DRS Parsing as Sequence Labeling

Minxing Shen
Heinrich Heine University Düsseldorf
Universitätsstraße 1
40225 Düsseldorf, Germany
minxing.shen@hhu.de

Kilian Evang
Heinrich Heine University Düsseldorf
Universitätsstraße 1
40225 Düsseldorf, Germany
kilian.evang@hhu.de

Abstract

We present the first fully trainable semantic parser for English, German, Italian, and Dutch discourse representation structures (DRSs) that is competitive in accuracy with recent sequence-to-sequence models and at the same time *compositional* in the sense that the output maps each token to one of a finite set of meaning *fragments*, and the meaning of the utterance is a function of the meanings of its parts. We argue that this property makes the system more transparent and more useful for human-in-the-loop annotation. We achieve this simply by casting DRS parsing as a sequence labeling task, where tokens are labeled with both fragments (lists of abstracted clauses with relative referent indices indicating unification) and *symbols* like word senses or names. We give a comprehensive error analysis that highlights areas for future work.[1]

1 Introduction

Semantic parsing is the task of mapping natural-language sentences to symbolic representations of their meaning. Although most current natural language understanding (NLU) applications are handled by end-to-end systems that solve specific tasks (such as machine translation, conversation, or sentiment analysis) without intermediate symbolic meaning representations, semantic parsing continues to attract research interest for good reasons: first, next-generation NLU systems may become more accurate and certainly more easily explainable and debuggable by combining symbolic representations with end-to-end techniques. Second, symbolic meaning representations are amenable to symbolic reasoning, which may be instrumental in enabling, e.g., digital assistants to solve more complex tasks. Third, better and more transparent computational models of text-meaning mapping can be a useful tool for semantics, i.e., to understand how natural-language semantics works.

In recent years, most work on annotating natural-language text with comprehensive, broad-coverage meaning representations has been performed in three frameworks: Abstract Meaning Representations (Banarescu et al., 2013), Universal Cognitive Conceptual Annotation (Abend and Rappoport, 2013), and Discourse Representation Structures (Abzianidze et al., 2017). Accurate parsers exist for all three (e.g., Lindemann et al., 2020; Oepen et al., 2020; van Noord et al., 2020). Each formalism has its specific strength: AMRs go very far in abstracting away from surface variation in how a certain meaning is expressed, UCCA has a clear mapping between form and meaning and a modular architecture, and DRSs ground natural language meaning in first-order logic, by explicitly representing the scopes of negation, quantification, disjunction, etc. In this paper, we focus on parsing to DRSs.

State-of-the-art DRS parsers follow the encoder-decoder paradigm pioneered for machine translation by Sutskever et al. (2014): the input sequence is encoded by a neural network into a vector, then another network predicts the output sequence (or in this case: output DRS) from that vector. Rather than improve upon the accuracy of such parsers on standard benchmarks, our aim in this paper is to achieve some of their benefits (ability to learn from examples, high accuracy, low computational complexity, robustness to atypical input, utilization of off-the-shelf language models, conceptual simplicity) while also having a degree of *compositionality*, traditionally a property of grammar-based systems. Specifically, our system learns to assign each token of an utterance one of a finite set of abstract meaning *fragments* that are deterministically combined to give the meaning of the whole utterance. While our system may not fulfill all criteria of compositionality according to some definitions, it can

[1] Our system is available at https://github.com/ShenMinX/DRS-parser

Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 213 - 225

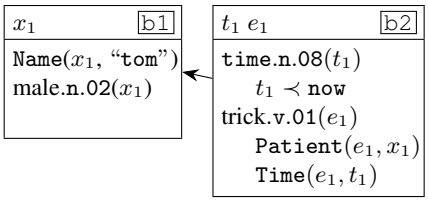

Figure 1: DRS for the sentence "Tom was tricked" in box notation

```
b1 REF x1                    % Tom [0...3]
b1 Name x1 "tom"             % Tom [0...3]
b1 PRESUPPOSITION b2          % Tom [0..3]
b1 male "n.02" x1             % Tom [0..3]
b2 REF t1                    % was [4...7]
b2 TPR t1 "now"              % was [4...7]
b2 Time e1 t1                % was [4...7]
b2 time "n.08" t1            % was [4...7]
b2 REF e1                % tricked [8...15]
b2 Patient e1 x1         % tricked [8...15]
b1 trick "v.01" e1       % tricked [8...15]
                               % . [15...16]
```

Figure 2: DRS for the sentence "Tom was tricked" in clause notation

arguably reap some of compositionality's benefits, which make it suitable for use in semi-automatic annotation workflows. We discuss this further in Section 5.

Previous work has introduced trainable compositional semantic parsers for AMR (Lindemann et al., 2020) and DRS (Evang, 2019; Bladier et al., 2021). In this paper, we improve upon the latter parser using a novel way to encode anchored DRSs as sequences, and thereby cast DRS parsing simply as a sequence labeling task (§2). We use a standard transformer-based model to learn this task, followed by post-processing to ensure well-formed DRSs (§3). We use training data from the Parallel Meaning Bank (§4). The accuracy of our model approaches the state of the art with the additional benefit of being, to a degree, compositional (§5). We give an error analysis in §6 and conclude in §7.

2 Encoding Anchored DRSs as Sequences

Gómez-Rodríguez and Vilares (2018); Strzyz et al. (2019); Vilares et al. (2020) encode syntax trees as token labels to cast syntactic parsing as a sequence labeling task. We apply a similar method to DRS parsing. We will use a simplified example from the Parallel Meaning Bank (PMB; Abzianidze et al., 2017) for exposition.

