Semantic Parsing

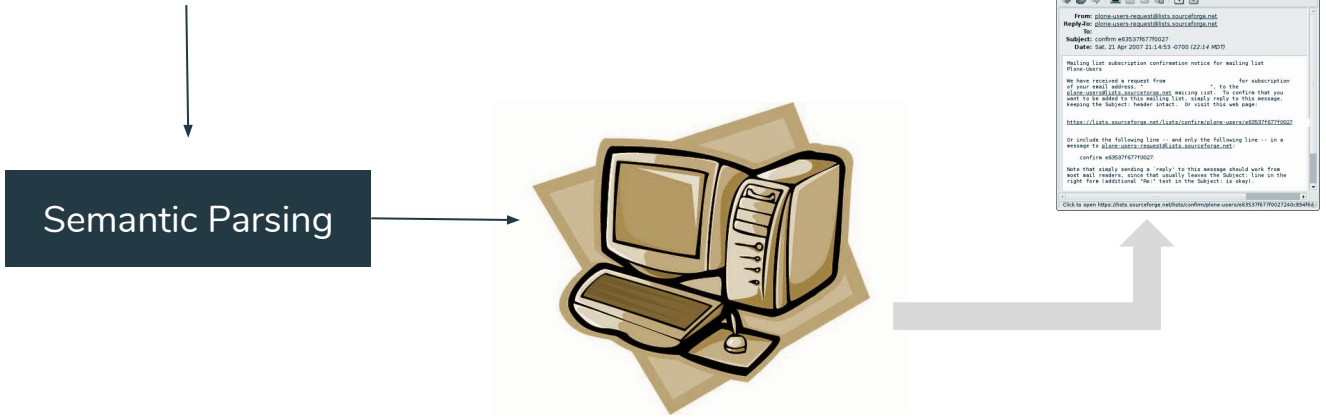
Overview (Part II)

Rexhina Blloshmi
SapienzaNLP Reading Group
18 March 2020

Previously on Semantic Parsing...

Deeper Semantic Analysis

Show the long email Alice sent me yesterday.

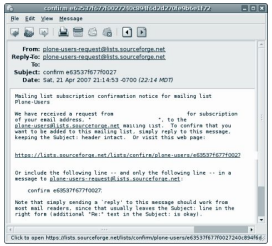


Deeper Semantic Analysis

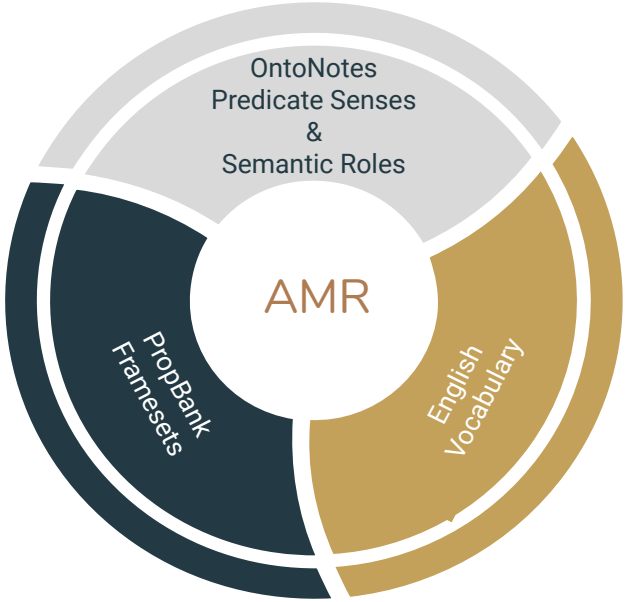
1 - Critical to get the exact output.
2 - Strict evaluation.
3 - No partial credit.

Show the long email Alice sent me yesterday.

Semantic Parsing



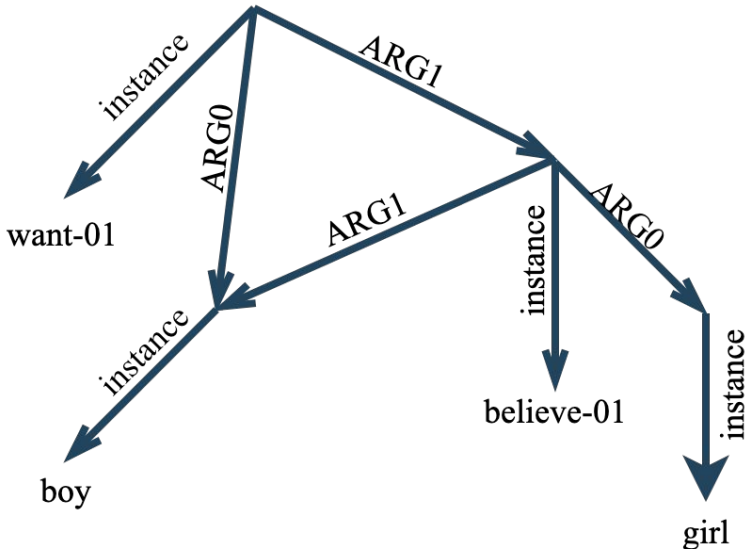
Semantic Representations: AMR



Semantic Representations: AMR Example

The **boy wants** the **girl** to **believe** him.

```
(w / want-01
   :ARG0 (b / boy)
   :ARG1 (b2 / believe-01
            :ARG0 (g / girl)
            :ARG1 b))
```

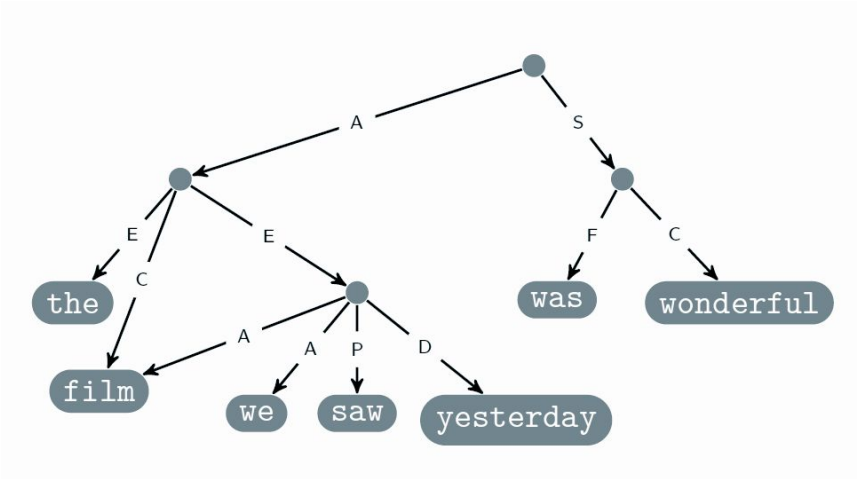


Semantic Representations: UCCA

Scene Elements	Elements of Non-Scene Units	Inter-Scene Relation	Other
Process (P)	Center (C)	Parallel Scene (H)	Function (F)
State (S)	Elaborator (E)	Linker (L)	
Participant(A)	Connector (N)	Ground (G)	
Adverbial (D)	Relator (R)		

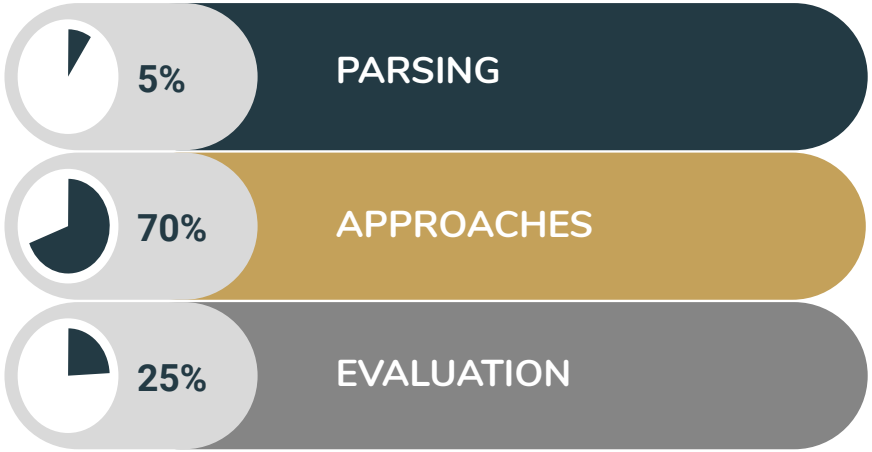
Semantic Representations: UCCA Example

The film we saw yesterday was wonderful.



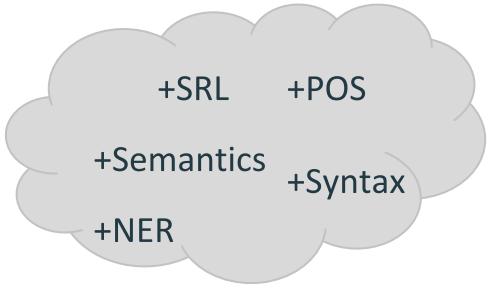
On this presentation ...

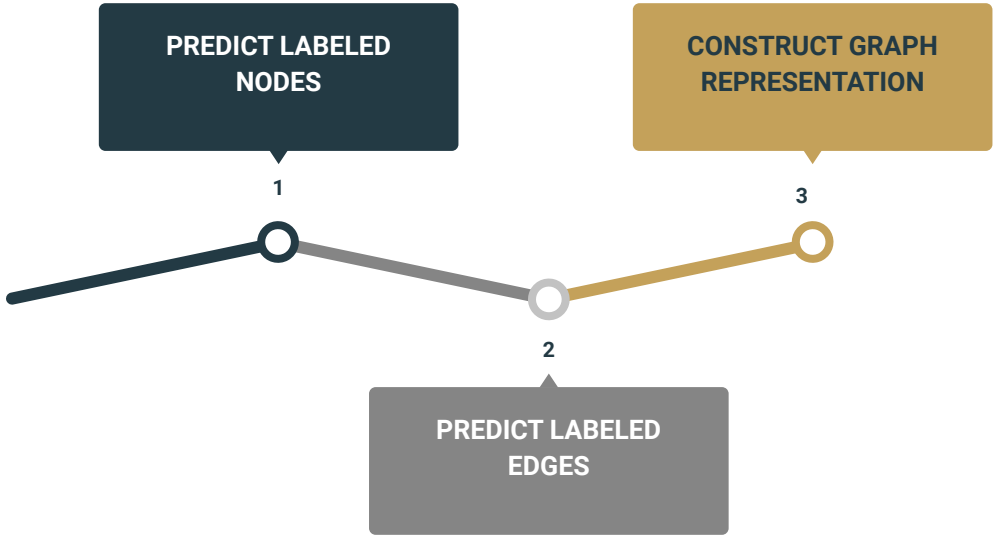
Outline



Parsing

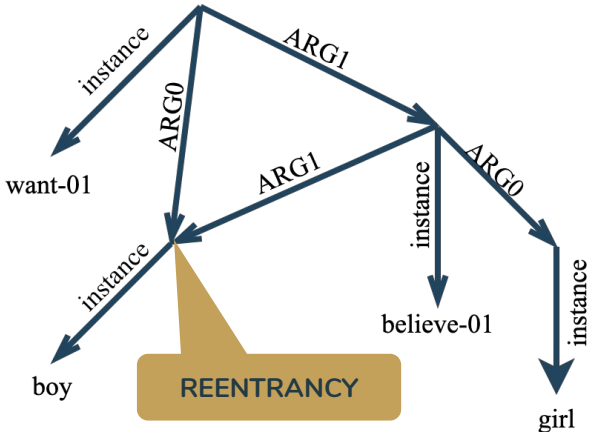
The **boy wants** the **girl** to **believe** him.



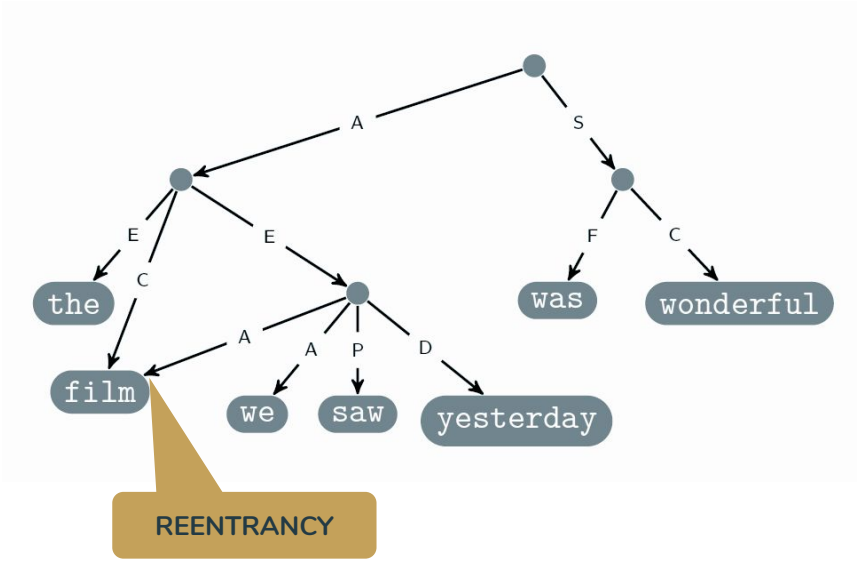


Semantic-Graph Representations

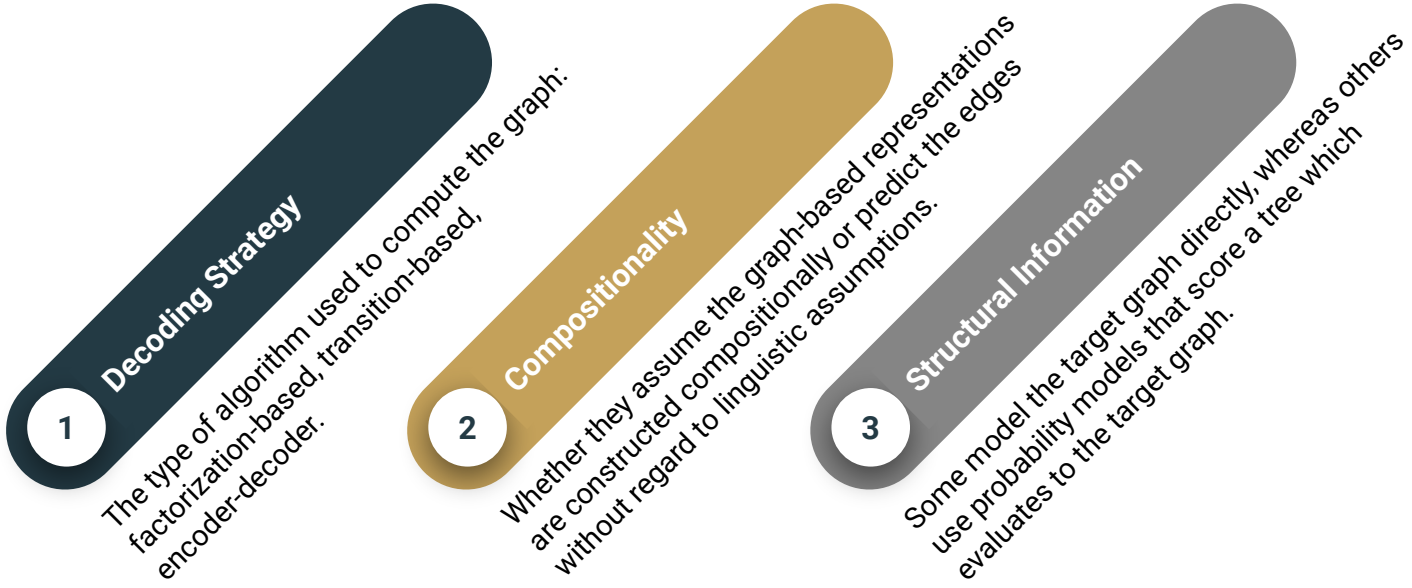
The boy wants the girl to believe him.



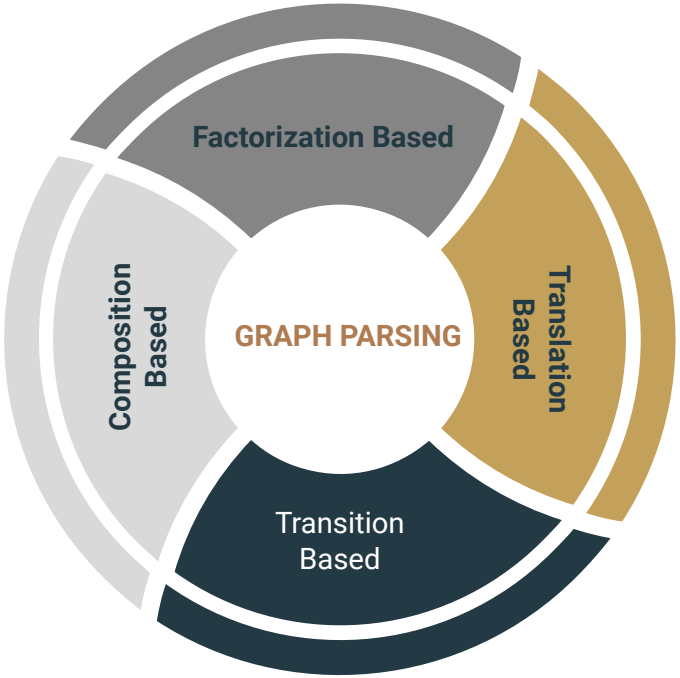
The film we saw yesterday was wonderful.



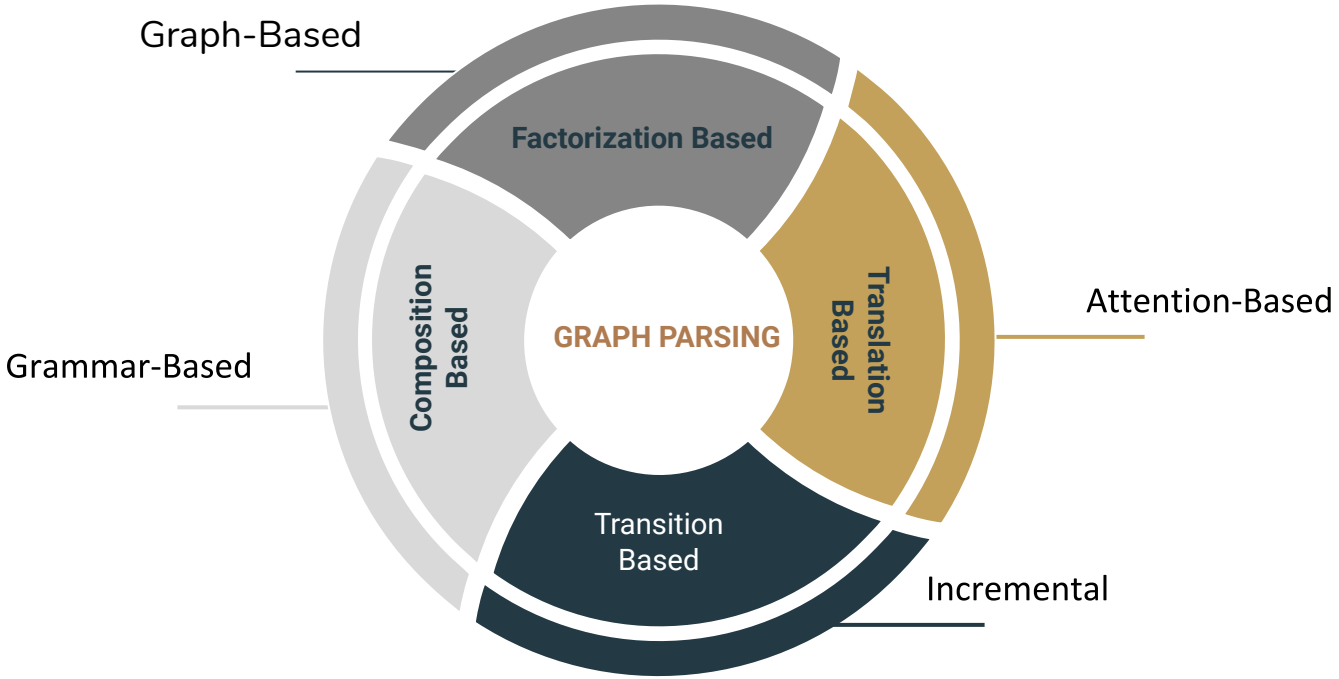
Semantic Parsers Key Differences



Graph Parsing Approaches



Graph Parsing Approaches



Factorization-Based

Factorization-Based

Characteristics

- Explicitly models the target semantic graph structures.
- Searching the highest-scoring graphs from a large set.
    - Searching for a subset $E^* \subseteq E$ with the maximum score.
- Identifies the relations by searching for the maximum spanning connected subgraph from an edge-labeled, directed graph representing all possible relations between the identified concepts.

