UNIVERSITY OF CALIFORNIA

Los Angeles

The Joint Training of Transition-Based AMR Parser

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Computer Science

by

Guangxuan Xu

2022

ABSTRACT OF THE THESIS

The Joint Training of Transition-Based AMR Parser

by

Guangxuan Xu
Master of Science in Computer Science
University of California, Los Angeles, 2022
Professor Nanyun Peng, Chair

Abstract Meaning Representation(AMR) parsing converts a natural language sentence into a specially designed semantic graph(AMR), which captures the most essential semantic entities and relations of the input sentence. While the recent introduction of pretrained sequence-to-sequence models have brought performance improvement and pipeline simplification, the problem of how to best encode structural information into seq2seq models remains. This exploratory work proposes joint training of transition-based AMR parsers that incorporates not only the parsing objective, but also a denoising objective into training; it seeks to answer whether the improved understanding of structural alignment can benefit sequence-to-sequence AMR parsers. It also shows potential application of the joint-trained models: the joint-training setup can greatly liberate the transition-based parsers from State Machine's alignment constraints and allow them to be easily repurposed for a set of related tasks that could theoretically benefit from the structural training, such as paraphrase generation and generation from keywords.

The thesis of Guangxuan Xu is approved.

Kai-wei Chang

Baharan Mirzasoleiman

Nanyun Peng, Committee Chair

University of California, Los Angeles

2022

To my beloved mother . . .

and father . . .

who accompanied and supported me for ups and down as always

To my dearest grandpa . . .

whose love is my eternal source of inspiration

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This work receives tremendous inspiration and guidance from my IBM Research mentors and colleagues, Dr.Tahira Naseem, Dr.Ramón Fernandez Astudillo, Dr.Young-Suk Lee, Dr.Radu Florian, Dr. Salim Roukos. I also appreciate Jiawei Zhou's detailed documentation and open access to code repository for the Structured BART paper, on which joint-training is based upon. Moreover, I also want sincerely thank my academic advisor Prof. Nanyun Peng for providing me the freedom to choose research topic, supporting my research direction, providing guidance, funding, and facility that make this research possible. Lastly, I appreciate all members of my thesis committee, Prof. Nanyun Peng, Prof. Baharan Mirzasoleiman, and Prof. Kai-wei Chang for their time and effort in reviewing and advising this thesis.

VITA

2017-2021	B.A. (Computer Science) and B.A. (Political Science), Wesleyan University.
2020	Research Intern and Exchange Student, Dartmouth College. Worked on mitigation of political bias under the direction of Dr. Ruibo Liu and Prof.Soroush Vosoughi.
2021	Research Intern (NLP), Tsinghua University, Beijing, China.
2021-2022	M.S. (Computer Science), UCLA.
2021–2022	Graduate Student Researcher, Computer Science Department, UCLA. Worked on NLP fairness, dialogue system evaluation, information extraction under direction of Professor Nanyun Peng.
2023 -	Research Software Engineer in AI, IBM Research, Yorktown Heights, NY. I work in the Multi-Lingual NLP group under the supervision of Senior Research Manager Radu Florian.

PUBLICATIONS

Enhanced Offensive Language Detection Through Data Augmentation. 2020. The International AAAI Conference on Web and Social Media.

Data boost: Text data augmentation through reinforcement learning guided conditional generation. 2020. Proceedings of the 2020 Conference on Empirical Methods in Natural

Language Processing.

Mitigating Political Bias in Language Models through Reinforced Calibration. 2021. Proceedings of the AAAI Conference on Artificial Intelligence, 2021.

On the safety of conversational models: Taxonomy, dataset, and benchmark 2021. Findings of the Association for Computational Linguistics 2022

Can Model Compression Improve NLP Fairness 2022. Arxiv

Quantifying and alleviating political bias in language models 2022. Artificial Intelligence, Volume 304, Pages 103654, Publisher Elsevier

Non-Parallel Text Style Transfer with Self-Parallel Supervision 2022. International Conference on Learning Representations

EnDex: Evaluation of Dialogue Engagingness at Scale 2022. Findings of EMNLP 2022

CHAPTER 1

Introduction

1.1 Motivation

Abstract Meaning Representation(AMR) Parsing is one of the most popular and well-studied approach of semantic parsing. Transition-based parsing[Niv03] is a common strategy to enforce structural guidance for graph generation. This work is built upon the current SOTA approach that combines transition-based guidance and a sequence-to-sequence BART model[LLG20] to parse AMR graphs[ZNF21]. Compared with non-transition based methods that directly finetune on a seq-to-seq model, having the transition-based constraint at each time step of the generation have the added benefit of inductive bias from incremental graph construction and a guarantee for graph well-formedness. Moreover, lacking structural guidance of transtion-based systems, non-transition based methods rely on large amount of training data, which is unfortunately very expensive to obtain, requiring expert knowledge. A number of complex preprocessing, postprocessing, and dependencies were introduced to AMR parsers to address data scarcity and lack of word-wise alignment issue; the State Machine in transition-based parsers also introduces significant complexity for modeling and inference, since it is dynynamically updated at each time step to create target-side vocabulary masking. High performance AMR parsers are often very complex system, requiring dependencies and multi-step procedures to train and use.

Meanwhile, AMR parsers are powerful tools that could find promising application for many domains. The AMR graph was designed in the background when syntactic parsers are already in widespread usage, but semantic tools are still in the state of "Balkanization," where correference resolution(CR), named entity recognition(NER), temporal relations prediction were independently developed with distinctly different annotation data and evaluation procedure. AMR parsing is an attempt to bring together those broadly categorized semantic tasks into one comprehensive umbrella and evaluate their performance together in one setting. From a users' perspective, they no longer need to understand the specific detail, and evaluation methods for all the semantic subtasks, and run multiple tools to assemble a pipeline; AMR parsing can come in to make comprehensive and general predictions on salient entities and important relations in sentences, and summarize them into the AMR graph format. For example, in an automatic diagnosis system that takes as input a textual medical case of illness for patients; it needs to automatically identify the disease and syndromes suffered by the target patients. AMR parsing is able to identify the disease and patients as entities, and predict semantic relations between those them; so such system may directly extract the needed information from an AMR parser. While it is also true that AMR do not cover all the relations that the users care about, reseachers have developed domain-specific annotation of AMR graph to support specialized application, such as Biomedical AMR[RMK17] and Dialogue AMR[BDA20].

1.2 Contributions

This work proposes a joint-training scheme built upon the SOTA transition-based AMR parsing model. Its contribution can be summarized as follows: 1. It proposes the first joint-trained transition-based AMR parser and have performed extensive experiments to explore its impact on structural learning and AMR parsing results.
2. It proposes an updated State Machine, Oracle, and linearization actions sets, which reduces constraint for input-to-graph alignment and enable more flexibility and expressiveness of the AMR model. 3. The prototype joint-trained model shows promising results in para-

phrase generation and generation from keywords tasks, and this could be a promising areas of research to explore the effect of structural training on generation tasks.

1.3 Thesis Statement

This work proposes a novel joint-training scheme for transition-based AMR parsers, and explores 2 research questions: 1.Whether joint-training allows better learning of structural alignment and improve parsing performance? 2.Whether joint-trained transition-based AMR parsers can be successfully repurposed for paraphrase and generation from keywords tasks to take advantage of its structural knowledge?

1.4 Dissertation Outline

The Background section will introduce the AMR parsing tasks, the application of AMR parsing, and important existing methods for parsing. It also describes some complex techniques and dependencies that were popularly adopted by previous AMR parsers, and discusses the current challenges of AMR parsing task. The method section will provide problem formulation of the joint-training approach, and introduces the several tasks that the joint-model can be deployed towards. The experiment section will introduce the experiment setup, evaluation metric, benchmark datasets, and then present the experiment results. It also discusses whether the result matches our hypothesis, and the research implications of the result. Lastly, the conclusion provides an overview of the joint-training method, and discusses future promising directions using the joint-training idea.

CHAPTER 2

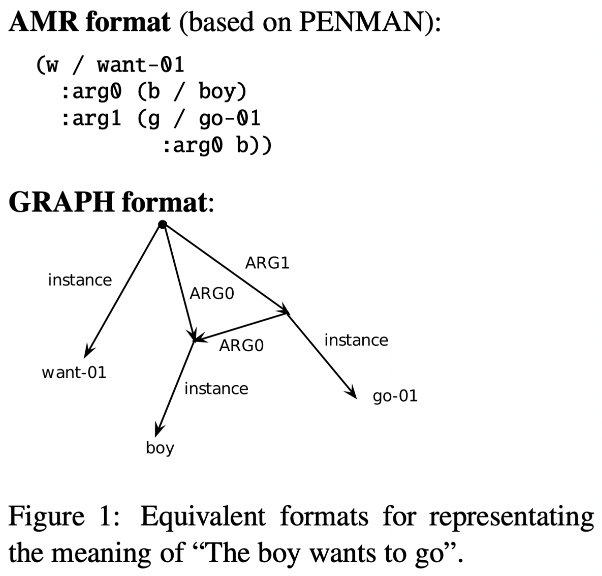
Background

This chapter offers a gentle introduction for the general audience about AMR parsing, including the definition of Abstact Meaning Representation, its real-life applications, a brief overview of recent and the state-of-the-art approaches to AMR parsing.

2.1 AMR Parsing

AMR parsing is a type of semantic parsing(as opposed to syntactic parsing), which takes a text sequence as input, and outputs a rooted directed acyclic graph. Semantic parsers emphasize semantic relations, such as agent and patient[DM18], while syntactic parsers focus on the functional relations between entities, such as subject and object. It is called AMR parsing because the resulting structure is the AMR graph, which was designed by [BBC13] as a general-purpose meaning representation [ALZ15]. AMR graph is a semantic formalism that encodes the core meaning of natural language text[KIY17]. There are also other proposed types of semantic representations of text, notably bi-lexical Semantic Dependencies(SDP) graphs [OKM16], and Universal Conceptual Cognitive Annotation (UCCA) graphs [AR17]. They propose different grammars and formats, and orients towards different focuses, but much of the semantic content is shared[AR17, HAR18].