Figure 1 shows the DRS for the sentence "Tom was tricked" in *box notation*. It consists of two sub-DRSs or *boxes*, `b1` and `b2`. `b1` introduces an entity named "Tom" `x1`. `b2` introduces a "tricking" event `e1` (an event of type `trick.v.01` in the WordNet ontology, Fellbaum (2000)) whose Patient role is filled by `x1`. Because "Tom" is a definite NP, it introduces a *presupposition*: `b2` presupposes `b1`. The event is in the past, i.e., its Time role is filled by a time entity (an entity of type `time.n.08` in WordNet) `t1` which precedes the time `"now"`.

Figure 2 shows the same DRS in *clause notation*. Here, a DRS is a set of clauses. A clause consists of a *box label* indicating which box the clause is part of, a *predicate* such as a word sense, a semantic role, or a discourse relation, and one or two *arguments*, which may be *referents* such as `e1` or `x1`, or *constants* such as `"hearer"`, `"now"`, or `"+"`.

Our sequence-labeling method assumes training DRSs to be *anchored*, that is, each clause must be aligned to one (or more) input token. Thanks to the grammar-based annotation method of the PMB, this is approximately the case, as can be seen in the clause representation. We thus encode the DRS as a sequence of labels, one for each token, where each label consists of zero or more clauses, as row (1) of Figure 3 shows. We call these labels *fragments*. Although labels are complex because they can consist of multiple clauses, our sequence labeling model treats them as atomic.

In prediction tasks, it is important that label predictions generalize to unseen data. In contrast to this, the numeric part of referent labels in clauses are not meaningful and depend on the number of referents that were introduced before in the same sentence, so they would generalize poorly. Thus, in row (2), we change the referents to be *relative*, inspired by Bos (2021): referents that have not occurred before get the index `0` and referents that have occurred get a negative index, indicating how long ago the same referent last occurred (counting back among all occurrences of referents of the same type).

To further reduce proliferation of different fragments, we also experiment with factorizing fragments into fragments proper and *integration labels*. In this factorization, the first backreference of every type in a fragment always has index `-1`, and a separately predicted integration label specifies how much to subtract from that to get to the actual index. This can be seen in row (3), where the first `b` label for the word *was* has index `-1` instead of `-2`, and

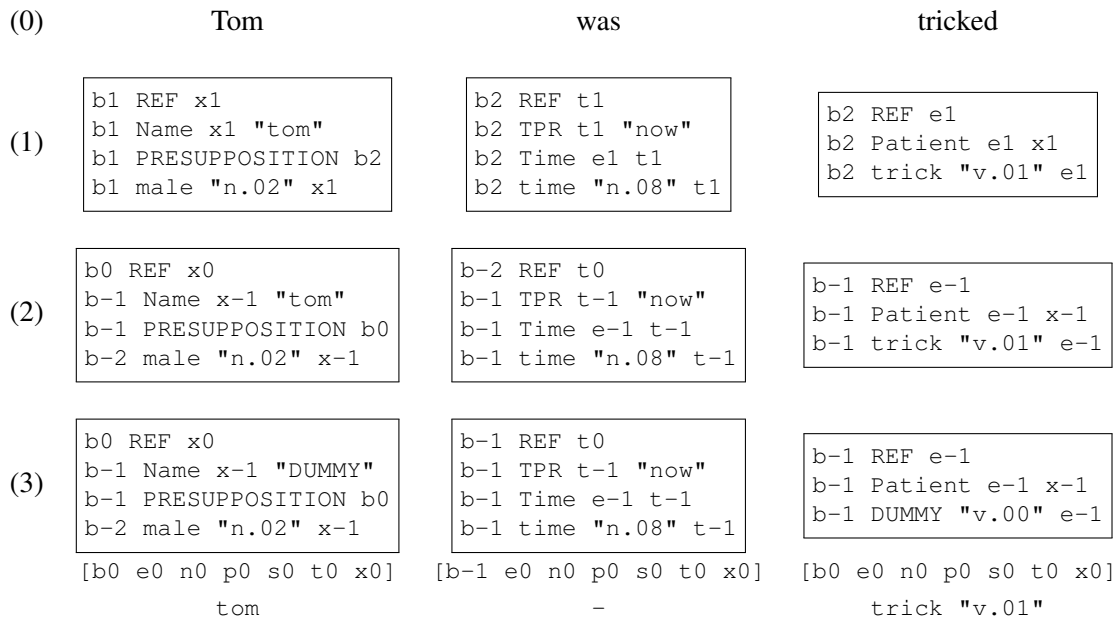

Figure 3: Sequence encoding of anchored DRSs. From top to bottom: (0) the sentence, (1) basic sequence encoding, (2) relative sequence encoding, (3) factored sequence encoding with separate integration and symbol labels.

the integration label `[b-1 e0 n0 p0 s0 t0 x0]` indicates that 1 should be subtracted from that to get to the actual relative index. This allows *was* in our example to have the same fragment as in *Someone was tricked*, where the subject does not introduce a presupposition and the actual index is thus `-1` rather than `-2` because there is one less box intervening.[2]

Another important factorization concerns large-class and open-class symbols, *viz.* (content-word) word senses, names, numbers, and time expressions. We follow Evang (2019) in replacing these with dummy expressions in the fragments and predicting them separately, as explained below in Section 3. We also follow them in heuristically changing the representation of first and second person pronouns, which introduce `"speaker"` and `"hearer"` constants instead of discourse referents in the PMB, for more consistent representation of predicates.

