Understanding Disrupted Sentences Using Underspecified Abstract Meaning Representation

Angus Addlesee[1], *Marco Damonte*[2]

[1]Heriot-Watt University, United Kingdom
[2]Amazon Alexa, United Kingdom

a.addlesee@hw.ac.uk, dammarco@amazon.co.uk

Abstract

Voice assistant accessibility is generally overlooked as today's spoken dialogue systems are trained on huge corpora to help them understand the 'average' user. This raises frustrating barriers for certain user groups as their speech shifts from the average. People with dementia pause more frequently mid-sentence for example, and people with hearing impairments may mispronounce words learned post-diagnosis. We explore whether semantic parsing can improve accessibility for people with non-standard speech, and consequently become more robust to external disruptions like dogs barking, sirens passing, or doors slamming mid-utterance. We generate corpora of disrupted sentences paired with their underspecified Abstract Meaning Representation (AMR) graphs, and use these to train pipelines to understand and repair disruptions. Our best disruption recovery pipeline lost only 1.6% graph similarity f-score when compared to a model given the full original sentence.

Index Terms: accessibility, semantic parsing, spoken dialogue systems, human-computer interaction

1. Introduction

People use voice assistants to set timers while their hands are dirty preparing food, or to turn up the volume of the TV from the warmth of their bed. These simple conveniences are undoubtedly pleasant, but they are also critical for the mental well-being of certain user groups [1]. People with dementia can seamlessly access their music [2], and people with limited mobility can regain their independence. Instead of having to ask a family member or carer to turn off their light before bed, they can do this whenever they want with just their voice. Charities promote the use of voice assistants for this reason, and the creators develop features targeting these specific user groups [1].

With this in mind, the accessibility of today's systems must be considered as speech production differs between the 'average' user and the user groups that, while benefiting the most from voice assistants, remain a minority in huge training datasets. For example, people with dementia pause more frequently and for longer durations mid-sentence due to word-finding problems [3, 4]. Non-standard speech is actually produced due to many different diseases and conditions like early-stage motor neurone disease, muscular dystrophy, down syndrome, etc... Even seemingly unrelated problems lead to speech production changes. For example, it is difficult to learn how to say new words with hearing difficulties, so people with hearing impairments often mispronounce words [5].

Similar to human speech processing, automatic speech recognition (ASR) can signal when it struggles to recognise a word in a spoken utterance [6]. This is indicated with a low word-level confidence score, much like when people know they are unsure of a particular word their interlocutor said. Both humans and ASR systems try to use the surrounding words as context to determine what word was likely uttered [7, 8], but unlike human speech processing, today's voice assistants do not attempt to repair instances in which the ASR word confidence remains low [9]. The natural language understanding components are sent the utterance text regardless, and users have to frustratingly repeat their entire utterance again if the mystery word's identity was guessed incorrectly [9, 10].

We propose that adapting voice assistants to be capable of semantically parsing and repairing spoken utterances, with low-confidence tokens, will make them more accessible for many user groups. Additionally, words are disrupted by dogs barking, sirens passing, doors slamming, or other family members mid-utterance [11]. This work can therefore improve robustness to noisy environments, benefiting all users. In this paper we generate several corpora to evaluate disruption recovery strategies, and experiment to explore the feasibility of such a solution.

In order to understand and repair sentences that are disrupted, we must carefully consider our choice of meaning representation language (MRL). It must be able to handle: *incrementality*, allowing structurally incomplete meanings to be established over time; and *conjunction*, enabling the semantic representation of both the disrupted utterance and follow-up repair to be consolidated into the representation of the full sentence. The chosen MRL must also be *transparent*, allowing system designers to establish reasons for particular behaviour. This is vital when interacting with potentially vulnerable groups, and for downstream reasoning that also improves accessibility [1].

One formalism that satisfies the above desiderata is the graph-based Abstract Meaning Representation (AMR) [12], shown in Figure 1. Previous work on incremental AMR parsing has exploited its underspecification and conjunction properties [13] that we require. AMR is also transparent by design [12], and has been successfully used for downstream reasoning [14, 15]. Finally, AMR has been used to pre-train large language models, enabling state-of-the-art (SotA) semantic parsing and text generation [16, 17] without sacrificing transparency.

