Gaining More Insight into Neural Semantic Parsing with Challenging Benchmarks

Xiao Zhang, Chunliu Wang, Rik van Noord, Johan Bos
Center for Language and Cognition, University of Groningen
{xiao.zhang, chunliu.wang, r.i.k.van.noord, johan.bos}@rug.nl

Abstract

The Parallel Meaning Bank (PMB) serves as a corpus for semantic processing with a focus on semantic parsing and text generation. Currently, we witness an excellent performance of neural parsers and generators on the PMB. This might suggest that such semantic processing tasks have by and large been solved. We argue that this is not the case and that performance scores from the past on the PMB are inflated by non-optimal data splits and test sets that are too easy. In response, we introduce several changes. First, instead of the prior random split, we propose a more systematic splitting approach to improve the reliability of the standard test data. Second, except for the standard test set, we also propose two challenge sets: one with longer texts including discourse structure, and one that addresses compositional generalization. We evaluate five neural models for semantic parsing and meaning-to-text generation. Our results show that model performance declines (in some cases dramatically) on the challenge sets, revealing the limitations of neural models when confronting such challenges.

Keywords: Annotated Corpus, Discourse Representation Theory, Semantic Parsing, Text Generation

1. Introduction

The Parallel Meaning Bank (PMB, Abzianidze et al., 2017) is a semantically annotated parallel corpus for multiple languages. It consists of a large collection of parallel texts, each accompanied by a formal meaning representation based on a variation of Discourse Representation Theory (DRT, Kamp and Reyle, 1993), called Discourse Representation Structure (DRS). It can be used for corpus-based studies on formal semantic phenomena, or to develop and evaluate semantic processing tasks such as text-to-meaning parsing and meaning-to-text generation. As a matter of fact, the PMB has been widely used in semantic parsing (Abzianidze et al., 2019; van Noord, 2019; van Noord et al., 2020; Wang et al., 2021b; Poelman et al., 2022), natural language generation (Wang et al., 2021a, 2023), and semantic tagging (Bjerva et al., 2016; Abzianidze and Bos, 2017; Abdou et al., 2018; Huo and de Melo, 2020).

The rapid development of neural models and their incredible performance seem to make the impression that tasks like semantic parsing are practically solved. For instance, the state-of-the-art DRS parser (Wang et al., 2023) achieves a remarkable score of approximately 95.0 on the English test set of the PMB and manual analysis reveals that the parser made very few errors except for words outside the vocabulary. Are neural models mastering semantic parsing (and indeed natural language generation), even for complex formal meaning representations like those present in the PMB? Or is there something else going on, and does this perception not align with the actual state of affairs?

We carried out a critical examination of the PMB and revealed three (related) problems: (1) there is a “data leakage” from the training data to the development and test splits; (2) the random splits of the data lead to a non-optimal division; and (3) the test set is often regarded as “easy” as it contains a large amount of relatively short sentences. Let us elaborate on this a bit.

In the current release of the PMB, the data splits were randomly decided and considered "standard". However, this random split may result in overlap and imprecise error estimates (Søgaard et al., 2021) and and cannot adequately represent the distribution of the dataset. For instance, the sentence “*I like chocolate ice cream!*” is allocated to the training set, while the very similar sentence “*I like chocolate ice cream.*” is assigned to the test set. Equally alarmingly, some instances in the development and test sets mirror those in the training set, potentially skewing parser evaluations. Consequently, this may lead to parser evaluation results that are overly optimistic. We completely agree with Opitz and Frank (2022) and Groschwitz et al. (2023), who both argue that "AMR Parsing is far from solved" hits the nail on the head, and even goes beyond Abstract Meaning Representation (AMR) and also applies to DRS. We think the current PMB test set lacks difficulty, because it puts emphasis on brief and simplistic sentences with an average length of less than ten words. The reason for this is that all instances of the test set have the “gold” annotation status, obtained via intensive manual correction, and the longer a document the harder it is to get an error-free annotation for it.

The aim of this paper is (a) to show that the

5th Workshop on Designing Meaning Representations (DMR 2024) @LREC-COLING-2024, pages 162–175
21 May, 2024.

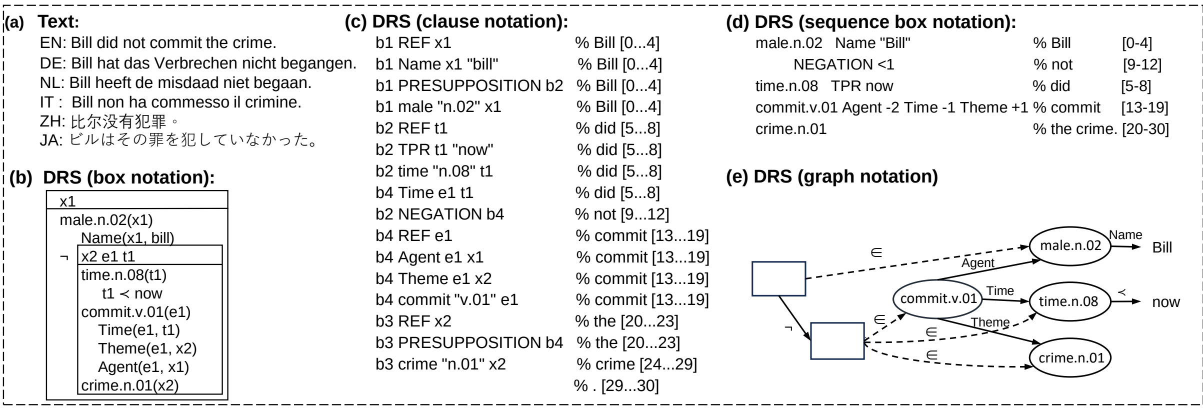

Figure 1: (a) An example sentence "*Bill did not commit the crime.*" taken from the PMB in six languages with its DRS in (b) box notation, (c) clause notation, (d) sequence box notation, and (e) graph notation.

random split indeed leads to an undesired simplification of the task, and (b) to demonstrate that the task of semantic parsing is far from being solved by providing a new challenging test set.

