ANLP: AMR Parsing and Generation

Neha Verma
09.17.2020

Outline

1. Introduction & Background
   a. AMR explained
   b. Parsing and Generation
2. Papers
   a. 2 AMR parsing papers
   b. 2 AMR-to-text generation papers
3. Future Directions
4. Discussion
5. Reference

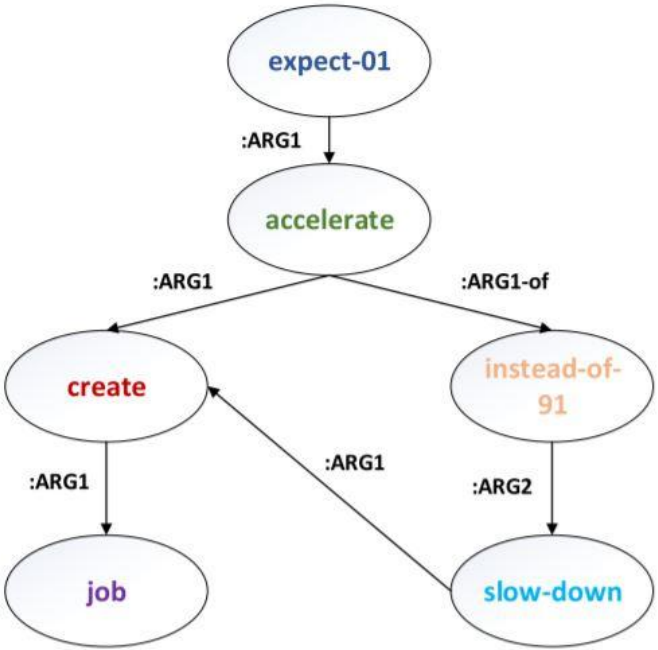
What is Abstract Meaning Representation?

- **Semantic formalism** for sentence representation
  - Entities = nodes
  - Relations = edges
    - “Who is doing what to whom”
  - Rooted, directed graphs
- **Far from syntactic rep**
  - Multiple surface realizations

Creation of AMR

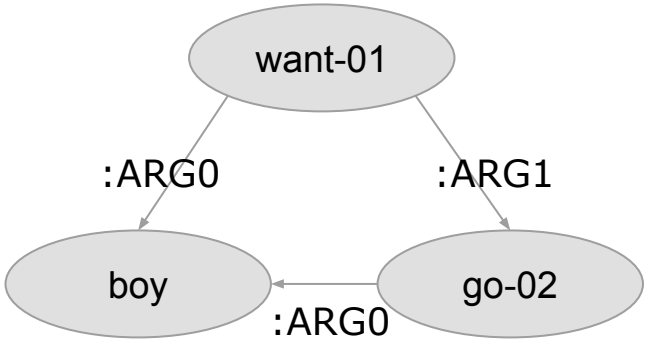
- Introduced in paper: Abstract Meaning Representation for Sembanking
  - Banarescu et al., 2013 (ACL)
- Proposed to unify semantic tasks using solid representation
  - Named entities
  - Coreference
  - Semantic relations
- Proposed with SemBank (semantic bank) to encourse semantic parsing
  - Like Penn TreeBank with syntactic parsing

AMR Examples



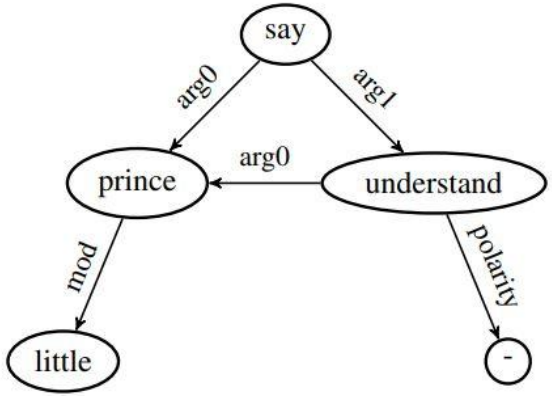
Job creation is not expected to slow down but will instead accelerate.

https://www.aclweb.org/anthology/2020.tacl-1.2.pdf



The boy wants to go.

https://www.aclweb.org/anthology/W13-2322.pdf

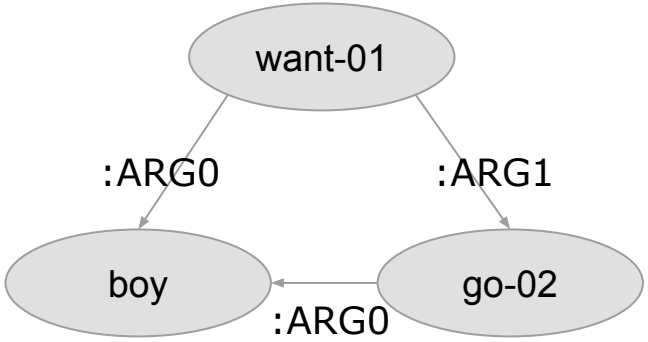


The Little Prince said, "I do not understand."

https://www.melanietosik.com/files/amr.pdf

AMR Contents

- Approximately 100 relations + inverses
  - Frame arguments
    - :arg0, arg1, ...
  - Date-entities relations
    - :year, :timezone
  - List relations
    - :op1, op2, ...
  - Quantity relations
    - :quant, :unit
  - General semantic relations
    - :topic, :frequency, :employed-by
- PropBank Frames (verbs with sense enumeration)
  - Dictates associated arguments



The boy wants to go.

https://www.aclweb.org/anthology/W13-2322.pdf

X ARG0-of Y = Y ARG0 X

https://github.com/nschneid/amr-tutorial/

Propbank examples

run-01 - **"operate, proceed, operate or proceed"**

- ARG0: operator
- ARG1: machine, operation, procedure
- ARG2: employer
- ARG3: coworker
- ARG4: instrumental

Aliases: run (v), run (n), running (n)
more

---

run-02 - **"walk quickly, a course or contest, run/jog, run for office"**

- ARG0: runner theme
- ARG1: course, race, distance location
- ARG2: opponent

Aliases: run (v), run (n), running (n)
more

---

run-03 - **"cost"**

- ARG1: commodity
- ARG2: price
- ARG3: buyer

Aliases: run (v), running (n)

Roleset Id: leave.01 *the act of moving away from*
Roles: Arg0: *entity leaving*
Arg1: *place, person, or thing left*
Arg2: *destination*
Example: *John left Texas for Colorado.*
Roleset Id: leave.02 *give, bequeath*
Roles: Arg0: *giver/leaver*
Arg1: *thing given*
Arg2: *benefactive, given-to*
Example: *Mary left her daughter the diamond pendant.*

- Martha Palmer, Dan Gildea, Paul Kingsbury, The Proposition Bank: A Corpus Annotated with Semantic Roles *Computational Linguistics Journal, 31:1,* 2005.
- Paul Kingsbury and Martha Palmer. From Treebank to PropBank. 2002. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, Spain.

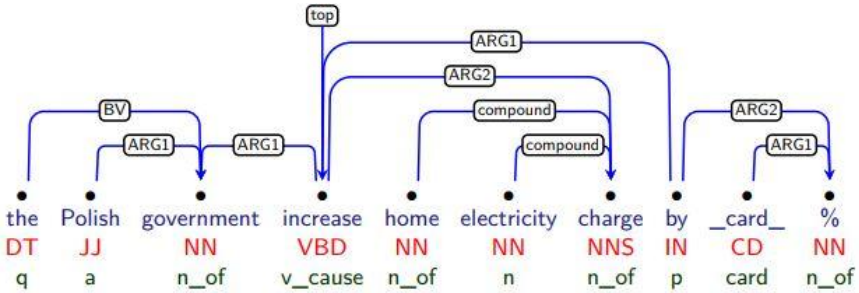
Other semantic representations

What is the meaning of a sentence?

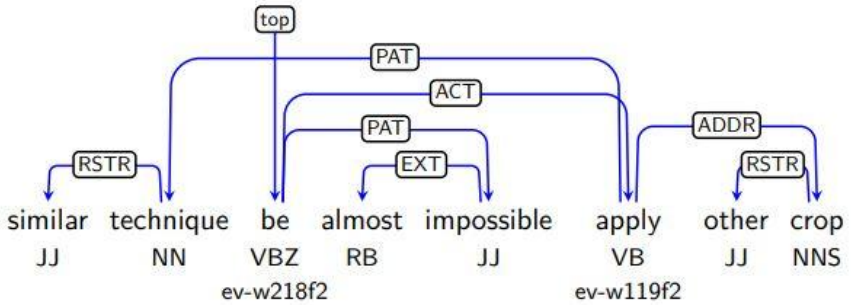
Examples:

- (Baker et al., 1998), PropBank (Palmer et al., 2005)
  - Semantic Role Labeling: explicit predicate/argument structure
  - I gave her a book.
- Bilexical Semantic Dependency
  - Simplest
- Prague Semantic Dependency
  - ACT(or), PAT(ient), ADDR(essee), ORIG(in), EFF(ect),EXT(end) or RSTR(iction)

The Polish government increased home electricity charges by 150%



A similar technique is almost impossible to apply to other crops.