AMR graph is proposed in 2013, the earliest among the three above mentioned semantic text representations. The motivation was from the 'Balkanization' of the semantic annotation landscape, where different semantic task, coreference resolution, named entity recognition,



AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

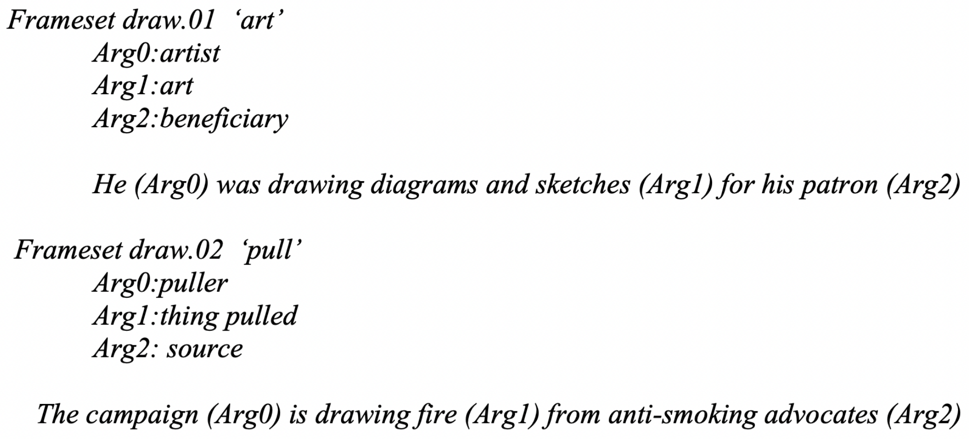
Figure 1: Equivalent formats for representating the meaning of "The boy wants to go".

Figure 2.1: A demonstration of AMR representation, from the original AMR paper.

semantic relations, temporal entities, discourse connectiveness, just to name a few, have distinctly different training data, and evaluation methods. The AMR inventors wanted to create one general framework that captures the logical meanings of the whole sentence, regardless of the specific subtasks involved. They cited the wide adoption of syntactic parsers illustrate the application prospect of quality semantic parsers built using AMR graphs.

AMR graphs are rooted, directed, acyclic graphs, whose edges and leafs are labeled[BBC13]. It is a traditional format for representing graph, equivalent to the PENMAN [BMN91] input format. There are some important high level properties of AMRs: 1. identical semantic content of different syntactic/grammatical variances should share one identical AMR representation. 2. Propbank Framesets are widely adopted; a frameset file contains multiple senses of a verb predicate. For example, two senses of draw in framenet is shown in 2.2. 3. It does not provide a word to graph mapping. 4. The design is heavily biased towards English.

The contents of the graph can be roughly divided into nodes and edges; nodes are made of either English word("boy"), Framenet concept("want-01"), or AMR defined keywords(eg. "date-entity"). Edges are relations for which there are about 100 in total. Those relations


Frameset draw.01 'art'
Arg0:artist
Arg1:art
Arg2:beneficiary

He (Arg0) was drawing diagrams and sketches (Arg1) for his patron (Arg2)

Frameset draw.02 'pull'
Arg0:puller
Arg1:thing pulled
Arg2: source

The campaign (Arg0) is drawing fire (Arg1) from anti-smoking advocates (Arg2)

Figure 2.2: A example of two senses of draw in Propbank Frameset.

ranges over frame relations(arg0, arg1), common semantic relations, quantities, dates, lists. Notably, AMR is a graph having the property of re-entrancy; in the example of 2.1, the instance boy can participate in two relations with respect two other nodes. For more details about AMR composition, please refer to [BBC13].

2.2 Applications of AMR Parser

Machine Translation [SGZ19] found semantic representations from AMR useful for machine translation because it helps preserve meaning of original sentence, and alleviate data sparsity issue of machine translation models.

Summarization [DK17] develops a method to prune summary AMR graph with which to generate summary text sequence, improving the summarization baseline on CNN-Dailymail benchmark corpus. [MG18] also proposes a novel pipeline for lossless summarization by merging parsed AMR graphs, and summary generation from merged graphs. Recent work by also [LKV21] extends of AMR summarization to document-level.

Event extraction AMR parsing is also leveraged by InfoForager [BLD20] to find answers to research questions by scanning scientific and medical journals. A joint information extraction framework [ZJ21] consisted of a semantic graph aggregator and an AMR guided graph decoder achieves state-of-the-art result on many IE subtasks.

Biomedical AMR parsers are combined with biomedical knowledge bases to perform domain specific information extraction, demonstrating quality results on COVID-19 scientific literature[ZPJ21]. AMR parsers can also be deployed to identify events related to molecular events/interactions in biomedical text [RMK17], where the authors develops an approach to align AMR subgraphs with biomedical events.

Dialogue systems [BDA20] offers a schema that supplements the current AMR schema with a set of speech concepts and relations, in the hope the AMR parsers can be applied to Human-Robot conversation. Ghazarial et al.[GWG22] use AMR parsers to create high quality negative samples to train dialogue coherence classifier.

Commonsense Reasoning and question answering AMR parsing is used as an intermediate result to construct a reasoning graph called ACP, with which the model can make produce quality answers in the CommonsenseQA task[LOJ20]. Knowledge Based Question Answering task can also use AMR parses to improve question understanding and to transform AMRs to align with query formats[KAR21].

Argument comparison The work by [OHW21] shows the AMR give quality representation of arguments, and is applied for comparison of the similarity of arguments , boasting high performance and explainability.

NLG Structural information was shown by RNNG [DKB16] to benefit not only parsing, but general language modeling tasks. Work by [TSO16] leverages an AMR encoder-decoder model for news headline generation and shows performance improvements over a neural attention thanks to structural syntactic and semantic information given by AMRs.

Paraphrase AMR parser is used to generate latent semantic representation which is shown to boost performance on paraphrase detection task[IDC18]. AMRPG[HC21] extracts encoded AMR and constituency parsing results and trains a decoder to recover the input sentence, which can serve as a paraphrase generation model.

2.3 An Overview of AMR Parsing Models and Methods

This section offers a bird's-eye view of current AMR parsing methods. Flanigan et al. [FDS16] proposed the first AMR parser baseline that releases a Smatch score[CK13] result. They divided the Parsing task into the subtasks of concept identification(nodes) and relation identification(edges). This method uses a semi-Markov model to find concepts, and an MSCG(maximum spanning, connected subgraph) algorithm to generate relations. Notebly, JAMR, the first AMR parser, is already relying on automatic aligner that maps word span to subgraphs. [WAM15] proposes an improvement to the JAMR's concept identification algorithm by using a simple classifier and generative actions to create subgraghs. [ALZ15] proposes a lambda-calculus representations of AMR, and uses a set of customized CCG(Combinatory Categorial Grammar) rules to generate lambda-calculus terms and resolve coordination and long-term dependencies.

More modern AMR parser uses pretrained neural models. [CL20] categorizes recent works on AMR parsers into attention-based sequence-to-sequence models[BG16, KIY17, BA17], attention-based sequence to graph transduction models[CL19, ZMD19b], and models that leverage text-to-graph alignments as latent variables[LT18]. [BBN21] suggests a concept of pure seq2seq models, which are end-to-end and make graph action predictions without external constraints. On the other end, it is methods that develops incremental graph construction and constraints for graph generation, such as the transition-based parsing systems.

Looking from a higher perspective, despite complexities in graph action constraints and pre-processing/post-procesing procedures, AMR parsing is still essentially a sequence-to-sequence task where the input is a text, and output is a linearized action sequence that needs to post-processed to graph; or, it is directly a sequence to graph tasks, where the output is a sequence of tuples(eg.(source-node, relation, target-node)) which are components of a graph.

As for whether external constraints or structural assistance is needed, it is debatable when the current large pre-trained models have demonstrated impressive power in extracting implicit relations. Pure seq2seq models often suffer more from data sparsity due to the lack of the above mechanisms, and would require additional pre/post-processing techniques to achieve good performance, such as character-level networks [NB17] and graph recategorization [PWG17]. Recent work SPRING [BBN21], nonetheless, achieves SOTA results on AMR parsing using a pure seq2seq BART model without such complexities.

On the other hand, transition-based AMR parsers[Niv03] use a state machine to guide model generation at each time steps to guarantee graph well-formedness. The transition-based algorithms process words by sentence order, and generate graph actions, graph nodes, and graph action, as it pops words from the buffer and add actions into the stack. The algorithm terminates when the buffer is empty, or all words in the sentence has been processed. Different works normally will define their unique set of graph actions, but the general buffer and stack configuration and the resulting sequential processing is preserved. It is argued that transition-based system's alignment based linearization and sequential processing procedure are beneficial inductive biases [MB16, ZNF21, ZNA21, NRM21].

Text-to-graph transduction models represent another category of AMR parsers. [ZMD19a] proposes a two stage parsing algorithm, which first employs a pointer generator network [SLM17] to generate a complete list of graph nodes; then, it apply a graph-based parser and the maximum spanning tree algorithm to separately generate the edges of the graph. A subsequent work proposes to jointly generate both nodes and edges, using an encoder-decoder framework. As shown in figure 2.3, it adopts a formulation almost exactly the same as seq2seq one, the only difference it that the target Y represents a tuple of graph relations, consisting of the source_node, the relation, and the target_node. One advantage of most text-to-graph transduction models is that it doesn't reply on token level alignment, which is needed by transition-based systems to produce the linearization. [ZMD19b] also mentions

$$\begin{aligned}\hat{Y} &= \underset{Y \in \mathcal{Y}}{\arg\max}\, \mathbf{P}(Y \mid X) \\ &= \underset{Y \in \mathcal{Y}}{\arg\max} \prod_{i}^{m} \mathbf{P}(y_i \mid y_{<i}, X)\end{aligned}$$

Figure 2.3: Graph transduction problem formulation, taken from [ZMD19b], where input X is a sequence of tokens ⟨x1, x2, ..., xn⟩ and output Y is a sequence of semantic relations ⟨y1 , y2 , ..., ym ⟩. One semantic relation y is a tuple ⟨u, du, r, v, dv⟩, where u and du represent source node, r represents relation type, v and dv represent the target node.

that building graph along sentence order offers good inductive bias for the parsing task, and thus proposes a broad coverage transduction parser that can also generates graph incrementally. [CL20] proposes another graph transduction approach, but added an addition graph encoder and two attention mechanism to locate text sequence and graph nodes to focus on for each time step.

2.4 Techniques in AMR Parsing

2.4.1 Alignment

Alignment for AMR refers to the mapping between concept/entity nodes in the graph and word in the input sequence. Alignment information is not provided by the AMR training data, but have been popularly adopted by most AMR parsing systems[ZMD19a] from very beginning. The first AMR parser[FDS16] requires alignment to train its concept identification and relation prediction models. They built an automatic alignment system based on a set of heuristic rules.

For following parsing systems alignment remains important because it tells the model which text it should focus on when generating the current graph action. All transition-based systems require alignment in its data-preprocessing to linearize graph. Alignments are further leveraged to control cross-attention in transformer models[FBN20, ZNF21], or being used as latent variables in training[LT18].

2.5 Graph re-categorization and Subgraph action

Re-categorization[LT18] has been a commonly used technique in high performance AMR parsers. Before training, it pre-processes the AMR graph data by removing node senses and grouping nodes together into a subgraph. After inference, it post-processes the output and uses rule-based algorithms to recover to original format. Subgraph action[BA17] is another method to group nodes into a single action, and uses the Subgraph action to generate an entire subgraph.