The most recent AMR corpus, called AMR 3.0 (LDC2020T02) [18], contains over 59,000 sentences paired with their AMR graph representations. AMR can represent *all* sentences, taking *all* words into account in a reasonably consistent manner, and this makes it a popular resource for the semantic parsing task. Two AMR parsing models are currently the non-ensemble SotA: AMRBART [17] and ATP [19]. These are closely followed by SPRING [16], the SotA without using additional training data. In fact, ATP actually uses the SPRING model, but outperforms it by 1% by training it on auxiliary tasks. AMRBART could not be retrained on modified AMR corpora due to issues open in their GitHub repository, we therefore re-implemented the SPRING system as our AMR parsing baseline on a P3 AWS machine. The disruption recovery experiments in this paper will be compared to this baseline as an upper bound, with an ideal pipeline understanding and repairing disrupted sentences as successfully as the SPRING model parses the full original sentence. The SPRING model relies on the BART-Large [20] pre-trained large language model, further fine-tuned on linearised AMR graphs with the RAdam optimiser [21]. The novel linearisation algorithm was then used by both AMRBART and ATP.

In this paper we: (1) show that token-masking improves a model's ability to generate underspecified AMR; and (2) build, evaluate, and analyse recovery pipelines that repair disrupted sentences. Our top-performing pipelines can repair sentences spoken by people with non-standard speech, or spoken in a noisy environment, only losing 1.6% graph similarity f-score.

2. Disrupting AMR

To evaluate disruption repair pipelines, we need a corpus of disrupted sentences paired with their underspecified AMR representations and completions. As this does not exist, we have generated several corpora from the original AMR 3.0 corpus to run our experiments. In this section we detail the general approach. Experiment-specific details are in their respective sections[1]

Each word in a sentence carries specific meaning, which is then represented by nodes and/or edges in an AMR graph. We must therefore ensure that when we disrupt words in the text, it is the semantic meaning of those exact words that we underspecify in the graph. For this, we have re-implemented a recent SotA AMR alignment model [22]. In Figure 1 we show a coloured diagram of a text/AMR alignment to illustrate our disruption approach. If we chose to disrupt the word "invented" in this example, the alignment model would identify which edges and nodes need to be underspecified in the AMR (dark blue edge and node in Figure 1). Following similar work to represent incomplete instructions given to robots [23], we take advantage of underspecification in AMR to represent the missing information with a 'NOTKNOWN' argument. If this tag is present in our model's semantic parse, information must be missing due to disruption in the spoken utterance, and repair is required.

In order for our recovery pipelines to be successful, they need to first parse a disrupted sentence into AMR. The underspecified graph should not just identify that information is missing, but critically, where that missing information belongs in the graph structure. From Cognitive Science we know that clarification requests are used to communicate and deal with misunderstandings on the fly by eliciting a repair from the interlocutor [25]. Our pipeline must therefore also correctly parse this repair, and then conjoin the two AMR parses into its full form – ideally the correct AMR representation of the full sentence.

3. Representing Interrupted and Disrupted Sentences

As detailed in Section 1, we can detect disrupted words in a spoken utterance (e.g. mispronunciations or door-slamming) as ASR will output predictions with a low word-level confidence score [6]. Words with a low score can be replaced with an '[UNK]' mask in this case. People pause mid-sentence due to word-finding problems however, and this is particularly common in speech produced by people with dementia. Words are therefore entirely missing from the ASR output and cannot be replaced by a mask [10]. In this section we: (1) check that underspecified AMR can be generated when no mask is present, ensuring people with dementia can benefit from semantic repair pipelines; and (2) confirm our hypothesis that the mask tokens do improve model performance.