Factorization-Based

Sub-tasks

Word Alignment Problem

1. Heuristic rules.
2. Linearize graphs and reuse word alignment tools (NMT).
3. Consider all possible alignments and let the parser choose the best.
4. Used to train Concept Identification.

Factorization-Based

Sub-tasks

Word Alignment Problem

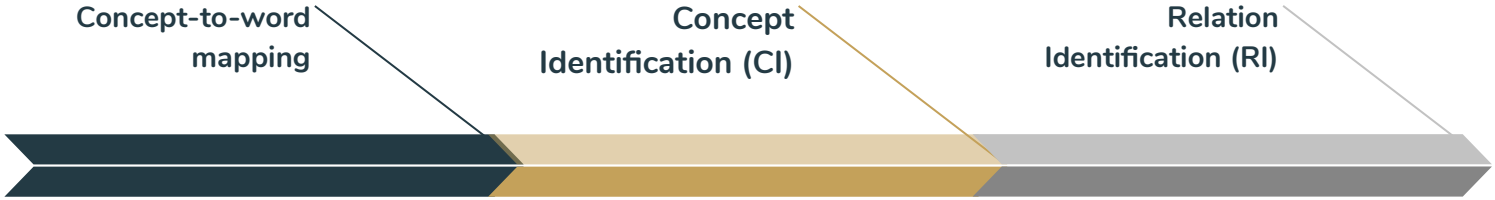
1. Heuristic rules.
2. Linearize graphs and reuse word alignment tools (NMT).
3. Consider all possible alignments and let the parser choose the best.
4. Used to train Concept Identification.

Sequence Labelling Problem

1. Some nodes are linked to sub-words.
2. Some nodes are linked to multiple words.

Factorization-Based

Sub-tasks



Word Alignment Problem

1. Heuristic rules.
2. Linearize graphs and reuse word alignment tools (NMT).
3. Consider all possible alignments and let the parser choose the best.
4. Used to train Concept Identification.

Sequence Labelling Problem

1. Some nodes are linked to sub-words.
2. Some nodes are linked to multiple words.

Linking Nodes Problem

1. Search for a subgraph $G_0 = (V, E^* \subseteq E)$ that maximizes the score function:

Factorization-Based

AMR Parsers

ACL, 2014	ACL, 2017	ACL, 2018	ACL, 2019
Flanigan et al., JAMR **Focus: 1st AMR parser**	**Foland & Martin** **Focus: BiLSTM parser**	**Lyu & Titov** **Focus: Latent Alignment**	**Zhang et al.** **Focus: Reentrencies**
1 - A set of rules to greedily align concepts to spans (prior to parsing). 2 - CI using a dynamic programming algorithm (semi-Markov model). 3 - RI finding the Maximum Preserving, Simple, Spanning, Connected Subgraph (Kruskal's algorithm).	1 - JAMR alignments enhanced to identify a subgraph type to associate with each span. 2 - A BiLSTM layer to identify concepts or subgraphs. 3 - 4 BiLSTM layers to identify relations and attributes.	1 - Alignments as latent variable in a joint model with parsing. 2 - CI as predictions based on RNN states. 3 - RI as predictions of edges based on concepts and aligned words representations.	1- Aligner free. 2 - CI by using an encoder-decoder architecture to predict nodes sequentially. 3 - RI searching for the highest scoring parse tree given a node list, their indices and valid edge scores.

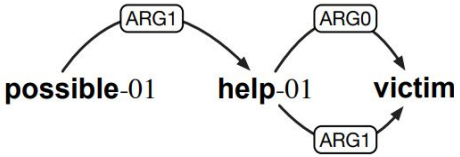
AMR Parsing as Sequence-to-Graph Transduction

[Zhang et al., ACL 2019]

Sequence-to-Graph	An attention-based model that treats AMR parsing as sequence-to-graph transduction.
Main focus	Handling Reentrancy explicitly. Alignment-free approach, supported by an extended pointer-generator network.
Parsing	Applied to AMR framework and efficiently achieve SOTA without data augmentation.

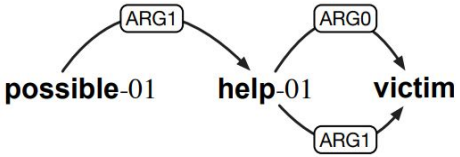
Another view of Reentrancy

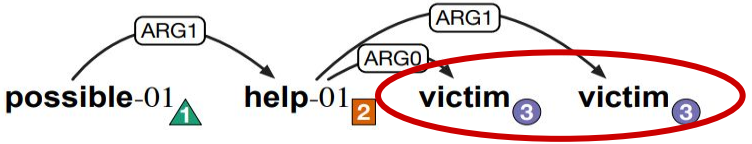
The victim could help himself.



Another view of Reentrancy

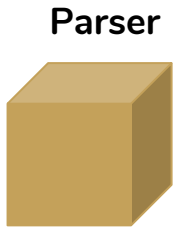
The victim could help himself.

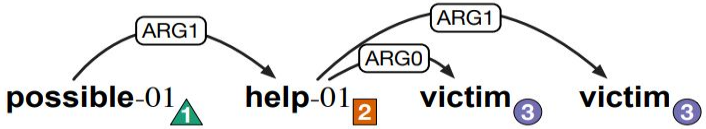




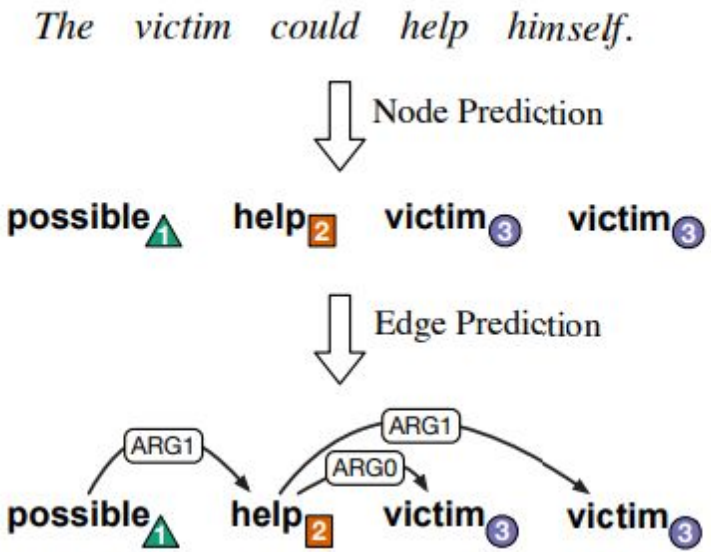
Sequence-to-Tree Transduction

The victim could help himself.

