74.2	94.2	83.1	79.7	79.7	79.7	0.0	0.0	0.0
Names_triple	60.0	73.3	66.0	60.8	82.6	70.1	1.1	14.2	2.1
Members_triple	83.0	90.5	86.6	79.7	92.8	85.7	5.8	55.2	10.5
Operators_triple	86.2	92.0	89.0	82.3	94.5	88.0	6.6	61.1	11.9
Discourses_triple	75.5	96.4	84.7	79.7	97.1	87.5	10.7	65.0	18.3

Table A.15: SMATCH Scores for Neural-Boxer trained on gold+silver SBNs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

	dev			test			test_long		
	P	R	F	P	R	F	P	R	F
Smatch	90.4	94.7	92.5	88.7	94.4	91.5	5.5	81.6	10.2
No Roles	91.3	95.5	93.3	89.6	95.3	92.4	-	-	-
No Discourse	90.4	94.6	92.5	88.7	94.4	91.5	-	-	-
No Operators	90.4	94.7	92.5	88.8	94.4	91.5	-	-	-
No Senses	91.9	96.2	94.0	90.4	96.1	93.2	-	-	-
Names	88.3	92.0	90.1	83.7	88.4	86.0	-	-	-
Negation	92.4	99.5	95.9	85.5	100	92.2	-	-	-
Discourse	89.4	100	94.4	89.6	100	94.5	-	-	-
Roles	89.8	94.4	92.0	88.3	94.2	91.1	-	-	-
Ana	93.9	83.8	88.6	100	80.0	88.9	-	-	-
Members	94.6	99.2	96.9	93.2	99.4	96.2	-	-	-
Concepts	86.3	90.5	88.4	84.2	89.8	86.9	-	-	-
Con_noun	89.7	94.2	91.9	87.5	93.3	90.3	-	-	-
Con_adj	76.4	81.0	78.6	75.2	83.2	79.0	-	-	-
Con_adv	81.5	88.3	84.8	77.2	88.0	82.2	-	-	-
Con_verb	76.1	78.6	77.4	74.7	78.5	76.5	-	-	-
Roles_triple	86.0	90.1	88.0	83.4	88.8	86.0	3.7	66.9	7.0
Ana_triple	90.9	81.1	85.7	100	80.0	88.9	9.1	73.3	16.2
Names_triple	88.5	92.2	90.3	84.9	89.4	87.1	3.9	71.8	7.3
Members_triple	91.3	95.6	93.4	89.8	95.5	92.5	5.1	86.4	9.6
Operators_triple	92.4	97.7	95.0	90.4	97.4	93.7	4.7	79.5	8.9
Discourses_triple	89.2	99.7	94.1	88.8	99.2	93.7	8.7	97.9	16.0

Table A.16: SMATCH Scores for DRS-MLM trained on gold+silver+bronze SBNs in PMB5 (- means there is an error when running the evaluation script and the result is unavailable)

Appendix B

B.1 Hyperparameters in AM-Parser

The hyperparameters used in the experiments that show the best performance on the scopeless SBN training data are summarized in Table B.1.

Hyperparameter	**Value**
Activation function	tanh
Optimizer	Adam
Learning rate	0.001
Epochs	100
Early Stopping	20
Dim of lemma embeddings	64
Dim of POS embeddings	32
Dim of NE embeddings	16
Minimum lemma frequency	7
Hidden layers in all MLPs	1
Hidden units in LSTM (per direction)	256
Hidden units in edge existence MLP	256
Hidden units in edge label MLP	256
Hidden units in supertagger MLP	1024
Hidden units in lexical label tagger MLP	1024
Layer dropout in LSTMs	0.35
Recurrent dropout in LSTMs	0.4
Input dropout	0.35
Dropout in edge existence MLP	0.0
Dropout in edge label MLP	0.0
Dropout in supertagger MLP	0.4
Dropout in lexical label tagger MLP	0.4

Table B.1: Common hyperparameters used in all experiments in AM-Parser.

B.2 hyperparameters of (Dozat and Manning, 2018)

We adopt the same configuration of the model except we only take character and tags into embedding and we fine-tune a RoBERTa-large model instead of BiLSTM. The hyperparameter table below is copied from Dozat and Manning (2018) Table 2.

Parameter	**Value**
Hidden Sizes	
Word/Glove/POS/Lemma/Char	100
GloVe linear	125
Char LSTM	1 @ 400
Char linear	100
BiLSTM	3 @ 600
Arc/Label	600
Dropout Rates (drop prob)	
Word/GloVe/POS/Lemma	20%
Char LSTM (FF/recur)	33%
Char linear	33%
BiLSTM (FF/recur)	45%/25%
Arc/Label	25%/33%
Loss & Optimizer	
Interpolation (λ)	.025
L_2 regularization	$3e^{-9}$
Learning rate	$1e^{-3}$
Adam β_1	0
Adam β_2	.95

Table B.2: Final hyperparameter configuration in Dozat and Manning (2018).