Inspired by the work of Søgaard et al. (2021), we design three new test sets: one standard test set and two challenge sets. The former is implemented by a two-round sorting approach to establish a more systematic split, ensuring the reliability and independence of standard development and test sets. The latter comprises a test set with substantially *longer texts* and a test set based on *compositional recombination*. The long-text set is derived by choosing documents with long texts from the PMB and manually correct the automatically assigned meaning representation. This set aims to assess the parser's performance on long and multi-sentence texts. The compositional set consists of texts formed by recombining the Combinatory Categorical Grammar (CCG, Steedman, 1996) derivation tree that is provided with the PMB data. This kind of tree recombination technique has been empirically validated for semantic data augmentation by Juvekar et al. (2023). Differently, we employ this technology for the creation of test sets, with the intent of assessing the semantic parser model's capability in compositional generalization (Furrer et al., 2020). To our knowledge, we are the first to utilize CCG to create data for compositional generalization testing. By empirical analysis of the performance of neural semantic parsers and generators based on five different language models, we show the effect of our newly created systematic split and challenge sets.

2. Background and Related Work

In this section, we first provide an overview of DRS, PMB, and CCG, review the works in parsing and generation, and introduce different data split methods. Subsequently, we introduce existing tasks and corpora related to long text semantic and compositional generalization.

2.1. Discourse Representation Structure

DRS is the formal meaning representation in the PMB, capturing the essence of the text and covering linguistic phenomena like anaphors and temporal expressions. Unlike many other formalisms such as Abstract Meaning Representation (AMR, Banarescu et al., 2013) used for large-scale semantic annotation efforts, DRS covers logical negation, quantification, and discourse relations, has complete word sense disambiguation, and offers a language-neutral meaning representation.

DRS can be represented in multiple formats as is shown in Figure1. In the box notation, DRS uses boxes containing discourse referents and conditions. Discourse referents, like *x1*, serve as markers for entities introduced in the discourse. Conditions convey information over the referents: to what concepts they belong and what relations they have to other referents, expressed by roles or comparison operators. Concepts are grounded by WordNet synsets, such as *male.n.02*. Thematic roles are derived from VerbNet (Bonial et al., 2011), for instance *Agent*. Operators, like $\prec$, $=$, $\neq$ and $\sim$, are utilized to formulate comparisons among entities. Furthermore, conditions can also be complex, serving to represent logical (negation, $\neg$) or rhetorical relations among different sets of conditions.

The clause notation is converted from box notation to adapt to machine learning models (van Noord et al., 2018). In the conversion, the label of the box, wherein the discourse referents and conditions are located, is positioned to precede them.

To simplify DRS, Bos (2023) introduced a variable-free DRS format called Sequence Box Notation (SBN), where the sequencing of terms is important. The meaning of each word adheres to an entity-role-index structure, with indices connecting entities and roles decorating connection.

The discourse relations (such as NEGATION and ELABORATION) are slightly different, indicating the beginning of a new context. The subsequent indices, marked with comparison symbols ($<$,$>$), link the newly established context to another context. SBN can also be interpreted as a directed acyclic graph, as depicted in Figure 1(e).

2.2. Combinatory Categorical Grammar

CCG is a lexicalised grammar formalism (Steedman, 1996) used in the PMB to steer the compositional semantics. It comprises just a few basic categories — N (noun), NP (noun phrase), PP (prepositional phrases) and S (sentence) — from which function categories can be composed using the backward slash for combining with phrases to the left and the forward slash for combining with phrases to its right. For instance, a typical determiner gets the lexical category NP/N to look for a noun (N) on its right resulting in a noun phrase (NP). CCG expressions can be combined with each other obeying the combinatorial rules, of which there are just a handful. The most common rules are forward and backward application:

$$\text{Forward App.} \quad (>): \quad (X/Y)\; Y \Rightarrow X \tag{1}$$

$$\text{Backward App.} \quad (<): \quad Y\; (X\backslash Y) \Rightarrow X \tag{2}$$

In the PMB, each CCG category is paired with a meaning representation with a semantic type that mirrors the internal structure of the category. This makes it a formidable linguistic formalism to implement compositional semantics.

2.3. The Parallel Meaning Bank

The PMB has evolved through four versions. Originating from the English-specific Groningen Meaning Bank (GMB, Basile et al., 2012), the PMB expanded it by embracing multiple languages. The initial version introduced German, Dutch, and Italian with their gold standard DRS in box format. The second version added silver and bronze standard data, which are partially corrected and uncorrected. Subsequent versions, namely the third and fourth versions, have witnessed an increased volume of manually annotated data and a shift from box to clause notation.

The PMB employs seven layers to process raw text, with each layer contributing an additional piece of syntactic/semantic information, building upon the results from the preceding layer (Abzianidze et al., 2020). The seven layers encompass tokenization, symbolization, word sense disambiguation, co-reference resolution, thematic role labeling, syntactic analysis and semantic tagging. Manual corrections are allowed at every layer. The final layer yields a CCG derivation tree, which is then utilized as input for the Boxer (Bos, 2015) and is converted into DRS. Initially tailored for English, PMB aligns it with other languages using an annotation projection method (Abzianidze et al., 2020).

In the field of semantic-related tasks, PMB has been widely used. However, it is not without limitations. Haug et al. (2023) emphasizes that a large portion of PMB data consists of short sentences, which compromises its ability to accurately represent real-world data.

2.4. Parsing and Generation with DRS

Semantic parsing with DRS initially employed rule-based parsers, such as Boxer (Bos, 2008). With the advent of neural models, the focus shifted to seq2seq approaches using LSTMs (van Noord et al., 2019, 2020). However, recent innovations include tree-based (Liu et al., 2018, 2019; Poelman et al., 2022) and graph-based techniques (Fancellu et al., 2019; Fu et al., 2020). In the ongoing exploration of neural networks, parsers have increasingly embraced transformer-based models like T5 (Raffel et al., 2019), BART (Lewis et al., 2020), and their variants. A significant breakthrough was DRS-MLM (Wang et al., 2023), a model that pre-trained mBART on PMB data and achieved state-of-the-art results in multiple languages. For meaning-to-text generation, Wang et al. (2021a) utilized a bi-LSTM on DRS's linearized format and found character-level decoders optimal. The mentioned DRS-MLM can also be used for DRS-to-text generation in pre-training steps outperforming other generators.