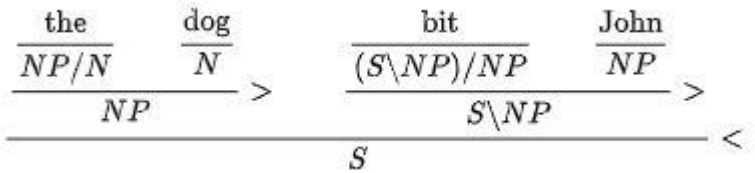
https://github.com/cfmrp/tutorial

Other semantic representations (cont)

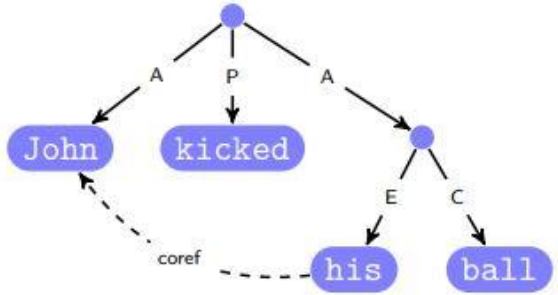
What is the meaning of a sentence?

Examples:

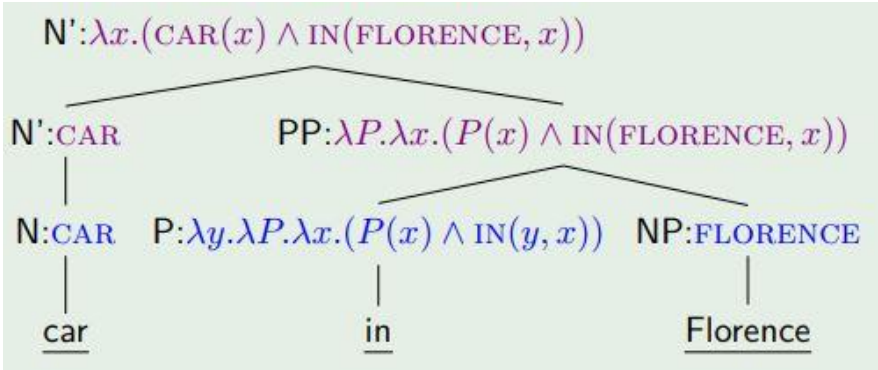
- Universal Conceptual Cognitive Annotation (UCCA)
- Lambda expressions
- Combinatory Categorial Grammar
  - predicate/argument reps



https://en.wikipedia.org/wiki/Combinatory_categorial_grammar



John kicked his ball.

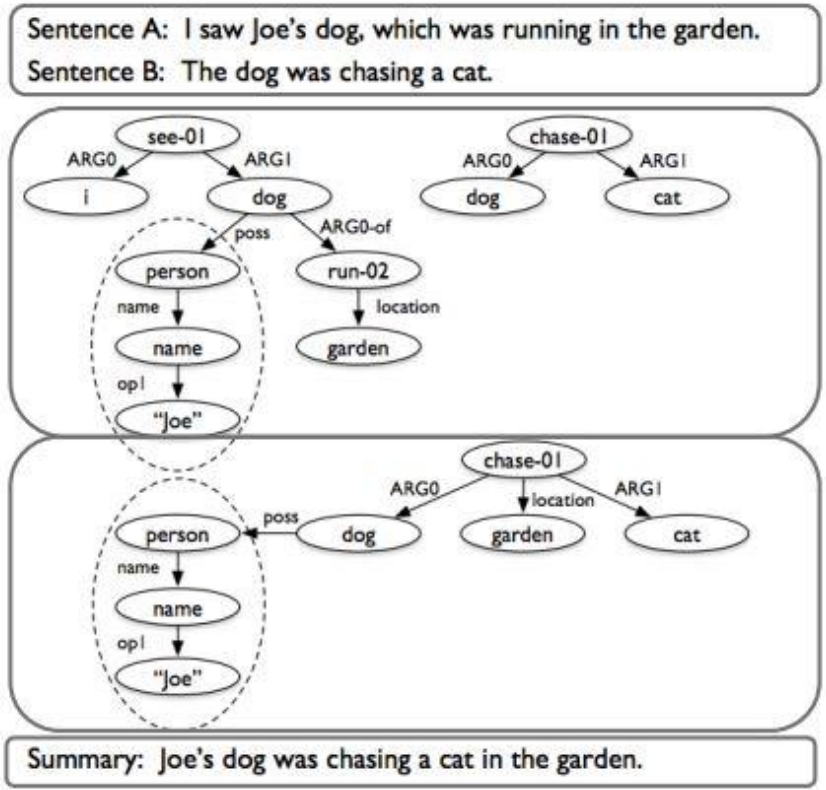


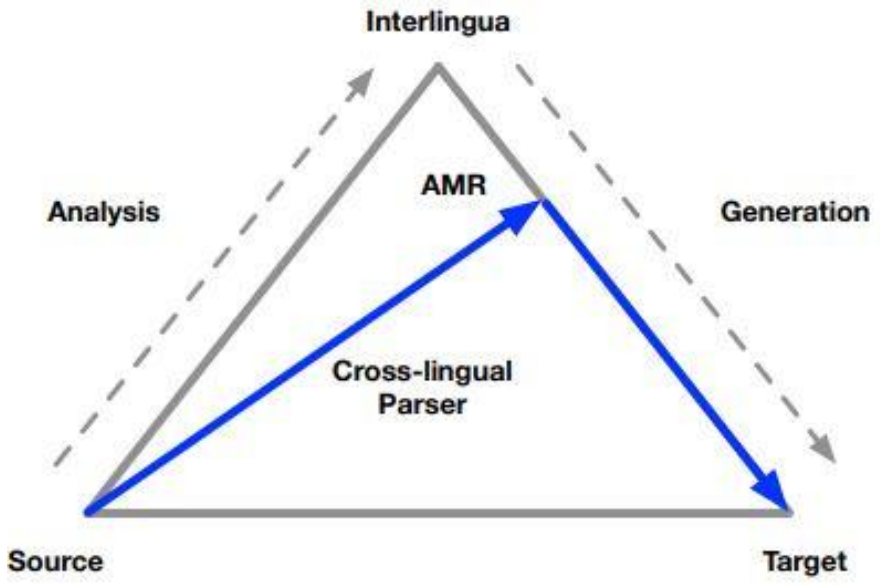
https://github.com/cfmrp/tutorial

(Some) Uses of AMR

- Summarization
  - Merge graphs
  - Condense
- Machine translation
  - Semantics based
- Entity linking
  - Identifying (named) entities in sentence
- Question Answering
  - Can benefit from interpretable semantic rep
- Generation
  - TableSumm!

(Some) Uses of AMR cont.

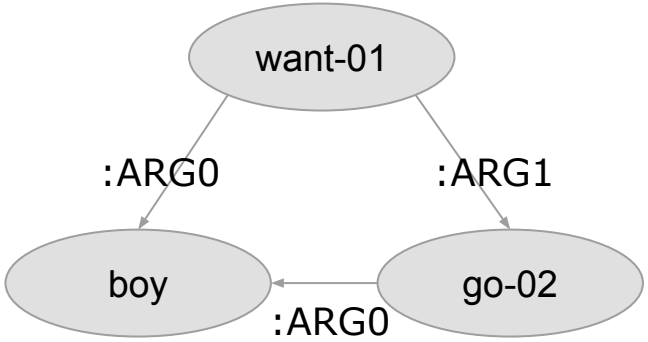




https://github.com/nschneid/amr-tutorial

Reentrance

- Represent coreference
  - Entity plays multiple roles
  - sentence level only
- Directed graph, not tree
  - Node *n* reentrant iff $d_{in}(n) > 1$



The boy wants to go.

https://www.aclweb.org/anthology/W13-2322.pdf

Task: Parsing

- Given a sentence, produce the corresponding AMR graph
  - Hard problem
  - No explicit alignment - heavily meaning based
- Previous work
  - SMT-based
  - Alignment between sentence + nodes
- SMATCH (semantic match) metric:
  - Overlap of logical triples:
  - *relation(variable, concept)*
  - F1 score wrt triples
  - Cai & Knight, 2013

```
instance(a0, read) ∧ instance(a1, girl) ∧ instance(a2, book) ∧ arg0(a0, a1) ∧ arg1(a0, a2)

                    => The girl is reading a book.
```

Task: Parsing

- Older work on AMR Parsing
  - JAMR
    - Concepts identified via semi-Markov model (probabilistic)
    - Relations identified via Maximum Spanning Connected Subgraph Algorithm
      - Like MST
    - https://www.aclweb.org/anthology/P14-1134.pdf
  - Transition based AMR parser
    - Generates dependency parse
    - Transition based algorithm converts dep. parse to AMR
      - Ex: REATTACH, DELETE-NODE, REPLACE-HEAD, SWAP
    - https://www.cs.brandeis.edu/~cwang24/files/transition-based-amr-parsing-naacl2015.pdf

Task: Generation

- Creating text for corresponding AMR graph
- Major classes of methods:
  - Probabilistic
  - Seq2seq
  - Graph2seq
  - Hybrids

Task: Generation

- AMR abstracts away many function words
  - Multiple realizations
- Function words helpful for generation!
  - Making AMR-to-text generation a hard problem
- Flanigan et al., 2016 was first AMR-to-text model
  - https://www.cs.cmu.edu/~jgc/publication/flanigantree.pdf
  - Statistical model
  - Graph → tree → string (via transducer + LM)
    - Like SMT!