Interrupted Corpus	Details	Smatch on Test Set
with NO mask	Full corpus	82.2
with NO mask	Incomplete	**80.5**
with mask	Full corpus	**82.4**
with mask	Incomplete	79.8

Table 1: *SPRING models trained on interrupted AMR corpora. 'Full' indicates that incomplete, complete, and repair sentences are present in the corpus (as opposed to incomplete only).*

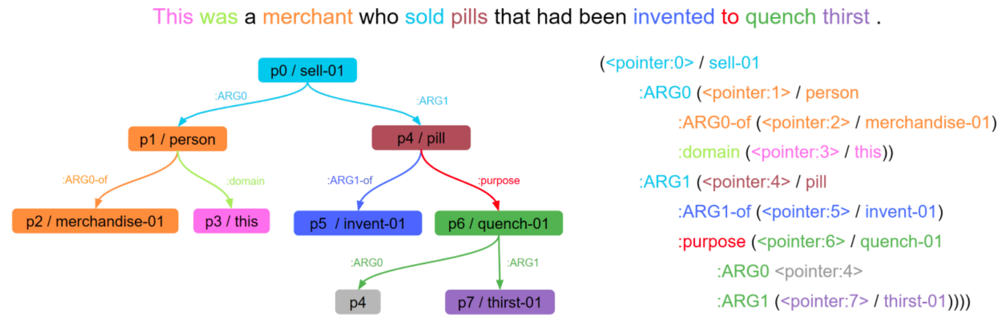
Word-finding problems, and therefore long mid-utterance pauses, typically precede nouns [26, 27], and this linguistic observation has been used to improve entity recognition on Siri's user data in English and French [28]. With this in mind, we created two 'interrupted' corpora where the missing words are nouns at the ends of sentences. We consistently refer to these corpora as 'interrupted' as opposed to 'disrupted', given that disruptions can occur at any point during a spoken utterance. The process in Section 2 was used to create two interrupted corpora, with the only difference being the presence of a token mask or not. For example, "I put the book on the" vs "I put the book on the UNK". Given the interruption constraints, the corpora were smaller than the original AMR corpus, each containing 13,896 train, 654 dev, and 693 test instances. For error analysis, we additionally train models on *only* the incomplete sentences in the interrupted corpus, this is one third of the interrupted corpus (as it excludes full sentences and repair turns). Following all previous AMR literature, we use Smatch [29] as the evaluation metric to measure the semantic overlap between the predicted and gold AMR graphs (graph similarity f-score).

We retrained four SPRING models for comparison, the results and differences are detailed in Table 1. Interestingly, when the model is only shown incomplete sentences paired with underspecified AMR, the corpus without masking outperforms the model that is given the mask. We argue that this is because SPRING exploits the BART-Large pre-trained large language model [20] which has never seen the 'UNK' mask previously. We confirm this later when training our disruption models with the much larger 'disrupted' corpora. Turning to the full corpus results, we can see that the absence of a mask impacts performance. We expect a recovery pipeline to identify when repair is needed in an interaction. Without a mask, the model confuses incomplete sentences with full sentences.

In this section we have determined that underspecified meaning representations can be generated from incomplete sentences without the presence of a mask. We see later, however, that the confusion with full sentences strongly impacts the full interruption recovery pipeline designed for people with dementia. In addition, we have shown that for other user groups and in noisy environments, where masking is possible, the semantic parser performs promisingly well.

We have only explored 'interrupted' sentences so far, but disruptions can occur at any point during a spoken utterance. We therefore used the process in Section 2 to disrupt sentences in the original AMR 3.0 corpus. Given that some sentences are

[1] Our AMR disruption code and example dialogues can be found at: `https://github.com/amazon-science/disrupt-amr`



This was a merchant who sold pills that had been invented to quench thirst .

```
(<pointer:0> / sell-01
    :ARG0 (<pointer:1> / person
        :ARG0-of (<pointer:2> / merchandise-01)
        :domain (<pointer:3> / this))
    :ARG1 (<pointer:4> / pill
        :ARG1-of (<pointer:5> / invent-01)
        :purpose (<pointer:6> / quench-01
            :ARG0 <pointer:4>
            :ARG1 (<pointer:7> / thirst-01))))
```

Figure 1: *A colour coded diagram of text/AMR alignment [24]. The sentence is at the top, the AMR graph is on the left, and the linearised text representation of the graph is on the right. The word "quench" in green is represented by the green nodes and edges.*

very long, we disrupted some sentences multiple times, generating multiple underspecified parses. To give a simple example, generating: "I put the book on the UNK" and "I put the UNK on the table". This resulted in a much larger corpus containing 76,168 train, 4,155 dev, and 4,451 test instances.