While such subgraph grouping methods have shown performance gain in benchmark datasets, they are also criticized for pipeline complexity and generalization issues. The decision to group a certain nodes and the algorithm to revert back to original graph are both rule-based. Those rules may not generalize well in new domains. Recent works[BBN21, ZNF21] have shown that graph re-categorization only boost score on the benchmark data, but lowers performance on other test data.

CHAPTER 3

Method

3.1 Introduction

3.1.1 Problem Formulation

If we assume a correct linearization of the AMR graph into an output space of $y = y_1, y_2, ..., y_T$, and the input text is $x = x_1, x_2, ..., x_S$, then the objective of the AMR parsing problem can be represented as the maximization of the log likelihood equation in 3.1. For transition-based parsing, there's a caveat that the linearization is always organized in sentence order(graph is constructed incrementally in sentence order). Moreover, in a transition-based system, the output space is not always the full output vocabulary; at different time step of the generation, the output space is constrained to only the allowed set of vocabularies, so that graph well-formedness is always maintained; this constraint is always deployed at decoding time, and is optional for training.

$$LL(D) = \sum_{x^i \in D} \log \prod_{t=1}^{T} P(y_t^i | y_{<t}^i, x^i) \tag{3.1}$$

By joint-training of AMR parsers, we proposes a joint(ACTION+Text) linearization scheme that includes input text into the output space, so that we can train on an additional denoising objective on top of the parsing objective3.1, and jointly train the model to parse text into graph, and recover masked input tokens.

3.1.2 Motivation for Joint-Training

There are 3 main motivations for the joint-training of AMR parsers.

The first one is that joint-training can potentially produce a more robust model, with better understanding of input text, and the structural alignment between input tokens and following graph actions. Moreover, masked tokens also represent a real life scenario where input tokens were out-of-vocabulary(OOV), and the AMR graph needs to be trained to fill in an appropriate substitute.

The second motivation is that the output space design of joint-trained models can encode alignment information perfectly by itself, without need for external memory for buffer, stack, and alignment. It fits in the larger trend to reduce dependencies, and simplification of transition-based parser to approximate the standard sequence to sequence task.

The third motivation is that the flexibility of a joint-trained model bring possibility for one unified AMR parser that could simultaneously trained on AMR parsing, paraphrase parsing, and keyword generation parsing at the same time.

3.2 Base Model: Structured BART transition-based AMR parser

> "Abstract Meaning Representation parsing is a sentence-to-graph prediction task where target nodes are not explicitly aligned to sentence tokens. However, since graph nodes are semantically based on one or more sentence tokens, implicit alignments can be derived. Transition-based parsers operate over the sentence from left to right, capturing this inductive bias via alignments at the cost of limited expressiveness. " – Quote from *AMR Parsing with Action Point Transformer [ZNA21].*

Our joint-training scheme is built on top of a State-of-the-Art transition-based parsing system, called Structured-BART[ZNF21]. And we will now introduce this important base model.

Structured BART is a transition-based AMR parser that uses a neural attention model BART to perform the sequence-to-sequence generation of AMR sequence. It proposes a novel way to use multi-head cross attention mask to encode alignment during training and decoding of the model. We choose Structured BART as the base model not only because it is the current SOTA, but also because its simple, neat, and flexible design, which makes transition-based parsers less complex, with less dependencies, and giving more expressiveness to model.

As for why transition-based parsers are popularly studied and deployed, there are two main advantages: graph well-formness and inductive bias in the linearization sequence. The state machine in a transition-based parser can dynamically update the state of existing graph, and guide the graph action generation by only allowing legal actions. While non-transition-based parser lack such constraint and the final outputs may not be recoverable to AMR graphs and will be subject to complex post-processing. For the second point, transition-based parsers have graph linearization organized in the sentence order, and the graph is generated incrementally. This linearization ordering helps preserve more structural information of the AMR after linearization, which is an inductive bias that is argued to benefit model training[ZNF21]. On the other hand, non-transition based parsers will normally resort to depth-first-search or breadth-first-search traversal to obtain a linearized action sequence for an AMR graph as the output space, or to decompose graph into a sequence of tuples of nodes and edges[ZMD19b]. From a human perspective, the dfs/bfs linearization is much less readable, but it is debatable whether such inductive bias from incrementally constructing linearization graph is necessary anymore when powerful transformer models have demonstrated impressive capability to understand and extract complex relations between the input and output space[ZNF21].

Having framed the AMR parsing problem as maximizing the following log-likelihood 3.1, researchers have tried using different models to improve parsing performance, including

Stack-LSTM models[BA17], Stack-Transformers models[FBN20], RoBERTa[ZNA21], and more recently, using BART model[ZNA21, ZNF21]; Structured-BART AMR parser uses BART to model the sequence generation task, with the help of an action-pointer based State Machine and a cross-attention mask that encodes alignment information during training and decoding.

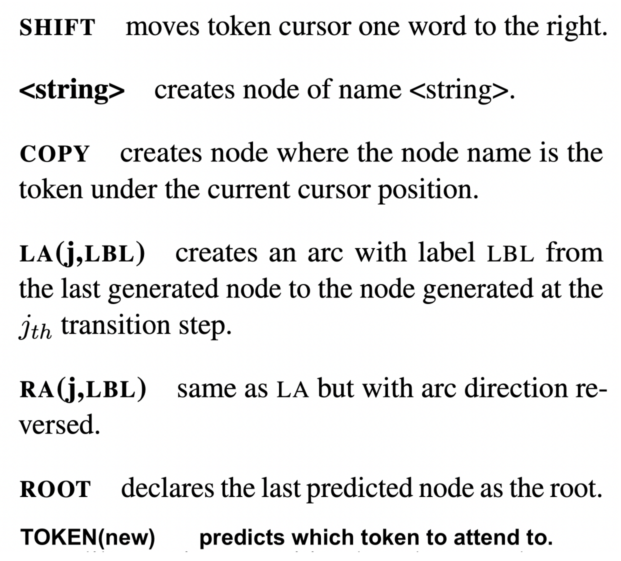
3.3 The Components of Joint-Trained Transition-based AMR Parser

3.3.1 Components Overview

This work proposes a joint-training scheme built on the Structured BART method [ZNF21]. The joint model largely inherits the State Machine, Oracle, and BART transformer model from the baseline Structured BART parser, but uses the joint(ACTION+Text) linearization strategy, which included the input texts in the output space to be learned along with graph actions. We also made many corresponding modifications on the Oracle and State Machine to accommodate new output space, and relax alignment requirements of the original method.

3.3.2 Oracle and ACTION+Text Linearization

ACTION+Text Linearization Our joint-model uses the much simplified set of actions define in 3.1, with the addition of TOKEN class to this set[ZNF21]. This is called the joint linearization or the ACTION+Text linearization. Each time the cursor moves, meaning that the parser moves to generate corresponding graph for the next token, TOKEN class is used to predict the next token to examine. When the input token is not masked, the decoder can access the token information from the encoder, but when the input token is masked, it must predict it from the entire 50k large vocabulary. With the inclusion of TOKEN into the output space, the model is trained to learn both parsing and an implicit alignment between the input text and corresponding graph.



SHIFT moves token cursor one word to the right.

<string> creates node of name <string>.

COPY creates node where the node name is the token under the current cursor position.

LA(j,LBL) creates an arc with label LBL from the last generated node to the node generated at the j_{th} transition step.

RA(j,LBL) same as LA but with arc direction reversed.

ROOT declares the last predicted node as the root.

TOKEN(new) **predicts which token to attend to.**

Figure 3.1: The ACTION+Text action set defined by the joint linearization scheme for joint-model training; The TOKEN class is newly added compared with the base action sets; TOKEN class always follows the SHIFT token, which indicate a shifting of cursor positioni, and the model needs a new token to attend to for the following generation. The TOKEN action class is created then to predict this new token.

Oracle The Oracle is a linearization tool that turns an AMR graph into an action sequence. It corresponds to a depth-first-search or breadth-first-search graph traversal algorithm for graph-based linearization. The Oracle needs to define a set of graph actions to make nodes, connect edges, and move on to another token, etc. It take an input graph, the text corresponding to the graph, and a text-graph alignment, and produces a linearized action sequence. For the joint-model, the Oracle is modified in that after each SHIFT action, a TOKEN class must follow to indicate which text token the model should attend to for the following graph generation.

3.3.3 State Machine

The State Machine is used to dynamically track the state of generation, with a cursor for input position, an action history stack to store generated sequence, and a heuristic algorithm to provide legal actions for the next step. The original Structured BART model also needs to store the alignment between generated sequence and tokens, but this is no longer needed in the joint-model and alignment contraints are removed.

A major modification in the State Machine is to learn a different vocabulary mapping between NODE and TOKEN classes. NODE class used to be generated from the entire BART vocabulary; however, since AMR node names are mostly lemmatized or taken from Propbank Framesets, we can learn the NODE vocabulary without using the entire 50 BART vocabulary. We also need design algorithm to explicitly differentiate the TOKEN and NODE state, since a single ¡string¿ can be both.

3.3.4 Model

The base model is still BART[LLG20], initialized with pretrained BART encoder and decoder embedding. The joint model modifies the data collator to include a dynamic masking algorithm that masks a specified amount of input texts at each training epoch, while, at the same time, it updates the vocabulary masks to reflect changes in masking(only a subset of allowed vocabulary is unmasked at each time step to ensure graph well-formedness). We also tried turning off the cross-attention mask, and instead, uses the standard self-attention and multi-head cross attention mechanism for transformer models. The cross-attention mask was designed in Structured BART[ZNF21] to feed structural text alignment information into the decoder, which has shown performance improvements. This is arguably no longer needed for joint-trained models, since alignment is already implicit in the new ACTION-Text linearization design.

The joint-training scheme of transition-based AMR parsers brings a number of exciting possibilities to explore. The following sections will describe why and how the joint-trained model can be tailored for those tasks, and their respective application and research inspirations, advantages and disadvantages.

3.4 Tasks for Joint Model–AMR Parsing

At each time step, our model performs multiple rounds of attention, reasoning, and composition that aim to answer two critical questions: (1) which part of the input sequence to abstract; and (2) where in the output graph to construct the new concept. – Quote from Cai et al., AMR Parsing via Graph-Sequence Iterative Inference [CL20]

3.4.1 Motivation

AMR parsing task, due to its complexity, are often decomposed into smaller subtasks, such as concept prediction and relation prediction [FDS16, CL19]; in the case of the above quote [CL20], where to look and where to build. Before the era of large-scale transformer models, researchers had been been relying on token-to-subgraph alignments to inform the model which part of the input text to attend to during graph generation.

For transition-based models, where-to-look problem is baked into its linearization strategy, where alignment information is required input for an AMR Oracle to perform linearization. Moreover, even for the training of attention-based transformer models, those alignments information are often used for better empirical performance. Among recent works on transtion-based parsers, the Stack Transformer paper [FBN20] proposes the using hard attention obtained from alignment to inform the model where to look. In Structured BART[ZNF21],they apply multi-head cross attention mask to provide structural alignment information to the decoder. Drozdoz et al. [DZF22] suggests the usage of neural aligner's posterior distribution over alignments during AMR model training.