3 Parsing Model

Our parsing model consists of a standard transformer sequence labeling model, followed by post-processing to assemble the predicted labels into a DRS.

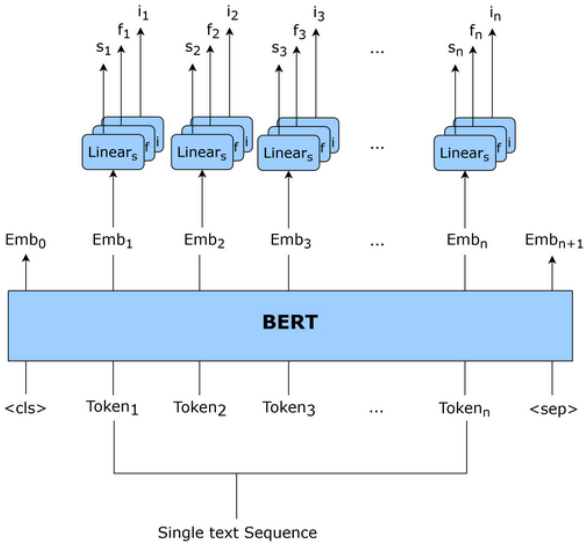

Figure 4: Neural model

[2] As pointed out by a reviewer, an even better factorization of fragments could potentially be achieved by indexing not with respect to linear position but with respect to the syntactic head word. This would require introducing a dependency parsing component. We leave this for future work.

Sequence Labeling Transformer Model Our model is schematically depicted in Figure 4. It takes an input sequence of tokens $X = w_1 \dots w_n$ and produces aligned output sequences Y_s, Y_f, Y_i, which are word senses, fragments, and integration labels. Our model simply consists of a pre-trained BERT model (Devlin et al., 2019) and three linear classifiers. Each classifier can be seen as a sub-system of the semantic parser that produces one of the three labels (word sense, fragment, and integration label).

Each input token must further be tokenized into *wordpieces* before they can be fed into the BERT model. To obtain a single representation for a token that consists of N wordpieces and thus produces N embedding vectors, we experiment with two commonly used strategies: taking only the first wordpiece, or averaging the embeddings of all N wordpieces.

Post-processing After the neural model predicts a fragment and a word sense for each token, we assemble these predictions into a complete clause list by choosing unique new names for discourse referents with index `0` and unifying other discourse referents with them according to their relative indices. We also replace `DUMMY` strings in clauses by the predicted word senses and by symbols for names, cardinalities, and date/time expressions, which are predicted from the tokens by a rule-based system similar to that of Evang (2019). For example, for the proper name *Tom* it predicts the symbol `"tom"`, for the numeral *two* it predicts `"2"`, and for the time expression *five o'clock*, it predicts `"17:00"`. Special clauses like `b1 "speaker" x1` and `b1 "hearer" x1` are removed and the corresponding referent (`x1` in the example) replaced by the symbols `"speaker"` and `"hearer"`. Finally, we use a set of postprocessing rules similar to that of van Noord et al. (2020) to ensure the validity of the resulting DRS: if there is a loop in the subordination relation among boxes, an arbitrary box in the loop is chosen, and all its clauses are removed to break the loop (cf. Figure 9 in the Appendix). Furthermore, a `REF` clause is introduced for each referent that is now used but not introduced by a `REF` clause, in the box where it first occurs. Finally, connectedness of all boxes is ensured by introducing `CONTINUATION` relations between top-level unconnected boxes.

	gold	silver	bronze	total
English	8 403	97 958	146 371	252 372
German	1 979	5 250	121 111	128 340
Italian	1 062	2 772	64 305	68 139
Dutch	1 012	1 301	21 550	23 863

Table 1: Numbers of DRSs in the PMB 3.0.0

4 Experimental Setup

Data and Splits We train and evaluate our models on the Parallel Meaning Bank (PMB; Abzianidze et al., 2017), version 3.0.0. This sembank contains sentences annotated with anchored DRSs in four languages (English, German, Italian, Dutch) and three annotation statuses: *gold* DRSs have been fully corrected by human annotators, *silver* ones have been partially corrected, and *bronze* ones are the unchecked outputs of rule-based pre-annotation. Table 1 gives an overview. We use the standard split into training, development, and test data suggested in the PMB release. Note that for Italian and Dutch, the number of gold DRSs is very small and they are only used for development and testing, leaving only bronze and silver data for training.

PLMs and Hyperparameters The backbone of our PyTorch (Paszke et al., 2019) implementation is the *Transformer* and *WordpieceTokenizer* classes offered by Hugging Face (Wolf et al., 2019). We use pre-trained BERT models provided on `huggingface.co`: `bert-base-cased`, `dbmz/bert-base-german-cased`, `dbmz/bert-base-italian-cased`, and `Geotrend/BERT-base-nl-cased` (Abdaoui et al., 2020), keeping their default configuration. The only hyperparameters we choose ourselves are the batch size (24), the learning rate, and the number of epochs. We used the Adam optimizer to train all the parameters in our model including the pretrained BERT. To ensure stability and avoid overfitting, we used a linear scheduler with no warm-up step, which gradually reduces the learning rate from 0.0015 to 0 for each training iteration. During preliminary experiments on the development set, we found that training loss barely changed after five epochs.