2.5. Data Split Methods

In most of the standardized datasets (Marcus et al., 1994; Fares et al., 2018), a consistent test set is typically maintained to enable comparisons between models (van der Goot, 2021). Traditionally, this kind of test set is created by random sampling (Elazar and Goldberg, 2018; Poerner et al., 2018), as is the current practice in the PMB. However, as we mentioned in the introduction, this random selection will lead to a data leakage from train to test. Multiple random split (Gorman and Bedrick, 2019) may be a fairer approach, but this will make comparison of models more difficult. To address these problems, Søgaard et al. (2021) advocates for the utilization of a biased or adversarial split besides the standard split, aiming to reduce the deviation between the test set and real-world data. We adopted this suggestion and developed an unbiased standard test set along with two biased challenge test sets, as detailed in Section 3.

2.6. Semantic Corpora with Long Texts

Few corpora focus on the semantics of long texts, primarily because of difficult annotations and constraints in meaning representation itself (For instance, AMR was initially designed for single sentences). O'Gorman et al. (2018) addressed this by manually annotating coreference, implicit roles, and bridging relations to create the multi-sentence AMR corpus. Other annotated corpora address discourse structure and rhetorical structure (Prasad et al., 2008), but ignore sentence semantics. As mentioned in Section 2.1, DRS is naturally designed for discourse, eliminating the need for additional annotation rules when annotating the meaning of long texts. Therefore, our annotation is more straightforward, as introduced in Section 3.

2.7. Compositional Generalization

Several studies have demonstrated that neural models tend to memorize patterns observed during training, struggling to generalize effectively to unfamiliar patterns (Lake and Baroni, 2018; Furrer et al., 2020). The combinationality in language significantly exacerbates this struggle. To assess this, tasks and datasets like the SCAN (Lake and Baroni, 2017) and the COGS (Kim and Linzen, 2020) have been developed. Kim and Linzen (2020) pointed out despite excellent standard test performances, their models reveal gaps in compositional generalization ability. This kind of gap led to our creation of the second challenge test set in Section 3 and experiments in Section 4.

3. Improving Semantic Evaluation

In this section we outline the methods to create better test sets. Besides the standard test set created with a different data split, we also show how we built additional challenge test sets. The resulting data set will be released as PMB 5.0.0[1].

3.1. Splitting Data Systematically

As mentioned in Section 1, the random split method employed by the PMB requires improvement. We have devised a strategy that reduces overlap between training and standard development/test sets, without introducing additional biases.

Our data split strategy involves two rounds of sorting. First, documents are sorted by character length. Afterward, the ordered collections are divided into groups of ten documents, which are then re-sorted based on their internal edit distances. The first sorting aims to maintain a consistent length distribution across the training, development, and test sets, while also ensuring some degree of uniformity in their semantic distribution. This is crucial to minimize bias introduced in the standard test data. The second sorting is particularly designed to create a certain degree of separation between the datasets, aiming at decreasing the word overlap. We allocate the first eight documents to the training set, and the remaining two are randomly distributed between the development and test sets. In Section 4, our experiments and analysis prove that the systematic split reduces the overlap between the training and development/test sets.

The distributions of gold data under the systematic split are shown in Table 1. For English, we adopt an 8:1:1 split ratio, while for the other three languages, we use a 4:3:3 ratio to ensure the test data is sufficient.

3.2. Creating Challenge Sets

We create two challenge sets for English: one focusing on long texts and another dedicated to compositional recombination by CCG.

3.2.1. Long-Text Challenge Set

Given that the gold data in the PMB predominantly consists of short sentences, with an average sentence length ranging between five and six words, it constrains our evaluation of the model's capability with long texts. In response, we select silver documents that notably exceed this average length for manual annotation, and change these into gold by correcting discourse structure, rhetorical relations, ellipsis, and inter-sentential pronouns (see Appendix A.2 for an example). Our long-text set includes 138 data samples with an average text length of 61 words, roughly ten times longer than the standard test set. The average lengths of train, development and test sets are shown in Table 1.

3.2.2. Compositional Challenge Set

As introduced in section 2, the final layer of the PMB produces the CCG derivation tree that is enriched with syntactic and semantic information, which is subsequently passed to the boxer to produce DRS. Therefore, recombining the gold CCG tree with other trees can yield distinct CCG trees, with associated text and DRS. In contrast to the creation of the long-text set, the quality of the DRS produced by this method closely approximates the gold standard, which greatly reduces the need for further manual annotation.

The original CCG derivation tree contains the compositional categories of words and phrases in a sentence, as shown in Figure 2 (a). We introduce two recombination operations: substitution and extension, shown in Figure 2 (c) and (d). In the

[1] The release is available at https://pmb.let.rug.nl/releases/

	Train	Dev	Standard Test	Long Test	Compositional Test
English (EN)	9,057 (5.64)	1,132 (5.38)	1,132 (5.15)	138 (60.78)	1,148 (6.48)
German (DE)	1,206 (5.06)	900 (4.79)	900 (4.87)	—	—
Dutch (NL)	586 (5.62)	435 (5.09)	435 (5.08)	—	—
Italian (IT)	745 (4.73)	555 (4.52)	555 (4.53)	—	—

Table 1: Distribution of train, development, and test sets in PMB 5.0.0 using the systematic split, together with two challenge sets. The average sentence length of each set are provided in brackets.