The joint-training task aims to integrate the problem of where to look, and what to look, into the AMR training objective. The problem is framed as the follow: after each SHIFT action, which means the end of generating graph with outbound or inbound relation for the current cursor position(input token under cursor), what is the next token that the model should attend to, given the full input sequence and output actions up to the current time-step? When the input sequence is given, the problem is easy, it is just the next token from input; but if that token happens to be masked, the model must predict a hypothetical text token to anchor the following generation of corresponding AMR graphs. The goal of joint-training is to make the neural model more robust in understanding the input text, and explicitly encoding the structural alignment information between graph actions and input texts into the output graph sequence. Even in situations where the high proportion of input texts are corrupted, the model will be able to infer the corrupted input and generate corresponding graphs.

3.4.2 Problem Formulation

We have two main research question in mind for joint-modeling of the AMR parsing task: the first one is whether the new ACTION+Text linearization itself can provide beneficial structural alignment information for training and inference; the second one is whether joint-training of maximizing action sequence likelihood and the denoising input corruption makes the parser better and more robust.

For the first question, the objective function to maximize is provided in Equation 3.2, where x is the original input sequence(uncorrupted), and $\hat{y}$ is the ACTION+Text linearized gold output sequence, consisting of both baseline linearization of action sequence y and original input sequence x. We want to explore whether the new output space design can effectively incorporates graph-token alignment information, such that neither hard attention[FBN20] nor cross-attention masking according to external alignment will be needed for model training.

$$LL(D) = \sum_{x^i \in D} \log \prod_{t=1}^{T} P(\hat{y}^i_t | \hat{y}^i_{<t}, x^i) \tag{3.2}$$

$$LL(D) = \sum_{\hat{x}^i \in D} \log \prod_{t=1}^{T} P(\hat{y}^i_t | \hat{y}^i_{<t}, \hat{x}^i) \tag{3.3}$$

For the second question, we will incorporate the denoising task into the likelihood maximization objective. Let $\hat{x}$ be the randomly corrupted sequence of original text input x by any specified percentage; let $\hat{y}$ be the ACTION+Text linearized gold output sequence defined same as in Eq3.2. The training objective is given in Eq3.3, where the model is trained to recovered corrupted input tokens, and generate the correct corresponding AMR action sequence.

3.5 Paraphrase AMR Generation Task (PAG)

Researchers have been exploring whether the structural information encoded in AMR parsers can be beneficial for related NLU and NLG tasks. Theoretically, the paraphrase task require high level understanding of the semantics of the input sentence, and a strong capability to generate corresponding new sentences that are fluent and structural sound. Those qualities make AMR parsers appealing candidate for the task. However, the original transition-based parsers do not allow generation of graphs that are unaligned with the input sequence; and since those AMR parser do not have text generation capability, it normally requires a second step of AMR-to-Text generation to obtain a paraphrase. On the other hand, for the non-transition based parsers, their training do not benefit from the inductive bias of baked-in alignment information in transition-based parsers.

We present the first end-to-end transition-based system that can generate the paraphrase of an input sentence along with its corresponding AMR graph. The joint-model is a good fit for the task because its ACTION+Text output space is flexible enough to allow text generation; and after we relax the cursor-action constraint in the State Machine, it can easily generate free TOKEN generation that are unseen in the input text. Moreover, in model architecture we add an additional pooling layer over the encoder output, so that essentially only a distilled sentence embedding is fed to the decoder, and it is tasked to recover a sentence and the corresponding AMR graph with the distilled input sentence embedding as guidance. We can also control how difference level of the output paraphrase by adjusting the stride size and window size of the pooling layer. In our prototype model, we adopt average pooling of stride-size=sequence-length to encourage maximum creativity and variation in paraphrasing output.

3.6 AMR Generation from Keywords Task (AGK)

The task of AMR generation from keywords is a similar task to AMR paraphrase generation, with the same modeling setup, but different input and output. For training of the paraphrase model, the same input and output data is used as the joint AMR parsing task. However, for the AMR Generation from Keywords(AGK) task, we develop a sampling algorithm to create shuffled keywords from the AMR graph as the input sequence, and the output sequence is the same as the joint AMR parsing task. The keywords sampling algorithm exploits the structure of AMR graphs where more salient words are closer to the root of the graph, it uses a breadth first search algorithm to sample keywords by a probability $2/depth$ at each depth level. More important words are more likely to be included as keywords, and stopwords and common words are removed as in CommonGen Dataset[LSX19].

Generation from keywords is a controlled generation task that requires a high level understanding keywords embedding, and strong generation capability to organize input keywords into a coherent and meaning sentence. We train the joint-AMR model such that it takes the shuffled keywords embedding as the flavor vector to guide generation of an AMR graph

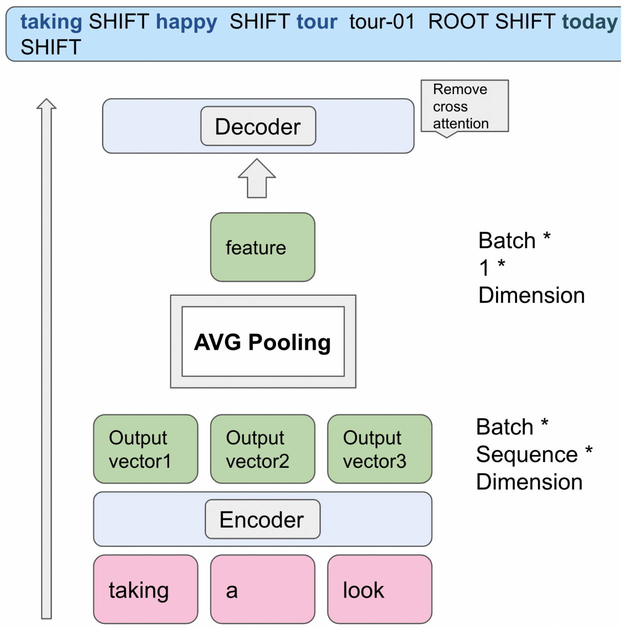


Figure 3.2: An illustration of the model architecture for paraphrase and keywords generation; The only modification of this architecture from the transformer architecture is the pooling layer added after the Encoder output. The original output size is Batch-size * Sequence-length * Dimension; the 1d-pooling layer will squeeze the sequence length dimension to a smaller size, and in this example, to 1. The pooling is performed to obtain a flavor vector that guides the decoder generation, rather than feeding it the entire encoder output.

using those keywords. The flavor vector is similarly created by adding a pooling layer over the BART Encoder output, as the example shown in Fig3.2. The original encoder output dimension is goes through a 1-dimensional average pooling layer, that reduces the sequence length dimension to 1. We can also change the stride size or window size of the pooling layer to obtain different degrees of distillation of the full input embedding as flavor vector.

CHAPTER 4

Experiments

4.1 AMR Parsing Task

4.1.1 Experiment Setup

Dataset We evaluate the performance of joint-model on the AMR 2.0(LDC2017T10) dataset of 39K sample, with 36521 samples for training, 1368 for validation, and 1371 for test. AMR 2.0 dataset have wikification nodes, and the pretrained aligner is the same as Structured BART[ZNF21].

Evaluation Metric SMATCH score[CK13] is the consensus and standard evaluation metric for AMR Parsing task, which is designed to capture the amount of overlap between two semantic structures. SMATCH computes the precision, recall, and f-score between AMR triples of the first AMR and the target AMR. AMR triples are made of (relation, variable, concept) or (relation, variable1, variable2). For more details, please refer to [CK13].

Model Architectures and hyper-parameters There are several importantchoices with respect to the training of our neural parser. The first one is target vocabulary masking: while masking is always enforced in decoding to guarantee graph well-formedness, it is optional for training(previous work reports higher performance when it is on during training). Our experiments found that for joint-trained models, target-vocab-mask do not affect the final result. So, we only report numbers where target vocabulary masking is turned off for efficiency. The second technique is cross-attention masking by leveraging a pre-computed alignment file: using alignment cross-attention masks has been shown to enhance performance, as is used in Stack-Transformer[FBN20] and Structured-BART[ZNF21]. While we try both masking and not masking, we hypothesize it shouldn't make a difference for joint-trained models, because the new linearization scheme already has alignment information implicit encoded, and should be directly accessible from transformer cross-attention and self-attention.

4.1.2 Baselines and Techniques

Recent works and Baselines We included 4 strong baselines, representing the State of the Art approaches. They are all finetuned on large-scale pre-trained neural models, such as RoBERTa[LOG19] and BART[LLG20]. All of the selected baseline are transition-based parsers, except SPRING, which is a graph-based AMR parser that also achieves very good result.

Graph re-categorization and dependencies Graph re-categorization is a pre-processing and post-processing technique adopted by most high performing AMR parsing systems, such as [LT18, CL20, ZMD19a, BBN21]. However, it also attracts criticism for implementation complexity and dependency complexity; moreover, recent works [BBN21, ZNF21] have shown that the rule-based re-categorization techniques cause out-of-domain adaptation issues in Bio-AMR data and AMR2.0 data, despite boosting score on benchmark dataset. In the example of a popular re-categorization pipeline[ZMD19a], Standford CoreNLP was used for lemmatization and POS-tagging training data. Some common subgraphs are then collapsed into a single graph action to train the model. In post-processing after inference, the collapsed subgraph action will need to be recovered to a real sub-graph using DBpedia Spotlight API for wiki link generation and number of other algorithms to create attributes. In our experiment result Table4.1, usage of re-categoritzation and dependencies are indicated by the **Collapse Subgraph** and **Dependency** columns. None of the joint-trained models need graph re-categorization, thanks to the strong baseline of Structured BART that our approach builds upon.

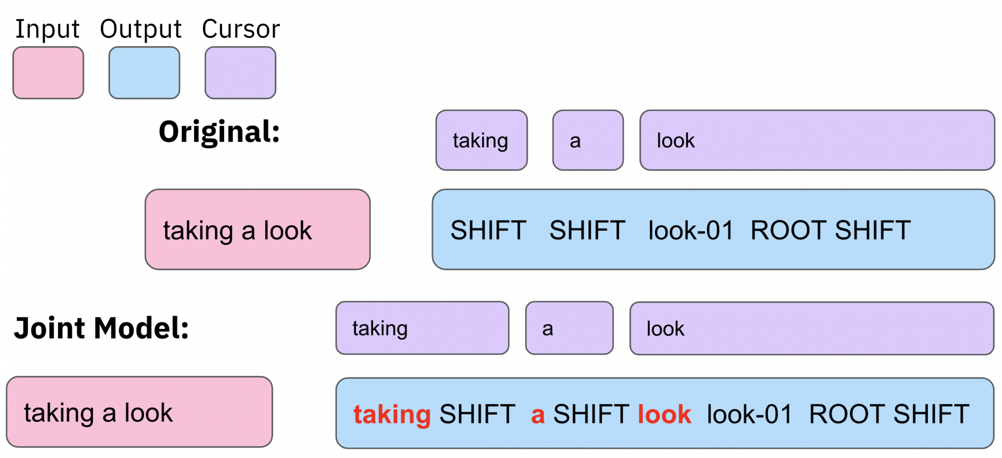


Figure 4.1: A demonstration of the difference between original linearization and the joint-model's ACTION+Token linearization. The red words are the TOKEN class added by the joint(ACTION+Token) linearization method.