BERT has 12 layers, each of which has a 768-dimensional output embedding per wordpiece. There is some mixed information in the literature as to which layer's output is most suitable for seman-

Parameters	114 M
Training time	12 mins
Word senses	5 864
Fragments w/ integration labels	1 864
Fragments w/o integration labels	2 694
Integration labels	100

Table 2: Initial model statistics for English

all concepts	0.7584
nominal	0.8217
verbal	0.6173
adjectival	0.5861
adverbs	0.5977

Table 3: Word sense f-scores in the initial model for English

Layer Wordpiece	7 initial	7 mean	12 mean
sense acc.	0.8663	0.8670	0.8648
fragment acc.	0.8630	0.8659	0.8651
integration acc.	0.9461	0.9475	0.9436
Counter f1	0.7873	0.7882	0.7836

Table 4: Choice of BERT output layer and wordpiece embeddings

tic parsing tasks. According to Chronis and Erk (2020), the middle layer is most transferable for downstream semantic tasks, while van Noord et al. (2020) claim that the last layer provides the best results for their DRS parser, so we experimented with both.

Evaluation We evaluate the performance of our parser using Counter (van Noord et al., 2018a), an extension of the Smatch evaluation metric (Cai and Knight, 2013). Counter approximates an optimal mapping between the referents in the gold DRS and the predicted DRS using hill-climbing, then outputs recall, precision, and f-score for the predicted clauses compared to the gold clauses.

5 Results and Discussion

Integration Labels We trained an initial model on the English gold training data, for which we give some statistics in Table 2. As can be seen, factoring fragments leads to 100 distinct integration labels and reduces the number of distinct fragments from 2 694 to 1 864. We found however that the factorization does not necessarily help the model, as the integration labels are extremely unbalanced. In fact, 80.1% of tokens in the training data have the "empty" integration label `[b0 e0 n0 p0 s0 t0 x0]`. In a direct comparison, we found that factoring out integration labels improves the prediction accuracy on the fragments by 3%. However, since prediction of integration labels is not perfect, the overall Counter f-score is not improved significantly (the difference in f-score is smaller than 0.01%). We nevertheless conduct all further experiments with integration labels enabled.

Word Senses The next label we take a closer look at is the word senses. Table 3 shows the f-score of our model's sense predictions, as reported by Counter, overall and broken down into nominal, verbal, adjectival, and adverbial word senses. The accuracy is much higher for nouns than for verbs, which reflects the fact that the former are less polysemous than the latter according to WordNet statistics.[3] Another possible reason is that many nominal senses do not stem from predictions of the word sense layer but from "function" senses that appear in many fragments, such as `time "n.08"` in the fragment for *was* in Figure 3. The lower scores for adjectival and adverbial can be explained with data sparsity, for there only 1 593 adjectives and 210 adverbs in the gold data. For comparison, the number of nouns and verbs are 20 192 and 6 108.

Choice of BERT Output Layer and Wordpiece Embeddings We were interested in how the choice of BERT output layers and word piece embeddings impacts performance of our model. Hence, we did the following experiments with our base model, shown in Table 4. First, we use BERT's middle (7th) output layer, using the embedding of the initial word piece for each word as input to the classifiers. Second, we used the middle layer, but with the mean vector of all word pieces (this is the method we used in all previous experiments). Third, we used the mean value of the final (12th) BERT output layer, which helped van Noord et al. (2020) build their best model, yet according to Chronis and Erk (2020) contains too much "information residual", hence is more suitable for syntactical tasks. To minimize the effect of

[3] https://wordnet.princeton.edu/documentation/wnstats7wn, retrieved 2022-03-11

	g	g+s	g+s+b
# senses	5 864	42 147	60 740
# fragments	1 864	20 170	27 949
# integrations	100	2 901	4 121
Counter f1	0.7896	0.8554	0.8640

Table 5: Training on silver and bronze data

Model	dev	test
Bladier et al. (2021)	81.4	81.4
van Noord et al. (2018b)	84.3	84.9
van Noord et al. (2019)	86.8	87.7
van Noord et al. (2020) (base)	87.6	88.5
van Noord et al. (2020) (best)	**88.4**	**89.3**
Pro Boxer	88.2	88.9
this work	86.4	88.4

Table 6: Comparison of our English parser with prior art (Counter f-scores on PMB 3.0.0)

random errors, we did five trials on each of these embedding approaches and averaged the results. Although the differences are rather small, the mean vector of the middle layer seems to provide the best scores across the board. Therefore, we stuck to this setting for subsequent experiments.

Bronze and Silver Training Apart from the small gold set whose quality is guaranteed by human annotators, PMB 3.0.0 also contains silver and bronze data with partial or no manual checking of the annotations. Their lower quality is compensated for by quantity. Liu et al. (2019) report a large improvement for their DRS parser when first training on the bronze and silver data, then "fine-tuning" on gold data. Since we are using a Transformer model like them, we expected this technique could also boost our parser's performance. Thus, we tested our model with 5 epochs training on silver and bronze followed by 5 epochs on gold. The results are shown in Table 5. They confirm that more data means better results even when the data is not perfect. Although the bigger training set also increases the number of classes for all three labels more than 10-fold, the model seems to handle it just fine. The only downside is the longer training time: as the silver and bronze sets for English are, respectively, 21 and 25 times larger than the gold one, the time consumption jumps from a few minutes to more than 10 hours.