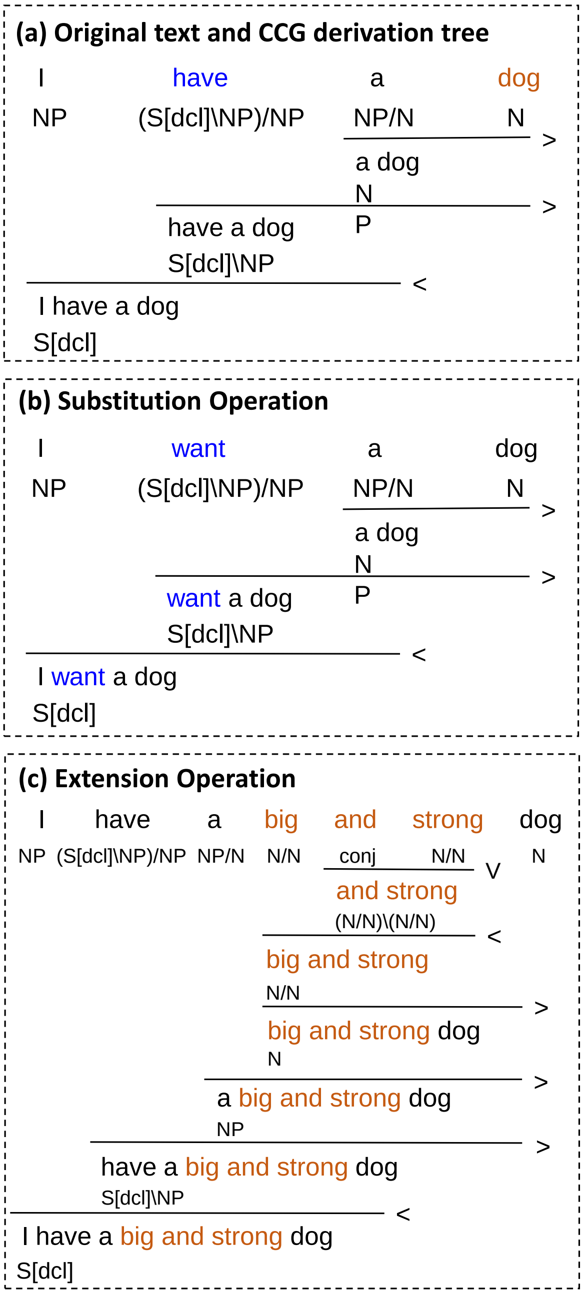

Figure 2: Two recombination operations performed on the CCG derivation tree of example sentence "*I have a dog*": (b) substitution (c) extension. We retained only the CCG categories and their corresponding words/phrases, excluding other semantic information.

substitution operation, the leaves or subtrees within a CCG derivation tree are replaced by counterparts from other different trees, provided they share the same CCG category. For instance, the word *have* swaps with *want*, as highlighted in blue. The extension operation takes a singular leaf from the tree and develops it into a larger subtree. As shown in Figure 2 (c), *dog* with the N category is extended to a subtree rooted at N, resulting in the phrase *big and strong dog*. The pseudo-code detailing these two operations is provided in Appendix A.1.

However, this method will generate many semantically abnormal sentences though they adhere strictly to syntactic structure. In this case, we use masked language models to estimate sentence pseudo-log-likelihood (PLL) scores (Salazar et al., 2020; Kauf and Ivanova, 2023). In practice, BERT (Devlin et al., 2018) is utilized as the scoring model, with a manually determined threshold. Specifically, the threshold is adjusted to eliminate 95% of the generated sentences, retaining only the top 5% that are highly deemed semantically correct.

Using this approach, we recombine the CCG trees of training samples and choose from the generated data, with the details presented in Table 1. Table 2 and 3 show some example texts produced through substitution and extension operations. Beyond individual operations, we also conduct multiple iterations on a sentence. The symbol $\times$ indicates the number of times an operation is applied to the same sentence.

4. Experiments and Analysis

This section offers an introduction to the selected seq2seq models, experimental settings, results and analysis for the text-to-DRS parsing and DRS-to-text generation.

4.1. Model Selection

The current approach to semantic parsing and text generation with DRS mainly involves fine-tuning a pre-trained language model. Our initial experiment employs a model based on BERT embeddings and LSTM architecture, following the methodology of van Noord et al. (2020). Then we utilize T5 and BART, two pre-trained transformer-based models. Specifically, we choose their multilingual variants:

Category	Operation	Training Set	Compositional Set
Noun	N⇒N	Bill was killed by an intruder.	Bill was killed by an Irishman.
Pronoun	NP⇒NP	My bag is very heavy.	His bag is very heavy.
Verb	(S\NP)/NP⇒(S\NP)/NP	The police are following us.	The police are visiting us.
Adjective	S\NP⇒S\NP	My tie is orange.	My tie is wet.
Adverb	(S\NP)/(S\NP)⇒(S\NP)/(S\NP)	The rent is very high.	The rent is extremely high.
Preposition	PP/NP⇒PP/NP	The boy bowed to me.	The boy bowed behind me.
Determiners	NP/N⇒NP/N	The answer is clear.	Neither answer is clear.
Modal	(S\NP)/(S\NP)⇒(S\NP)/(S\NP)	It will be scary.	It should be scary.
Substitution×2	N⇒N + (S\NP)/NP⇒(S\NP)/NP	Russia fears the system.	Cuba replaced the system.
Substitution×3	NP⇒NP + PP/NP⇒PP/NP + S\NP⇒S\NP	I took the elevator to the fourth floor.	They took another elevator to the last floor.

Table 2: Examples of substitution operations with CCG categories and operations. Note the table only shows the most common combinations for both two-fold (substitution × 2) and three-fold (substitution × 3) iterations. The color blue indicates the operation depicted in Figure 2 (b).

Category	Training Set	Compositional Set
Noun	My brother is rich.	My bad brother is rich. My brother who is speaking English is rich.
Verb	Coffee will be served after the meal.	Coffee will be secretly served after the meal. Coffee will be served by Elizabeth after the meal.
Adjective	Tom was thoughtful.	Tom was very thoughtful. Tom was thoughtful and innocent.
Extension×2	Tom is courteous.	Tom himself is more courteous. Tom who did it is courteous.
Extension×3	There are thirty names on the list.	There are about thirty new names on the short list. There are over thirty other names by Berlioz on the list.