4.1.3 Two Questions and Two Task Formulations

Leak There are two variants of the Joint model, leak and MLM, which corresponds to the two questions we have for joint-trained parser. The first one is whether the joint(ACTION+Text) linearization strategy itself provide beneficial structural alignment for training and inference, where the training objective is outlined in Eq3.2. The Joint-Model(Leak) means that input texts are not masked, but, rather leaked to the decoder. More specifically, for all the time-steps that a TOKEN class is expected, the decoder will obtain the golden token from the input sequence and mask out all remaining vocabularies except the golden one. In this formulation, the model do not learn to predict the input token, but rather only passively receives the token from the State Machine and insert it in the priors for the generation of the corresponding graph. In the example of Fig4.1, the sequence of graph actions related to LOOK(LOOK-01, ROOT, SHIFT) have the LOOK Token immediately in the priors before their generation. This joint-model training scheme is named **Joint-Struct.B(Leak)**.

MLM We are also interested in whether the addition of the denoising objective on top of

Model name	Pre-trained Model	Transition	Collapse Subgraph	Attention Alignment	Dependency	Smatch
Stack-Transformer (2020)	RoBERTa	✓	✓	✓	✓	80.2
Act.Pntr-Transformer (2021)	RoBERTa	✓	✓	✓	✓	82.6
SPRING-1 (2021)	BART	x	x	x	x	83.8
SPRING-2 (2021)	BART	x	✓	x	✓	84.5
SPRING-3 + extra-data(2021)	BART	x	x	x	x	84.3
Struct.B-1 (2021)	BART	✓	x	x	x	83.4
Struct.B-2 (2021)	BART	✓	x	✓	x	84.2
Joint-Struct.B-1(leak)	BART	✓	x	✓	x	83.9
Joint-Struct.B-2(MLM)	BART	✓	x	✓	x	84.0
Joint-Struct.B-3(Leak)	BART	✓	x	x	x	84.1
Joint-Struct.B-4(MLM)	BART	✓	x	x	x	83.85

Table 4.1: The main AMR Parsing experiment table; we added indices after model name to indicate that they are the same model family; Transition refers to whether it is a transition-based parser; Collapse Subgraph refers to whether the model uses graph collapsing or re-categorization techniques; Attention Alignment refers to whether model uses alignment information to modify cross-attention; dependency means whether model uses external dependency package such as lemmatizer, NER, or POS taggers. The Smatch score for Struct.B and Joint-Struct.B are reported as 3-seeds average.

the parsing objective can lead to performance boost, due to the intended training on where and what the model should look at each inference time-step. The problem formulation is given in Eq3.3. This problem will require randomly adding noise to the input training data. Our current experiment tries dynamically masking 15% of input at each epoch, and trains the model on recovering those masked input tokens. This joint-model training scheme is named **Joint-Struct.B(MLM)**.

4.1.4 Experiment Results

The experiment result is reported in Fig4.1, where the Smatch score are calculated as three seeds average. First, we can compare Joint-Struct.B-1 with the baseline Struct.B-2, where the only difference between the two is the linearization methods. We observe little difference in Smatch score, so it seems our joint(ACTION+Token) linearization strategy do not impact parsing performance. However, if we compare Struct.B-1 and Joint-Struct.B-3 where cross-attention masking for alignment is turned off, the joint-model shows a much higher result. It shows that cross-attention mask is indeed useful for the original Struct.B model, but it no longer benefits the joint-model where alignment is already implicit in its joint linearization design. Our hypothesis is confirmed by the experiment result.

However, if we shift our attention to the MLM training result, our expectation to see improvement does not bear fruit. The joint-trained model with a combined objective as in Eq3.3 does not seem to produce better parsers than the baseline Struct.B-2, nor does it seem to improve upon the Joint-model(leak) variants. Moreover, the MLM trained joint-model also seem to be more unstable across different initialization seeds. There are a number of possible reasons for the failure of the MLM approach: 1.MLM has been shown to work well for training on large-scale data; 35k of AMR training may be too small for the denoising objective to show improvement. 2. The limited vocabulary within the current AMR training set is already well-learned; MLM objective may help for training on new domain and new set of vocabularies. It is also notable that the joint-trained model has a longer linearization sequence, so it is more costly to train and conduct inference than the baseline Struct.BART model.

4.2 Paraphrase AMR Generation Task(PAG)

Firstly, it's worth noting that the model we present is the first one of its kind, taking a sentence input, and generate both the AMR graph and output paraphrase sequence incrementally and interdependently. It is a task that trains the model to understand, interpret, and construct a sequence as well the structural relations between entities in the sequence. We are here only presenting a prototype model, and do not claim superiority among related works. This exploratory attempt aims to demonstrate that it is feasible and promising to use structured trained models for related task such as paraphrasing. An example of the paraphrase generaetion is given in Fig4.2; we also include more generation outputs in the **Appendix A.2** for the readers who are intersted in eye-balling the quality of generation. The prototype paraphrase model is a slight modification of the joint-trained paring model in above section, with average pooling layer added and no MLM is performed.

A simple set of automatic evaluation is performed to measure the quality of the paraphrase generation, as shown in first two rows of Table4.2. The Gold-paraphrase row represents the upperbound results, where the gold data from the AMR corpus is used as predictions. The BLUE score is 1.0 because the similarity between we compare the gold with itself. The perplexity score is calculated using pretrained GPT2-Large model[RWC19] from huggingface, where a lower score can be seen as more fluent and more likely to be generated by a competent language model. The second row shows the automatic evaluation results for our Joint-paraphrase model; The BLEU score[PRW02] falls to 0.366, which still indicates good translations according to standard interpretation of the score. The joint-paraphrase model seems to perform quite well on the perplexity metric, with very modest increase in perplexity compared to the gold data. It indicates the generated sentences have reasonable quality, comparable to the gold data.

Our evaluation of the joint-paraphrase model also reveals clear weaknesses. With the current average pooling model architecture, the model has a tendency to copy a portion of input sequence, which is undesirable for the paraphrase tasks that seeks to see diversity in structure and phrasing. A potential future improvement is to remove the positional embedding for the BART model to prevent exact memorization. Moreover, by eyeballing the model generation results and testing it in out-of-domain data, we find that the model is more likely to generate inconsistent sentences when input text contains words unseen in the training set.

We acknowledge the current state of the paraphrase model do not have ideal results, but

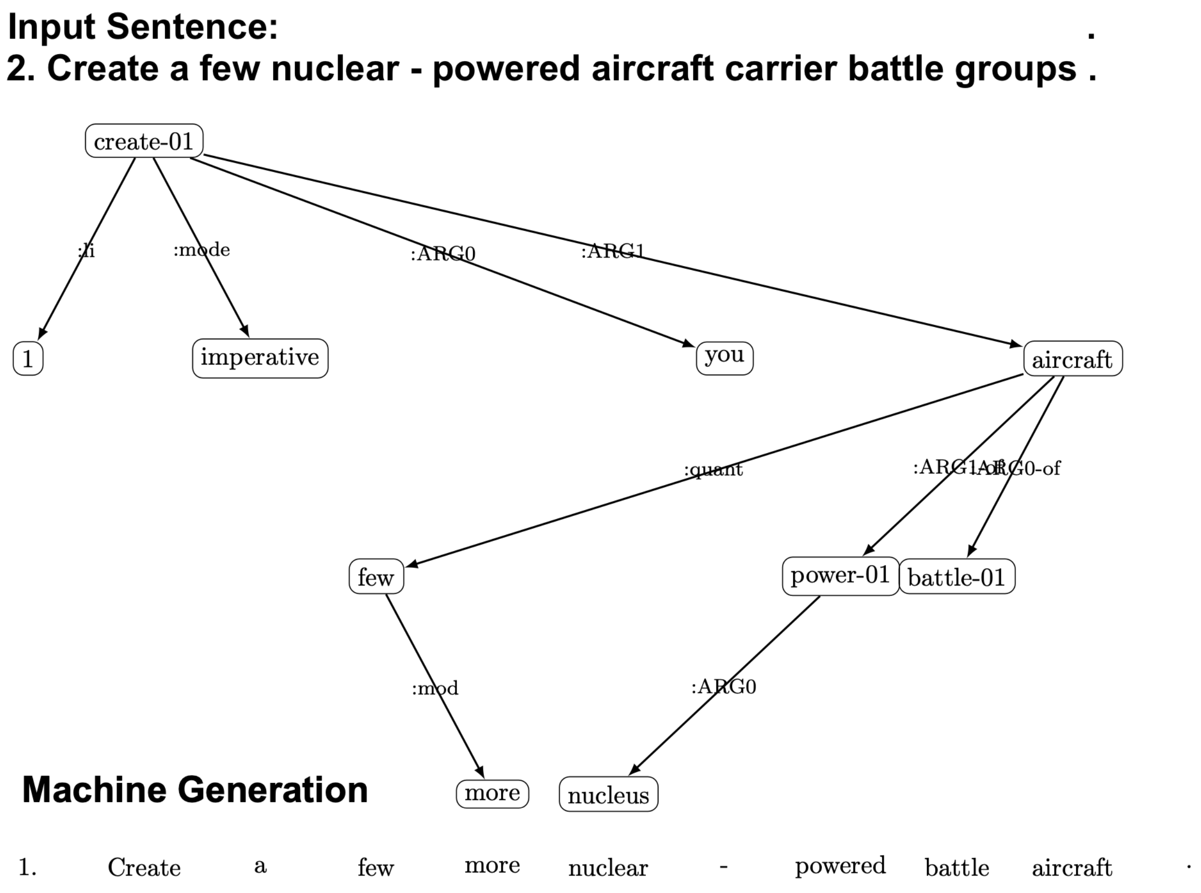


Figure 4.2: An example of the paraphrase result by the Joint-model.

it still holds theoretical significance. Existing paraphrasing approaches have tried to leverage structural information of sentence by using AMR as intermediate resulst, such as [IDC18] for paraphrase detection and [HC21] for paraphrase generation. Our approach presents a more straightforward way to directly train a paraphrase model with built-in structural learning; the resulting AMR graph that is produced along with the paraphrased text can serve both as avenues to control paraphrase generation, and as a window to understand how the model is interpreting the structural relations of its own generation, making the seq-to-seq process more explainable and visible.