Resources

- 3 Linguistic Data Consortium AMR Banks
  - AMR 1.0 (2014)
  - AMR 2.0 (2017)
  - AMR 3.0 (2020)
- SemEval task, 2017
  - Focuses:
  - Parsing biomedical texts
  - Generation
- Portal
  - Abstract Meaning Representation (AMR)
- Tutorials
  - cfmrp/tutorial: Tutorial on 'Graph-Based Meaning Representations: Design and Processing' (ACL 2019)
  - nschneid/amr-tutorial: Abstract Meaning Representation (AMR) tutorial slides

Papers

1. Abstract Meaning Representation for SemBanking
   a. https://www.aclweb.org/anthology/W13-2322.pdf
2. A Discriminative Graph-Based Parser for the Abstract Meaning Representation (JAMR)
   a. https://www.aclweb.org/anthology/P14-1134.pdf
3. A Transition-based Algorithm for AMR Parsing
   a. https://www.cs.brandeis.edu/~cwang24/files/transition-based-amr-parsing-naacl2015.pdf
4. Generation from Abstract Meaning Representation using Tree Transducers
   a. https://www.cs.cmu.edu/~jgc/publication/flanigantree.pdf
5. AMR Parsing as Sequence to Graph Transduction
   a. https://arxiv.org/pdf/1905.08704v2.pdf
6. AMR Parsing via Graph-Sequence Iterative Inference
   a. https://arxiv.org/pdf/2004.05572.pdf
7. AMR-to-text Generation with Graph Transformers
   a. https://www.aclweb.org/anthology/2020.tacl-1.2.pdf
8. Investigating Pretrained Language Models for Graph-to-Text Generation
   a. https://arxiv.org/pdf/2007.08426v1.pdf

Paper 1:

AMR Parsing as Sequence-to-Graph Transduction

Sheng Zhang, Xutai Ma, Kevin Duh, & Benjamin Van Durme, ACL 2019

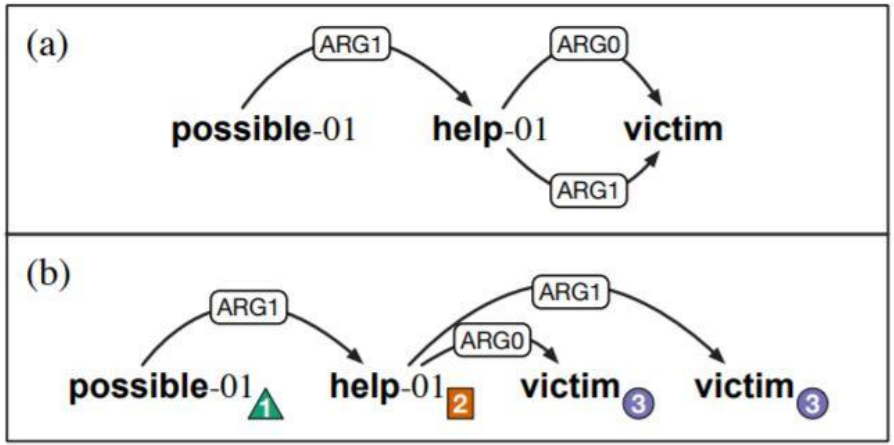
Motivation

- Aligner-free parsing
- Tackle reentrancy directly
- Sequence-to-graph transduction approach
  - attention-based

Another view of reentrance

- Copy node with $d_{in}(n) > 1$
  - Now, $d_{in}(n_{copy}) = 1$
- Annotate with indices
  - Node identity preserved by same index
- **Creates tree structure**

The victim could help himself.



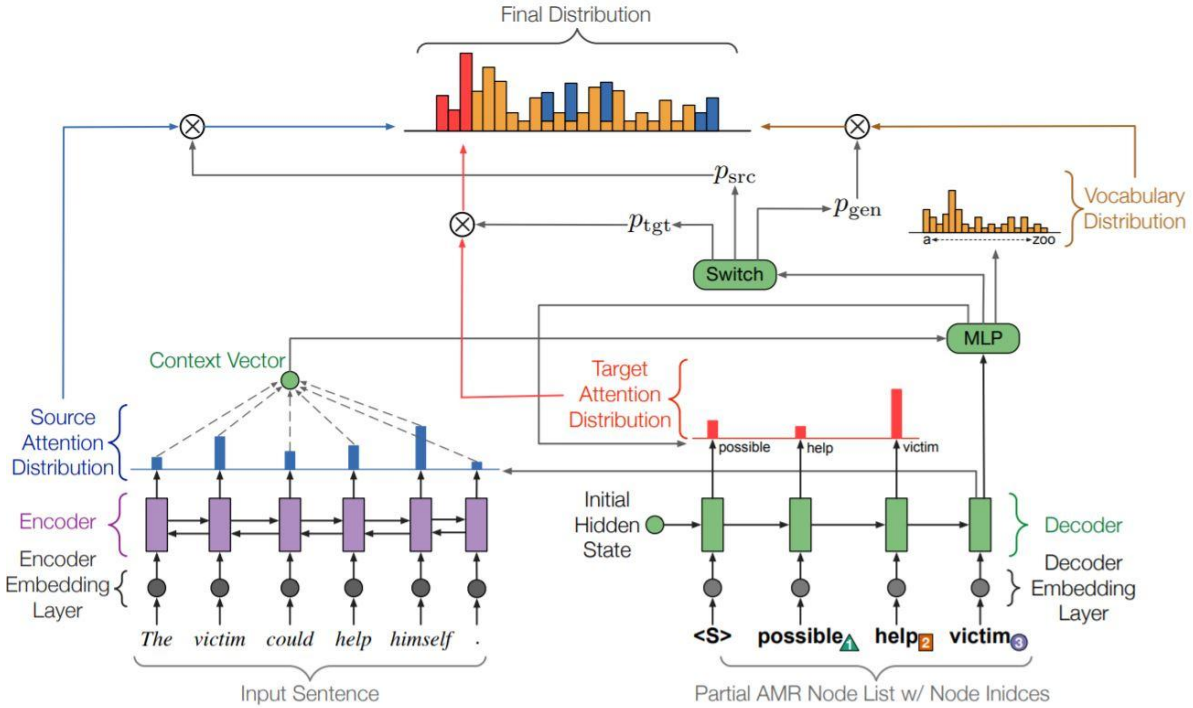
Parsing task

Treated parsing as sequential steps:

1. Node prediction
    a. Given sentence, predict sequence of AMR nodes & indices
2. Edge prediction
    a. Given sentence & node/index list, find best tree with relation edges