Table 3: Examples of extension operations. We have excluded the operations of CCG categories due to the vast number of extension variations, which are nearly impossible to cover comprehensively. Instead, we present the most prevalent extension types for each category. The color orange indicates the operation depicted in Figure 2 (c).

mT5 (Xue et al., 2021), byT5 (Xue et al., 2022), mBART (Liu et al., 2020), and DRS-MLM (Wang et al., 2023) which is pre-trained on DRS data using the mBART architecture. In the case of DRS-MLM, for it is initially pre-trained on a train set under random split, we re-pre-train it using the train set based on our systematic split. To maintain consistent model sizes, we selected the large version across all models.

4.2. Evaluation Metrics

The evaluation process for Text-to-DRS parsing consists of two primary phases (Poelman et al., 2022). Firstly, the generated DRSs and gold standard DRSs are transformed into Penman notation (Kasper, 1989). Subsequently, we utilize SMATCH (Cai and Knight, 2013), an evaluation tool for AMR parsing, to calculate the match between the output and the gold standard by quantifying the overlap of triples. Evaluation of the generation task is conducted using BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), and COMET(Rei et al., 2020).

4.3. Experiment Settings

We carried out three primary experiments. (1) We fine-tuned the selected language models for four languages: EN, DE, NL, and IT, and evaluated them using the standard test set. Following the training configurations set by van Noord et al. (2018); Poelman et al. (2022); Wang et al. (2023), we trained the models on gold and silver data for EN, and trained on gold, silver, and bronze data for DE, NL, and IT. This was subsequently followed by a fine-tuning phase exclusively on gold data; (2) We calculated and compared the word overlap rate of the train sets and test sets under systematic and random split. Then, we showed the performance of the two top-performing models from the first experiments under these two splits. To ensure the assessment was solely influenced by the data split, we only tested on the English (only English has sufficient gold data) and fine-tuned exclusively on the gold data, and (3) We tested all fine-tuned models in the first experiments on the long-text set and compositional set. We divided the compositional set into two subsets: substitution and extension, to assess the difficulty produced by these two operations.

For all experiments and models, uniform hyperparameters were employed, and the presented results are the average scores derived from three parallel experiments.[2]

4.3.1. Standard Test

Table 4 shows the results of the text-to-DRS parsing task. Across the four languages, both byT5 and DRS-MLM models stood out, with byT5 attaining 88.0 in German, slightly surpassing DRS-MLM's 87.1, and both models achieving the same F1 of 87.2 in Italian. However, in English and Dutch, DRS-MLM takes the lead with F1 91.5 and 85.5 respectively. mT5 and mBART closely follow, but their performance in Dutch is significantly weaker, possibly due to the limited Dutch data in their pre-training corpus.

Table 5 shows the results of DRS-to-text generation. ByT5 surpasses other models in all languages except for Dutch. Particularly in English, ByT5 achieves top scores with 71.9, 54.9, and 93.0 in three metrics, respectively. However, for the Dutch, DRS-MLM remains the superior model across these three metrics.

The standout performance of byT5 and DRS-MLM can be attributed to byte-level tokenization and specific pre-training, respectively. Unlike other tokenization methods, like Byte Pair Encoding (BPE, Sennrich et al., 2016), byT5's byte-level tokenization, which can be seen as character-level within our four target languages, results in a smaller dictionary and has the ability to handle unseen words. DRS-MLM employs several pre-training tasks on the PMB data, making the model better suited for the DRS data format. This advantage is most obvious when dealing with Dutch, which has the least training data among the four languages.

4.3.2. Systematic Split vs. Random Split

Figure 3 displays the distribution of word overlap rates between train and development/test sets under random and systematic split. The word overlap rate, defined in Equation 3, measures the word-level sentence similarity. According to the figure, the systematic word overlap distribution is further to the left than the random split, indicating that it has less overlap. And as outlined in Section 3, the systematic split does not simply reduce overlap by indiscriminately adding bias. It also guarantees that each set has a consistent length distribution, which can also be viewed as a semantic distribution to a certain extent. Therefore, in the case of PMB, a systematic split is a more effective method for dividing the dataset compared to the random split.

$$\text{overlap} = \frac{\text{sentence1} \cap \text{sentence2}}{\text{sentence1} \cup \text{sentence2}} \tag{3}$$

We further proved the advantage through experiments. The parsing and generation results under these two splits are shown in Table 6 and 7. The model's performance on the random split exceeds that on the systematic split for both tasks, suggesting the systematic approach presents more rigorous challenges.

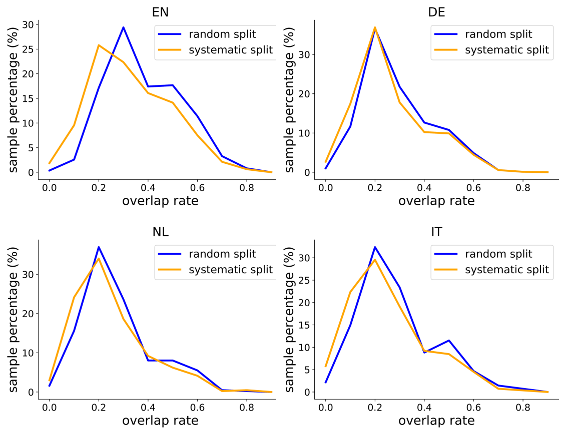

Figure 3: Distribution of word overlap rates between train and test sets in EN, DE, NL, IT. Lower overlap rates signify fewer words occurring in both train and test sets.

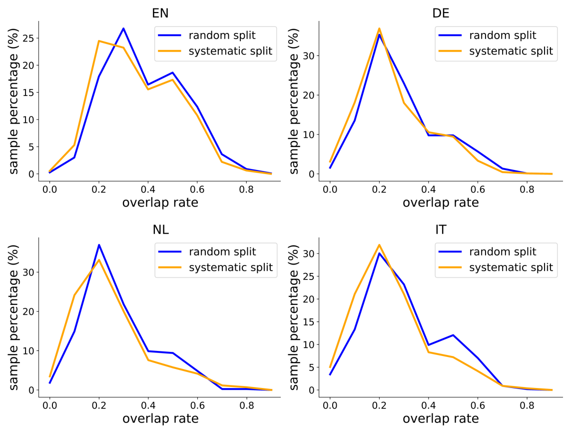

Figure 4: Distribution of word overlap rates between train and development sets in EN, DE, NL, IT.