Model	Eval-Data	# of Samples	BLEU	PPL	Coverage
Gold-paraphrase	AMR	1368	1.0	2.03	N/A
Joint-paraphrase	AMR	1368	0.366	2.18	N/A
Gold-keywords	AMR	1368	1.0	2.23	55.20
Joint-keywords	AMR	1368	0.05	2.18	41.42
Gold-keywords	CommonGen	4018	1.0	6.08	99.65
Joint-keywords	CommonGen	4018	0.261	6.93	64.86

Table 4.2: Experiment results for joint-paraphrase model and joint-keywords model; The models are evaluated on two different datasets. BLEU score uses equal weights for the 4 types; Perplexity score is computed using GPT2-Large model; Coverage is a metric that computes how percentage of lemmatized keywords captured in the generated composition.

4.3 AMR Generation from Keywords Task(AGK)

4.3.1 Background

The ability to compose sentences by using a select set of concepts is considered an advanced task of intelligence and a milestone of human intelligence development[LSX19]. The task of generation from keywords asks machine to produce meaningful and consistent sentences from a given set of keywords. An example from the CommonGen paper is that given a set of keywords {DOG, FRISBEE, CATCH, THROW}, machines then generate the output sentence A MAN THROWS A FRISBEE AND HIS DOG CATCHES IT IN THE AIR. It is argued that relational reasoning and compositional generalization are two key capability required for Generation from keywords task. Relational reasoning means the model needs to first construct an overall overview of the relations between the given keywords; compositional generalization means the model's ability to infer the relations between unseen combinations of keywords[LSX19]. Those two requirements are exactly the strong suits of structural trained models. AMR parsers are trained on the structural relations between salient semantic entities. The AMR task is also built for general-purpose semantic parsing, implying that it can handle unseen entities.

Our joint-keywords model conducts a more complex task than existing generation from keywords models, producing not only a coherently composed sentence, but also the corresponding AMR graph, as shown in example illustrated in Fig**??**. Our joint-keywords model enjoys theoretical advantage in structural learning and tractability just as the previous paraphrasing task. The AMR graph generated along with the composed sentence will first bring increased visibility for the generation processing, and along with it, providing the levers to control sentence composition. For example, we can edit the graph rules by preventing certain entities from establishing relations, and thus controlling the direction of generation.

4.3.2 Experiment Setup

Model Architecture The prototype model is trained using maxpooling of the encoder output with stride size $sequence_length/3$ and window-size 3, following Fig3.2. We trained the model for 40 epochs and choose the top model with the highest smatch score that matches the golden AMR graph, that is picking the model generating most similar AMR graph to the golden graphs. We acknowledge that it is not the ideal way of picking the best model, which should take into account not only AMR graph quality, but also sentence composition quality.

Dataset The joint-keyword model is trained on the self-created AMR-keyword dataset and is evaluated on two datasets, our self-created AMR-keyword dataset and the CommonGen dataset[LSX19](A portion of the AMR-keywords is set aside for evaluation). The AMR-keywords dataset, also mentioned in Section3.6, contains input and output pairs, where the input is a set of keywords and the output is the joint(ACTION+Token) style AMR linearization. The keywords are selected from entity nodes of AMR graphs, using a breadth first algorithm that samples by a probability 2/depth at each depth level. The AMR graph typical assign more salient nodes closer to the root, so we give higher proability of sampling for top nodes. Stopwords and common words are filtered. The current algorithm still do not remove all AMR-specific entity names, which needs further improvements. CommonGen is a benchmark dataset[LSX19] developed for the similar task that takes common concepts as keywords, and ask the model to produce sentence composition of those concepts.

Metric We selected 3 metrics(BLEU[PRW02], perplexity, and Coverage[LSX19]) to evaluate the similarity with gold, fluency, and coverage of input keywords. The BLEU score assigns equal weights for the 4 types. Perplexity is calculated from GPT2-Large[RWC19] using huggingface. Coverage is proposed by [LSX19] to calculate the percentage of lemmatized input keywords that can be found in the output. Our experiment find that BLEU isn't the most appropriate metric for our task, and Coverage metric misses semantic variations of keywords.

4.3.3 Results

We conducted in-group evaluation of generated samples from our prototype joint-keywords model, and were pleasantly surprised by the interestingness, diversity, and quality of the generation. We can hardly distinguish human written samples and ones that machine composed. In the **Appendix A.1**, we provide an entire page of keyword generation samples for interested audience to review.

The automatic evaluation shows mixed results. The low BLEU score do not match the impression of human evaluation. We find that our model generates very creative contents that differs from the golden reference data. The generation from keywords task do not have a deterministic best answer, so reference-based metrics such as BLEU are not the most appropriate ways of evaluation. The perplexity score confirms human impression that the quality is high, matching the gold referenced data. The coverage metric understates the true coverage of our model because our joint-keyword model often replaces keywords with synonyms or verb/noun semantic equivalence. For example, given the keywords COMPEL SENSE forget China person country, the model produces Don't forget that China has enormous economic and political power. They will not be bullied by the US.. The model uses bullied to interpret compel. Another reason for the low Coverage score may be our keywords creation algorithm, which uses the lemmatized or Propbank Framesets of the keywords, which are not exact matches with the true output space words. This may be a desirable property because we would want the model to generate along the semantic direction, not the following strict forms of keywords. In summary, above reasons explain why our gold data do not score high in Coverage, and the fact that the generated sentences only fall slightly short of gold data illustrates the good semantic coverage of the joint-keyword model.

Again, this work do not claim superiority in generation from keywords task. This work hopes to shed light on the possibility for direct structural training of generation from keywords model, which is a promising direction to explore.

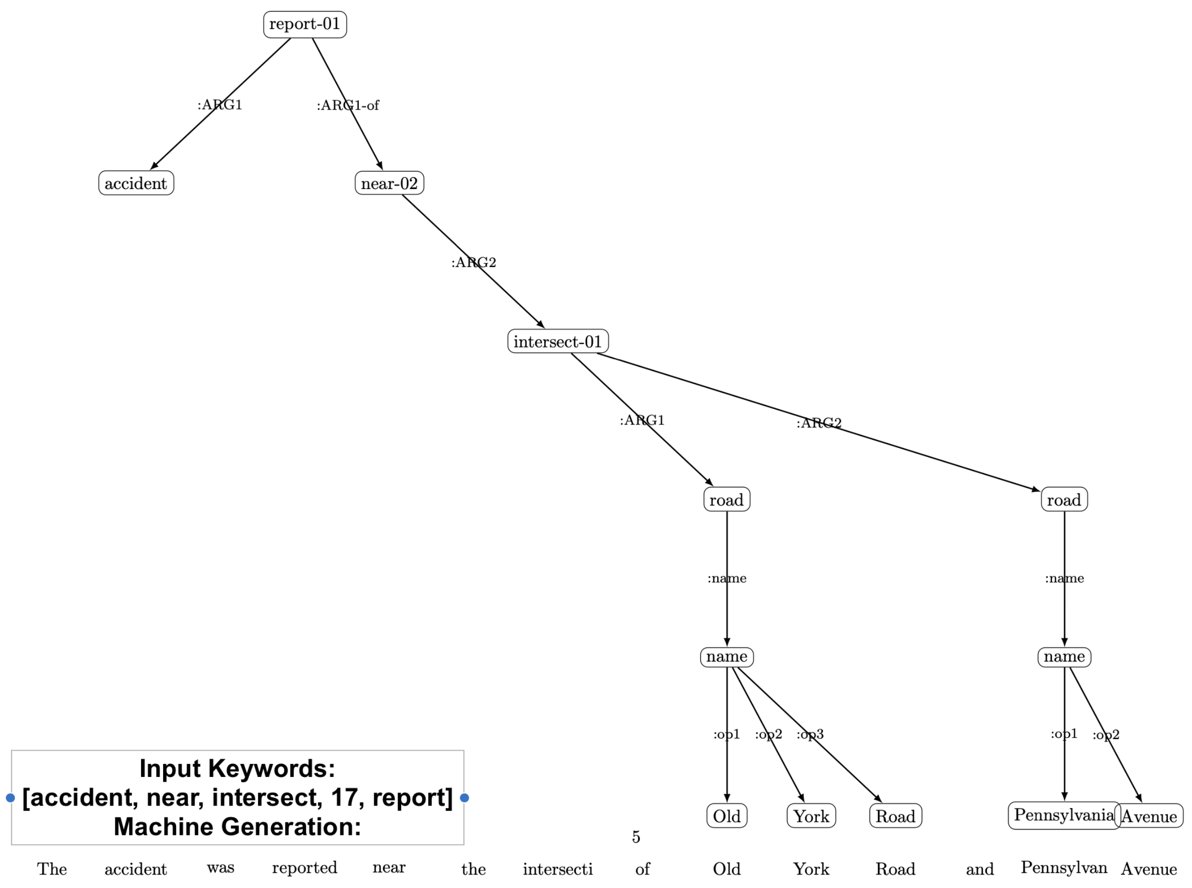


Figure 4.3: A generation from keywords example using the Joint-model, using keywords ACCIDENT, NEAR, INTERSECT, 17, REPORT; the composed sentence is: THE ACCIDENT WAS REPORTED NEAR THE INTERSECTI OF OLD YORD ROAD AND PENNSYLVAN AVENUE; we can notice a typical error here with unfinished word-piece that should follow intersection and Pennsylvania.