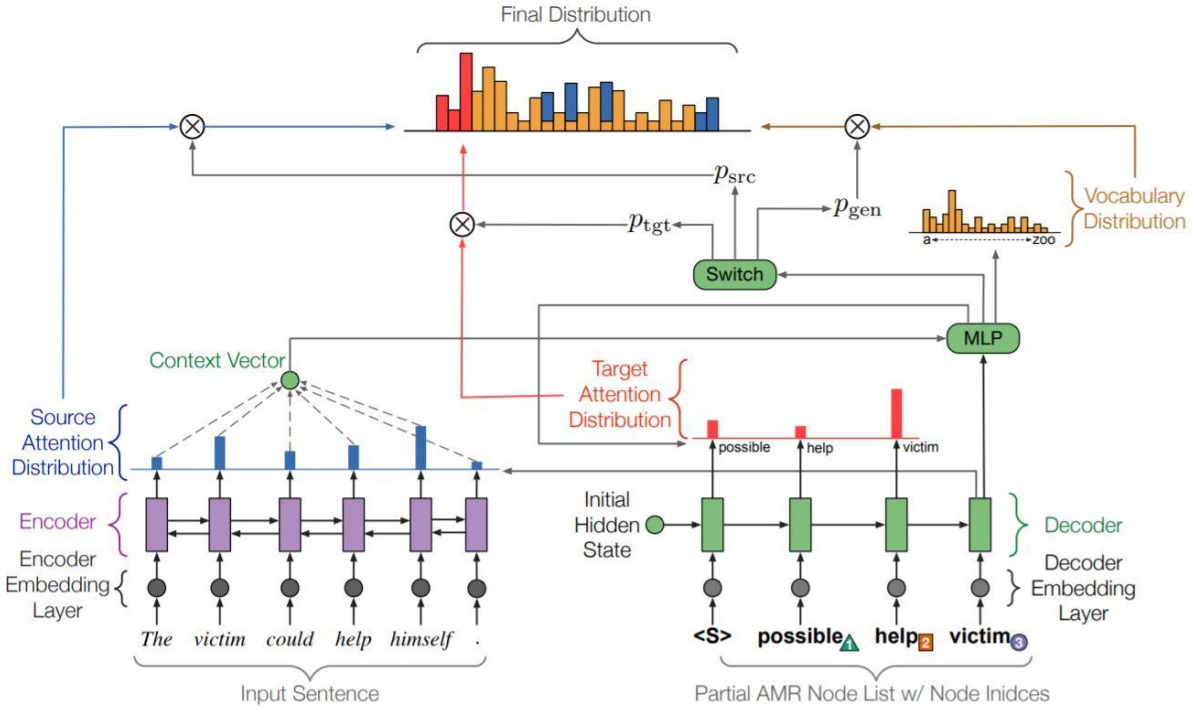
Node prediction

- **Pointer generator**
  - Copy from input via pointing OR
  - Generate novel words
  - Including previous nodes!
    - Reentrance
- **Architecture**
  - Pretrained embeddings
  - LSTM Encoder & Decoder



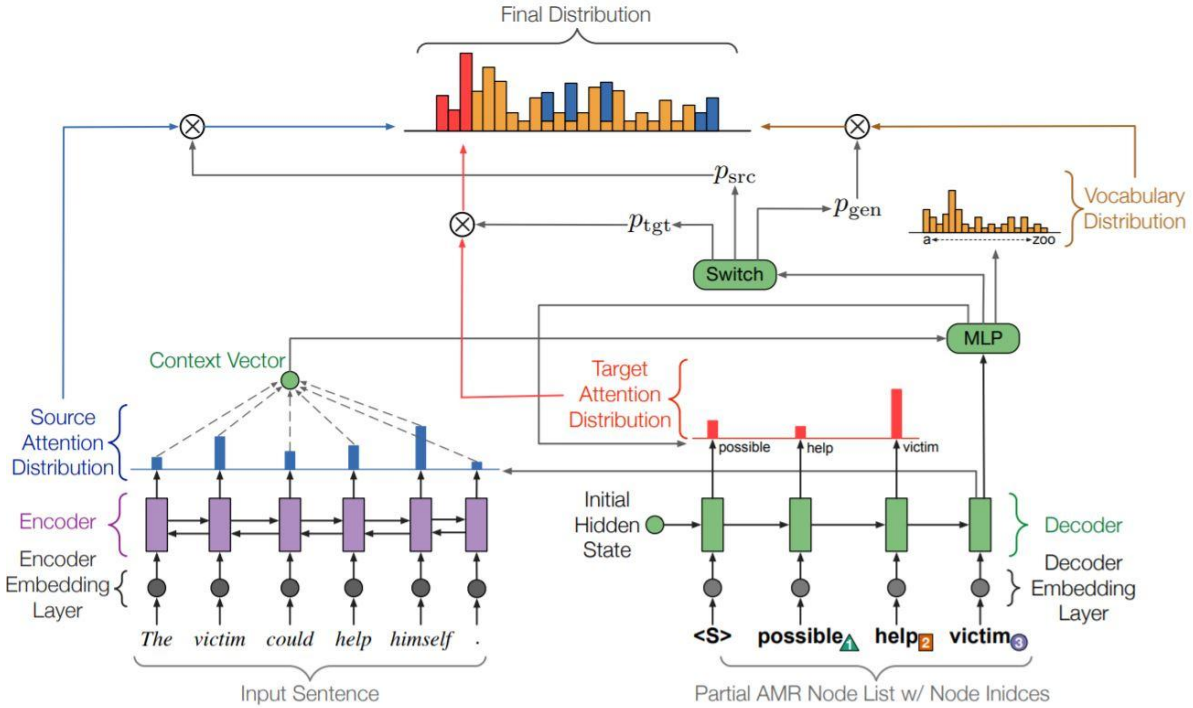
Node prediction - attentions

- **Source attention**
  - Additive attn, for context vector
- **Attention vector**
  - MLP computes
  - Source + target info
  - Vocab distribution
  - Target attn dist
  - Fed to switch for pointer-generator step



Node prediction - emission

- **Generation**
  - p_{tgt} emit from already emitted (reentrance!)
  - p_{src} copy node from source
  - p_{gen} emit from vocab distribution

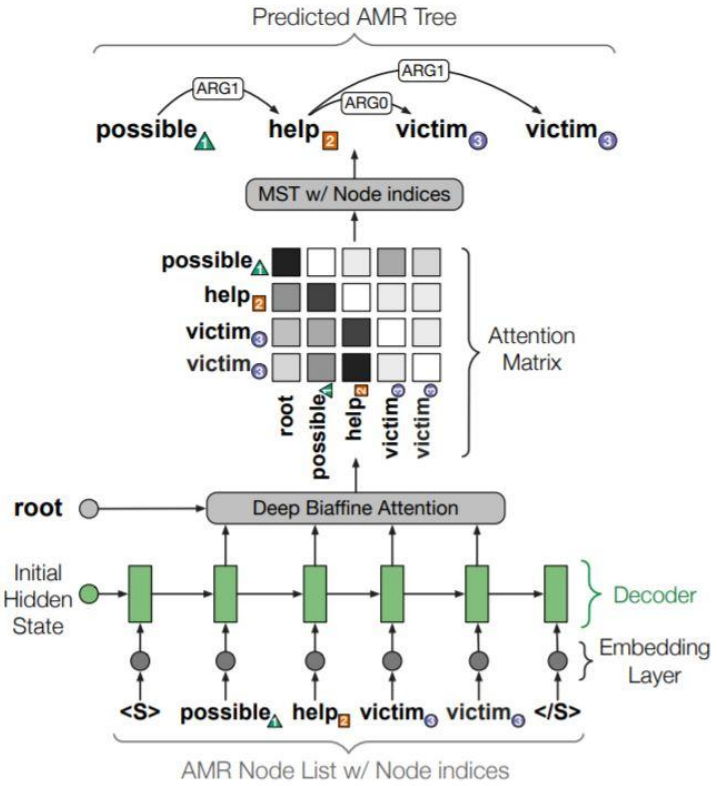


Edge Prediction

- Uses decoder hidden states $\{s_i\}$
- Joint training
- Find maximum spanning tree
  - Decide if edge is in tree
  - Then label it

$$\text{Biaffine}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^\top \boldsymbol{U} \boldsymbol{x}_2 + \boldsymbol{W}[\boldsymbol{x}_1; \boldsymbol{x}_2] + \boldsymbol{b}$$

$$\text{score}_{k,t}^{\text{edge}} = \text{Biaffine}(\text{MLP}^{(\text{edge-head})}(s_k), \text{MLP}^{(\text{edge-dep})}(s_t))$$



Results

- Corpora
  - LDC2017T10
  - LDC2014T12
- Significant increase in reentrance Smatch

Corpus	Parser	F1(%)
AMR 2.0	Buys and Blunsom (2017)	61.9
	van Noord and Bos (2017b)	71.0*
	Groschwitz et al. (2018)	71.0±0.5
	Lyu and Titov (2018)	74.4±0.2
	Naseem et al. (2019)	75.5
	Ours	**76.3**±0.1
AMR 1.0	Flanigan et al. (2016)	66.0
	Pust et al. (2015)	67.1
	Wang and Xue (2017)	68.1
	Guo and Lu (2018)	68.3±0.4
	Ours	**70.2**±0.1

Metric	vN'18	L'18	N'19	Ours
SMATCH	71.0	74.4	75.5	**76.3**±0.1
Unlabeled	74	77	**80**	79.0±0.1
No WSD	72	76	76	**76.8**±0.1
Reentrancies	52	52	56	**60.0**±0.1
Concepts	82	**86**	**86**	84.8±0.1
Named Ent.	79	**86**	83	77.9±0.2
Wikification	65	76	80	**85.8**±0.3
Negation	62	58	67	**75.2**±0.2
SRL	66	70	**72**	69.7±0.2

Summary & contributions

- Attention-based AMR parser
  - No aligners
  - Handles reentrance well
  - Separates node & edge prediction in 2 steps
- Best AMR parser at time of publication

Paper 2:

AMR Parsing via Graph-Sequence Iterative Inference

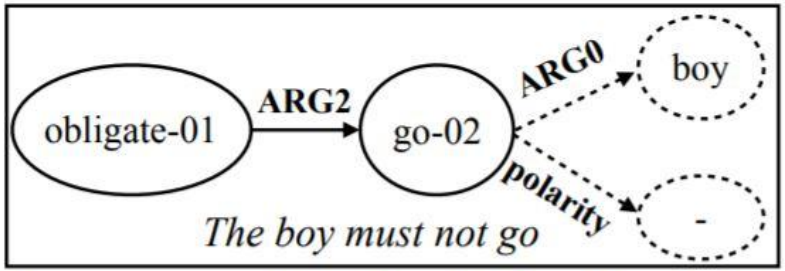
Deng Cai & Wai Lam, arXiv 2020

Motivation

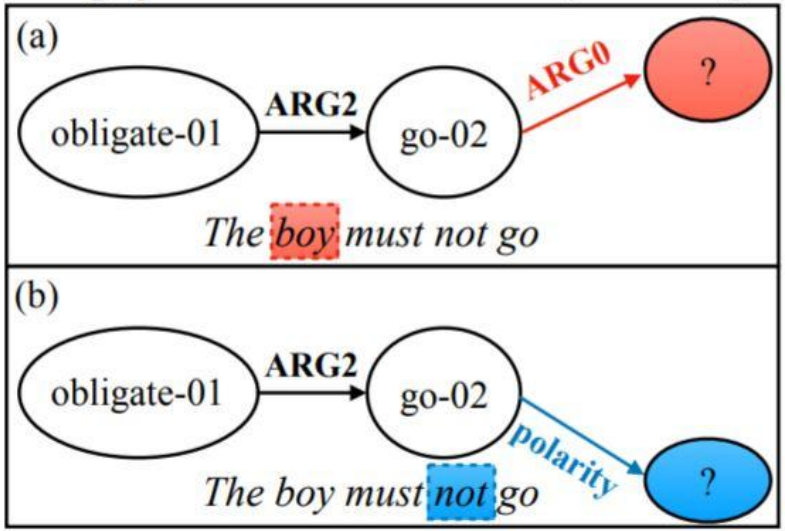
- Previous graph-seq transduction models don’t predict in one step
  - We’ve seen split 1) node prediction and 2) edge prediction steps
- Iterating between input sequence & current constructed subgraph
  - Combines the steps more
  - Closely models human deduction process

Iterative decision making

- Dual graph-sequence decisions
- Iterative inference
- Graph structure informs sequence scope
- Sequence scope informs graph structure



The current partial (solid) and full (solid + dashed) AMR graphs for the sentence "*The boy must no go*"

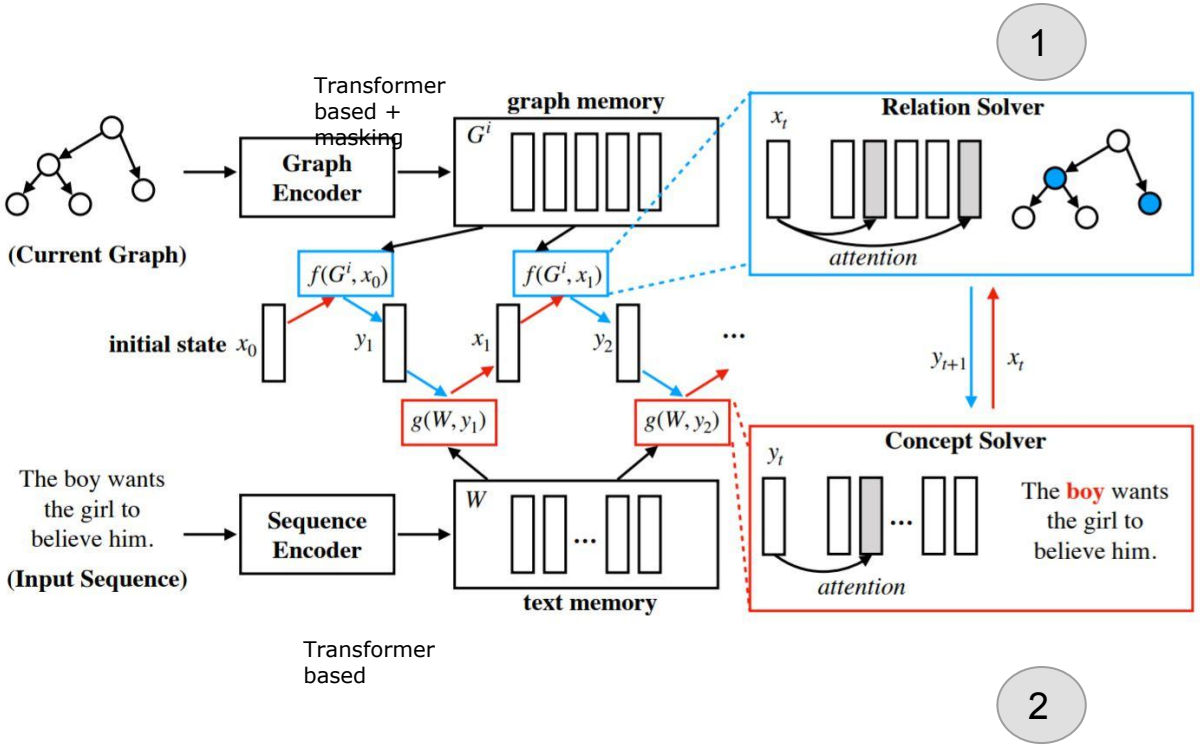


Model

- Steps: $\{G^0, G^1, \ldots\}$
- Input sequence W

states

$$y_t^i = g(G^i, x_t^i),$$
$$x_{t+1}^i = f(W, y_t^i),$$



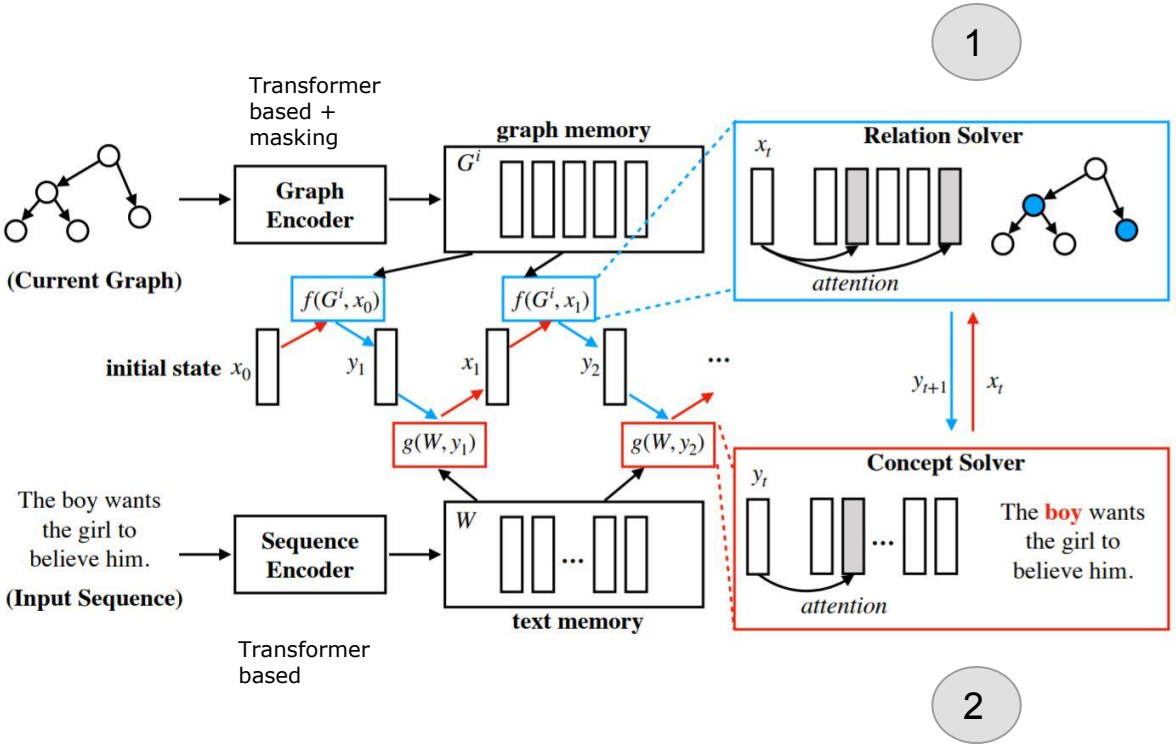
Model

Graph encoder

- Transformer
- Masked self-attn & source attn
  - Look at nodes up to that position
  - Look at input sequence

Sequence encoder

- Transformer



Concept solver (node prediction)