4.3.3. Challenge Test Sets

The results of the models on the challenge test sets are shown in Tables 8 and 9. The performance on

[2]We provide the most recent experimental results for all test sets, available at `https://pmb.let.rug.nl/models.php`.

Parser	English		German		Dutch		Italian	
	F1	ERR	F1	ERR	F1	ERR	F1	ERR
LSTM	78.6	8.4	80.2	4.0	74.4	8.5	79.6	5.0
mT5	88.8	2.8	86.7	1.9	47.0	16.0	82.0	2.8
byT5	91.4	2.1	**88.0**	**0.7**	79.8	5.0	**87.2**	**0.7**
mBART	89.1	2.3	86.1	1.8	64.5	3.4	86.2	1.8
DRS-MLM	**91.5**	**1.5**	87.1	2.1	**85.5**	**2.0**	**87.2**	0.9

Table 4: Evaluation results for neural text-to-DRS parsing on the standard test sets of four languages. Note: ERR is the ill-formed rate (%) of generated DRSs that fail to transform into a graph structure.

Generator	English			German			Dutch			Italian		
	B	M	C	B	M	C	B	M	C	B	M	C
LSTM	33.8	32.4	72.5	24.9	25.4	67.1	19.0	21.6	63.2	28.2	24.7	72.2
mT5	69.9	53.4	92.8	47.8	37.5	84.8	11.3	15.2	63.6	48.8	36.3	86.0
byT5	**71.9**	**54.9**	**93.0**	**50.9**	**39.1**	**85.2**	41.8	34.2	82.1	**53.2**	**38.5**	**87.5**
mBART	51.8	43.5	88.1	40.8	33.4	79.9	38.1	32.0	80.6	45.8	34.5	84.7
DRS-MLM	67.5	52.4	92.2	47.6	36.6	84.4	**49.4**	**37.5**	**86.0**	46.3	34.2	86.3

Table 5: Evaluation results for neural DRS-to-text generation on the standard test sets of four languages. Note: B = BLEU; M = METEOR; C = COMET.

Parser	Random split		Systematic split	
	F1	ERR	F1	ERR
byT5	87.1	5.0	**83.5**	**6.0**
DRS-MLM	88.9	1.9	**87.3**	**4.1**

Table 6: Results of parsing under random and systematic split. Lower scores are marked.

Generator	Random split			Systematic split		
	B	M	C	B	M	C
byT5	66.1	52.2	91.7	**64.7**	**51.0**	**89.0**
DRS-MLM	65.8	51.4	91.7	**60.2**	**48.4**	**87.9**

Table 7: Results of generation under random and systematic split.

the long-text test set is significantly inferior, marked by a high incidence of ill-formed outputs[3]. The most pronounced drop is observed in ByT5, which shows a reduction of 86% compared to the standard test set. In the generation task, although truncation does not hugely impact on evaluation, the models still grapple with long sequences, reflecting decreases of at least 29.9, 11.9, and 16.2 across three metrics. Notably, neural models struggle with the long set, primarily because their tokenization significantly amplifies both input and output lengths. For example, while the average sentence lengths in the long set stand at 61 for text and 253 for DRS, these numebrs increase to 98 and 503 after BPE tokenization (mT5, mBART, and DRS-MLM) and even further to 410 and 1370 with character-level tokenization (ByT5). Obviously, these models can not handle such long sequences as effectively as the short sequences in the standard test.

For the compositional challenge set, it's crucial to note that all semantic components in the test sets were also in the training. Therefore, we expect near-perfect scores from the models. They perform well on the *compositional-substitution* set, showcasing their ability to learn and apply word meanings in known sentence structures. Among these models, byT5 performs the best with 93.1 F1 in parsing, while mT5 and DRS-MLM show similarly strong performance in generation. When testing on the *compositional-extension* set, the performance of the models dropped by around ten points in both tasks. Most parsing or generation errors were in the newly added parts in the texts, likely due to the introduction of more intricate sentence structures, especially compound predicate adjectives and attributive clauses, as shown in the examples in Table 3. The most frequent errors of the models are provided with examples in Appendix A.2.

5. Conclusion

Past performance of neural semantic parsers and meaning-to-text generators have been slightly in-

[3] SMATCH employs a hill-climbing technique to identify the optimal match, which may introduce inaccuracies when evaluating the output of the model for long texts (Opitz and Frank, 2022). In this case, the results for long texts should be considered as reference only.

Parser	en-long		en-substitution		en-extension	
	F1	ERR	F1	ERR	F1	ERR
LSTM	**43.7**	**19.2**	90.8	2.8	82.7	**3.5**
mT5	38.8	34.6	88.9	2.9	80.3	8.9
byT5	5.5	65.4	**93.1**	**0.5**	**84.8**	5.0
mBART	22.0	53.8	89.7	1.4	80.4	7.6
DRS-MLM	20.0	57.7	90.3	2.8	81.1	7.7

Table 8: Evaluation results for text-to-DRS parsing on the challenge test sets.

Generator	en-long			en-substitution			en-extension		
	B	M	C	B	M	C	B	M	C
LSTM	5.48	14.6	40.3	58.7	43.6	82.1	49.1	41.3	77.6
mT5	31.4	40.3	**76.6**	75.2	**55.6**	**92.7**	67.3	52.9	**90.0**
byT5	14.1	28.3	59.3	75.7	54.7	92.5	66.7	53.0	89.8
mBART	15.7	28.7	60.6	68.8	51.8	89.8	58.4	48.8	86.1
DRS-MLM	**32.6**	**40.5**	75.4	**76.0**	54.9	92.5	**69.4**	**53.2**	**90.0**

Table 9: Evaluation results for DRS-to-text generation on the challenge test sets.

flated (or at best, made the suggestion that these semantic computational tasks were close to being “solved”) due to data leakage from training to test and non-representative test sets. At least, that is what our empirical study on the Parallel Meaning Bank showed. We created a more realistic assessment of performance by refining the data split and formulating challenge sets. A systematic split for the PMB yields a test set that is harder for semantic parsers and generators. The introduction of two further challenge sets, one with manually corrected longer documents and one with automatically derived compositional recombination using categorical grammar, are indeed way more challenging than the standard test set. Hence, semantic parsing and text-to-meaning generation can not be considered “solved” yet.