Final Model for English We compare our final best model for English to previous work, shown in Table 6. Note that Bladier et al. (2021) is an improved version of Evang (2019)'s transition-based DRS parser. The models presented by van Noord et al. (2018b, 2019, 2020) are all character-wise sequence-to-sequence models. No results on the same data are available for the encoder-decoder model of Liu et al. (2019); however, on PMB 2.2.0 its difference in Counter f-score with van Noord et al. (2019) was less than 1% on the dev and test set. The "base" model of van Noord et al. (2020) is the character-wise sequence-to-sequence parser of van Noord et al. (2019) with the addition of BERT embeddings, and their "best" model encodes the character embedding and the BERT embedding separately before feeding their concatenated vector into the decoder, which achieved state-of-the-art results. Worth noting is their claim that it's best to keep BERT parameters "frozen", which we did not find to be the case for our model: in preliminary experimentation, finetuning BERT parameters with our model outperformed a corresponding frozen model by 20% in Counter f-score.

We also compare with the semi-rule-based system used for pre-annotating the Parallel Meaning Bank (Abzianidze et al., 2017). Van Noord et al. (2020) call this system "Pro Boxer". In a sense, Pro Boxer is closest in approach to ours because it makes use of neural taggers for making token-level tagging predictions. It differs from ours and all other systems however in that it is not fully trainable from examples; the translation from tags to DRSs is done via hand-crafted rules. Moreover, it relies on a CCG parser that creates explicit syntactic representation which is perhaps more complexity than needed. As van Noord et al. also point out, the comparison with Pro Boxer is not quite fair because it is the system that produced the PMB pre-annotations and thus profits from anchoring bias.

The results in Table 6 show that our best model beats all available previous scores on the English PMB 3.0.0 test set except for Pro Boxer and van Noord et al. (2020) and is also very competitive on the dev set. Its difference with the state-of-the-art model on the test set is within 1%. Compared with the best previous fully trainable *compositional* model in Bladier et al. (2021), our model improves performance by a large margin.

	en-dev	en-test	de-dev	de-test	it-dev	it-test	nl-dev	nl-test
van Noord et al. (2020)	88.4	89.3	82.4	82.0	80.0	80.5	71.8	71.2
our system	86.4	88.4	79.2	78.3	79.5	80.4	72.5	72.1

Table 7: Comparison of our German, Dutch, and Italian models with prior art (Counter f-scores on PMB 3.0.0)

Results for German, Italian, and Dutch Although most DRS parsers to date have only been evaluated on English, the PMB also contains data in German, Italian, and Dutch. We trained our best model on the German (gold, silver, bronze), Italian (silver, bronze), and Dutch (silver, bronze) data and compared the results with the current state of the art in van Noord et al. (2020), shown in Table 7. The performance of both models is aligned with the amount of data available for each language, and also the proportion of manually corrected (gold) data. Another source of variation (and possible reason for the large gap in accuracy between the two parsers for German) is the choice of pretrained BERT model. For consistency, we only used the cased models that are available in the Hugging Face library, and if possible from the same source.

Compositionality and Its Benefits Is our semantic parser compositional? Bender et al. (2015) provide a definition of compositionality in meaning systems, which we summarize as follows: (1) there is a finite set of atomic word-meaning pairings, (2) there is a finite number of rules combining constituent-meaning pairings into larger constituent-meaning pairings, and any non-atomic constituent-meaning pairing is a function of the constituent-meaning pairings from which it is created and of the rule that creates it, (3) meaning representations are not changed destructively. They argue that compositional aspects of meaning such as predicate-argument structure should be processed by compositional systems, whereas non-compositional aspects such as anaphora or word senses should be handled by different mechanisms. Our parser largely follows these recommendations: ad (1), the fragments that represent abstract word meanings are drawn from a finite set, learned from the training data, while non-compositional word senses, names, etc. are handled by separate mechanisms. Ad (2), our system does away with the notion of constituent by not using syntactic structure, but it is trivial to express the mechanism that combines the word meanings into an utterance meaning in terms of a single rule that iteratively combines adjacent words into larger structures, fulfilling this criterion as well. Ad (3), our combining rule amounts to unifying discourse referents which is perhaps not strictly non-destructive, as it involves renaming them. However, unification can also be expressed in terms of adding variable bindings or combining graphs, so this criterion should be considered fulfilled too. Of course, the post-processing heuristics that are occasionally needed to obtain valid DRSs do not fit into a compositional framework. Furthermore, we do not currently have any dedicated mechanisms to handle partially compositional or non-compositional layers of meaning such as scope or anaphora.

Why care about compositionality in semantic parsing? If the goal of semantic parsing is not merely to automatically obtain a representation of the meaning of an utterance but also to understand why the parser produced that answer, i.e., an explainable and transparent system, compositionality can help. In particular, in the output of our parser, every token is mapped to one of a finite number of meaning fragments (unlike a sequence-to-sequence system where a single token can in principle give rise to an unbounded number of output symbols), every clause belongs to one of these fragments (unlike a sequence-to-sequence system where the output is not usually anchored), and there is a straightforward rule that combines fragments into utterance meanings (unlike sequence-to-sequence systems where the interactions between tokens are opaque). This type of transparency is especially important in human-in-the-loop annotation, where parsers produce an initial annotation and annotators correct them. To do this efficiently and consistenly, annotators need to pinpoint where an error arises, and word-meaning pairings with a finite number of meanings seem a good handle on that. Bender et al. (2015) make a similar argument about grammar-based sembanking, pointing out the consistency, comprehensiveness, and scalability that compositionality afford.