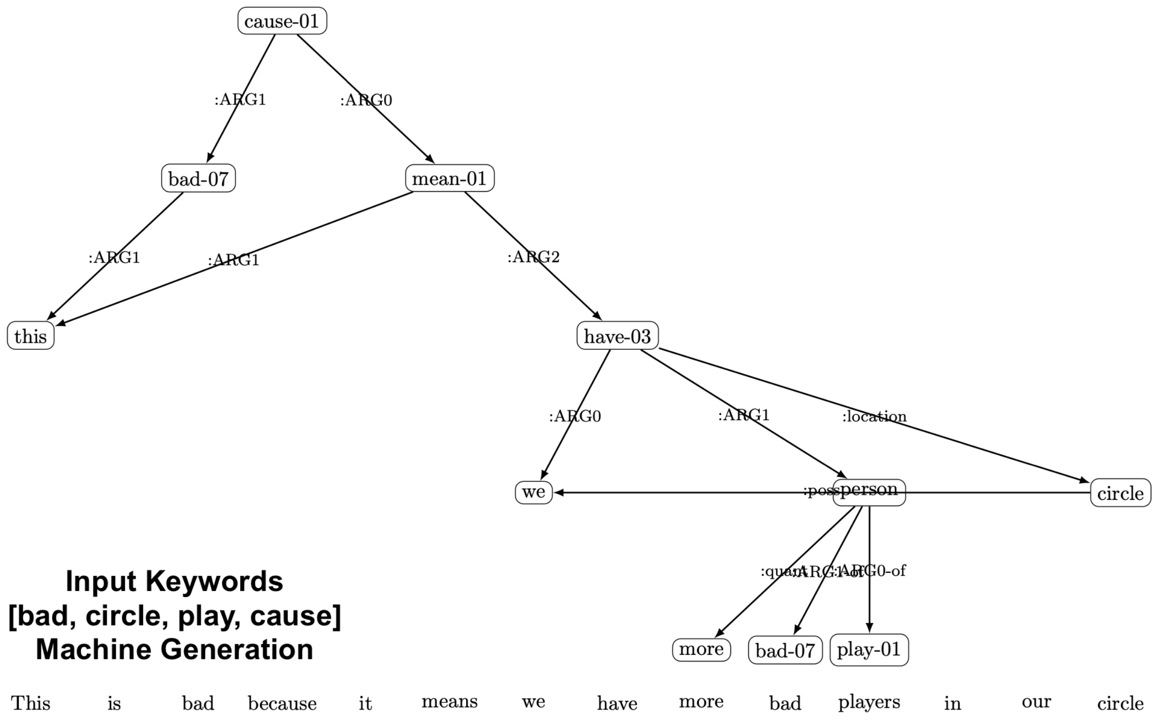


Figure 4.4: Generation from keywords result by the Joint-model for keywords BAD, CIRCLE, PLAY, CAUSE.; The Machine generated output is THIS IS BAD BECAUSE IT MEANS WE HAVE MORE BAD PALYERS IN OUR CIRCLE.

CHAPTER 5

Conclusion

Even though the current joint-trained transition-based AMR parser does not outperform existing State-of-the-Art model, such as Structured BART[ZNF21] and SPRING[BBN21], it is, nevertheless, an innovation to further simply AMR parsing, providing implicit alignment in its linearization; we have shown that in the joint-model design, standard transformer architecture can already leverage alignment information effectively, without needs for hard-coded cross-attention masking. Moreover, by relaxing the alignment constraints between output action sequence and input text, this work liberates imagination for AMR parsing models, from just parsing to any sequence-to-sequence tasks that may benefit from structural learning, such as paraphrase generation and keywords composition. This finding is significant because previous approaches only leverage the structural knowledge encoded by AMR parsers as intermediate result to train their respective model. The joint-model shows the prospect of directly training model with both a specific task objective and structural knowledge of AMRs. It even brings hope for a T5[RSR19] style unified AMR model that is simultaneous trained on a set of different tasks to benefit from both transfer learning and structural learning.

APPENDIX A

Appendix

A.1 Keywords Generation Results of Prototype Joint-Keyword Model

WARNING: there may be inappropriate content in the below samples; they do not represent the author's opinion by any means.

Corresponding AMR graphs are not included for clarity.

Keywords: turn body still person suffer fish

Machine Generation: Still , he turned his suffering into a laughing fish .

Keywords: accuse free foreign person differ speak call slash express view

Machine Generation: A foreign minister speaking on behalf of China called for Americans to express their views on the various issues raised by the Chinese government .

Keywords: US cause need deal fight country result name achieve

Machine Generation: The US and EU need to achieve mutual respect and cooperation in fighting terrorism .

Keywords: good article interrogative too country intense name United

Machine Generation: Is it too much of a good thing that the US is doing these kinds of things ?

Keywords: imperative ear exist tiger person assume create

Machine Generation: Please create a new type of tiger that does n't exist in the real world so that people can assume the responsibilities for it .

Keywords: express

Machine Generation: I express my sincere condolences to the families and loved ones of these brave young men .

Keywords: learn study

Machine Generation: You learn something new every day at school .

Keywords: attract

Machine Generation: To do so , you have to first attract people to your side . That 's what happened

Keywords: recommend advocate

Machine Generation: He should have just left it up to the experts to come up with a solution .

Keywords: compel sense forget China person country

Machine Generation: Do n't forget that China has enormous economic and political power . They will not be bullied by the US .

Keywords: imperative person name hate legitimate eat country especially really dog drink

Machine Generation: Do n't drink or eat dog meat , and especially do n't hate on people from other countries because they have different tastes .

Keywords: contrast forget possible feel country name

Machine Generation: But I forgot you ca n't feel sorry for Americans / Canadians / Australians living in these countries .

Keywords: Mao possible nation name save

Machine Generation: The only way to save mankind is to bring Mao Zedong down . Only through the intervention of the National Assembly can China do this .

Keywords: betray person too name

Machine Generation: He was too much of a cheat to betray his own country .

Keywords: prove line obligate deserve

Machine Generation: Bottom line , he does n't deserve to win .

Keywords: elite contrast person

Machine Generation: But that 's just me , you know .

Keywords: remove imperative group

Machine Generation: remove them from the discussion .

Keywords: cause obligate collude prevent person Africa name run

Machine Generation: The South African government must run an anti - corruption program to combat the problem .

Keywords: resemble possible mad else disappoint change

Machine Generation: Like everyone else here , he disappointed me by not being able to change the subject .

Keywords: intend world direction lead country long

Machine Generation: The United States intends to lead the world in the direction of sustainable development .

Keywords: follow interest

Machine Generation: I 'm interested to know what others think of the whole thing .

Keywords: possible specialize country

Machine Generation: France may specialize in this kind of surgery .

Keywords: betray person compatriot

Machine Generation: compatriots betray their country .

Keywords: free need account relative speak

Machine Generation: You need to use a relative 's free pass to get to freedom of speech .

Keywords: free slight foreign person somewhat slave put

Machine Generation: They were somewhat put in a somewhat similar situation to slaves , with slightly less freedom and slightly less slave - like immigration .

Keywords: ask person mention

Machine Generation: I mentioned to my parents that when I was younger I would go to Costa Rica to visit the springs .

Keywords: affair become soldier enter cause person Alliance fan group join ask patriotic name ally military permit fascinate armament obligate system quality

Machine Generation: As soon as the invasion was over , Abather enlisted in the US Air Force with the understanding that it would only be a matter of time before he would be able to join the elite US Air Force . Since then , he has joined other NATO member states in the Far East , the Far East , and the Far East . He has even become a fan of the US Air Force .

Keywords: bump support

Machine Generation: I would support any effort to get rid of these people .

Keywords: moment person name critical descend

Machine Generation: At this moment , Abather is descended upon a critical moment in his life .

Keywords: imperative Communist cooperate name

Machine Generation: Look at the Communist Party 's cooperation with the US in the Vietnam War .

Keywords: influence peaceful cooperate create

Machine Generation: We want to create a peaceful and trusted environment in which business and investor confidence can flourish .

Keywords: terrify good vest column interest person name

Machine Generation: Also , I find it interesting that your column says Zimmerman was " the most feared " bomber in the history of the US .

Keywords: enter obligate destroy language current nation culture name

Machine Generation: If we destroy this nation we must first enter the culture in which we live .

Keywords: turn

Machine Generation: He turned to me and said : " I want to help people in need , and

Keywords: imperative dragon coil wait sense

Machine Generation: Wait a minute and see if this thread really makes you feel this way

.

Keywords: US enemy cause need country result name force reach

Machine Generation: Therefore , the US and EU need to reach an agreement to end the war in Iraq .

Keywords: recommend foreign yes slave

Machine Generation: Yes , they should have slaves in foreign countries as well .

Keywords: Deng Communist attend need person comrade name allow ambassador Huntsman betray let country recommend once citizen

Machine Generation: The Chinese people should not be deceived by the propaganda put out by the US . Let 's not forget that when former US president Nixon was captured in Cambodia , he also said that the Vietnamese people should not let down their fellow countrymen .

Keywords: dawn society name allow

Machine Generation: Dawn comes and goes daily in China .

Keywords: slow politics cause avoid domestic possible interrogative develop too blame

Machine Generation: Could it be that domestic politics in China is too slow to react to international affairs ?

Keywords: interrogative proper bribe

Machine Generation: Is n't that a proper bribe ?

Keywords: reality gray person selfish

Machine Generation: Reality is a selfish thing .

Keywords: become soon activity joke propaganda name organization view

Machine Generation: The EDL soon became viewed as a joke by the mainstream media .

Keywords: environment criticize wrong

Machine Generation: Critics say it is wrong because it does n't do anything about the environment around it .

Keywords: Japan account possible war

Machine Generation: I can account for Japan 's rapid surrender .

A.2 Paraphrase Generation Results of Prototype Joint-Paraphrase Model

WARNING: there may be inappropriate content in the below samples; they do not represent the author's opinion by any means.

Corresponding AMR graphs are not included for clarity.

Input Sentence: I remember , 60 - some years ago the US had just fought a war with China , and still threatened China with the use of nuclear weapons ;

Paraphrase Generation: I just remember ; 60 years ago , the US had fought a nuclear war with China , and had now involved some kind of cooperation with China ;

Input Sentence: 40 - some years ago , the US entered Vietnam with the intention of containing China ;

Paraphrase Generation: 40 years ago , the US entered the anti - China war with intention of containing China ;

Input Sentence: Why are there still so many people ignoring their conscience and speaking on the behalf of the US ?

Paraphrase Generation: Why are there so many people still speaking against their conscience and ignoring the conscience of the US ?

Input Sentence: Without taking a big loss , people do n't improve their abilities to remember - this is possibly the greatest weakness of us Chinese .

Paraphrase Generation: Without taking a big step , people do n't remember to improve their skills - this is possibly the greatest weakness of us Chinese .

Input Sentence: It 's entirely possible - do n't we ourselves raise " traitors to China " among primary school students ?

Paraphrase Generation: It 's entirely possible - we do n't raise ourselves to be " traitors " among primary school children ?

Input Sentence: He is planning on moving out of the house by the end of the year , he and his wife do not see eye to eye .

Paraphrase Generation: He is planning to move out of the house by the end of year , he does not see his wife and little brother eye on a resolution .

Input Sentence: The cyber attacks were unprecedented .

Paraphrase Generation: The cyber attacks were unprecedented since the Soviet Union were in operation .

Input Sentence: Estonia is a Baltic nation .

Paraphrase Generation: Estonia is a Baltic nation of pacifist people .

Input Sentence: Officials in Afghanistan 's neighboring countries regularly announce large - scale arrests and seizures of narcotics made after illegal convoys cross the border .

Paraphrase Generation: Officials in neighboring countries frequently make large - scale reports of cross - border illegal smuggling and drug smuggling despite the Governments ' efforts to enforce such arrests .

Input Sentence: Dmitry Medvedev stated that the Russian Federation 's task for the next few years is to make sure that the Strategic Missile Forces receive the necessary funding to respond to modern threats .

Paraphrase Generation: Russian Deputy President Dmitry Medvedev stated that the Russian Federation 's task for the next few years is to make sure that the necessary technologies are supplied to defend the Far East 's missile system .