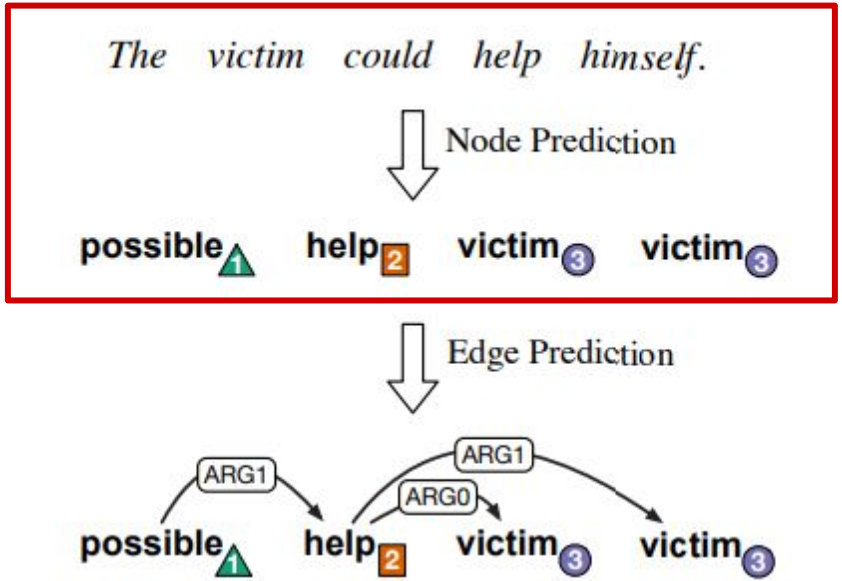




Two-stage AMR parsing

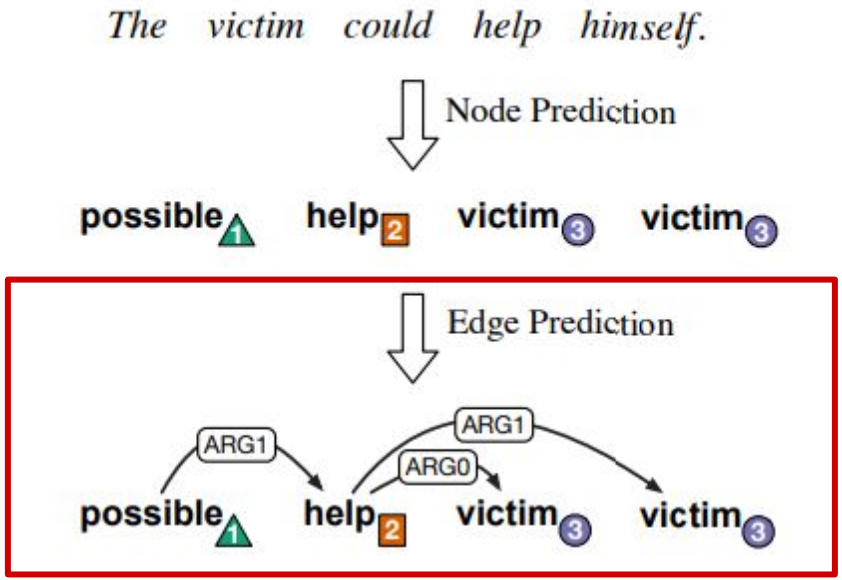


Two-stage AMR parsing



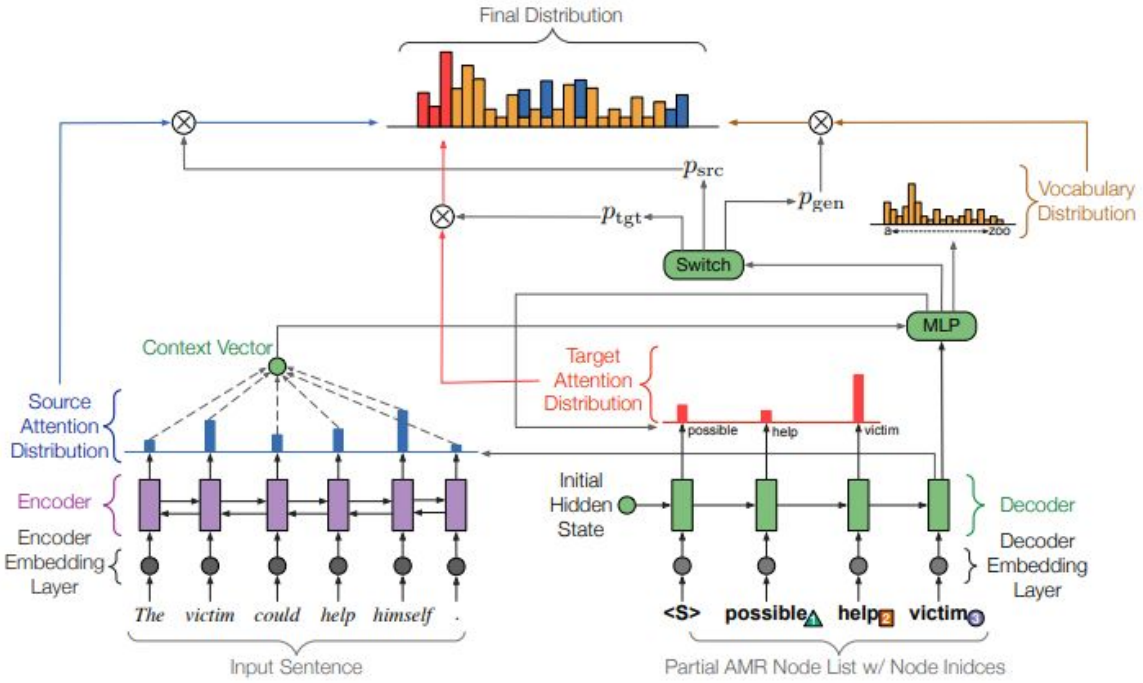
Extended Pointer-Generator Networks

Two-stage AMR parsing



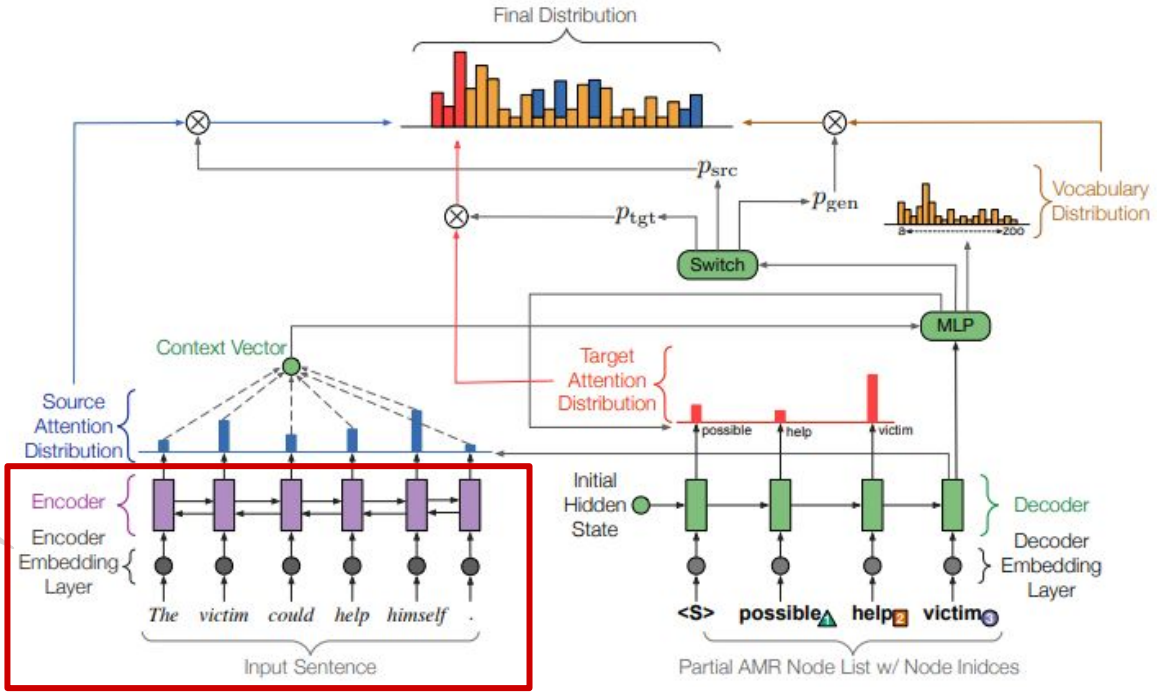
Deep Biaffine Classifier

Model for Node Prediction



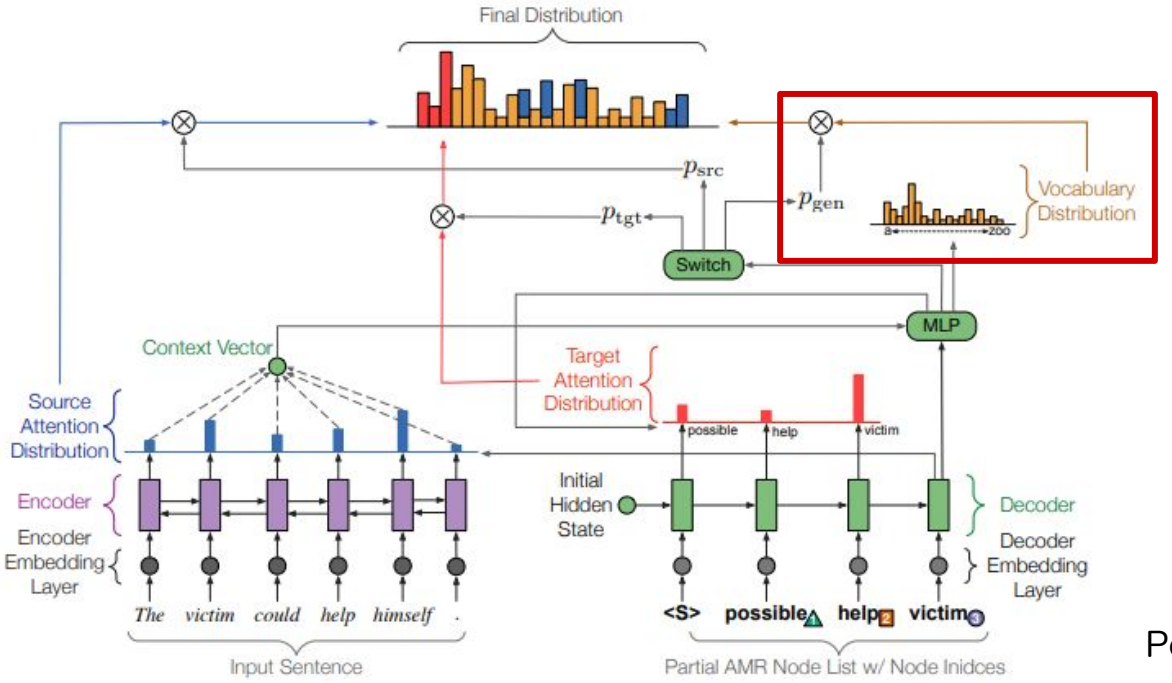
Model for Node Prediction

Embedding:
- BERT
- GloVe
- POS
- CharNN



BiLSTM Encoder

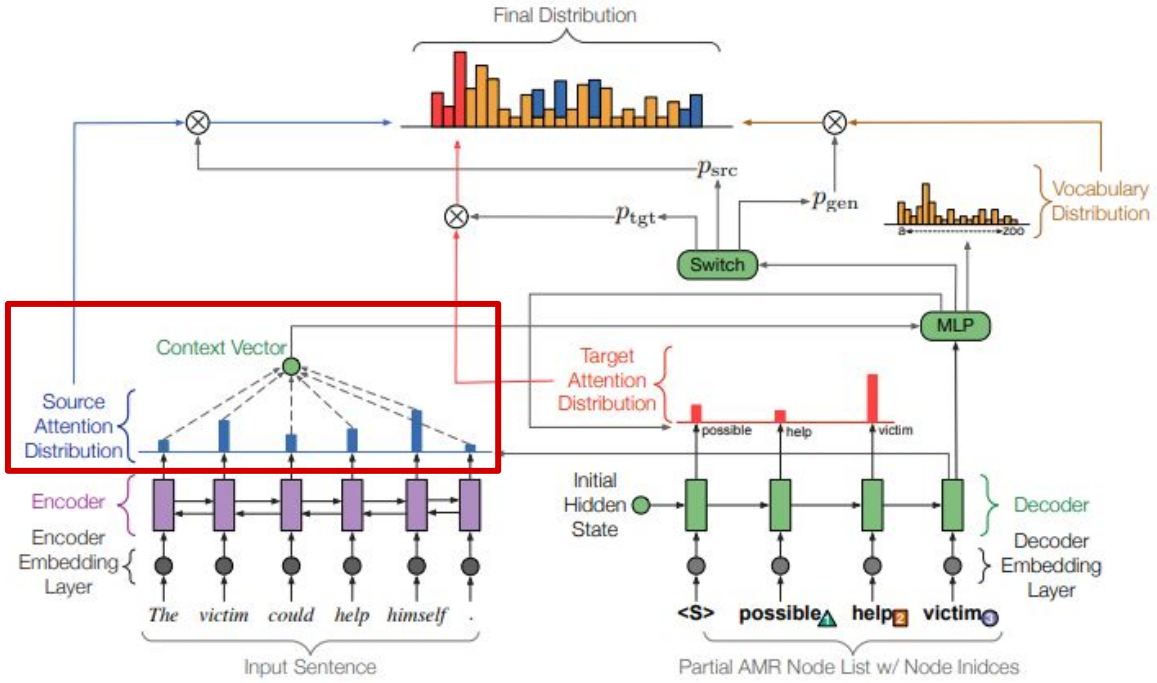
Model for Node Prediction



Decoder:
1 - Generation

Pointer-Generator Decoder

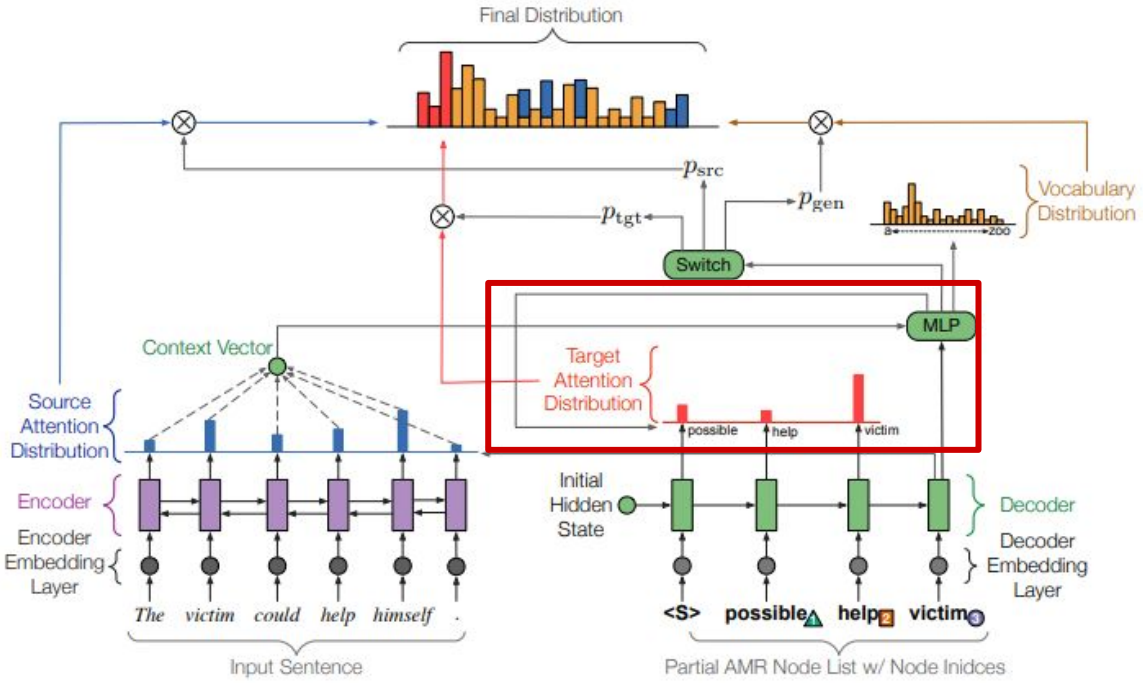
Model for Node Prediction



Decoder:
1 - Generation
2 - Source-Copy attention

Pointer-Generator Decoder

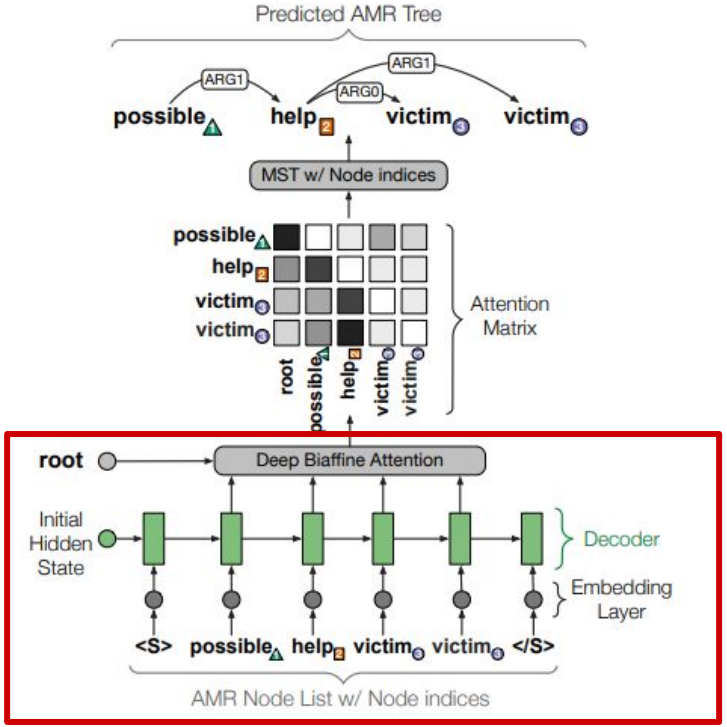
Model for Node Prediction



Decoder:
1 - Generation
2 - Source-Copy attention
3 - Target-Copy attention

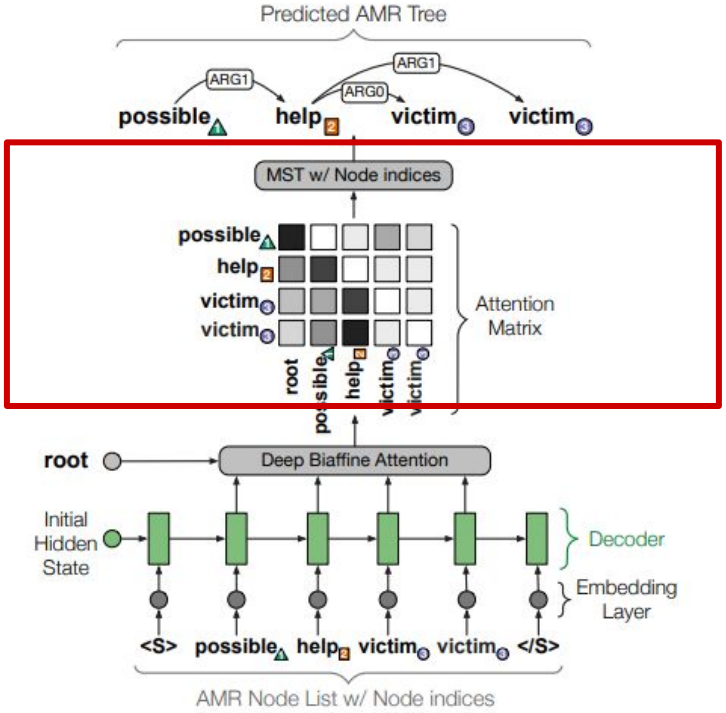
Pointer-Generator Decoder

Model for Edge Prediction



Reuse decoder states of pointer-generator network.

Model for Edge Prediction



1 - Score edges between nodes.
2 - Look for the highest scoring parse tree in the space of valid parse trees using a greedy algorithm, i.e. Chu-Liu-Edmonds.

Recap steps

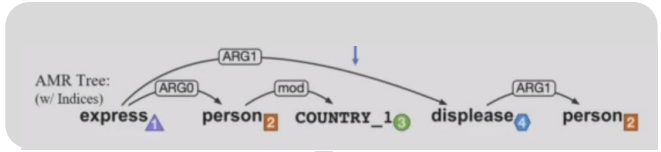
Chinese people expressed their displeasure.

Lemmatization & Anonymization
COUNTRY_1 people express they displeasure .

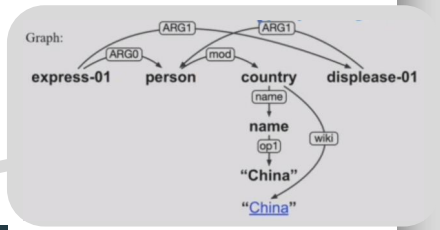
Preprocessing

express 1 person 2 COUNTRY_1 3 displease 4 person 2

Node Prediction



Edge Prediction



Post-processing

- Merge duplicate Nodes
- Process Word Senses
- Entities & Wikification

Transition-Based

Transition-Based

Characteristics

A **transition system** is an abstract machine characterized by a set of configurations and transitions between them.

- A configuration is composed of a stack of partially processed words and a buffer of unseen input words.
- Starting from an initial configuration, the system applies transitions until a terminal configuration is reached.

Transition-Based

Tree-to-Graph

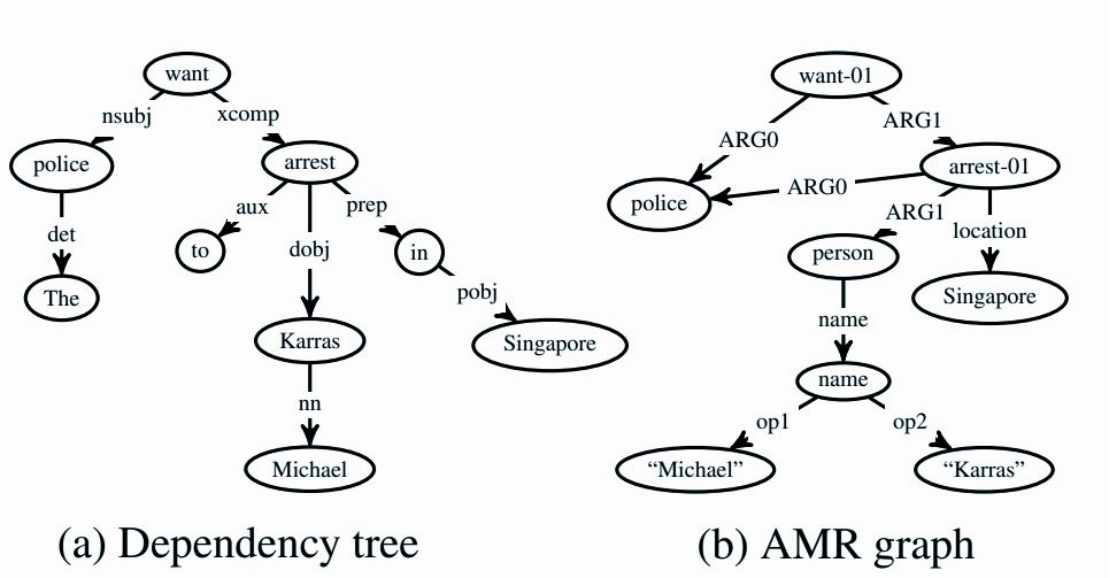
Parse a sentence into an AMR by taking the dependency tree of that sentence as input and transforming it to an AMR representation via a series of actions.

- The sentence is parsed into a dependency tree with a dependency parser.
- The dependency tree is transformed into an AMR graph.

From Dependency to AMR Parsing

Tree-to-Graph: Similarities

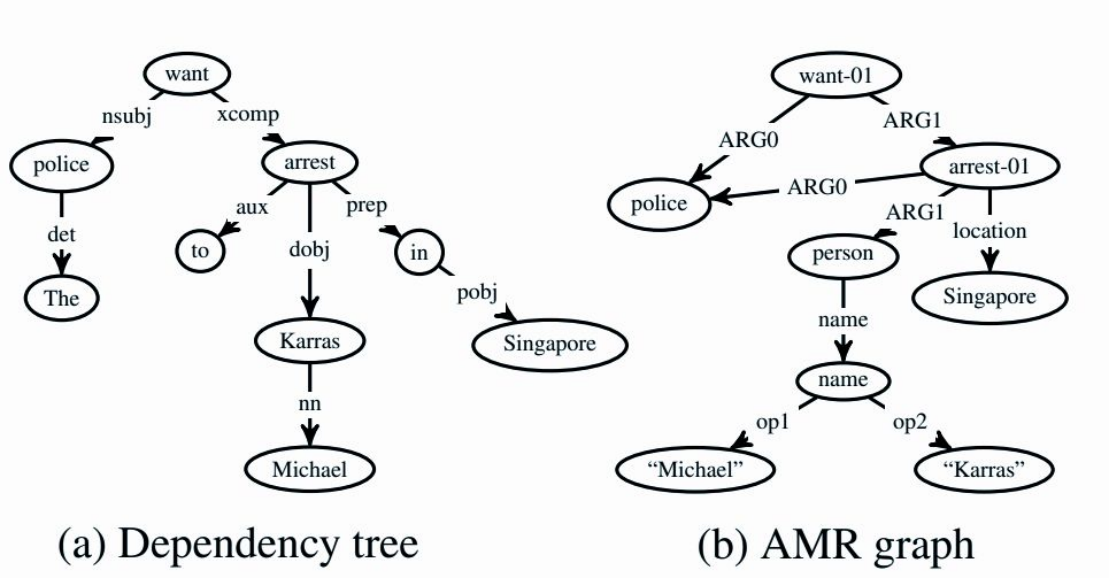
- Both describe relations as holding between a head and its dependent, or between a parent and its child.



From Dependency to AMR Parsing

Tree-to-Graph: Differences

- AMR concepts and relations abstract away from actual word tokens.
- Content words become concepts while function words either become relations or get omitted.
- Reentrancy is also used to represent co-reference.



Transition-Based

Sequence-to-Graph

Transition-based systems configured to directly parse a sequence into a graph by taking into consideration key features of AMR.

- Non-projectivity of AMR.
- Reentrancy.
- Word-to-concept alignment.

Transition-Based

AMR Parsers

NAACL, 2015	ACL, 2017	EMNLP, 2018	EMNLP, 2019
Wang et al., CAMR **Focus: Tree-to-Graph Transition** 1 - Compute the dependency tree using a pretrained parser. 2 - Collapse relations and concepts in one node to jointly predict them. 3 - JAMR aligner. 4 - Transition-based alg. initialized with a post-order traversal of dependency tree.	**Damonte et al., AMREager** **Focus: Seq-to-Graph Transition** 1 - Left-to-Right, linear-time transition system. 2 - CI at each Shift step of the Transition system. 3 - A classifier for the next transition. 4 - A classifier for edge label on each LArc and RArc step.	**Liu et al.** **Focus: Tunable Alignment** 1 - Enhanced Rule-Based JAMR aligner with rich semantic resources, GloVE & WordNet for multiple alignments. 2 - Parser chooses among multiple alignments. 3 - Sentence-to-Graph transition system for parsing along with its oracle parser.	**Zhang et al.** **Focus: One-stage Parsing** 1 - Directly transform a text into a MR in one stage. 2 - Incrementally builds a MR via a sequence of semantic relations. 3 - Aligner-free. 4 - Applied to three separate broad-coverage semantic parsing tasks – AMR, SDP & UCCA.

Example Transition Actions*

Transition	Current State	Resulting State	Description
DROP	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, \beta, A]$	pops out the word that doesn't convey any semantics (e.g., function words and punctuations).
MERGE	$[\sigma \mid s_0, \delta, b_0 \mid b_1 \mid \beta, A]$	$[\sigma \mid s_0, \delta, b_0\_b_1 \mid \beta, A]$	concatenates a sequence of words into a span, which can be derived as a named entity (`name`) or `date-entity`.
CONFIRM(c)	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, c \mid \beta, A]$	derives the first element of the buffer (a word or span) into a concept `c`.
ENTITY(c)	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, c \mid \beta, A \cup \text{relations}(c)]$	a special form of CONFIRM that derives the first element into an entity and builds the internal entity AMR fragment.
NEW(c)	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, c \mid b_0 \mid \beta, A]$	generates a new concept `c` and pushes it to the front of the buffer.
LEFT(r)	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A \cup \{s_0 \xleftarrow{r} b_0\}]$	links a relation `r` between the top concepts on the stack and the buffer.
RIGHT(r)	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A \cup \{s_0 \xrightarrow{r} b_0\}]$	
CACHE	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma, s_0 \mid \delta, b_0 \mid \beta, A]$	passes the top concept of the stack onto the deque.
SHIFT	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma \mid s_0 \mid \delta \mid b_0, [\,], \beta, A]$	shifts the first concept of the buffer onto the stack along with those on the deque.
REDUCE	$[\sigma \mid s_0, \delta, b_0 \mid \beta, A]$	$[\sigma, \delta, b_0 \mid \beta, A]$	pops the top concept of the stack.

s = (σ, δ, β, A)
- **σ** is a stack holding processed words
- **δ** is a deque holding words popped out of σ that will be pushed back in the future
- **β** is a buffer holding unprocessed words.
- **A** is a set of labeled relations.

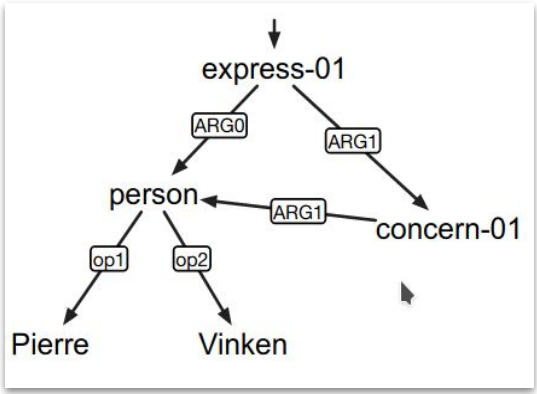
*Liu et al, EMNLP 2018.