4. Building Disruption Recovery Pipelines

The interrupted corpus *with masks* was only required for the experiments in Section 3. We want to build recovery pipelines for accessibility purposes, and interruption masks are not possible, so we will therefore evaluate interruption pipelines with no masks for people with dementia, and disruption pipelines for the other groups discussed in Section 1, where masks are possible.

ASR outputs low word-level confidence scores that can be masked before semantic parsing, so a disruption is known to have occurred before recovery is initiated. A model trained on full sentences and incomplete sentences could parse all user utterances, but a pipeline with specialised models could be used instead. That is, the disrupted sentence could be parsed by a model that has only seen disrupted sentences during training, and the repair could then be parsed by another model that has only been trained on repair utterances. This 'split' pipeline contains specialised models, but each would be trained on less data.

Given the surrounding context of a misunderstood word, humans sometimes try to predict what was said [7, 8]. We hypothesise that guessing the misunderstood word may frustrate the user further when interacting with a voice assistant, but we deemed it was important to include this human-interaction approach for completeness. We fine-tuned two T5 models [30] on our corpora, as they are particularly good at text generation [31, 32]. One model was trained to complete interrupted utterances, and one to repair disrupted sentences.

Finally, we built a naive pipeline to confirm that our other recovery pipelines successfully repair disruptions. This pipeline parsed both the incomplete sentence and repair turn with the original SPRING model trained on AMR 3.0. It then conjoined the two parses at the root node (e.g. the 'top' in Figure 1).

5. Pipeline Evaluation

With each approach detailed in Section 4, we evaluated the following pipelines against their respective Upper Bounds (UB):

- Interrupted: UB - the SPRING model trained on the original AMR 3.0 corpus, given only full sentences in the interrupted corpus. Interrupted pipelines aim to match this upper bound.
- Interrupted: All - the interrupted sentence and repair turn are parsed by SPRING trained on the full interrupted corpus.
- Interrupted: Prediction - the interrupted sentence is completed by prediction using the T5 model fine-tuned on the interrupted corpus. This is then parsed by the SPRING model trained on the original AMR 3.0 corpus.
- Interrupted: Naive - the interrupted sentence and repair turn are both parsed by the SPRING model trained on the original AMR 3.0 corpus, and conjoined at the root node.
- Disrupted: UB - the SPRING model trained on the original AMR 3.0 corpus, given only full sentences in the disruption corpus. Disrupted pipelines aim to match this upper bound.
- Disrupted: All - the disrupted sentence and repair turn parsed by the SPRING model trained on the full disrupted corpus.
- Disrupted: Split - the disrupted sentence and repair turn parsed by specialised SPRING models (only possible for disruptions, due to masks, as aforementioned in Section 4).
- Disrupted: Prediction - the disrupted sentence is completed by prediction using the T5 model fine-tuned on the disrupted corpus. This is then parsed by the SPRING model trained on the original AMR 3.0 corpus.
- Disrupted: Naive - the disrupted sentence and repair turn are both parsed by the SPRING model trained on the original AMR 3.0 corpus, and conjoined at the root node.

Pipeline evaluation results can be found in Table 2. If we first explore the 'Interrupted' pipeline results, we can see that the model confusion identified in Section 3 has a serious impact on performance. That is, with no tokens to identify whether a sentence is interrupted or not, the semantic parser confuses incomplete sentences for complete sentences – failing to output an underspecified AMR graph. Both the prediction and naive recovery approaches outperform the semantic parsing model for this reason. In Section 6 we run an error analysis to explore what type of semantic information our model is failing to parse.