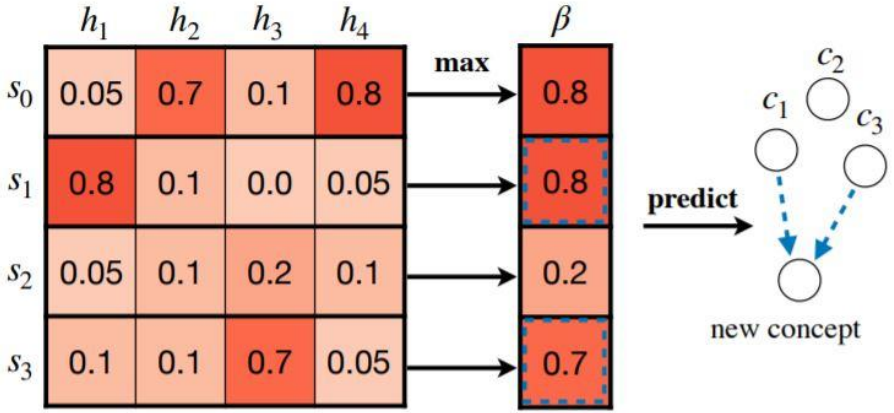
- Inputs: Sequence encoding (h), previous graph decision (y)
- Compute attention @ step t
- Form prob dist across vocab from attn.
- Node generation
  - Generate from prob dist across vocab
  - Copy from input text

$$\alpha_t = \text{softmax}(\frac{(W^Q y_t)^{\mathrm{T}} W^K h_{1:n}}{\sqrt{d_k}}),$$

Relation solver (edge prediction)

- Inputs: graph encoding (s), last node decision (x)
- Attach new node to graph
  - Figure out where is source node
  - Uses multihead attention
    - Max over heads for edge probability
- Biaffine classifier for edges

$$\beta_t^h = \text{softmax}\left(\frac{(W_h^Q x_t)^{\mathrm{T}} W_h^K s_{0:m}}{\sqrt{d_k}}\right).$$

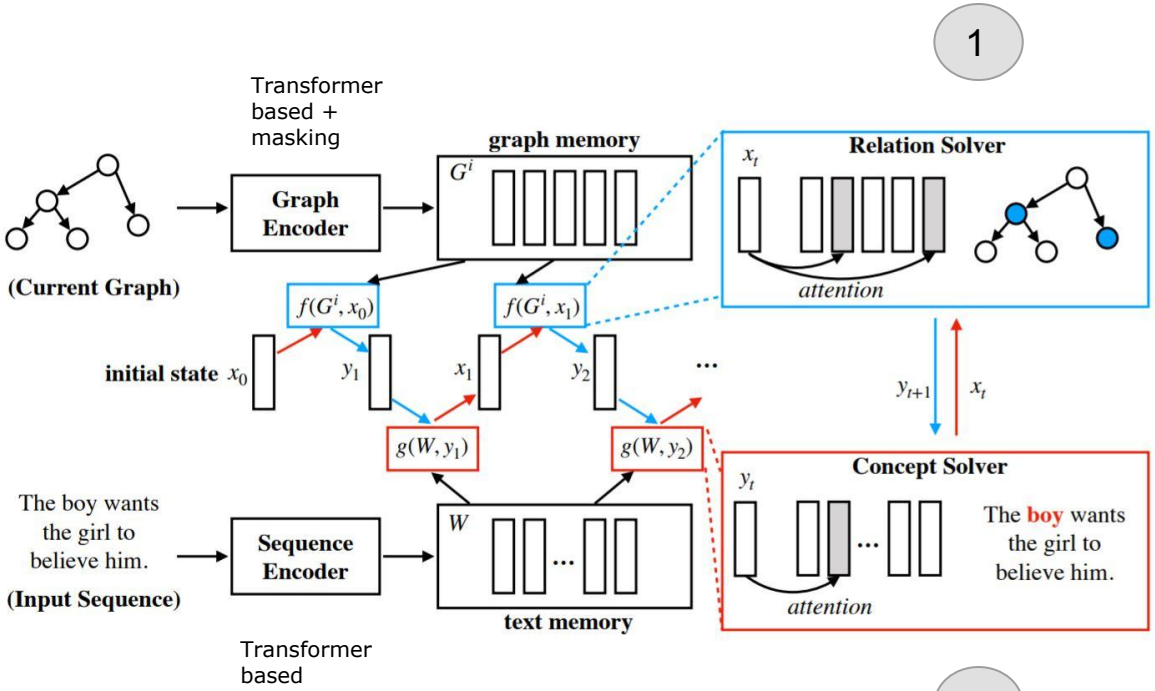


Model revisited

- How both solvers fit
- Iteratively predict
- Attns in solvers give state values

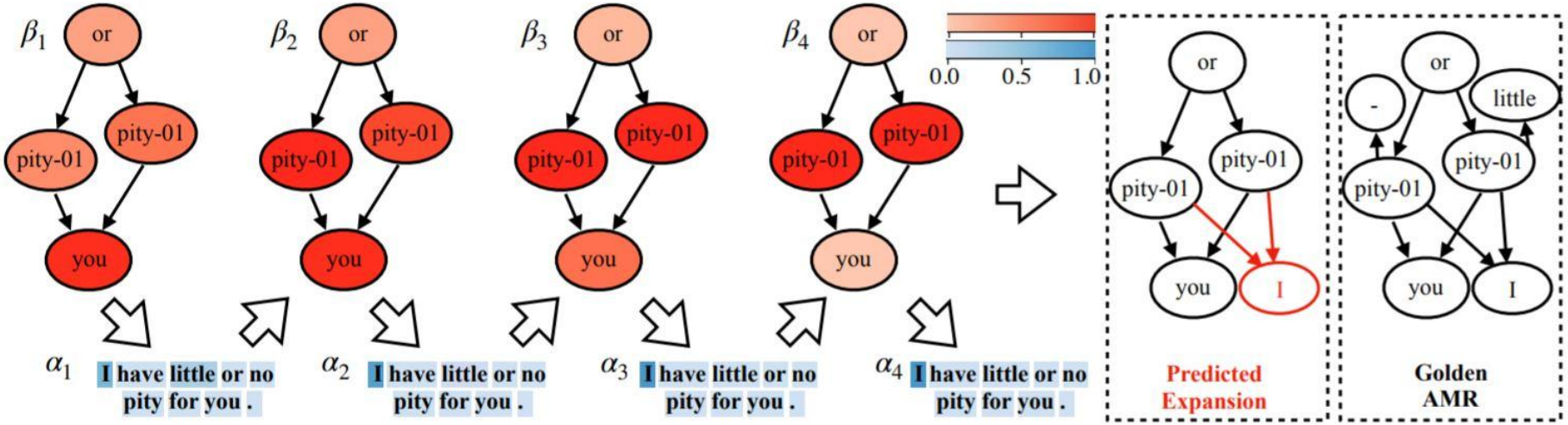
$$y_t^i = g(G^i, x_t^i),$$
$$x_{t+1}^i = f(W, y_t^i),$$

$$y_t = \mathrm{FFN}^{(y)}(x_t + (W^V h_{1:n})\alpha_t),$$
$$x_{t+1} = \mathrm{FFN}^{(x)}(y_t + \sum_{h=1}(W_h^V s_{0:n})\beta_t^h),$$



Example

alpha = attn for node prediction
beta = attn for edge attachments



Results

- Corpora
  - LDC2017T10 (2.0)
  - LDC2014T12 (1.0)

Model	G. R.	BERT	SMATCH	fine-grained evaluation							
				Unlabeled	No WSD	Concept	SRL	Reent.	Neg.	NER	Wiki
van Noord and Bos (2017)	×	×	71.0	74	72	82	66	52	62	79	65
Groschwitz et al. (2018)	✓	×	71.0	74	72	84	64	49	57	78	71
Lyu and Titov (2018)	✓	×	74.4	77.1	75.5	85.9	69.8	52.3	58.4	86.0	75.7
Cai and Lam (2019)	×	×	73.2	77.0	74.2	84.4	66.7	55.3	62.9	82.0	73.2
Lindemann et al. (2019)	✓	✓	75.3	-	-	-	-	-	-	-	-
Naseem et al. (2019)	✓	✓	75.5	80	76	86	72	56	67	83	80
Zhang et al. (2019a)	✓	×	74.6	-	-	-	-	-	-	-	-
Zhang et al. (2019a)	✓	✓	76.3	79.0	76.8	84.8	69.7	60.0	75.2	77.9	85.8
Zhang et al. (2019b)	✓	✓	77.0	80	78	86	71	61	77	79	86
Ours	×	×	74.5	77.8	75.1	85.9	68.5	57.7	65.0	82.9	81.1
	✓	×	77.3	80.1	77.9	86.4	69.4	58.5	75.6	78.4	86.1
	×	✓	78.7	81.5	79.2	88.1	**74.5**	63.8	66.1	**87.1**	81.3
	✓	✓	**80.2**	**82.8**	**80.8**	**88.1**	74.2	**64.6**	**78.9**	81.1	**86.3**

Summary & contributions

- Novel parsing model via iterative interface
  - Connects problems of node (concept) and edge (relation) prediction
  - Where to abstract (node) ⇄ Where to construct (edge)
- Current best AMR parser (SMATCH)
  - Best even without BERT! ⇒ very strong model approach
  - Improves further with BERT

Paper 3:

AMR-to-Text Generation with Graph Transformers

Tianming Wang, Xiaojun Wan & Hanqi Jin, TACL 2020

Motivation

- Gap between Graph-to-Seq and Seq-to-Seq performance was small
- Graph-to-Sequence models **directly encode graph structure**
  - Linearizing loses info

Model

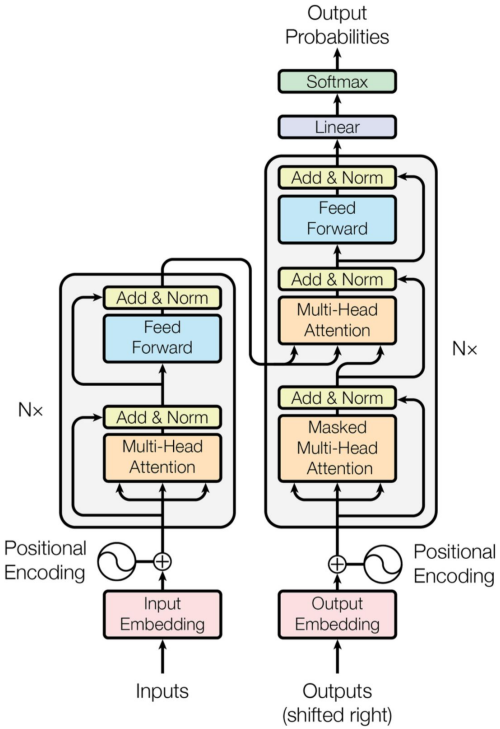
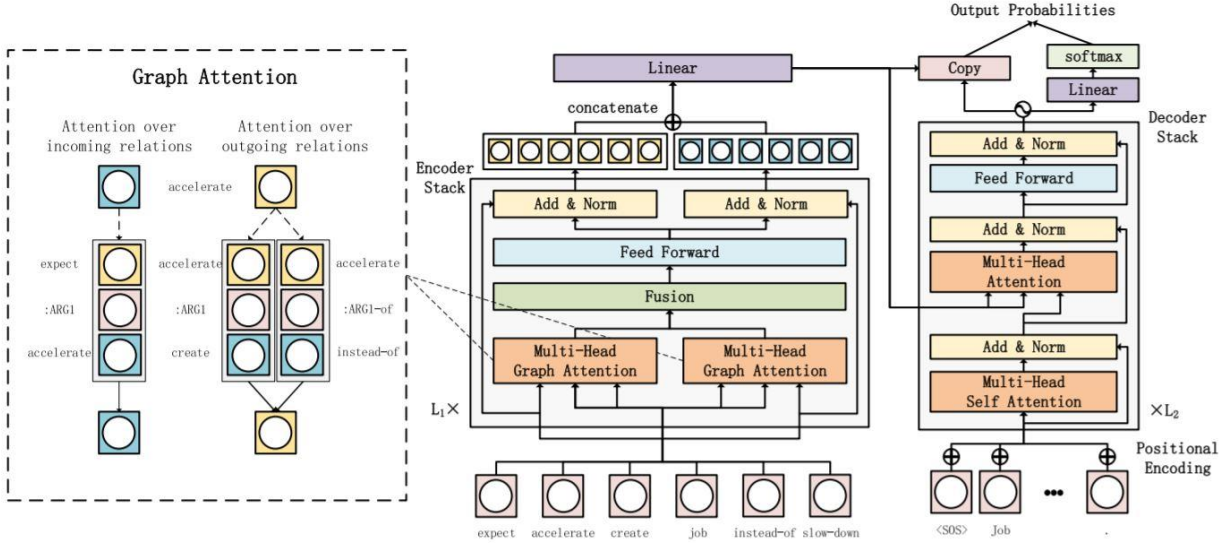
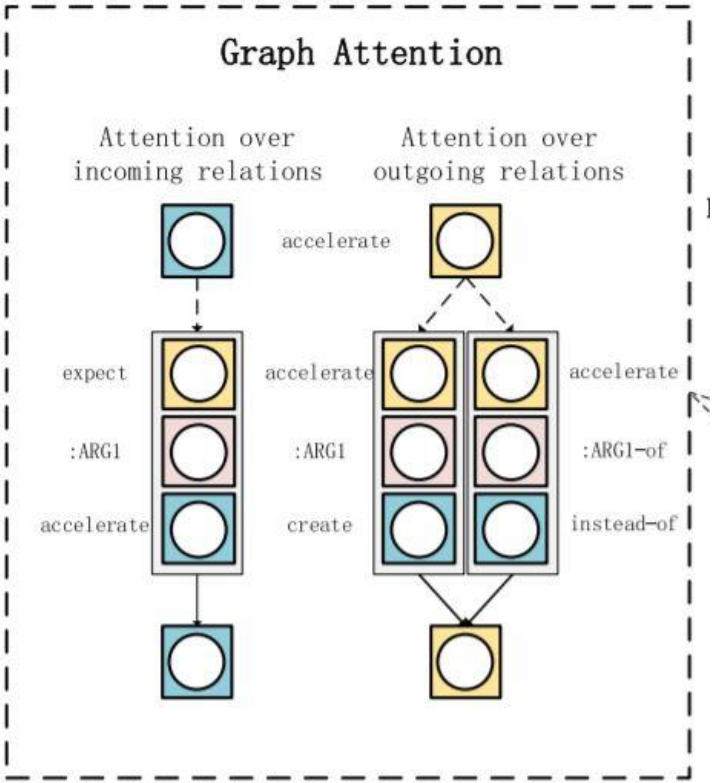


Figure 1: The Transformer - model architecture.

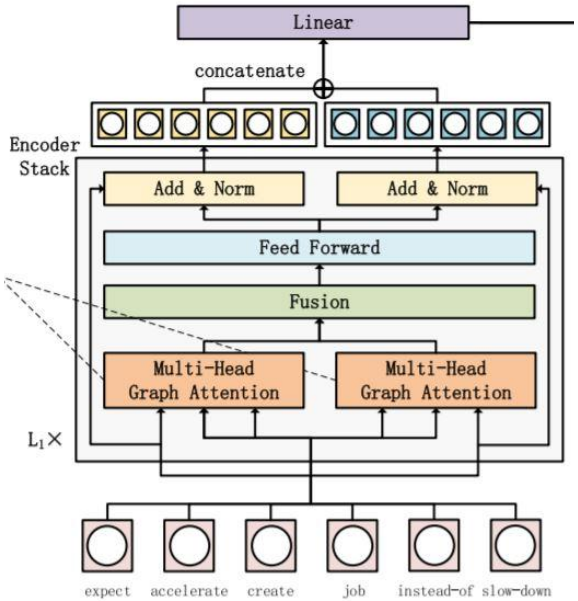


Dual Graph Attention



- Considers separately
  - $N_{in}(v)$
  - $N_{out}(V)$
- Multiple layers allow nodes > 1 step away to be in attn vector

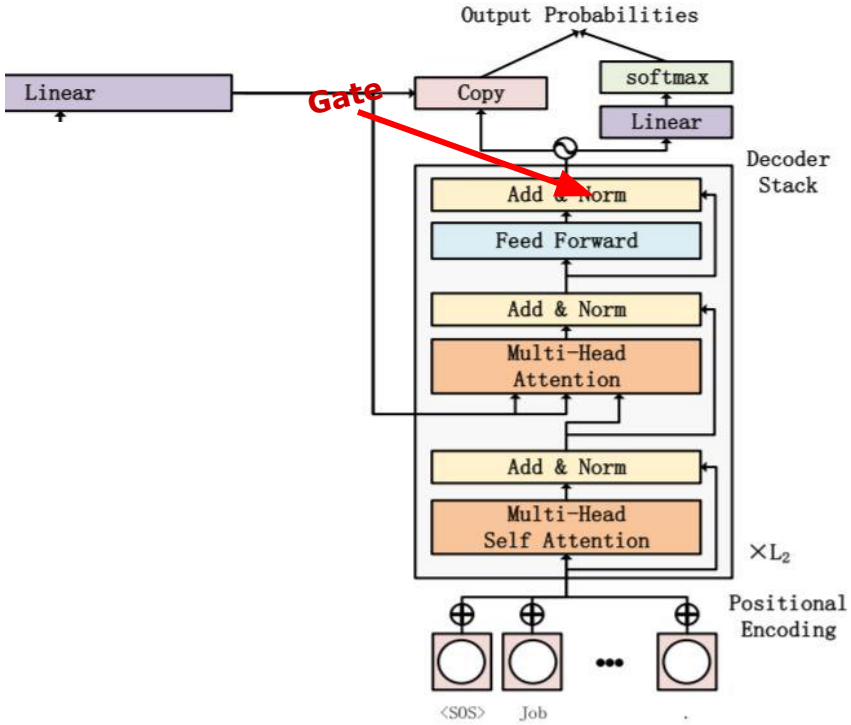
Encoder stack



- Fusion combines incoming and outgoing graph attns
- Final layer is to compress dimension

Decoder stack

- Copy mechanism
  - Gate decides vocab word or copy from AMR
- Otherwise, same as transformer



Results

- Corpora
  - LDC2015E86
  - LDC2017T10
  - “Silver” data
- **Outperformed methods:**
  - Vanilla Transformer
    - Linearized input
  - Seq2seq
  - Graph2seq
  - Hybrid

LDC2015E86 + silver data

Methods	BLEU
PBMT (Pourdamghani et al., 2016)	26.9
Tree2Str (Flanigan et al., 2016)	23.0
TSP (Song et al., 2016)	22.4
S2S+Anon (Konstas et al., 2017)	22.0
GraphLSTM (Song et al., 2018)	23.3
t-GCNSEQ (Damonte and Cohen, 2019)	23.9
g-GCNSEQ (Damonte and Cohen, 2019)	24.4
Transformer	17.7
Graph Transformer	25.9
S2S+Anon (2M) (Konstas et al., 2017)	32.3
S2S+Anon (20M) (Konstas et al., 2017)	33.8
S2S+Copy (2M) (Song et al., 2018)	31.7
GraphLSTM (2M) (Song et al., 2018)	33.6
Transformer (2M)	35.1
Graph Transformer (2M)	**36.4**

Table 1: Test results of models. ‘‘(2M)’’ / ‘‘(20M)’’ denotes using the corresponding number of automatically labeled Gigaword data instances as additional training data.