Broad-Coverage Semantic Parsing as Transduction

[Zhang et al., EMNLP 2019]

Unified Transduction Framework	Attention-based neural transducer to directly transform a text into a meaning representation (MR) in one stage.
Hybrid Graph- & Transition-based	Builds a MR incrementally via a sequence of semantic relations. Leverages multiple attention mechanisms used in graph-parsers.
Parsing	Requires only minimal task-specific processings to be applicable to three separate broad-coverage semantic tasks, i.e., AMR, UCCA, SDP.

Unified Arborescence Format: AMR



Remove Reentrancy

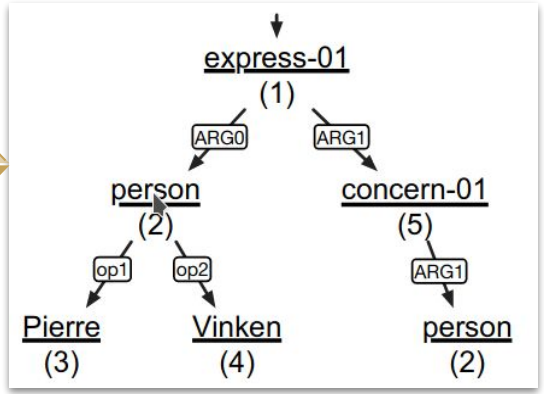
Preserve Reentrancy Information

Duplicate Nodes

Whenever a node has a reentrant relation, make a copy of that node and use the copy to participate in the relation.

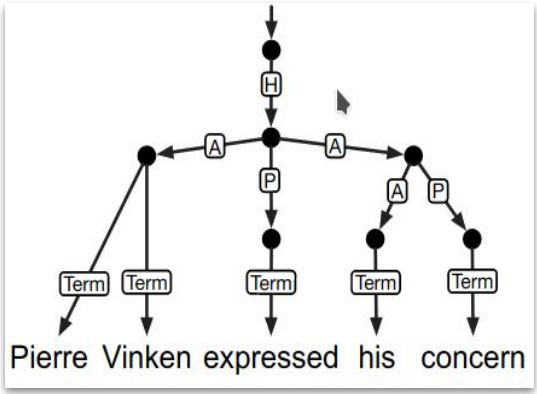
Index Nodes

Assign a node index to each node and assign the same index as the original to the duplicated copy.



Pierre Vinken expressed his concern.

Unified Arborescence Format: UCCA



Reduce terminal nodes

Add labels to nodes

Index Nodes

Collapsing subgraphs of terminal nodes

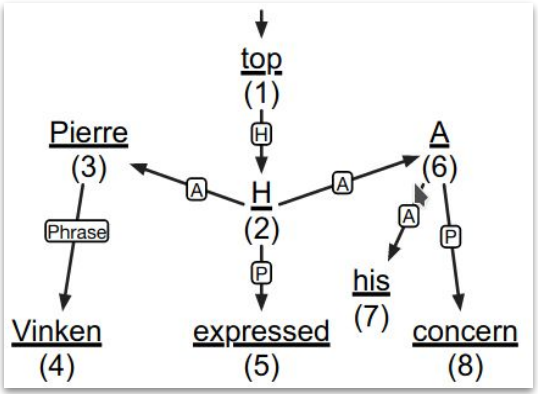
1) Replace each pre-terminal node with its terminal node.
2) Add *phrase* edge if multiple terminals.

Label non-terminal nodes

Each node label is simply the same as its incoming edge label.

Node indices

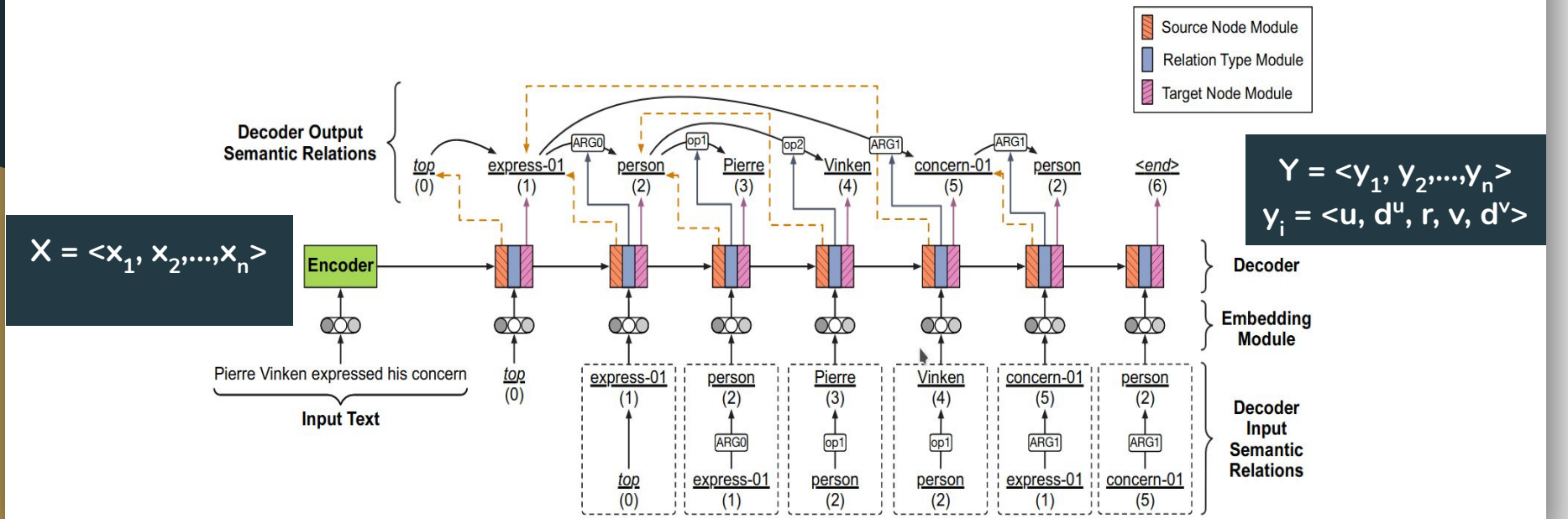
Similar to AMR case.



Pierre Vinken expressed his concern.

Model

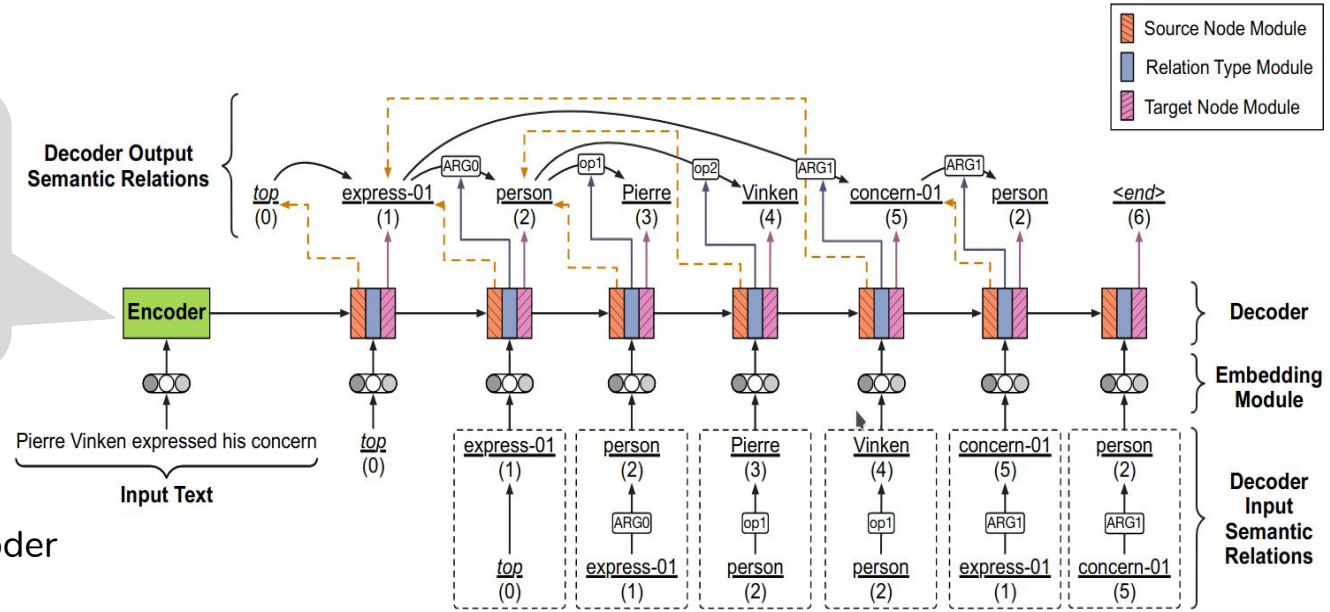
u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> target node index



Model

u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> target node index

Embedding:
- BERT
- GloVe
- POS
- CharNN



BiLSTM Encoder

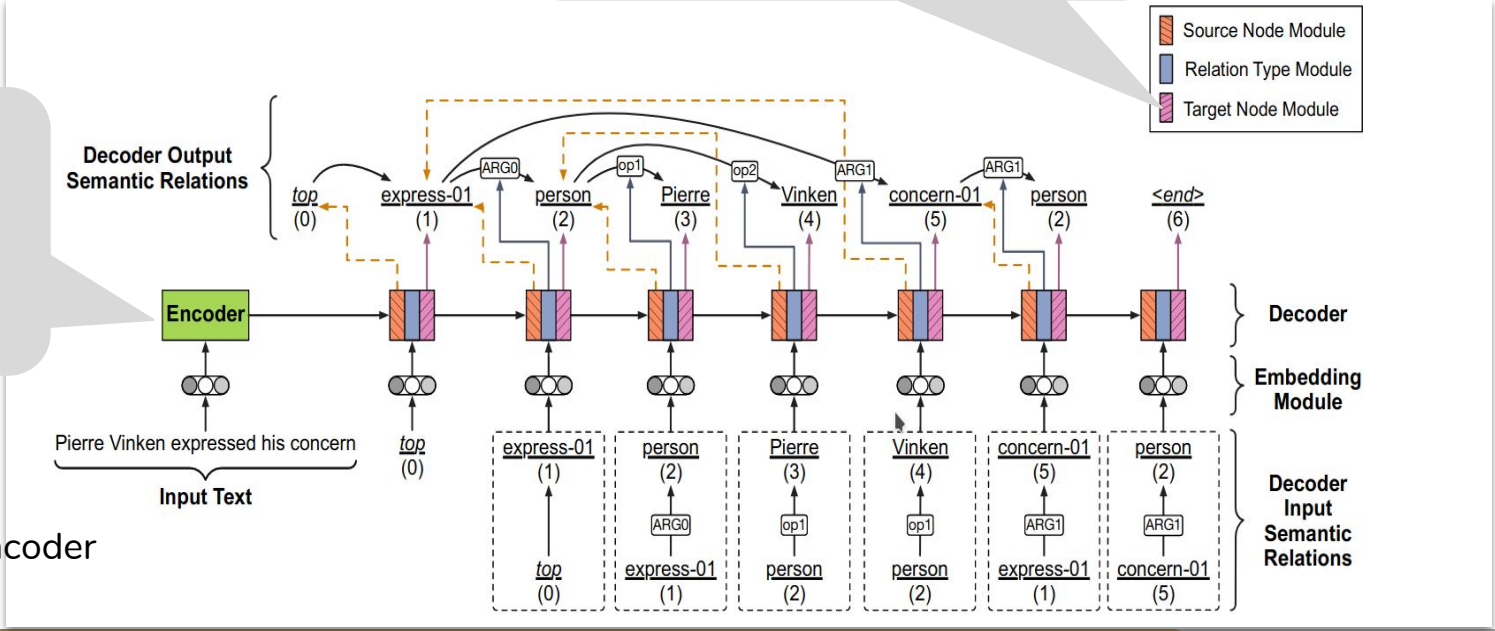
Model

u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> target node index

Takes the vector representations of the previous semantic relation, and predicts a target node label as well as its index.

$$\mathbf{z}_i = \mathrm{FFN}^{(\mathrm{relation})}([\mathbf{h}_i^l; \mathbf{c}_i; \mathbf{r}_i; \mathbf{u}_i; \mathbf{d}_i^u])$$
$$\mathbf{h}_i^l = \mathrm{LSTM}_1(\mathbf{h}_i^{l-1}, \mathbf{h}_{i-1}^l)$$
$$\mathrm{FFN}(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Embedding:
- BERT
- GloVe
- POS
- CharNN



BiLSTM Encoder

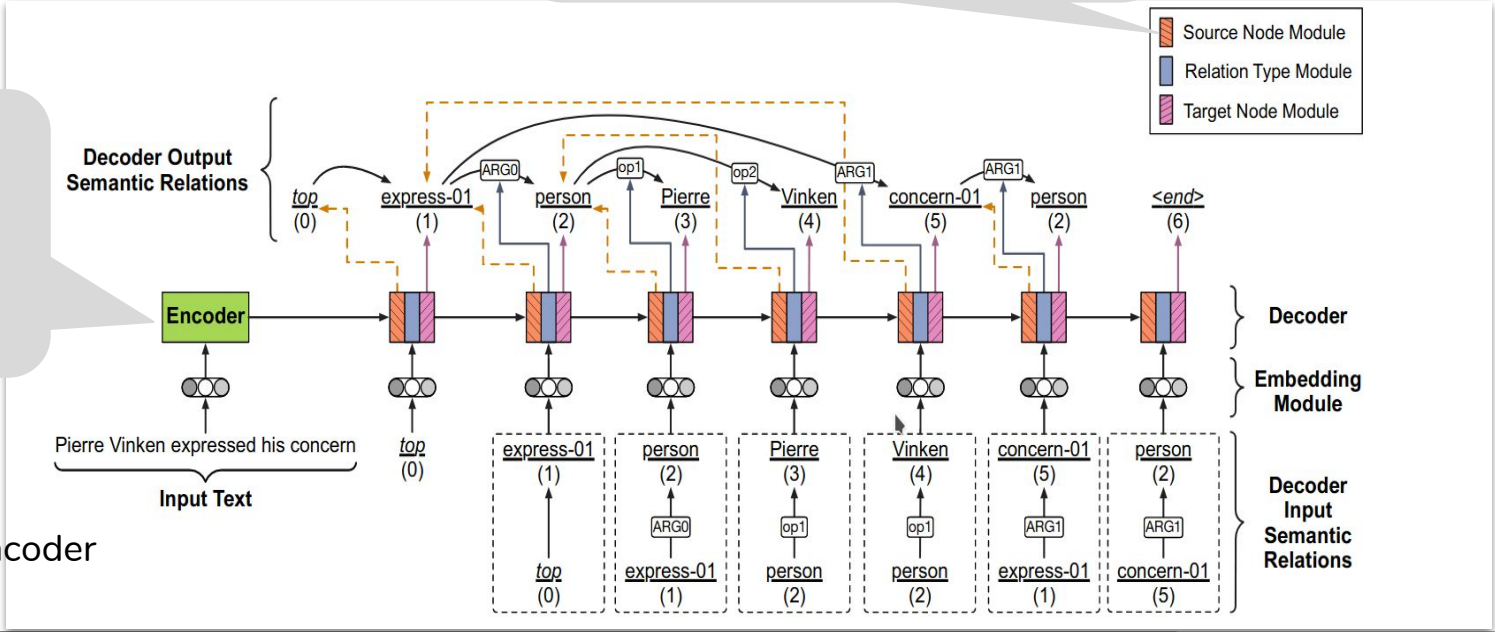
Model

u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> target node index

Predicts a source node via pointing to a node label among preceding target node labels.

$$\text{softmax}\left(\text{BIAFFINE}\left(\mathbf{h}_{i+1}^{(\text{start})}, \mathbf{h}_{1:i}^{(\text{end})}\right)\right)$$

Embedding:
- BERT
- GloVe
- POS
- CharNN



BiLSTM Encoder

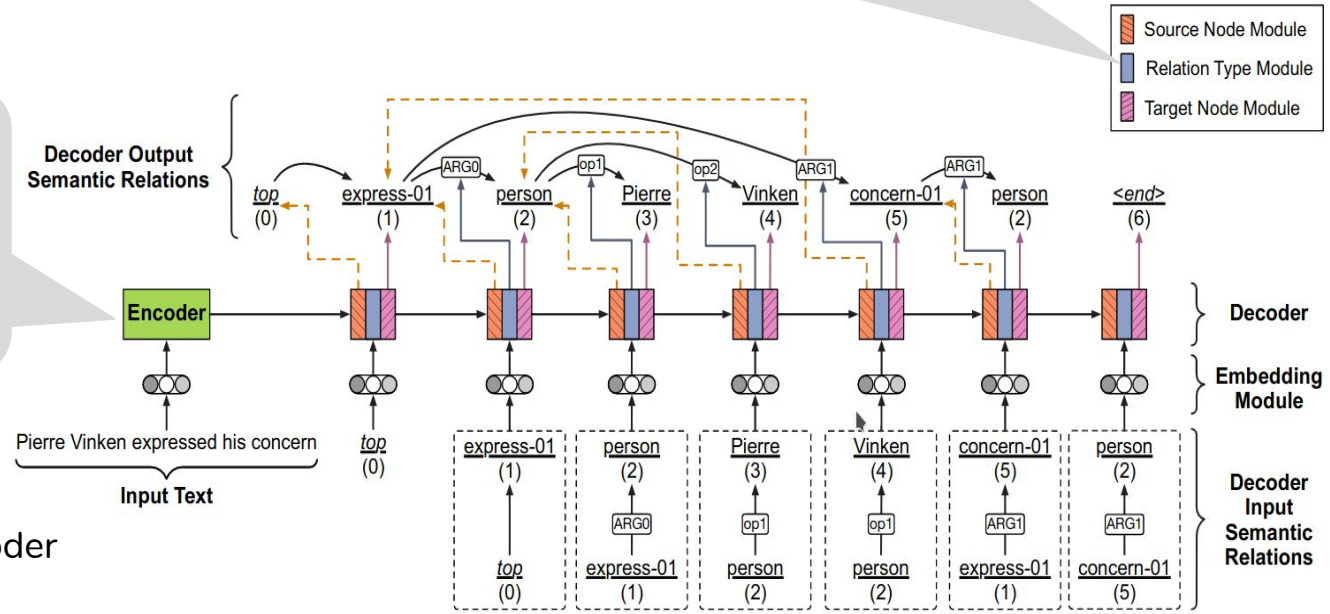
Model

u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> target node index

Reuses LSTM hidden states from the target node module to compute the probability distribution of next relation type.