The disruption recovery pipelines perform remarkably well (this contrast is explored in Section 6). It is worth highlighting that the naive pipeline is passed the same incomplete sentence and repair turn as the recovery pipelines, the main difference is how the representations are conjoined. Our recovery pipelines outperforming the naive pipeline, by up to 4.6%, highlights that our semantic parser accurately identifies where the missing information belongs in the semantic structure of the sentence.

The prediction pipeline generally repaired sentences with sensible completions, but as expected, these predictions were often arbitrary guesses due to ambiguity. To illustrate this point, consider completing the sentence: "She drove to UNK".

Finally, it appears that specialist models outperform more general semantic parsing models, even though the general

Pipeline	**Smatch**	**Smatch Loss**
Interrupted: UB	86.6	-
Interrupted: All	81.3	5.3
Interrupted: Prediction	82.3	4.3
Interrupted: Naive	**83.2**	**3.4**
Disrupted: UB	84.7	-
Disrupted: All	82.8	1.9
Disrupted: Split	**83.1**	**1.6**
Disrupted: Prediction	78.6	6.1
Disrupted: Naive	78.5	6.2

Table 2: *Evaluation of interruption and disruption recovery pipelines. The 'Smatch Loss' is the difference between the pipelines Smatch score and its respective upper bound, 'UB', score. The ideal loss would be 0, as that would indicate that the recovery pipeline successfully repaired all sentences.*

model was trained with a larger amount of data. From the results, we can conclude that when a system's ASR outputs an utterance with a disruption (indicated by low word-level confidence), this disrupted sentence should be sent to a specialist semantic parsing model trained specifically to generate underspecified meaning representations. This can make a voice assistant more accessible for people with non-standard speech, and more robust for use in noisy environments like a family home.

6. Findings from Deeper Analysis

Smatch is the standard AMR evaluation metric, but newer metrics were designed to provide a more fine-grained evaluation of AMR semantic parser performance [13]. These include scores measuring the parser's ability to predict graph structure, named entities, negation, concepts, and more. We ran these metrics across all of our pipelines to measure what certain pipelines parsed successfully, and what can be improved.

The interruption recovery pipeline performed very poorly, being outperformed by both the prediction and naive pipelines. Upon further inspection, the 'Interruption: All' pipeline struggled to parse negation and reentrancies when compared to the prediction and naive pipelines. We determined that this was simply due to the fact that the prediction and naive pipelines used the SPRING model trained on the original AMR 3.0 corpus, and that the interrupted corpus must contain fewer negations and reentrant edges. We trained a pipeline using the interrupted corpus *with masking* from Section 3 to confirm this pattern holds true. This is a positive finding, as AMR datasets are supersets of their older versions (AMR 2.0 is a subset of the larger AMR 3.0 corpus for example). In future, given a larger AMR corpus, the same interruption code could be used to recover interrupted sentences uttered by people with dementia.

Sticking with the interruption recovery results, the naive pipeline outperformed the prediction approach. We discovered that this was due to a large drop in the prediction pipelines ability to parse named entities, with a recall score drop of 8%. This makes sense following reasoning explained in Section 5. Nouns are often impossible to predict, resulting in the named entity being missing from the predicted AMR graph.

The disruption pipelines do not share the interruption pipeline's struggles with negation and reentrancies, further confirming our suspected diagnosis. In fact, the disruption recovery pipelines outperformed their respective prediction and naive pipelines at parsing negation, named entities, and unlabeled graph structure. The naive approach was particularly bad at predicting the semantic structure of the graph (an 8% precision drop). This is expected as the two parses are arbitrarily conjoined at the root node, resulting in structural issues.

7. Conclusion

In this paper we motivate the need for interruption and disruption recovery pipelines for voice assistant accessibility, and for general robustness in noisy environments like family homes. We found that disruption recovery is possible with a general model (one that can accurately parse disrupted sentences in addition to complete sentences), and with a pipeline containing specialist models (parsers specifically tailored to their specific role, e.g. parsing only disrupted sentences).