The fact that the accuracy of our compositional DRS parser now almost reaches that of the best

Phenomenon	with	without
NP coordination (2 conjuncts)	85.6	86.1
NP coordination (3 conjuncts)	54.1	87.5
Temporal expression	82.5	86.6
Cardinality	83.9	86.7
Named entity	86.1	86.7
Universal quantification	77.6	86.8
Presupposition	87.5	82.4
Rhetorical relation	84.1	86.5

Table 8: DRS clauses anchored to the conjunction *and* in the phrase *Lungs, heart, veins, arteries, and capillaries*

Table 9: Average f-scores for DRSs with and without certain phenomena

```
...
b1 Sub x1 x2 % and [30...33]
b1 Sub x1 x3 % and [30...33]
b1 Sub x1 x4 % and [30...33]
b1 Sub x1 x5 % and [30...33]
b1 Sub x1 x6 % and [30...33]
...
```

Figure 5: DRS clauses anchored to the conjunction *and* in the phrase *Lungs, heart, veins, arteries, and capillaries*

sequence-to-sequence ones is a big step ahead towards transparent DRS parsing. It is also worth noting that our sequence encoding scheme is equally applicable to incremental parsers, which potentailly afford a greater degree of psycholinguistic plausibility. In addition, the multi-task architecture of our approach is modular and allows for arbitrary additional sequence labeling tasks and factorizations.

6 Error Analysis

We were interested in which semantic phenomena present particular challenges to our parser and thus performed an error analysis of the output of our best model on the English development data, shown in Table 9. Each of the listed phenomena is identified by the presence of a particular type of clause in the gold DRS, such as a Sub relation for coordination, a Quantity relation for quantities, etc. For each phenomenon, we give the f-score for sentences with it vs. sentences without it.

While NP coordination with two conjuncts seems to be handled well, with three conjuncts, accuracy drops dramatically. This can partially be explained by poor generalization of conjunction fragments across different numbers of conjuncts, see, e.g., Figure 5. A realignment step similar to the one we use for first and second person pronouns could help here. Temporal expressions, cardinalities, and named entities all involve the prediction of open-class strings independently of the neural model. Considering that these strings typically only affect a single clause, the underperformance of our parser on sentences involving them is not small, thus improving the predictions—perhaps replacing rules with specialized neural transcoders—could be a worthwhile area for future work. Universal quantification (expressed using the `CONSEQUENCE` relation in DRSs) also correlates with significant difficulties, perhaps due to the diversity of lexical triggers (*one*, *everybody*, *both*, *everything*, *all*, *always*...) and associated fragments. Rhetorical relations present a difficulty because they are often not aligned to a token, therefore not seen in training by our parser. Presupposition on the other hand is correlated with higher scores, presumably because the vast majority of sentences contains at least one definite expression.

To gain a better understanding of common error types, we did an exploratory manual analysis, randomly sampling 100 DRSs produced by our best model on the English development set. Thanks to the compositional model structure, we could easily replicate the PMB-style word-clause alignment in the output, which makes these analyses much easier. The examples we refer to can be found in the Appendix.

In the sample, the most common errors we found were incorrect word senses, for which 36% of the sample DRSs had at least one instance. The second is semantic roles and discourse relations (30%). Despite our intention to separate them into two different sub-tasks, in our sample, these two error types often co-occur (cf. Appendix, Figure 6). In fact, in our sample, we could not find a single case where the predicted word sense of a verb and the predicted semantic roles are not compatible with each other. We hypothesize that correlations between both are learned well by the underlying BERT model, which informs both the fragment classifier and the word sense classifier. In a sense, word sense errors could be expected to be much more frequent than semantic role errors, because word senses form a larger class than verbal fragments. It could be that our model tends to produce internally "consistent" meanings (with matching senses and roles) even at the price of predicting incorrect roles, for which it

is penalized, since Counter does not reward consistency but only correctness. We leave a closer investigation of this hypothesis to future work.

Compared to verbs, noun fragments have less variation, thus we generally observe fewer errors with them. However, there is a noun-related error that consistently occurs in our sample, *viz.* failure to recognize demonyms as such and assign them the corresponding analysis, which involves a presupposed country (cf. Appendix, Figure 7).

Our parser also consistently fails to recognize generic *you* as opposed to deictic *you* (cf. Appendix, Figure 8), which points to the importance of discourse context for understanding the (speaker) meaning of even a single word, and perhaps to something that all current DRS parsers lack: an explicit distinction between sentence meaning and speaker meaning (cf. Bender et al., 2015).

Besides the very large class of word senses, there is also the completely open classes of symbols: names, cardinalities, and times. Our parser predicts them from the corresponding tokens using rule-based heuristics, which we have only implemented for English for now. Simply copying the token often gives the correct symbol, which is partly why we only saw a 1% difference for them in the previous evaluation and why other languages still have acceptable f-scores (the other reason being that Counter arguably underpenalizes incorrect symbols). Of course, things can also go wrong (cf. Appendix, Figure 7).

Finally, we look at fragment predictions with incorrect discourse referent indices, which lead to incorrectly unified discourse referents in the output. The tendency in our sample seems to be that things here go right most of the time, but when they go wrong, they go very wrong, leading to DRSs that are not just incorrect but *invalid* and can thus not be scored by Counter. One way for a DRS to be invalid is to have a loop in its subordination relation, e.g., when two boxes presuppose each other. The way our repair heuristics fix this is to completely delete one of the boxes, and then fix unintroduced referents by introducing new `REF` clauses, and fix a nonconnected subordination relation by introducing `CONTINUATION` relations between boxes (cf. Appendix, Figure 9). Although a bit crude and drastic, these fixing heuristics seem to hurt f-score less than one might expect, for they mainly affect DRSs that were quite wrong to begin with.