Input Sentence: Dmitry Medvedev promised to raise officers ' salaries .

Paraphrase Generation: Medvedev promised to raise officers ' salaries without any compromise .

Input Sentence: Dmitry Medvedev was sworn in on May 7 , 2008 as the Russian Federation 's President and has so far portrayed an image of a liberal and avoided the harsh anti - western rhetoric of Vladimir Putin .

Paraphrase Generation: Russian President Vladimir Putin was elected on May 7 , 2007 in the so - called Far East and has portrayed himself as a far from the neo - conservative rhetoric of the Kremlin .

Input Sentence: Vladimir Putin has threatened to point nuclear missiles at countries that take part in U.S. missile defense and opted out of a key Soviet - era arms control treaty .

Paraphrase Generation: Putin has pointed out that Russia threatens to take part in nuclear weapons control at a point that many European countries have relied on non - proliferation and missile defense programs .

Input Sentence: Large oil revenues in recent years have allowed the Government of the Russia Federation to buy weapons and fund the development of new missiles .

Paraphrase Generation: Large Government spending in the past has allowed the Government of Russia to fund and develop new weapons of mass destruction .

REFERENCES

[ALZ15] Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. "Broad-coverage CCG Semantic Parsing with AMR." In *EMNLP*, 2015.

[AR17] Omri Abend and Ari Rappoport. "The State of the Art in Semantic Representation." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 77–89, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[BA17] Miguel Ballesteros and Yaser Al-Onaizan. "AMR Parsing using Stack-LSTMs." In *EMNLP*, 2017.

[BBC13] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. "Abstract Meaning Representation for Sembanking." In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[BBN21] Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. "One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation without a Complex Pipeline." In *AAAI Conference on Artificial Intelligence*, 2021.

[BDA20] Claire Bonial, L. Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David R. Traum, and Clare R. Voss. "Dialogue-AMR: Abstract Meaning Representation for Dialogue." In *International Conference on Language Resources and Evaluation*, 2020.

[BG16] Guntis Barzdins and Didzis Gosko. "RIGA at SemEval-2016 Task 8: Impact of Smatch Extensions and Character-Level Neural Translation on AMR Parsing Accuracy." In **SEMEVAL*, 2016.

[BLD20] Claire Bonial, Stephanie M. Lukin, David Doughty, Steven Hill, and Clare R. Voss. "InfoForager: Leveraging Semantic Search with AMR for COVID-19 Research." In *DMR*, 2020.

[BMN91] John Bateman, Christian Matthiessen, Keizo Nanri, and Licheng Zeng. "The Re-Use of Linguistic Resources across Languages in Multilingual Generation Components." In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'91, p. 966–971, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[CK13] Shu Cai and Kevin 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)*, pp. 748–752, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[CL19] Deng Cai and Wai Lam. "Core Semantic First: A Top-down Approach for AMR Parsing." 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)*, pp. 3799–3809, Hong Kong, China, November 2019. Association for Computational Linguistics.

[CL20] Deng Cai and Wai Lam. "AMR Parsing via Graph-Sequence Iterative Inference." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1290–1301, Online, July 2020. Association for Computational Linguistics.

[DK17] Shibhansh Dohare and Harish Karnick. "Text Summarization using Abstract Meaning Representation." *ArXiv*, **abs/1706.01678**, 2017.

[DKB16] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. "Recurrent Neural Network Grammars." In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 199–209, San Diego, California, June 2016. Association for Computational Linguistics.

[DM18] Timothy Dozat and Christopher D. Manning. "Simpler but More Accurate Semantic Dependency Parsing." In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[DZF22] Andrew Drozdov, Jiawei Zhou, Radu Florian, Andrew McCallum, Tahira Naseem, Yoon Kim, and Ramón Fernández Astudillo. "Inducing and Using Alignments for Transition-based AMR Parsing." *ArXiv*, **abs/2205.01464**, 2022.

[FBN20] Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. "Transition-based Parsing with Stack-Transformers." In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1001–1007, Online, November 2020. Association for Computational Linguistics.

[FDS16] Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime G. Carbonell. "CMU at SemEval-2016 Task 8: Graph-based AMR Parsing with Infinite Ramp Loss." In **SEMEVAL*, 2016.

[GWG22] Sarik Ghazarian, Nuan Wen, A. G. Galstyan, and Nanyun Peng. "DEAM: Dialogue Coherence Evaluation using AMR-based Semantic Manipulations." In *ACL*, 2022.

[HAR18] Daniel Hershcovich, Omri Abend, and Ari Rappoport. "Multitask Parsing Across Semantic Representations." *ArXiv*, **abs/1805.00287**, 2018.

[HC21] Kuan-Hao Huang and Kai-Wei Chang. "Generating Syntactically Controlled Paraphrases without Using Annotated Parallel Pairs." *ArXiv*, **abs/2101.10579**, 2021.

[IDC18] Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. "Abstract Meaning Representation for Paraphrase Detection." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 442–452, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[KAR21] Pavan Kapanipathi, I. Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander G. Gray, Ramón Fernández Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, Dinesh Garg, A. Gliozzo, Sairam Gurajada, Hima P. Karanam, Naweed Khan, Dinesh Khandelwal, Young suk Lee, Yunyao Li, Francois P. S. Luus, Ndivhuwo Makondo, Nandana Mihindukulasooriya, Tahira Naseem, Sumit Neelam, Lucian Popa, Revanth Reddy Gangi Reddy, Ryan Riegel, Gaetano Rossiello, Udit Sharma, G. P. Shrivatsa Bhargav, and Mo Yu. "Leveraging Abstract Meaning Representation for Knowledge Base Question Answering." In *Findings*, 2021.

[KIY17] Ioannis Konstas, Srini Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. "Neural AMR: Sequence-to-Sequence Models for Parsing and Generation." *ArXiv*, **abs/1704.08381**, 2017.

[LKV21] Fei-Tzin Lee, Christopher Kedzie, Nakul Verma, and Kathleen McKeown. "An analysis of document graph construction methods for AMR summarization." *ArXiv*, **abs/2111.13993**, 2021.

[LLG20] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[LOG19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *ArXiv*, **abs/1907.11692**, 2019.

[LOJ20] Jung Hoon Lim, Dongsuk Oh, Yoonna Jang, Kisu Yang, and Heuiseok Lim. "I Know What You Asked: Graph Path Learning using AMR for Commonsense Reasoning." In *COLING*, 2020.

[LSX19] Bill Yuchen Lin, Minghan Shen, Yu Xing, Pei Zhou, and Xiang Ren. "CommonGen: A Constrained Text Generation Dataset Towards Generative Commonsense Reasoning." *ArXiv*, **abs/1911.03705**, 2019.

[LT18] Chunchuan Lyu and Ivan Titov. "AMR Parsing as Graph Prediction with Latent Alignment." In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[MB16] Arindam Mitra and Chitta Baral. "Addressing a Question Answering Challenge by Combining Statistical Methods with Inductive Rule Learning and Reasoning." In *AAAI Conference on Artificial Intelligence*, 2016.

[MG18] Ritwik Mishra and Tirthankar Gayen. "Automatic Lossless-Summarization of News Articles with Abstract Meaning Representation." *Procedia Computer Science*, **135**:178–185, 2018.

[NB17] Rik van Noord and Johan Bos. "Neural Semantic Parsing by Character-based Translation: Experiments with Abstract Meaning Representations." *ArXiv*, **abs/1705.09980**, 2017.

[Niv03] Joakim Nivre. "An Efficient Algorithm for Projective Dependency Parsing." In *Proceedings of the Eighth International Conference on Parsing Technologies*, pp. 149–160, Nancy, France, April 2003.

[NRM21] Tahira Naseem, Srinivas Ravishankar, Nandana Mihindukulasooriya, Ibrahim Abdelaziz, Young-Suk Lee, Pavan Kapanipathi, Salim Roukos, Alfio Gliozzo, and Alexander Gray. "A Semantics-aware Transformer Model of Relation Linking for Knowledge Base Question Answering." pp. 256–262, 01 2021.

[OHW21] Juri Opitz, Philip Heinisch, Philip Wiesenbach, Philipp Cimiano, and Anette Frank. "Explainable Unsupervised Argument Similarity Rating with Abstract Meaning Representation and Conclusion Generation." In *Workshop on Argument Mining*, 2021.

[OKM16] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Zdeňka Urešová. "Towards Comparability of Linguistic Graph Banks for Semantic Parsing." In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3991–3995, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA).

[PRW02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation." In *Annual Meeting of the Association for Computational Linguistics*, 2002.

[PWG17] Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. "Addressing the Data Sparsity Issue in Neural AMR Parsing." In *EACL*, 2017.

[RMK17] Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé. "Biomedical Event Extraction using Abstract Meaning Representation." In *Workshop on Biomedical Natural Language Processing*, 2017.

[RSR19] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *ArXiv*, **abs/1910.10683**, 2019.

[RWC19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners." 2019.

[SGZ19] Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. "Semantic Neural Machine Translation Using AMR." *Transactions of the Association for Computational Linguistics*, **7**:19–31, 2019.

[SLM17] A. See, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks." *ArXiv*, **abs/1704.04368**, 2017.

[TSO16] Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. "Neural Headline Generation on Abstract Meaning Representation." In *Conference on Empirical Methods in Natural Language Processing*, 2016.

[WAM15] Keenon Werling, Gabor Angeli, and Christopher D. Manning. "Robust Subgraph Generation Improves Abstract Meaning Representation Parsing." In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 982–991, Beijing, China, July 2015. Association for Computational Linguistics.

[ZJ21] Zixuan Zhang and Heng Ji. "Abstract Meaning Representation Guided Graph Encoding and Decoding for Joint Information Extraction." In *NAACL*, 2021.

[ZMD19a] Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. "AMR Parsing as Sequence-to-Graph Transduction." In *Annual Meeting of the Association for Computational Linguistics*, 2019.

[ZMD19b] Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. "Broad-Coverage Semantic Parsing as Transduction." In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[ZNA21] Jiawei Zhou, Tahira Naseem, Ramón Fernández Astudillo, and Radu Florian. "AMR Parsing with Action-Pointer Transformer." In *NAACL*, 2021.

[ZNF21] Jiawei Zhou, Tahira Naseem, Ramón Fernandez Astudillo, Young-Suk Lee, Radu Florian, and Salim Roukos. "Structure-aware Fine-tuning of Sequence-to-sequence Transformers for Transition-based AMR Parsing." In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6279–6290, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[ZPJ21] Zixuan Zhang, Nikolaus Nova Parulian, Heng Ji, Ahmed Elsayed, Skatje Myers, and Martha Palmer. "Fine-grained Information Extraction from Biomedical Literature based on Knowledge-enriched Abstract Meaning Representation." In *Annual Meeting of the Association for Computational Linguistics*, 2021.