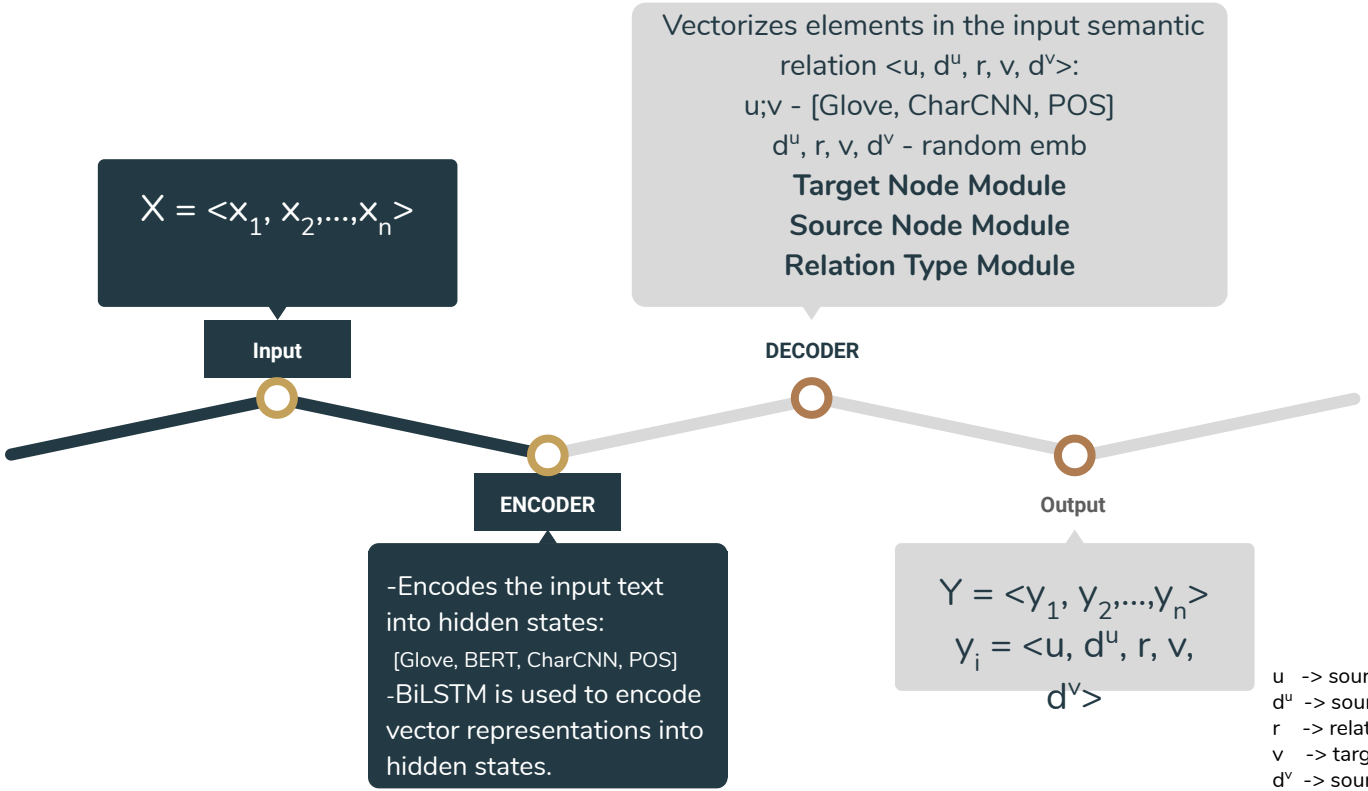
$$\text{softmax}\left(\text{BILINEAR}\left(\mathbf{h}_{i+1}^{(\text{rel-src})}, \mathbf{h}_{i+1}^{(\text{rel-tgt})}\right)\right)$$

Embedding:
- BERT
- GloVe
- POS
- CharNN



BiLSTM Encoder

Recap Steps



u -> source node label
d^u -> source node index
r -> relation type
v -> target node label
d^v -> source node index

Translation-Based

Translation-Based

Characteristics

- Inspired by the success of sequence-to-sequence models that are the heart of modern Neural Machine Translation.
- Semantic graphs as a foreign language.
    - Semantic graphs are encoded and then viewed as strings from another language.

Translation-Based

AMR Parsers

EACL, 2017	ACL, 2017	IJCAI, 2019
Peng et al. **Focus: AMR data sparsity** 1 - A stacked BiLSTM Seq2Seq. 2 - Categorization strategy which first maps low frequency concepts and entity subgraphs to a reduced set of category types. 3 - Linearization through depth-first traversal order.	**Konstas et al.** **Focus: Data augmentation** 1 - A stacked BiLSTM Seq2Seq. 2 - Data augmentation: rely only on a large unannotated corpus of English. 3 - Demonstrate that a seq2seq can be trained using any graph-isomorphic linearization. 4 - Self-training to bootstrap a high quality AMR parser from millions of unlabeled Gigaword sentences.	**Ge et al.** **Focus: Modeling syntax and semantics** 1 - Seq2Seq Transformer model for English to AMR translation. 2 - Enhance input with +Syntax and +Semantics. 3 - Syntactic and Semantic are either linearized or input as structure in a structure-aware network.

Modeling Source Syntax and Semantics for Neural AMR Parsing

[Ge et al., 2019]

Sequence-to-Sequence

AMR as translation task from a source sentence to a Linearized AMR graph.

Enhanced Input

Modeling Syntax and Semantics in a mixed sequence of words, syntactic labels and semantic roles.
Modeling Syntax & Semantics through a structure-aware attention.

Parsing

Applied to AMR Framework.

AMR Parsing as Seq2Seq Learning

via AMR Linearization

AMR Graph-to-Target Sequence

- AMR graphs are first converted into AMR trees by removing variables and duplicating the co-referring nodes.
- Remove wiki links.
- Newlines present in an AMR tree are replaced by spaces to get a sequence.

Target Sequence-to-AMR Graph*

- Assign a unique variable to each concept.
- Prune duplicated nodes and redundant material.
- Perform Wikification.
- Restore co-referring nodes.
- As the output may contain brackets that do not match, resulting in incomplete concepts, it should fix incomplete concepts.

*scripts by van Noord and Bos, 2017.

AMR Parsing as Seq2Seq Learning

via AMR Linearization

The airstrikes escalated conflict in a separatist area of Georgia

.

```
(e / escalate-01
    :ARG0 (s2 / strike-01 :path (a / air))
    :ARG1 (c / conflict-01
              :location (a2 / area
                         :mod (s / separatist)
                         : part-of (c2 / country
                                    :wiki "Georgia_(country)"
                                    :name (n / name :op1 "Georgia")))))
```

(escalate-01 : ARG0 (strike-01 : path (air)) : ARG1 (conflict-01 : location (area : mod (separatist) : part-of (country : name (name : op1 “ Georgia “)))))

Modeling Source Syntax and Semantics

via Sequence-Aware Linearization

Linearizing Source Syntax

- pre-order traversal to obtain the syntactic sequence, mixed with grammar labels, POS tags as well as words.

Linearizing Source Semantics

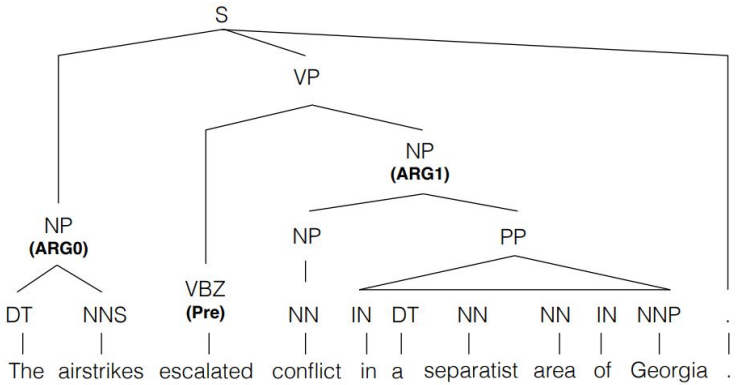
- insert semantic role labels at the corresponding segment's beginning position.

Linearizing Both

- pre-order traversal on the syntactic tree with semantic roles to obtain the linearized version: if a syntactic node is augmented with semantic roles, then it is sequenced as its syntactic label, followed by the semantic role labels.

Modeling Source Syntax and Semantics

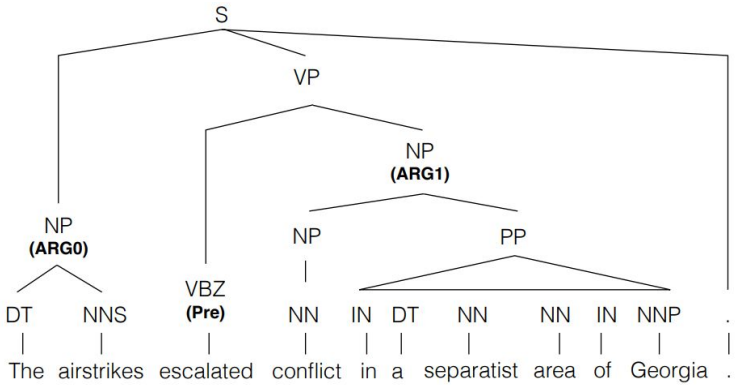
via Sequence-Aware Linearization



Linearizing Source Syntax	S NP DT **The** NNS **airstrikes** VP VBZ **escalated** NP NP NN **conflict** PP IN **in** NP NP DT **a** JJ **separatist** NN **area** PP IN **of** NP NNP **Georgia** . .
Linearizing Source Semantics	ARG0 **The airstrikes** Pre **escalated** ARG1 **conflict in a separatist area of Georgia .**

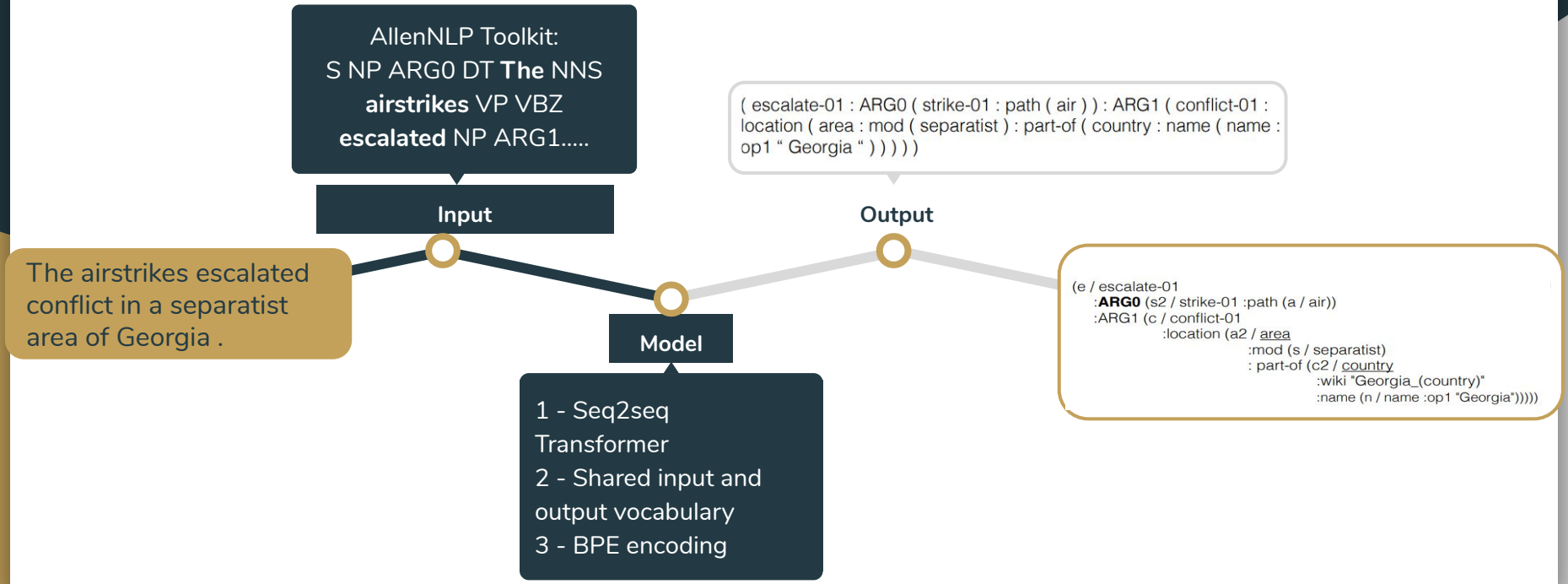
Modeling Source Syntax and Semantics

via Sequence-Aware Linearization



Linearizing Both	S NP ARG0 DT **The** NNS **airstrikes** VP VBZ **escalated** NP ARG1 NP NN **conflict** PP IN **in** NP NP DT **a** JJ **separatist** NN **area** PP IN **of** NP NNP **Georgia** . .
Linearizing Source Syntax	S NP DT **The** NNS **airstrikes** VP VBZ **escalated** NP NP NN **conflict** PP IN **in** NP NP DT **a** JJ **separatist** NN **area** PP IN **of** NP NNP **Georgia** . .
Linearizing Source Semantics	ARG0 **The airstrikes** Pre **escalated** ARG1 **conflict in a separatist area of Georgia** .

Recap Steps


AllenNLP Toolkit:
S NP ARG0 DT **The** NNS **airstrikes** VP VBZ **escalated** NP ARG1.....
Input
The airstrikes escalated conflict in a separatist area of Georgia .
Model
1 - Seq2seq Transformer
2 - Shared input and output vocabulary
3 - BPE encoding
Output
(escalate-01 : ARG0 (strike-01 : path (air)) : ARG1 (conflict-01 : location (area : mod (separatist) : part-of (country : name (name : op1 " Georgia ")))))
(e / escalate-01
:**ARG0** (s2 / strike-01 :path (a / air))
:ARG1 (c / conflict-01
:location (a2 / area
:mod (s / separatist)
: part-of (c2 / country
:wiki "Georgia_(country)"
:name (n / name :op1 "Georgia")))))

Composition-Based

Composition-Based

Principle of compositionality:
the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.

Composition-Based

Characteristics

Principle of compositionality:
the meaning of a complex expression is determined by the meanings of its constituent expressions and the rules used to combine them.

- A semantic graph can be viewed as the result of a derivation process, in which a set of lexical and syntactico-semantic rules are iteratively applied and evaluated.
- Explicitly models derivations that yield semantic graphs.
- Manipulating Graphs through:
  - Graph algebra (AM algebra)
  - Graph grammar (hyperedge replacement grammar)

Some Linguistic Rules*

Definitions

Transitivity is thought as a global property of a clause, by which activity is transferred from an agent to a patient.

- **Transitive verbs** accept one or more objects.
- **Intransitive verbs** do not have objects.

Control is a construction in which the understood subject of a given predicate is determined by some expression in context.

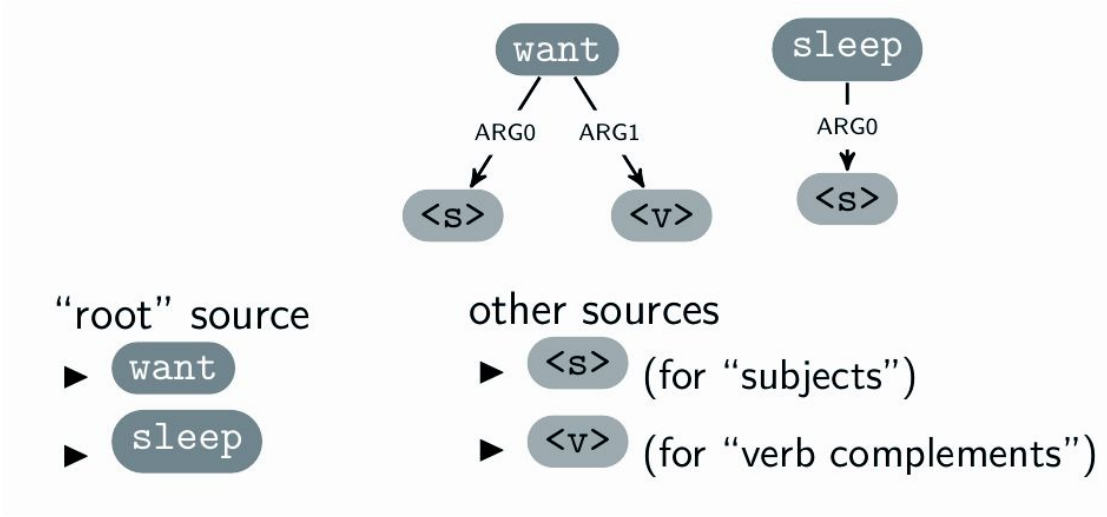
- **Control verbs** semantically select their arguments. Their appearance strongly influences the nature of the arguments they take.

*Wikipedia Definitions

Apply-Modify Algebra

S-Graph

Directed graphs with node and edge labels in which certain nodes have been designated as **sources** and annotated with type information.