Our top-performing pipeline lost only 1.6% Smatch when compared to the SPRING model given the full utterances. This recovery pipeline had to parse the disrupted utterance, correctly identify where the missing information belongs in the semantic graph structure of the sentence, parse the repair utterance, and conjoin these representations to recover the full semantic graph.

As explained in Section 1, ASR may output words with low confidence scores due to mispronunciation or background noise. We all mispronounce words (e.g. when asking the weather in Llanfairpwllgwyngyll), but people with speech impairments, muscular dystrophy, early-stage motor neurone disease, and even hearing impairments commonly mispronounce words [1]. Similarly, it is difficult to hear a word mid-utterance when a dog barks. This paper confirms that it is possible to repair misunderstandings naturally when this occurs.

We will release the code to generate all of the corpora used in this paper for reproducibility, and we will also release the hyperparameters used to fine-tune the T5 prediction models.

8. Future Work

To confirm that our pipelines do improve accessibility in practice, a user study must be designed, go through meticulous ethical approval, be carried out, and analysed. In this paper we showed that our approach was feasible, but response generation would also be needed for an actual user study. This response generator was out of the scope of this work, but would generate a clarification request to elicit the repair turn from the user.

From a very small number of test cases, we found that chatGPT was surprisingly good at generating clarification requests given only a few examples. We could not use chatGPT for our entire disruption recovery pipeline, however, as it breaks our requirement for MRL transparency. This is absolutely critical when designing systems to be accessible, as the target groups commonly contain vulnerable members. We must be able to establish reason for system behaviour, and control this behaviour to avoid harming the user. ChatGPT could potentially be used only to generate the clarification request, but further analysis is needed. We plan to evaluate which response generation approach would be the most appropriate (for example chatGPT or AMR to text generation [16, 17]). We are also working to use the findings in this paper in a future user study.

9. Acknowledgements

This project was completed during an Amazon Alexa internship in Cambridge. We would like to thank our colleagues for feedback and advice during the completion of this work. This work was also partially funded by the EU H2020 program under grant agreement no. 871245 (`https://spring-h2020.eu/`).

10. References

[1] A. Addlesee, "Voice assistant accessibility," in *Proceedings of The 13th International Workshop on Spoken Dialogue Systems (IWSDS)*, 2023.

[2] A. Pradhan, K. Mehta, and L. Findlater, ""accessibility came by accident" use of voice-controlled intelligent personal assistants by people with disabilities," in *Proceedings of the 2018 CHI Conference on human factors in computing systems*, 2018, pp. 1–13.

[3] V. Boschi, E. Catricala, M. Consonni, C. Chesi, A. Moro, and S. F. Cappa, "Connected speech in neurodegenerative language disorders: a review," *Frontiers in psychology*, vol. 8, p. 269, 2017.

[4] A. Slegers, R.-P. Filiou, M. Montembeault, and S. M. Brambati, "Connected speech features from picture description in alzheimer's disease: A systematic review," *Journal of Alzheimer's Disease*, vol. 65, no. 2, pp. 519–542, 2018.

[5] H. Pimperton and C. R. Kennedy, "The impact of early identification of permanent childhood hearing impairment on speech and language outcomes," *Archives of disease in childhood*, vol. 97, no. 7, pp. 648–653, 2012.

[6] D. Oneaţă, A. Caranica, A. Stan, and H. Cucu, "An evaluation of word-level confidence estimation for end-to-end automatic speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 258–265.

[7] M. Purver, J. Ginzburg, and P. Healey, "On the means for clarification in dialogue," in *Current and new directions in discourse and dialogue*. Springer, 2003, pp. 235–255.

[8] C. Howes, P. G. Healey, M. Purver, and A. Eshghi, "Finishing each other's... responding to incomplete contributions in dialogue," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 34, 2012.

[9] M. Nakano, Y. Nagano, K. Funakoshi, T. Ito, K. Araki, Y. Hasegawa, and H. Tsujino, "Analysis of user reactions to turn-taking failures in spoken dialogue systems," in *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, 2007, pp. 120–123.

[10] A. Addlesee and M. Damonte, "Understanding and answering incomplete questions," in *Proceedings of the 5th Conference on Conversational User Interfaces*, 2023.