Ablation Study

- Remove dual representation in graph encoder
  - Loss in BLEU score
- Inseparate Graph Attentions
  - Loss in BLEU Score

1. **Relations between outgoing and incoming edges differ**
2. **Representing them with separate graph attns is helpful**

Performance wrt Size & Structure of AMR

- Graph transformer compared to seq2seq and GraphLSTM
  - Outperforms wrt Depth
  - Outperforms wrt # edges/relations
  - Outperforms wrt # reentrancies

Model	Depth		
	1–5	6–10	11–
#count	278	828	265
S2S	37.2	21.2	19.3
GraphLSTM	+3.9	+2.1	+0.7
Graph Transformer	**+6.0**	**+4.5**	**+3.7**

Model	Number of edges		
	1–10	11–20	21–
#count	425	452	494
S2S	30.8	21.4	19.7
GraphLSTM	+4.3	+2.9	+0.8
Graph Transformer	**+6.2**	**+4.6**	**+3.8**

Model	Number of reentrancies		
	0	1–3	4–
#count	624	566	181
S2S	26.6	21.2	15.3
GraphLSTM	+2.9	+2.4	+0.1
Graph Transformer	**+7.1**	**+4.0**	**+2.9**

Paper 4:

Investigating PLMs for Graph-to-Text Generation

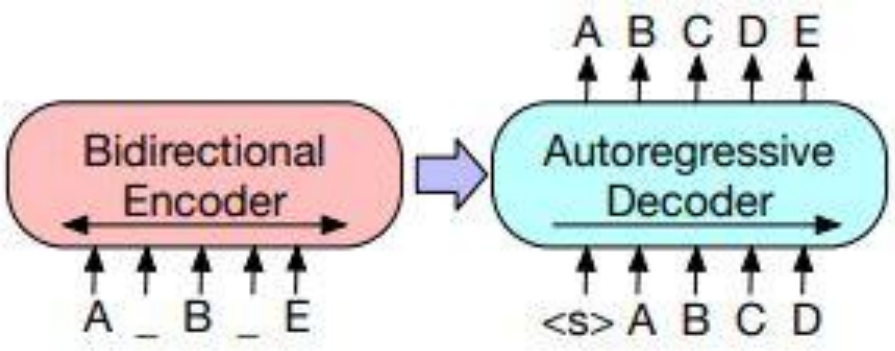
Leonardo Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych, arXiv 2020

Motivation + PLMs Used

- PLMs achieved SOTA in other data-to-text problems
- Exploit performance of large PLMs
  - BART
  - T5

BART (Facebook)

- Encoder-decoder Transformer architecture
- Text-to-text denoising autoencoder
  - Reconstruct original text
- ReLUs are replaced by GeLUs
- Corpora
  - Books + Wikipedia
- Like BERT + GPT
  - Encoder + masking
  - L2R Decoder



"BERT" "GPT"

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, https://arxiv.org/pdf/1910.13461.pdf

T5 (Google)

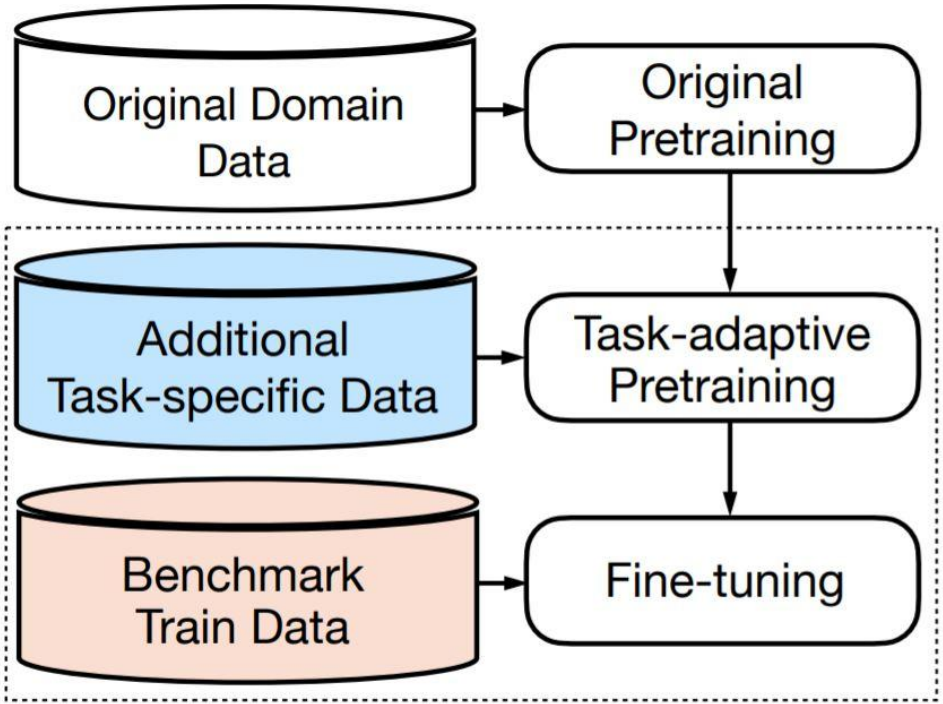
- Encoder-decoder Transformer architecture
- Positional embedding difference
  - Not fixed embedding, just a scalar
- Converts different problems into text-to-text format
- Corpora
  - Common Crawls
- Prefix for task

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, https://arxiv.org/abs/1910.10683

https://ai.googleblohttps://arxiv.org/abs/1910.10683g.com/2020/02/exploring-transfer-learning-with-t5.html

Fine-tuning & Pretraining

- Fine tuning:
  - Original training set
  - LDC2017T10
- Pretraining:
  - Supervised: graph + text pairs
  - Unsupervised: input corrupted AMR sentences, reconstruct input
- T5 pretraining:
  - “translate from Graph to Text:”
- “Silver” data
  - Gigaword for more pretraining data



Use of Graph Structure by PLMs

- Removing parentheses reduced BLEU by 6.35 points
  - PLM uses graph structure
  - Not explicitly encoded in model

AMR17			
Model	**BLEU**	**METEOR**	**chrF++**
BART	43.72	41.27	71.27
BART-TSU	43.94	41.15	71.14
BART-TSS	45.54	42.50	72.78
T5	43.18	40.10	66.79
T5-TSU	45.91	41.82	71.19
T5-TSS	**48.85**	**43.12**	**73.10**

AMR17				
Train	**Eval**	**BLEU**	**METEOR**	**chrF++**
order	order	36.83	38.17	65.57
shuf	order	17.02	27.79	51.00
order	shuf	05.18	19.63	39.97
shuf	shuf	15.56	27.23	50.20

Summary & contributions

- Demonstrated that powerful PLMs are adept at handling AMR-to-text
  - Even with complex directed graph structure
- Achieves SOTA AMR-to-text generation
- Opens possibility of further improving generation with more involved use of PLMs

Future work

- Multisentential AMR
- Incorporation of lexical semantics
- Gen:
  - Inject graph structure into PLM more elegantly
- Parsing:
  - Cross-lingual approaches

Discussion Questions 1-5

1. What are the limitations of AMR as a representation and how would these affect parsing and generation?

2. What other NLP tasks could take advantage of using AMR? How might the task incorporate this semantic representation into a model?

3. What other semantic representations do you know about? How can they be compared with AMR?

4. Text generation is associated with a lot of choices (for content, word choice, word ordering, etc). AMR representations are notoriously underspecified. Why is this the case? What techniques do you propose for adding the missing information during generation?

5. The previous topic was non-autoregressive machine translation. How would you design an AMR-to-text algorithm that generates words in various orders, not just left to right?