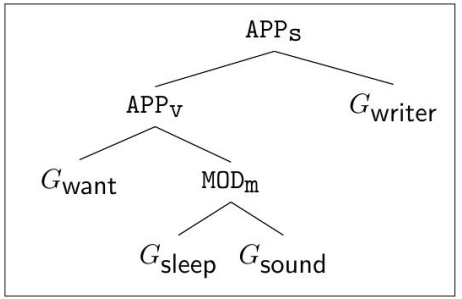
Apply-Modify Algebra

S-Graph

AM algebra provides a systematic way to construct graphs:

- **term (tree of operation symbols)**
- value (s-graph)

The writer wants to sleep soundly



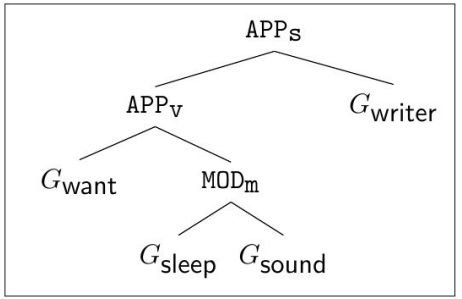
Apply-Modify Algebra

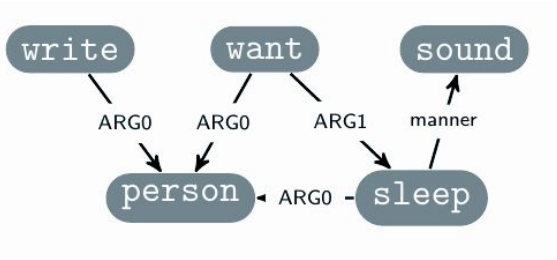
S-Graph

AM algebra provides a systematic way to construct graphs:

- term (tree of operation symbols)
- **value (s-graph)**

The writer wants to sleep soundly





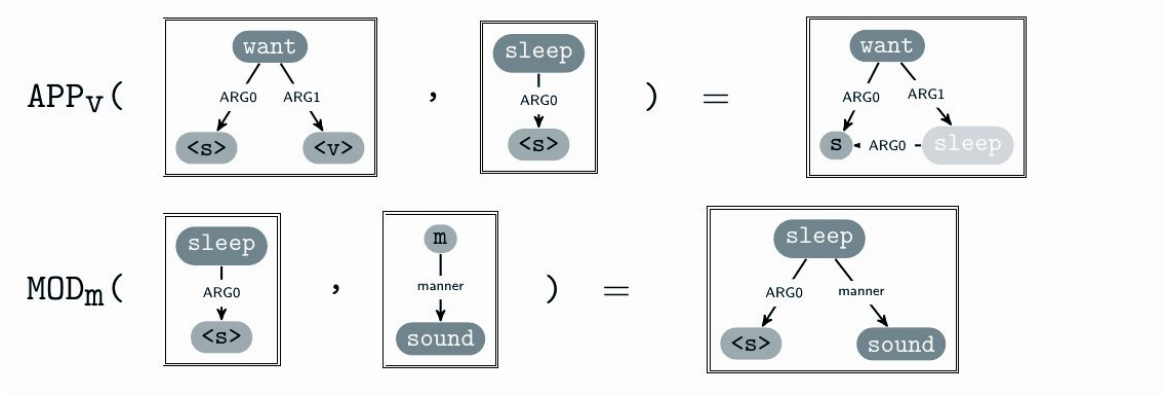
Apply-Modify Algebra

S-Graph

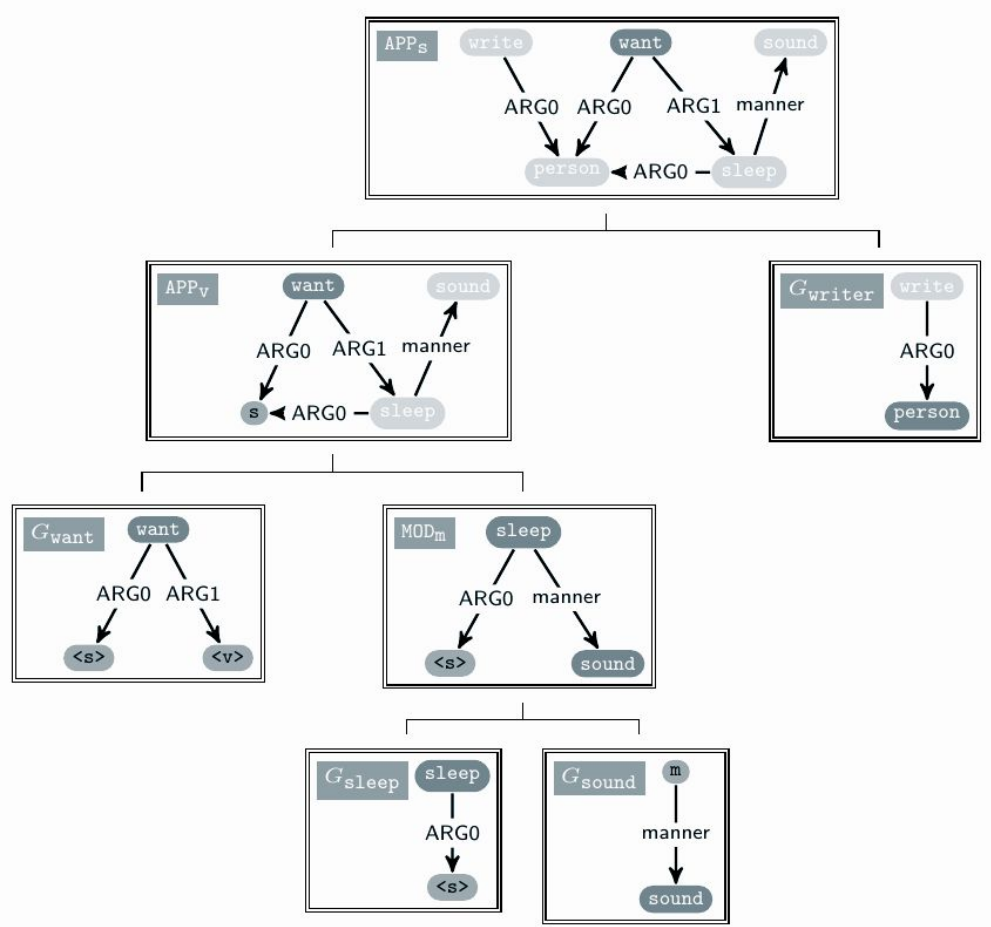
Two operations for combining s-graphs:

- Apply (head+complement)
- Modify (head+modifier)

The writer wants to sleep soundly



The writer wants to sleep soundly



Composition-Based

AMR Parsers

ACL, 2015	ACL, 2018	ACL, 2019
Artzi et al. **Focus: CCG for AMR Parsing** 1 - Lambda-calculus representations. 2 - Use Combinatory Categorial Grammars (CCG) to recover compositional aspects of meaning.	**Groschwitz et al.** **Focus: Linguistic rules for AMR Parsing** 1 - Derive AMR compositionally. 2 - Use Apply-Modify algebra to parse a string into a graph compositional structure. 3 - Use of Semantic Types to restrict AMR Parser.	**Lindenmann et al.** **Focus: Linguistic rules for meaning representation** 1 - Compositional-neural semantic parser. 2 - Build over Groschwitz et al., 2018. 3 - Incorporate BERT and Multi Tasking with Dependency parsing. 4 - Applied across a diverse range of GraphBanks with competitive accuracies.

AMR dependency parsing with a typed semantic algebra

[Groschwitz et al., 2018]

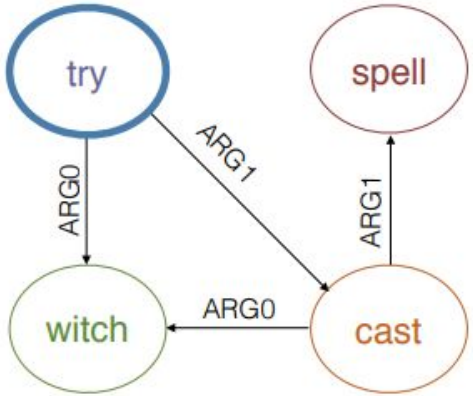
Typed Semantic Algebra	**Apply-Modify Algebra (AM) to parse a string into a graph compositional structure.**
Dependency-Semantic Combination	**Neural supertagger for identifying the elementary graphs for the individual words with a neural dependency model along the lines for identifying the operations of the AM.**
Parsing	**Applied to AMR Framework.** **Extended to other parses by Lindenmann et al., 2019.**

Classic AMR Parsing

The witch tried to cast a spell.

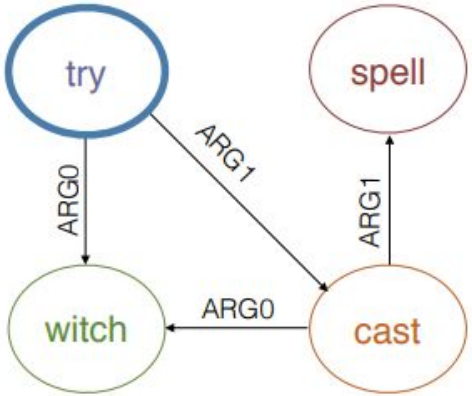
Node Prediction

Edge Prediction

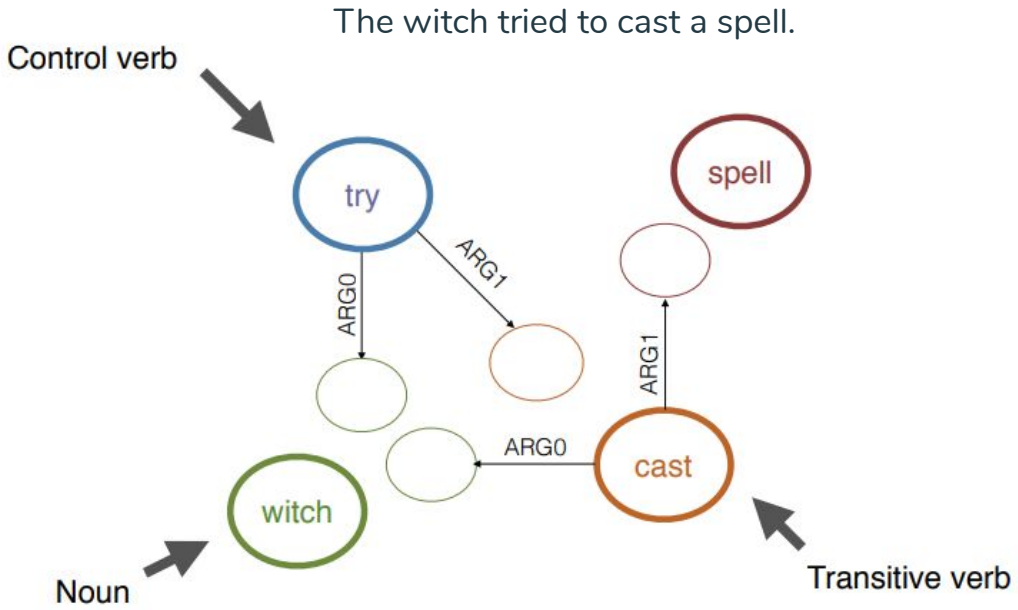


Not just nodes and edges

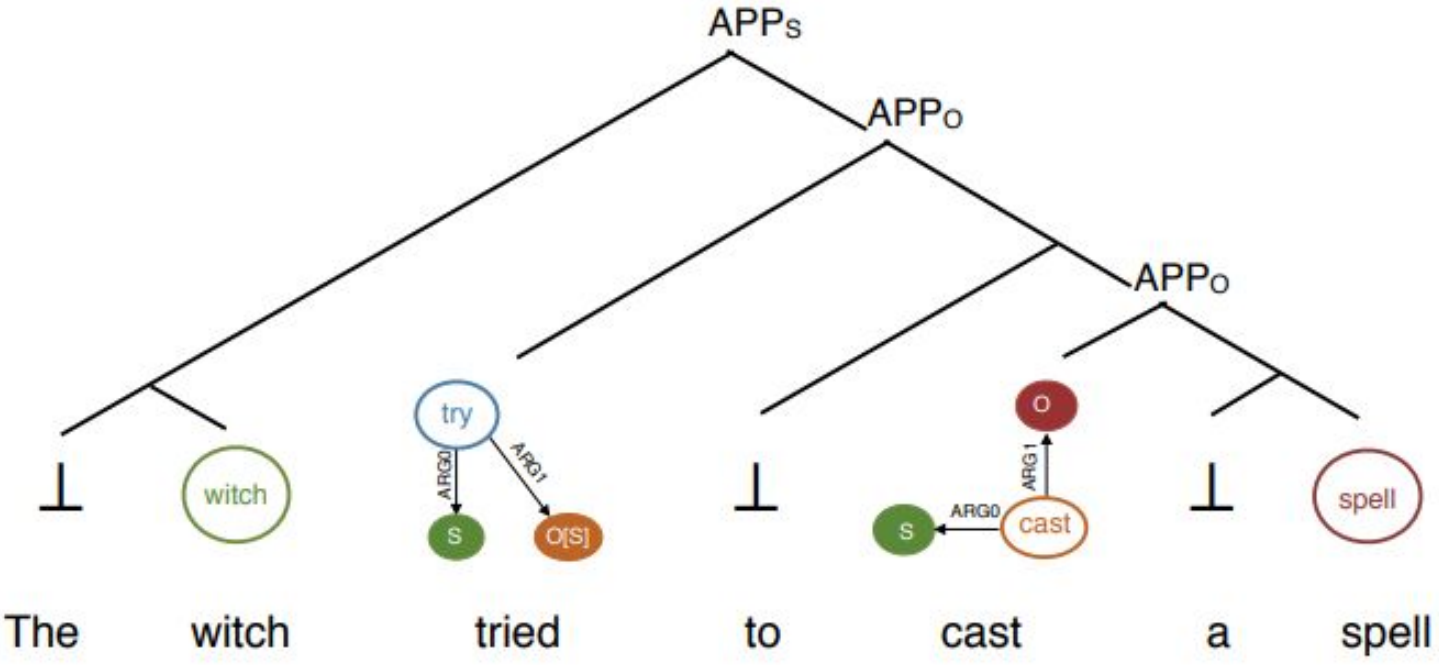
The witch tried to cast a spell.



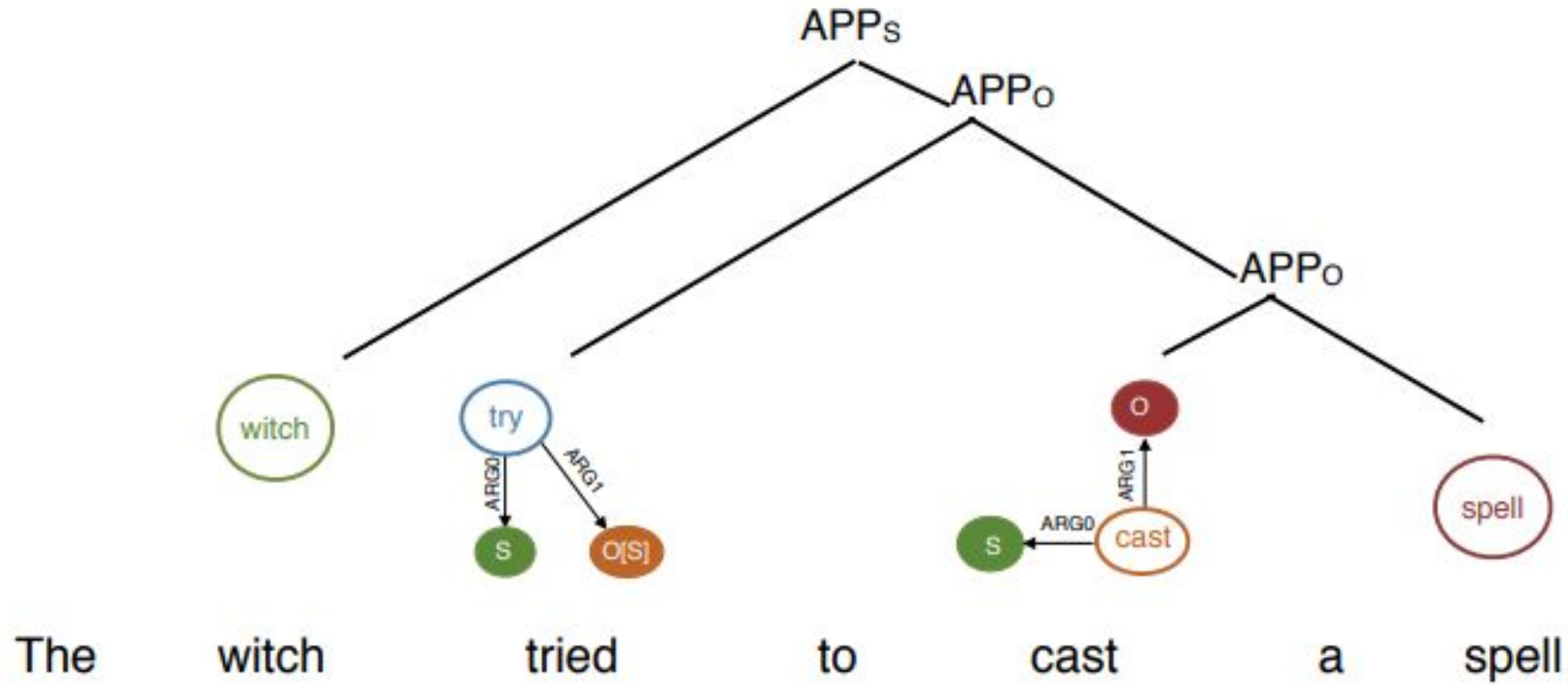
A closer look

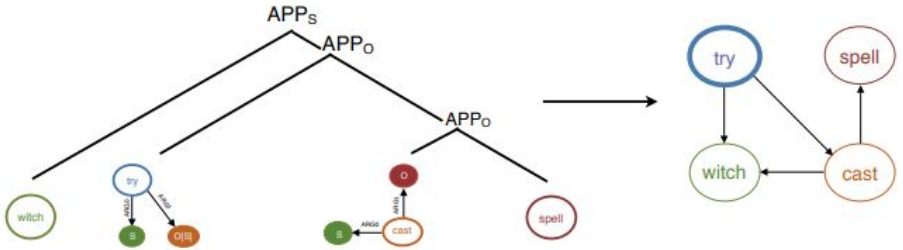


Compositionality



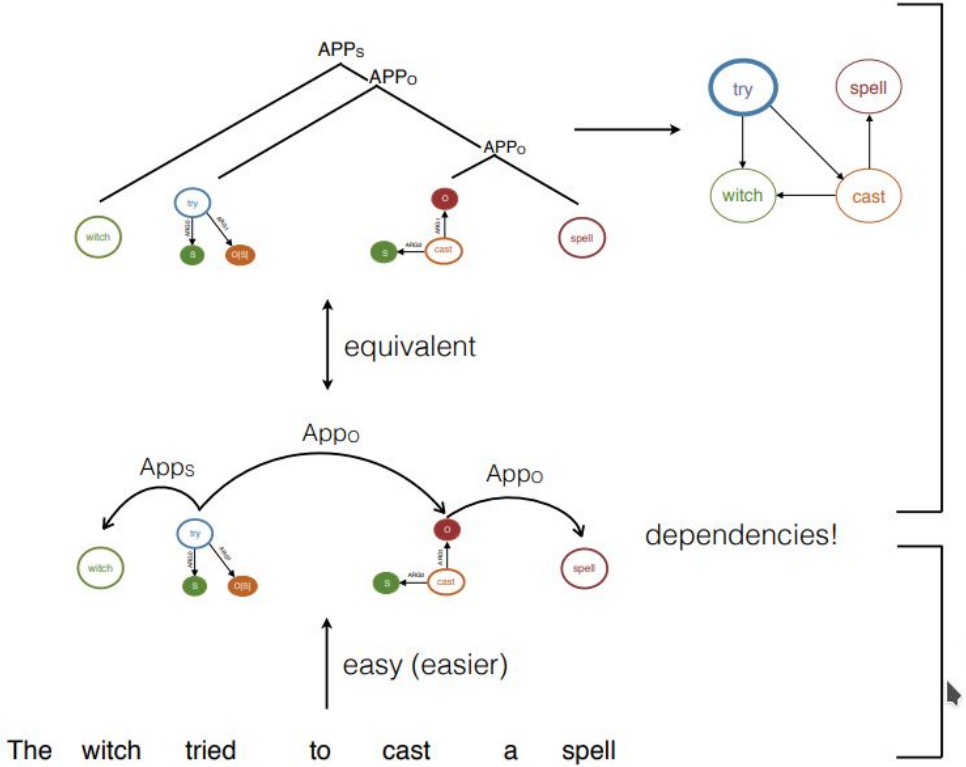
Compositionality





difficult

The witch tried to cast a spell



Authors assume that the formalism is equivalent to a dependency structure.
Dependency structures are easier to predict.

AM dependency trees for AMR

Decomposition

1. Align each node in the graph with a word token.
2. Group the edges together with either their source or target nodes, depending on the edge label.
3. Choose a source name for the open slot at the other end of each attached edge.
4. Match reentrancy patterns to determine annotations for each source.