7 Conclusions

We have presented the first fully trainable DRS parser that is both competitive with the state of the art and compositional. Unlike sequence-to-sequence models it provides an explicit mapping between tokens and clauses, and fixed fragments ensure consistent analyses. Unlike traditional pipelines, it does not make use of explicit syntactic representations or λ-expressions but uses a simple sequence factorization, and wraps up much of the complexity in a general-purpose BERT model. We argue that these characteristics make our model especially suitable for interactive annotation with humans in the loop, but is also good enough for other applications. Beyond producing more and better data, our error analysis suggests that the next frontier in DRS parsing will involve better modeling of discourse context, and perhaps an explicit separation of sentence meaning and speaker meaning.

Acknowledgments

We would like to thank the three anonymous reviewers for insightful comments. We also thank Laura Kallmeyer for her support. The second author's work on this paper was carried out as part of the research project TreeGraSP, funded by a Consolidator Grant of the European Research Council (ERC).

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Omri Abend and Ari Rappoport. 2013. UCCA: A semantics-based grammatical annotation scheme. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 1–12, Potsdam, Germany. 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.

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

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249, London, UK. Association for Computational Linguistics.

Tatiana Bladier, Gosse Minnema, Rik van Noord, and Kilian Evang. 2021. Improving DRS parsing with separately predicted semantic roles. In *Proceedings of the Workshop on Computing Semantics with Types, Frames and Related Structures*.

Johan Bos. 2021. Variable-free discourse representation structures. https://semanticsarchive.net/Archive/jQzMzJlY/, accessed 2022-02-25.

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.

Gabriella Chronis and Katrin Erk. 2020. When is a bishop not like a rook? when it's like a rabbi! multi-prototype BERT embeddings for estimating semantic relationships. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 227–244, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kilian Evang. 2019. Transition-based DRS parsing using stack-LSTMs. In *Proceedings of the IWCS Shared Task on Semantic Parsing*, Gothenburg, Sweden. Association for Computational Linguistics.

Christiane D. Fellbaum. 2000. Wordnet: an electronic lexical database. *Language*, 76:706.

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324, Brussels, Belgium. Association for Computational Linguistics.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2020. Fast semantic parsing with well-typedness guarantees. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3929–3951, Online. 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.

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

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Viable dependency parsing as sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 717–723, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Rik van Noord, Lasha Abzianidze, Hessel Haagsma, and Johan Bos. 2018a. 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).

Rik van Noord, Lasha Abzianidze, Antonio Toral, and Johan Bos. 2018b. 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.

David Vilares, Michalina Strzyz, Anders Søgaard, and Carlos Gómez-Rodríguez. 2020. Parsing as pretraining. pages 9114–9121.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

A Examples from Error Analysis

Gold (dev):
"Look out !"

b1 REF e1	% Look [0...4]
b1 Experiencer e1 "hearer"	% Look [0...4]
b1 look_out "v.01" e1	% Look [0...4]
	% out [5...8]
	% ! [8...9]

Prediction:
"Look out !"

b1 look "v.01" e1	% Look
b1 Agent e1 "hearer"	% Look
b1 REF e1	% Look

Figure 6: Gold and predicted DRS for the sentence "Look out!" Both the word sense and the semantic role were predicted incorrectly.

a. Gold (dev):
"He 's Argentinian."

b1 REF x1	% He [0...2]
b1 PRESUPPOSITION b2	% He [0...2]
b1 male "n.02" x1	% He [0...2]
b2 REF t1	% 's [2...4]
b2 EQU t1 "now"	% 's [2...4]
b2 Time e1 t1	% 's [2...4]
b2 time "n.08" t1	% 's [2...4]
b2 REF e1	% Argentinian [5...16]
b2 Source e1 x2	% Argentinian [5...16]
b2 Theme e1 x1	% Argentinian [5...16]
b2 be "v.03" e1	% Argentinian [5...16]
b3 REF x2	% Argentinian [5...16]
b3 Name x2 "argentina"	% Argentinian [5...16]
b3 PRESUPPOSITION b2	% Argentinian [5...16]
b3 country "n.02" x2	% Argentinian [5...16]
	% . [16...17]

b. Prediction (original dev):
"He 's Argentinian."

b1 male "n.02" x1	% He
b1 PRESUPPOSITION b2	% He
b1 REF x1	% He
b2 time "n.08" t1	% 's
b2 Time e1 t1	% 's
b2 EQU t1 "now"	% 's
b2 REF t1	% 's
b2 person "n.01" x1	% Argentinian
b2 location "n.01" x2	% Argentinian
b2 REF x1	% Argentinian
b2 Source x1 x2	% Argentinian
b2 Name x2 "argentinian"	% Argentinian
b2 REF x2	% Argentinian
b2 REF e1	% Argentinian

Figure 7: Gold and predicted DRS for the sentence "He's Argentinian". Our parser failed to choose the correct fragment and symbol for the demonym "Argentinian".