[11] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples, "Voice interfaces in everyday life," in *proceedings of the 2018 CHI conference on human factors in computing systems*, 2018, pp. 1–12.

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

[13] M. Damonte, S. B. Cohen, and G. Satta, "An incremental parser for abstract meaning representation," in *15th EACL 2017 Software Demonstrations*. Association for Computational Linguistics (ACL), 2017, pp. 536–546.

[14] J. Lim, D. Oh, Y. Jang, K. Yang, and H.-S. Lim, "I know what you asked: Graph path learning using amr for commonsense reasoning," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2459–2471.

[15] P. Kapanipathi, I. Abdelaziz, S. Ravishankar, S. Roukos, A. Gray, R. F. Astudillo, M. Chang, C. Cornelio, S. Dana, A. Fokoue-Nkoutche *et al.*, "Leveraging abstract meaning representation for knowledge base question answering," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 3884–3894.

[16] M. Bevilacqua, R. Blloshmi, and R. Navigli, "One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 12 564–12 573.

[17] X. Bai, Y. Chen, and Y. Zhang, "Graph pre-training for amr parsing and generation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6001–6015.

[18] K. Knight, B. Badarau, L. Baranescu, C. Bonial, M. Bardocz, K. Griffitt, U. Hermjakob, D. Marcu, M. Palmer, T. O'Gorman *et al.*, "Abstract meaning representation (amr) annotation release 3.0," 2021.

[19] L. Chen, P. Wang, R. Xu, T. Liu, Z. Sui, and B. Chang, "Atp: Amrize then parse! enhancing amr parsing with pseudoamrs," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 2482–2496.

[20] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, 2020, pp. 7871–7880.

[21] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *International Conference on Learning Representations*, 2020.

[22] A. Drozdov, J. Zhou, R. Florian, A. McCallum, T. Naseem, Y. Kim, and R. F. Astudillo, "Inducing and using alignments for transition-based amr parsing," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 1086–1098.

[23] H. Chen, H. Tan, A. Kuntz, M. Bansal, and R. Alterovitz, "Enabling robots to understand incomplete natural language instructions using commonsense reasoning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1963–1969.

[24] P.-L. H. Cabot, A. C. M. Lorenzo, and R. Navigli, "Amr alignment: Paying attention to cross-attention," *arXiv preprint arXiv:2206.07587*, 2022.

[25] P. G. Healey, G. J. Mills, A. Eshghi, and C. Howes, "Running repairs: Coordinating meaning in dialogue," *Topics in cognitive science*, vol. 10, no. 2, pp. 367–388, 2018.

[26] B. Croisile, B. Ska, M.-J. Brabant, A. Duchene, Y. Lepage, G. Aimard, and M. Trillet, "Comparative study of oral and written picture description in patients with alzheimer's disease," *Brain and language*, vol. 53, no. 1, pp. 1–19, 1996.

[27] F. Seifart, J. Strunk, S. Danielsen, I. Hartmann, B. Pakendorf, S. Wichmann, A. Witzlack-Makarevich, N. H. de Jong, and B. Bickel, "Nouns slow down speech across structurally and culturally diverse languages," *Proceedings of the National Academy of Sciences*, vol. 115, no. 22, pp. 5720–5725, 2018.

[28] S. Dendukuri, P. Chitkara, J. R. A. Moniz, X. Yang, M. Tsagkias, and S. Pulman, "Using pause information for more accurate entity recognition," in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, 2021, pp. 243–250.

[29] S. Cai and K. Knight, "Smatch: an evaluation metric for semantic feature structures," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 748–752. [Online]. Available: https://www.aclweb.org/anthology/P13-2131

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[31] L. F. Ribeiro, M. Schmitt, H. Schütze, and I. Gurevych, "Investigating pretrained language models for graph-to-text generation," in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, 2021, pp. 211–227.

[32] V. M. Andreas, G. I. Winata, and A. Purwarianti, "A comparative study on language models for task-oriented dialogue systems," in *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*. IEEE, 2021, pp. 1–5.