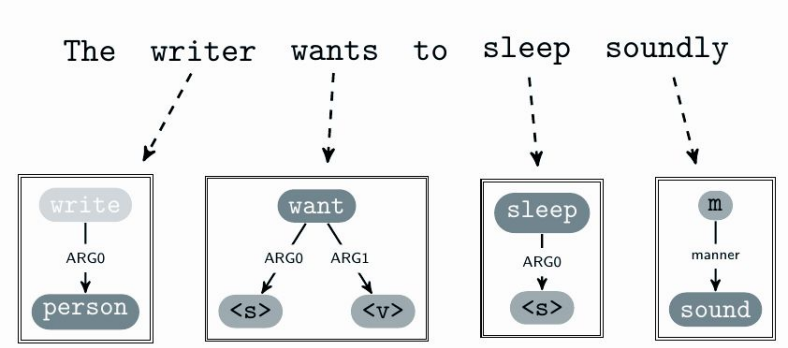
AM Algebra-Based Parsing

Prectiction Model

1. A supertagger predicts graphs for words.
2. A dependency parser predicts APP and MOD edges.
3. At test time, compute highest-scoring well-typed AM dependency tree for input sentence.

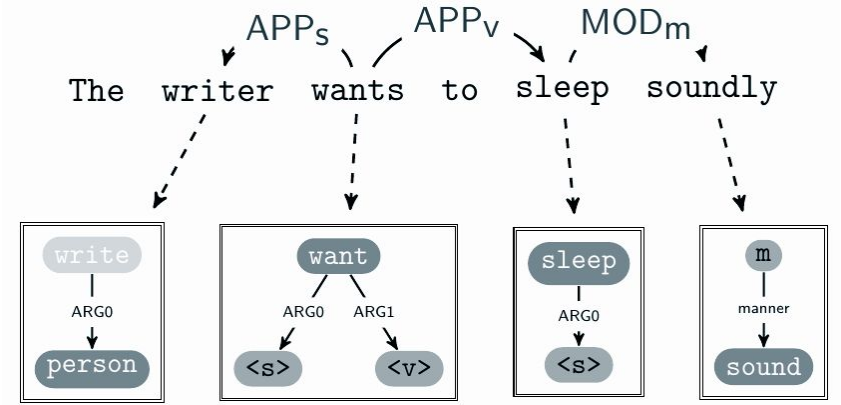
AM Algebra-Based Parsing

1. **A supertagger predicts graphs for words.**
2. A dependency parser predicts APP and MOD edges.



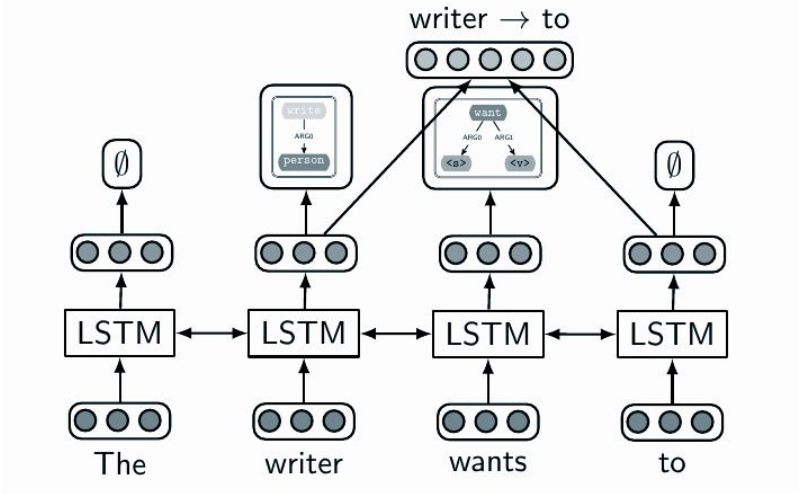
AM Algebra-Based Parsing

1. A supertagger predicts graphs for words.
2. **A dependency parser predicts APP and MOD edges.**



AM Algebra-Based Parsing

1. A supertagger predicts graphs for words.
2. A dependency parser predicts APP and MOD edges.



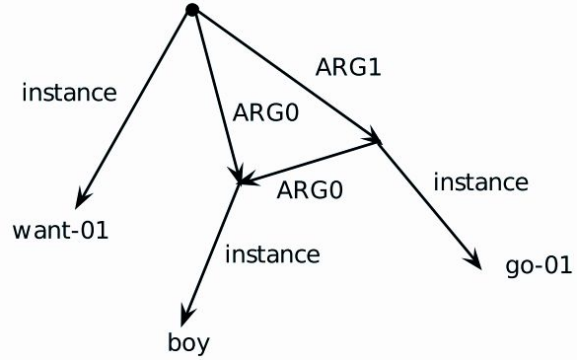
Evaluating Performance

AMR Evaluation

Gold parse: *the boy wants to go*

Semantic relationships encoded in the AMR graph can also be viewed as a conjunction of logical propositions, or triples:

instance(a, want-01) ∧
instance(b, boy) ∧
instance(c, go-01) ∧
ARG0(a, b) ∧
ARG1(a, c) ∧
ARG0(c, b)

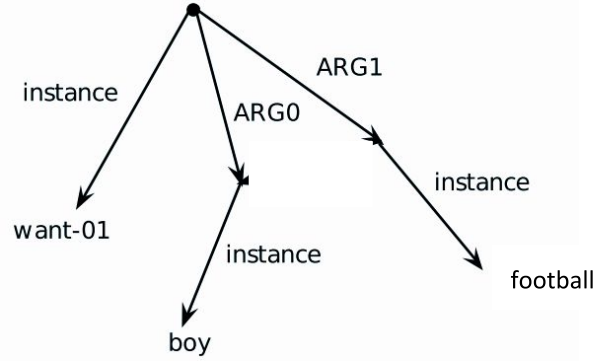


AMR Evaluation

Predicted parse: *the boy wants the football*

Semantic relationships encoded in the AMR graph can also be viewed as a conjunction of logical propositions, or triples:

instance(x, want-01) ∧
instance(y, boy) ∧
instance(z, football) ∧
ARG0(x, y) ∧
ARG1(x, z)



AMR Evaluation

Smatch (semantic match) [Cai and Knight, 2013]

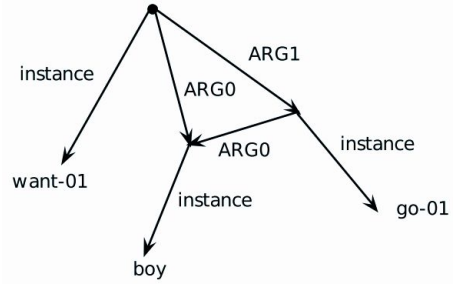
Smatch measures precision, recall, and f-score of the triples in the second AMR against the triples in the first AMR.

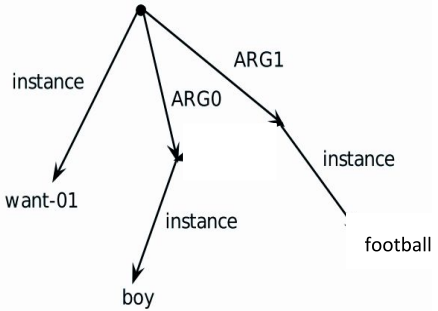
The boy wants to go

instance(a, want-01)
∧
instance(b, boy) ∧
instance(c, go-01) ∧
ARG0(a, b) ∧
ARG1(a, c) ∧
ARG0(c, b)

The boy wants football

instance(x, want-01) ∧
instance(y, boy) ∧
instance(z, football) ∧
ARG0(x, y) ∧
ARG1(x, z)





AMR Evaluation

Smatch (semantic match) [Cai and Knight, 2013]

Variable names are not shared between the two AMRs, so there are multiple ways to compute the propositional overlap based on different variable mappings.

The boy wants to go

instance(a, want-01)
∧
instance(b, boy) ∧
instance(c, go-01) ∧
ARG0(a, b) ∧
ARG1(a, c) ∧
ARG0(c, b)

The boy wants football

instance(x, want-01) ∧
instance(y, boy) ∧
instance(z, football) ∧
ARG0(x, y) ∧
ARG1(x, z)

AMR Evaluation

Smatch

Smatch score is defined as the maximum f-score obtainable via a one-to-one matching of variables between the two AMRs.

The boy wants to go

instance(a, want-01) ∧
instance(b, boy) ∧
instance(c, go-01) ∧
ARG0(a, b) ∧
ARG1(a, c) ∧
ARG0(c, b)

The boy wants football

instance(x, want-01) ∧
instance(y, boy) ∧
instance(z, football) ∧
ARG0(x, y) ∧
ARG1(x, z)

	M	P	R	F
x=a, y=b, z=c:	4	4/5	4/6	0.73
x=a, y=c, z=b:	1	1/5	1/6	0.18
x=b, y=a, z=c:	0	0/5	0/6	0.00
x=b, y=c, z=a:	0	0/5	0/6	0.00
x=c, y=a, z=b:	0	0/5	0/6	0.00
x=c, y=b, z=a:	2	2/5	2/6	0.36
smatch score:				0.73

AMR Evaluation

Fine-Grained [Damonte et al., 2017]

Smatch score (Cai and Knight, 2013) is not sufficient to evaluate AMR parsers.

1. AMR parsing involves a large number of subtasks.
2. The Smatch score consists of a single number that does not assess the quality of each subtasks separately.
3. The Smatch score weighs different types of errors in a way which is not necessarily useful for solving a specific NLP problem.

AMR Evaluation

Fine-Grained [Damonte et al., 2017]

Silvio Berlusconi gave Lucio Stanca his current role of modernizing Italy's bureaucracy.

```
(g / give-01
      :ARG0 (p3 / silvio :mod (n4 / berlusconi))
      :ARG1 (r / role
               :time (c2 / current)
               :mod (m / modernize-01
                       :ARG0 p4
                       :ARG1 (b / bureaucracy :part-of (c3 / italy)))
               :poss p4)
      :ARG2 (p4 / person lucio :mod stanca))
```

Smatch: 56

```
(g / give-01
      :ARG0 (p3 / person :wiki "Silvio_Berlusconi"
               :name (n4 / name :op1 "Silvio" :op2 "Berlusconi"))
      :ARG0 (r / role
               :ARG0 (c2 / current)
               :ARG0 (m / modernize-01
                        :ARG0 p4
                        :ARG0 (b / bureaucracy
                                 :ARG0 (c3 / country :wiki "Italy"
                                           :name (n6 / name :op1 "Italy")))))
               :ARG0 p4)
      :ARG0 (p4 / person :wiki -
               :name (n5 / name :op1 "Lucio" :op2 "Stanca")))
```

Smatch: 78

AMR Evaluation

Fine-Grained [Damonte et al., 2017]

Unlabeled

Smatch score computed on the predicted graphs after removing all edge labels.

Reentrancy

Smatch score only on reentrant edges.

Named Ent. & Wikification

F-score on the named entities and wiki roles for named entities (Wikification) that consider edges labeled with :name and :wiki.

Semantic Role Labelling

Smatch score on :ARG edges only.

No WSD

Smatch score ignoring the sense specified by the Propbank frame (e.g., duck-01/02).

Concepts

F-score on the list of predicted concepts.

Negations

An F-score that computes the correct-ness of all negated concepts by looking for the :polarity role.

NP-Only

Smatch score on noun phrases only.

AMR Evaluation

Fine-Grained [Damonte et al., 2017]

Silvio Berlusconi gave Lucio Stanca his current role of modernizing Italy's bureaucracy.

Metric	First parse	Second parse
Smatch	56	78
Unlabeled	65	100
No WSD	56	78
NP-only	39	86
Reentrancy	69	46
Concepts	56	100
Named Ent.	0	100
Wikification	0	100
Negations	0	0
SRL	69	54

```
(g / give-01
      :ARG0 (p3 / silvio :mod (n4 / berlusconi))
      :ARG1 (r / role
            :time (c2 / current)
            :mod (m / modernize-01
                  :ARG0 p4
                  :ARG1 (b / bureaucracy :part-of (c3 / italy)))
            :poss p4)
      :ARG2 (p4 / person lucio :mod stanca))
```

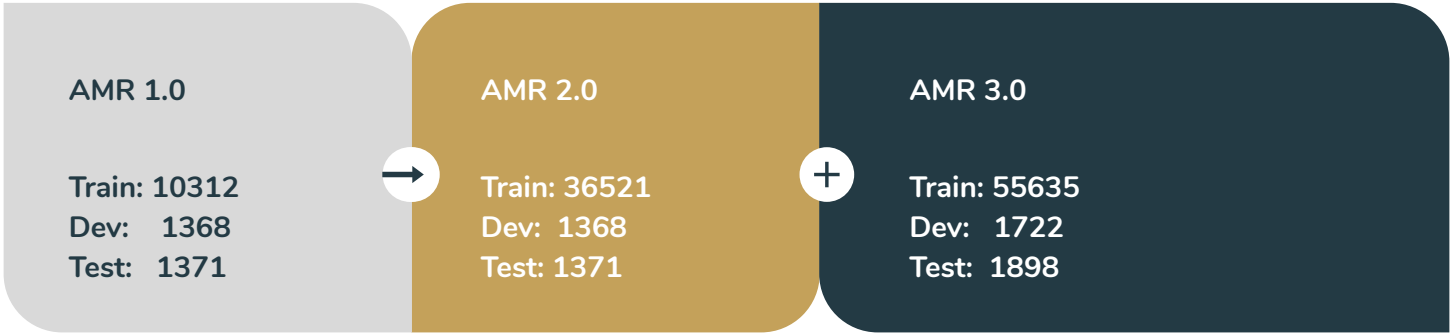
Smatch: 56

```
(g / give-01
      :ARG0 (p3 / person :wiki "Silvio_Berlusconi"
            :name (n4 / name :op1 "Silvio" :op2 "Berlusconi"))
      :ARG0 (r / role
            :ARG0 (c2 / current)
            :ARG0 (m / modernize-01
                  :ARG0 p4
                  :ARG0 (b / bureaucracy
                        :ARG0 (c3 / country :wiki "Italy"
                              :name (n6 / name :op1 "Italy"))))
            :ARG0 p4)
      :ARG0 (p4 / person :wiki -
            :name (n5 / name :op1 "Lucio" :op2 "Stanca")))
```

Smatch: 78

AMR GraphBanks

English



Additional AMR GraphBanks

GraphBanks

The Little Prince

Train: 1274
Dev: 145
Test: 143

=

AMR Chinese

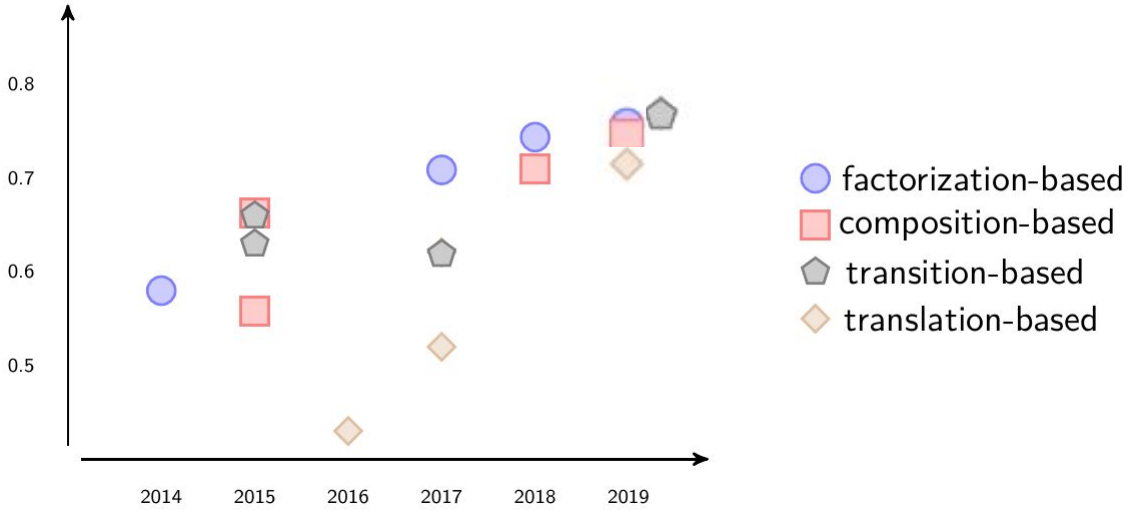
Train: 1274
Dev: 245
Test: 143

BIO AMR

Train: 5452
Dev: 500
Test: 500

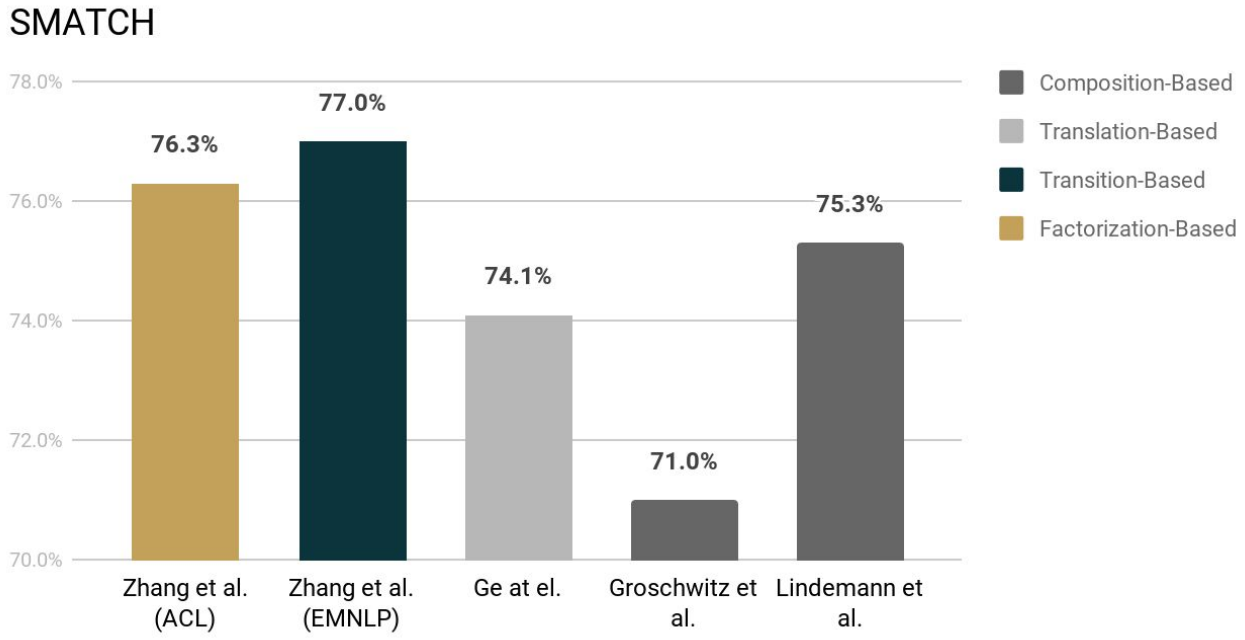
Performance

Smatch Trend (Different test sets over years)