Gold (dev):
"You can buy ø stamps at any post~office."

b4 CONDITION b5	%
b2 CONDITION b3	% You [0...3]
b3 REF x1	% You [0...3]
b3 CONSEQUENCE b4	% You [0...3]
b3 person "n.01" x1	% You [0...3]
b1 POSSIBILITY b2	% can [4...7]
b4 REF e1	% buy [8...11]
b4 Agent e1 x1	% buy [8...11]
b4 Theme e1 x2	% buy [8...11]
b4 buy "v.01" e1	% buy [8...11]
b4 REF x2	% stamps [12...18]
b4 stamp "n.04" x2	% stamps [12...18]
b6 Location e1 x3	% at [19...21]
b5 REF x3	% any [22...25]
b5 CONSEQUENCE b6	% any [22...25]
b5 post_office "n.01" x3	% post~office [26...37]
	% . [37...38]

Prediction:
"You can buy stamps at any post~office."

b1 POSSIBILITY b2	% can
b2 buy "v.01" e1	% buy
b2 Agent e1 "hearer"	% buy
b2 REF e1	% buy
b2 Theme e1 x1	% buy
b2 stamp "n.04" x1	% stamps
b2 REF x1	% stamps
b2 Location e1 x2	% at
b2 REF x2	% any
b2 post_office "n.01" x2	% post~office

Figure 8: Gold and predicted DRS for the sentence "You can buy stamps at any post office". Our parser did not recognize "you" as generic as opposed to deictic.

1. Gold (dev):
“ Tom is ø Mary 's stepson.”

b1 REF x1	% Tom [0...3]
b1 Name x1 "tom"	% Tom [0...3]
b1 PRESUPPOSITION b4	% Tom [0...3]
b1 male "n.02" x1	% Tom [0...3]
b4 REF e1	% is [4...6]
b4 REF t1	% is [4...6]
b4 Co-Theme e1 x3	% is [4...6]
b4 EQU t1 "now"	% is [4...6]
b4 Theme e1 x1	% is [4...6]
b4 Time e1 t1	% is [4...6]
b4 be "v.02" e1	% is [4...6]
b4 time "n.08" t1	% is [4...6]
b2 REF x2	% Mary [7...11]
b2 Name x2 "mary"	% Mary [7...11]
b2 PRESUPPOSITION b3	% Mary [7...11]
b2 female "n.02" x2	% Mary [7...11]
b3 REF x3	% 's [11...13]
b3 Of x4 x2	% 's [11...13]
b3 REF x4	% stepson [14...21]
b3 PRESUPPOSITION b4	% stepson [14...21]
b3 Role x3 x4	% stepson [14...21]
b3 person "n.01" x3	% stepson [14...21]
b3 stepson "n.01" x4	% stepson [14...21]
	% . [21...22]

2. Prediction (Loop):
“Tom is Mary 's stepson.”

b1 male "n.02" x1	% Tom
b1 PRESUPPOSITION b2	% Tom
b1 Name x1 "tom"	% Tom
b1 REF x1	% Tom
b2 time "n.08" t1	% is
b2 be "v.02" e1	% is
b2 REF e1	% is
b2 Time e1 t1	% is
b2 Theme e1 x1	% is
b2 Co-Theme e1 x2	% is
b2 EQU t1 "now"	% is
b2 REF t1	% is
b3 female "n.02" x3	% Mary
b3 PRESUPPOSITION b4	% Mary
b3 Name x3 "mary"	% Mary
b3 REF x3	% 's
b4 PRESUPPOSITION b2	% stepson
b4 REF x2	% stepson
b4 User x2 x3	% stepson
b4 person "n.01" x1	% stepson
b4 driver's_license "n.01" x2	% stepson
b4 PRESUPPOSITION b3	% stepson
b4 Role x1 x2	% stepson
b4 REF x1	% stepson

3. Prediction (Disconnects):
“Tom is Mary 's stepson.”

b1 male "n.02" x1	% Tom
b1 PRESUPPOSITION b2	% Tom
b1 Name x1 "tom"	% Tom
b1 REF x1	% Tom
b2 time "n.08" t1	% is
b2 be "v.02" e1	% is
b2 REF e1	% is
b2 Time e1 t1	% is
b2 Theme e1 x1	% is
b2 Co-Theme e1 x2	% is
b2 EQU t1 "now"	% is
b2 REF t1	% is
b3 female "n.02" x3	% Mary
b3 PRESUPPOSITION b4	% Mary
b3 Name x3 "mary"	% Mary
b3 REF x3	% 's
b2 REF x2	

4. Prediction (Postprocessing fixed):
“Tom is Mary 's stepson.”

b1 male "n.02" x1	% Tom
b1 PRESUPPOSITION b2	% Tom
b1 Name x1 "tom"	% Tom
b1 REF x1	% Tom
b2 time "n.08" t1	% is
b2 be "v.02" e1	% is
b2 REF e1	% is
b2 Time e1 t1	% is
b2 Theme e1 x1	% is
b2 Co-Theme e1 x2	% is
b2 EQU t1 "now"	% is
b2 REF t1	% is
b3 female "n.02" x3	% Mary
b3 PRESUPPOSITION b4	% Mary
b3 Name x3 "mary"	% Mary
b3 REF x3	% 's
b2 REF x2	
b2 CONTNUATION b3	

Figure 9: Gold, predicted, and fixed DRSs for the sentence “Tom is Mary’s stepson”. The initial prediction is invalid because boxes `b3` and `b4` presuppose each other. This is fixed by completely deleting `b4`, which leaves an unintroduced referent `x2` and two unconnected boxes `b2` and `b3` behind. These errors are fixed, respectively, by introducing a `REF` clause for `x2` where it first occurs (in `b2`) and introducing a `CONTINUATION` relation between `x2` and `x3`.