6. References

Mostafa Abdou, Artur Kulmizev, Vinit Ravishankar, Lasha Abzianidze, and Johan Bos. 2018. What can we learn from semantic tagging? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4881–4889, Brussels, Belgium. Association for Computational Linguistics.

Lasha Abzianidze, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 242–247, Valencia, Spain. Association for Computational Linguistics.

Lasha Abzianidze and Johan Bos. 2017. Towards universal semantic tagging. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Lasha Abzianidze, Rik van Noord, Hessel Haagsma, and Johan Bos. 2019. 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.

Lasha Abzianidze, Rik van Noord, Chunliu Wang, and Johan Bos. 2020. The parallel meaning bank: A framework for semantically annotating multiple languages. *Applied mathematics and informatics*, 25(2):45–60.

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

Valerio Basile, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. A platform for collaborative semantic annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 92–96, Avignon, France. Association for Computational Linguistics.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan. The COLING 2016 Organizing Committee.

C. Bonial, W. Corvey, M. Palmer, V.V. Petukhova, and H.C. Bunt. 2011. A hierarchical unification of lirics and verbnet semantic roles. In *Proceedings IEEE-ICSC 2011 Workshop on Semantic Annotation for Computational Linguistic Resources*, pages 1–7. Stanford University.

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

Johan Bos. 2015. Open-domain semantic parsing with boxer. In *Nordic Conference of Computational Linguistics*.

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

Shu Cai and Kevin Knight. 2013. 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)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21, Brussels, Belgium. Association for Computational Linguistics.

Federico Fancellu, Sorcha Gilroy, Adam Lopez, and Mirella Lapata. 2019. 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)*, pages 2769–2778, Hong Kong, China. Association for Computational Linguistics.

Murhaf Fares, Stephan Oepen, Lilja Øvrelid, Jari Björne, and Richard Johansson. 2018. The 2018 shared task on extrinsic parser evaluation: On the downstream utility of English Universal Dependency parsers. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 22–33, Brussels, Belgium. Association for Computational Linguistics.

Qiankun Fu, Yue Zhang, Jiangming Liu, and Meishan Zhang. 2020. DRTS parsing with structure-aware encoding and decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6818–6828, Online. Association for Computational Linguistics.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *CoRR*, abs/2007.08970.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, Florence, Italy. Association for Computational Linguistics.

Jonas Groschwitz, Shay Cohen, Lucia Donatelli, and Meaghan Fowlie. 2023. AMR parsing is far from solved: GrAPES, the granular AMR parsing evaluation suite. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10728–10752, Singapore. Association for Computational Linguistics.

Dag TT Haug, Jamie Y Findlay, and Ahmet Yıldırım. 2023. The long and the short of it: Drastic, a semantically annotated dataset containing sentences of more natural length. In *Proceedings of the 4th International Workshop on Designing Meaning Representations*, pages 89–98. Association for Computational Linguistics.

Da Huo and Gerard de Melo. 2020. Inducing universal semantic tag vectors. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3121–3127, Marseille, France. European Language Resources Association.

Mandar Juvekar, Gene Louis Kim, and Lenhart Schubert. 2023. Semantically informed data augmentation for unscoped episodic logical forms. In *15th International Conference on Computational Semantics*.

H. Kamp and U. Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Number pt. 2 in Developments in Cardiovascular Medicine. Kluwer Academic.

Robert T. Kasper. 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*.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring.

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

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International*

Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, pages 2873–2882. PMLR.

Brenden M. Lake and Marco Baroni. 2017. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019. Discourse representation parsing for sentences and documents. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6248–6262, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, page 114–119, USA. Association for Computational Linguistics.

Tim O'Gorman, Michael Regan, Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. 2018. AMR beyond the sentence: the multi-sentence AMR corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3693–3702, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Wessel Poelman, Rik van Noord, and Johan Bos. 2022. Transparent semantic parsing with Universal Dependencies using graph transformations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4186–4192, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Melbourne, Australia. Association for Computational Linguistics.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832, Online. Association for Computational Linguistics.

Mark Steedman. 1996. Surface structure and interpretation. In *Linguistic Inquiry*.

Rob van der Goot. 2021. We need to talk about train-dev-test splits. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4485–4494, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rik van Noord. 2019. Neural boxer at the IWCS shared task on DRS parsing. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018. Exploring Neural Methods for Parsing Discourse Representation Structures. *Transactions of the Association for Computational Linguistics*, 6:619–633.

Rik van Noord, Antonio Toral, and Johan Bos. 2019. Linguistic information in neural semantic parsing with multiple encoders. In *Proceedings of the 13th International Conference on Computational Semantics - Short Papers*, pages 24–31, Gothenburg, Sweden. Association for Computational Linguistics.

Rik van Noord, Antonio Toral, and Johan Bos. 2020. 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)*, pages 4587–4603, Online. Association for Computational Linguistics.

Chunliu Wang, Huiyuan Lai, Malvina Nissim, and Johan Bos. 2023. Pre-trained language-meaning models for multilingual parsing and generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5586–5600, Toronto, Canada. Association for Computational Linguistics.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021a. Evaluating text generation from discourse representation structures. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 73–83, Online. Association for Computational Linguistics.

Chunliu Wang, Rik van Noord, Arianna Bisazza, and Johan Bos. 2021b. Input representations for parsing discourse representation structures: Comparing English with Chinese. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 767–775, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A. Appendix

Appendix A.1 Pseudo-code for CCG recombination

Both substitution and extension operations begin with a standard pre-processing step: subtree set construction. This extracts all subtrees from the dataset's CCG derivation trees (For consistency, we treat leaves as subtrees with only the root). Substitution operation primarily involves randomly selecting subtrees, and then deleting and substituting them. The replacement subtree is chosen from the list in the first step. Extension operation involves forming child mappings and producing subtrees according to the mappings.

Algorithm 1 Extract Subtrees from CCG Trees

```
Variables:
SubtreeList ← empty list
AllCCGTrees ← CCG tree list

function ExtractSubs(node, currentPath)
    if node is null then return
    end if
    Add node to currentPath
    if node.left and node.right are null then
        Add currentPath to SubtreeList
    end if
    ExtractSubs(node.left, currentPath)
    ExtractSubs(node.right, currentPath)
end function

function SubtreesForTree(root)
    ExtractSubs(root, empty list)
return SubtreeList
end function

function SubtreesForTrees(AllCCGTrees)
    for each tree in AllCCGTrees do
        SubtreesForTree(tree)
    end for
return SubtreeList
end function
```

Algorithm 2 Substitution Operation

```
Variables:
SubtreeList ← list of subtrees

function GetParent(tree, childNode)
    for each node n in tree do
        if n.left = childNode or n.right = childNode then
            return n
        end if
    end for
    return null
end function

function DeleteAndAdd(tree, nodeToDelete)
    parent ← GetParent(tree, nodeToDelete)
    newSubTree ← randomly select from SubtreeList with same root of nodeToDelete
    if parent.left = nodeToDelete then
        parent.left ← newSubTree
    else if parent.right = nodeToDelete then
        parent.right ← newSubTree
    end if
end function

function Substitute(tree)
    nodeToDelete ← randomly select a node from tree
    DeleteAndAdd(tree, nodeToDelete)
end function
```

Algorithm 3 Extension Operation

```
Variables:
Subtrees ← list of subtrees
ChildMap ← dictionary of children

function Traverse(node)
    if node is null then
        return
    end if
    if node.left then
        ChildMap[(node, node.left)] ← node.right
    end if
    if node.right then
        ChildMap[(node, node.right)] ← node.left
    end if
Traverse(node.left)
Traverse(node.right)
end function

function CreateSubtree(parent, left, right)
    parent.left = left
    parent.right = right
end function

function Extension(tree)
    leaf ← RandomSelectLeaf(tree)
    if left then
        newSubRoot ← CreateSubtree(leaf, leaf, ChildMap[(leaf, leaf)])   ▷ To extend the node from right
    else
        newSubRoot ← CreateSubtree(leaf, ChildMap[(leaf, leaf)], leaf)   ▷ To extend the node from left
    end if
    choose the newSubtree from Subtrees according to newSubRoot
    replace leaf with newSubtree
end function
```

Appendix A.2 Case Study

In this appendix, we present some wrong generations by byT5 model in the semantic parsing task. Additionally, the gold-standard text and DRS can also be seen as examples of the challenge sets.

Test set	Gold Text	Gold DRS	Generated
Standard	Mary called us.	female.n.02 Name "Mary" call.v.03 Agent -1 Time +1 Co-Agent +2 time.n.08 TPR now person.n.01 Sub speaker	female.n.02 Name "Mary" call.v.03 Agent -1 Time +1 Theme +2 time.n.08 TPR now person.n.01 Sub speaker
Long Text	Recent studies show that children who do not get enough sleep tend to have some emotional problems as well as weight gain later in life. As VOA's Melinda Smith reports, the research seems to blame the parents.	recent.a.02 AttributeOf +1 study.n.01 show.v.02 Proposition >1 Experiencer -1 Time +1 time.n.08 EQU now CONTINUATION <0 child.n.01 tend.v.01 Agent -1 Time +1 Topic +2 time.n.08 EQU now have.v.01 Pivot -3 Theme +3 Theme +7 emotional.a.03 AttributeOf +1 problem.n.01 entity.n.01 Sub -1 Sub +2 weight.n.01 gain.n.01 Theme -1 later.r.01 EQU -6 life.n.01 NEGATION <1 time.n.08 EQU now get.v.01 Pivot -12 Time -1 Theme +2 enough.a.01 AttributeOf +1 sleep.n.01 CONTINUATION <3 agency.n.01 Name "VOA" female.n.02 Name "Melinda Smith" PartOf -1 report.v.01 Agent -1 Time +1 time.n.08 EQU now CONTINUATION <1 research.n.01 seem.v.01 Experiencer -1 Time +1 Stimulus +2 time.n.08 EQU now blame.v.01 Agent -3 Theme +1 person.n.01 Role +1 parent.n.01	recent.a.01 AttributeOf +1 study.n.04 show.v.04 Proposition >1 Experiencer -1 Time +1 time.n.08 EQU now CONTINUATION <0 child.n.01 NEGATION <1 time.n.08 EQU now get.v.01 Pivot -2 Time -1 Theme +2 enough.a.01 AttributeOf +1 sleep.n.01 tend.v.01 Agent -4 T
Substitution	Hungarian prisoners broke out of jail.	country.n.02 Name "Hungary" person.n.01 Location -1 Role +1 prisoner.n.01 break_out.v.03 Theme -2 Time +1 Source +2 time.n.08 TPR now jail.n.01	country.n.02 Name "Hungary" person.n.01 Source -1 Role +1 prisoner.n.01 break_out.v.01 Source -2 Time +1 Theme +2 time.n.08 TPR now jail.n.01
Extension	Mr. Smith who worked on that project asked Jane to marry him.	mr.n.01 male.n.02 Name "Smith" Title -1 work.v.01 Agent -1 Time +1 Theme +2 time.n.08 TPR now project.n.01 ask.v.02 Agent -4 Time +1 Recipient +2 Topic +3 time.n.08 TPR now female.n.02 Name "Jane" marry.v.01 Agent -1 Co-Agent +1 male.n.02 ANA -8	mr.n.01 male.n.02 Name "Smith" Title -1 work.v.02 Agent -1 Time +1 Theme +2 time.n.08 TPR now project.n.01 ask.v.02 Agent -4 Time +1 Patient +2 Result +3 time.n.08 TPR now female.n.02 Name "Jane" marry.v.01 Agent -1 Co-Agent +1 male.n.02 ANA -5

Table 10: Four examples in different test sets.