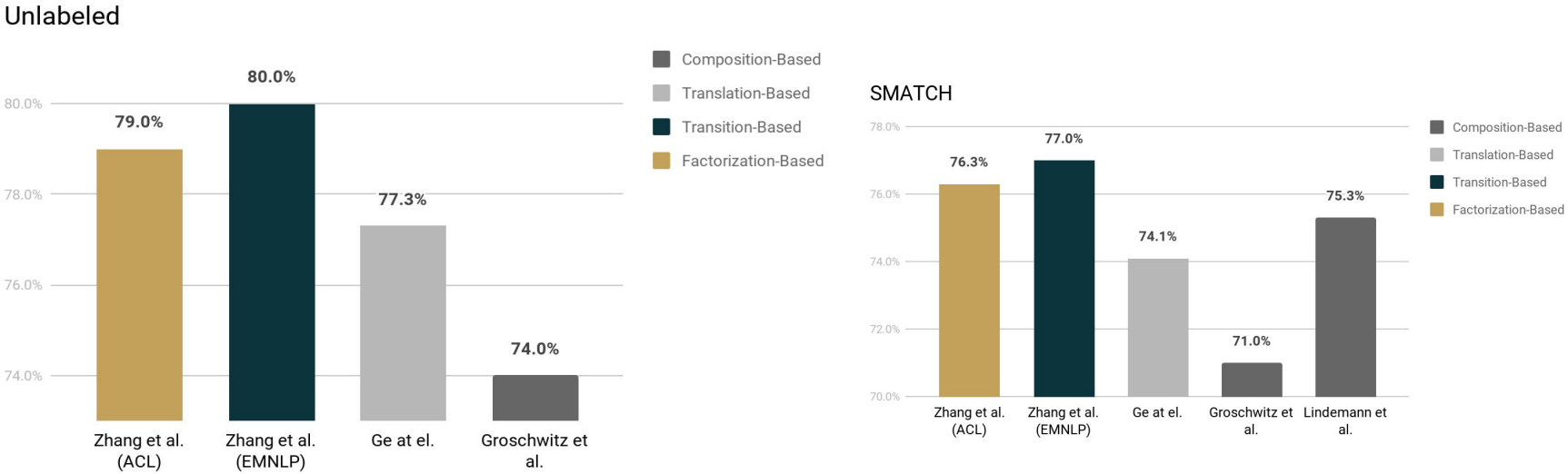
Performance

Smatch AMR 2.0



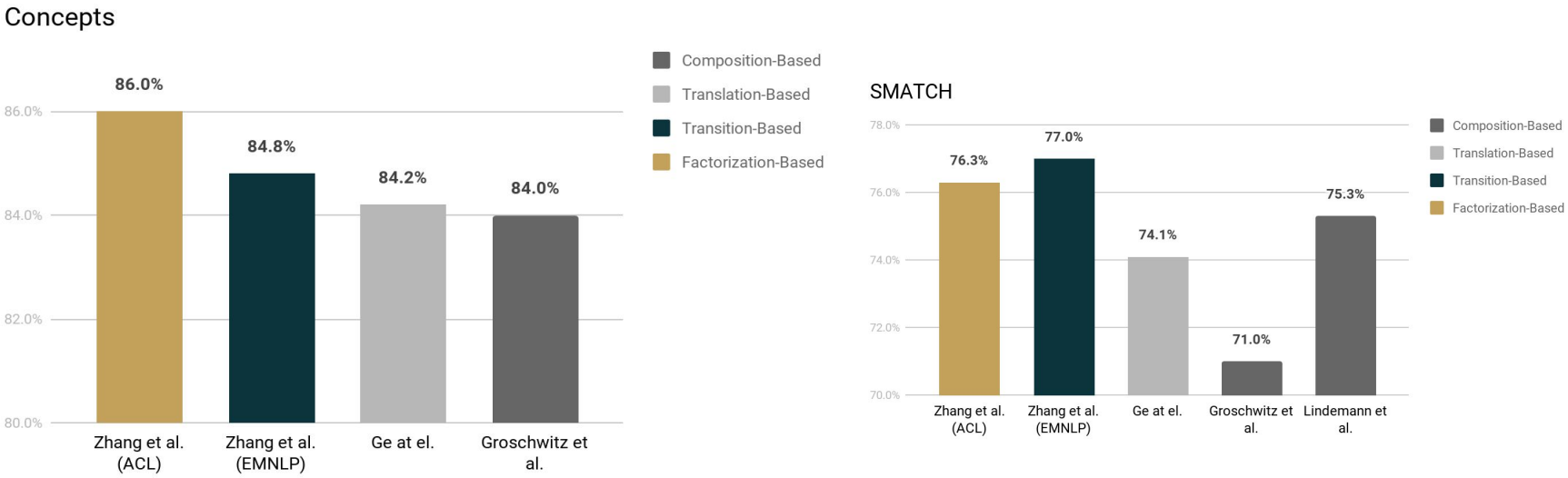
Performance

Unlabeled AMR 2.0 :
Smatch score computed on the predicted graphs after removing all edge labels.



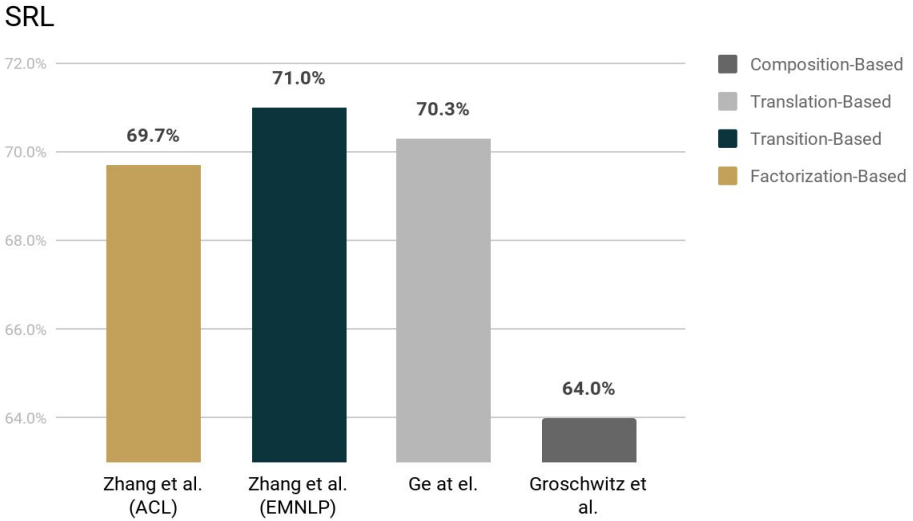
Performance

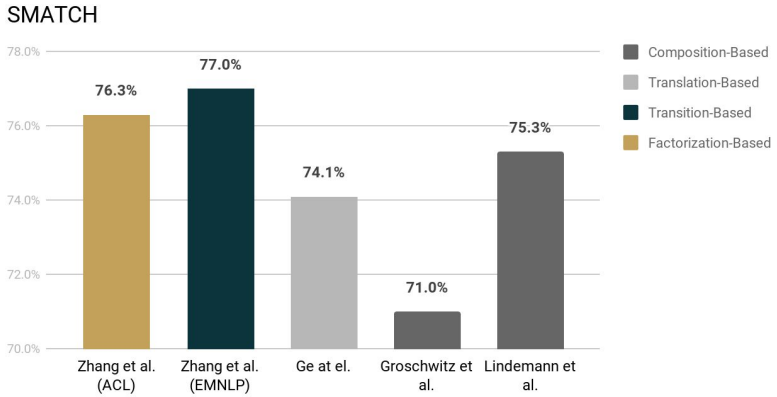
Concepts AMR 2.0:
F-score on the list of predicted concepts.



Performance

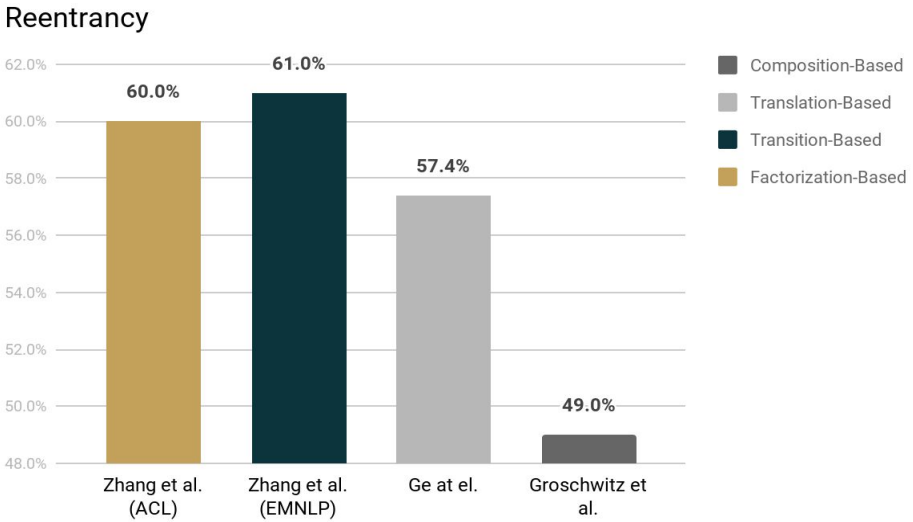
SRL AMR 2.0:
Smatch score on :ARG edges only.

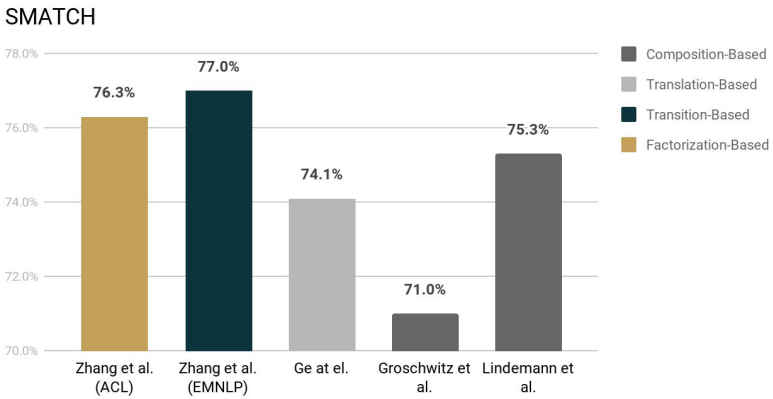




Performance

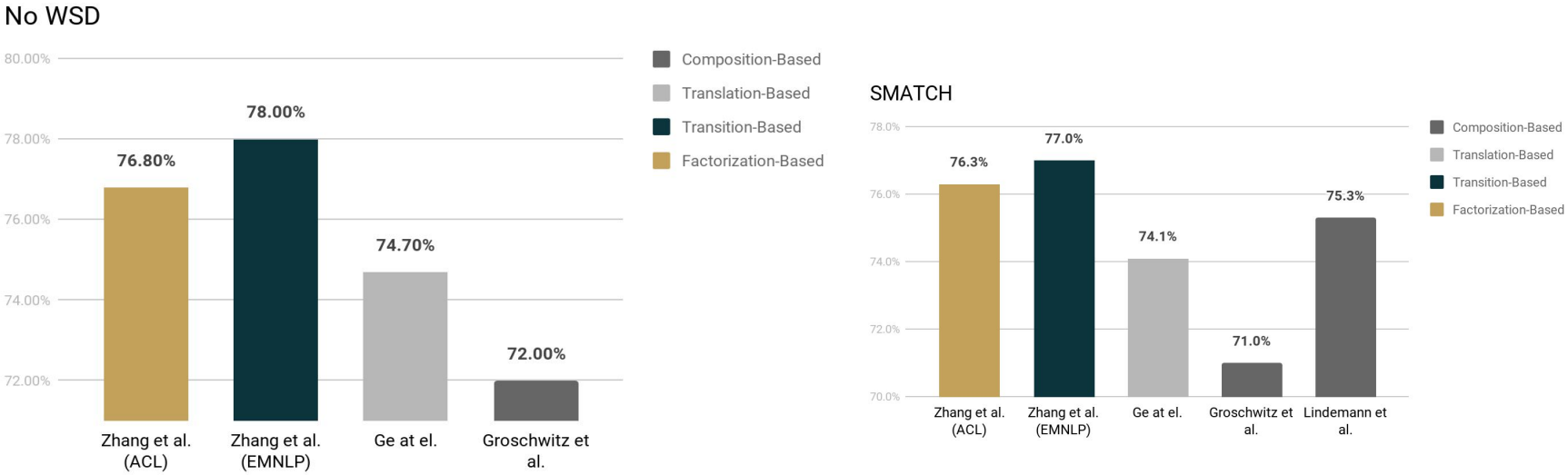
Reentrancy AMR 2.0:
Smatch score only on reentrant edges.





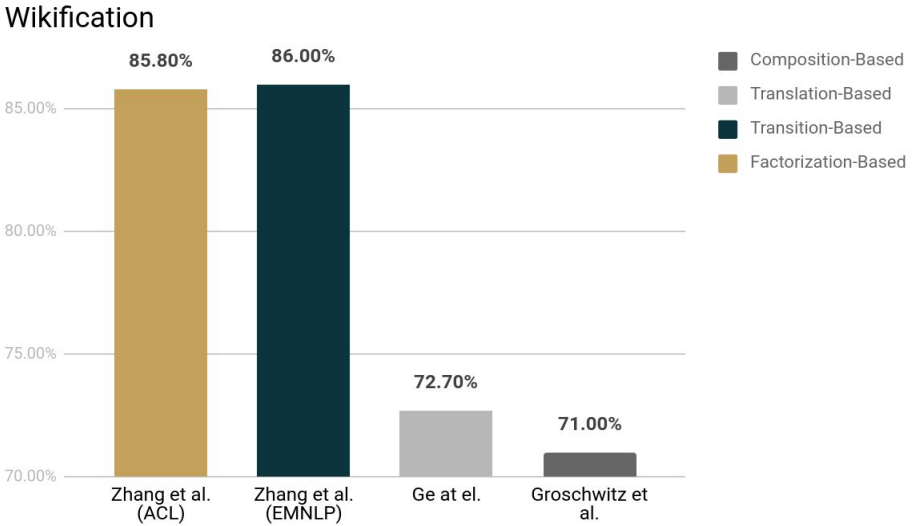
Performance

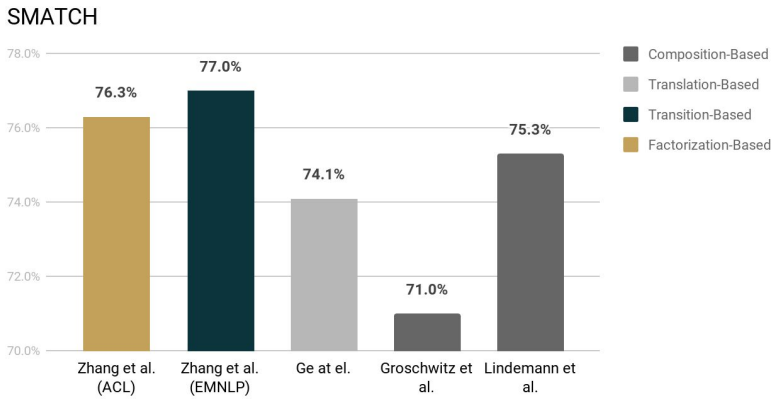
No WSD AMR 2.0:
Smatch score ignoring the sense specified by the Propbank frame (e.g., duck-01/02).



Performance

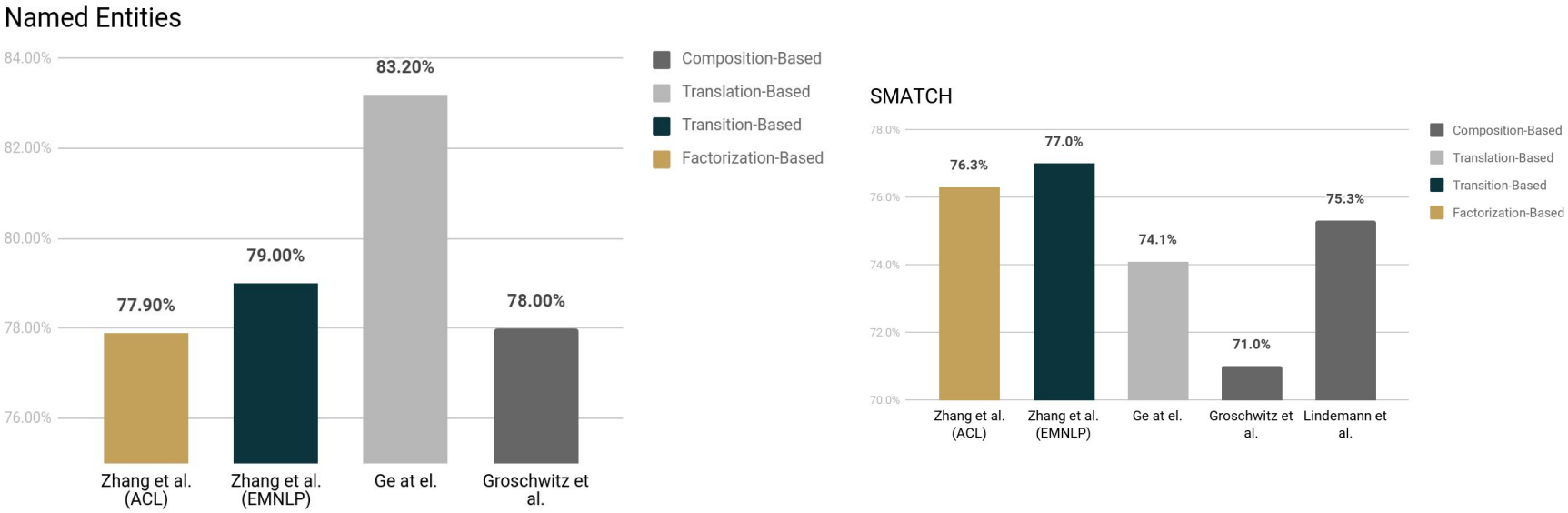
Wikification AMR 2.0:
Wiki roles for NM (Wikification) that consider edges labeled with :name and :wiki.





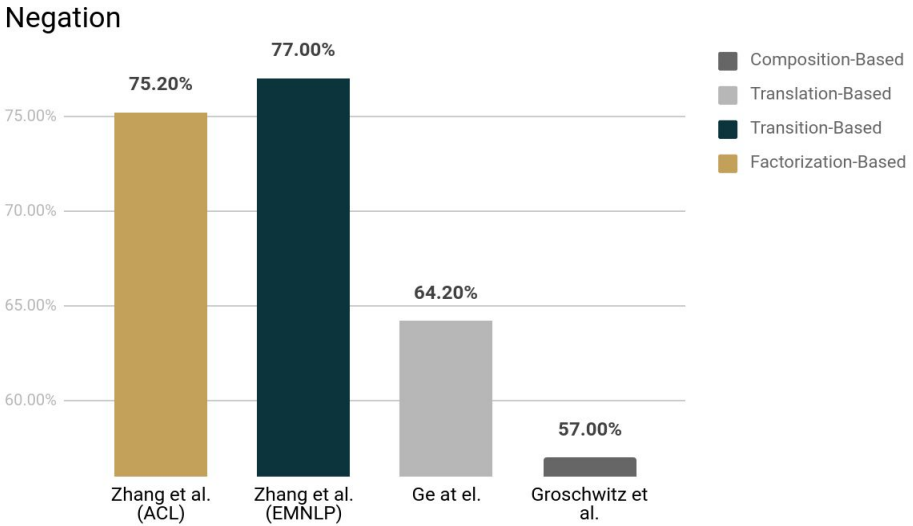
Performance

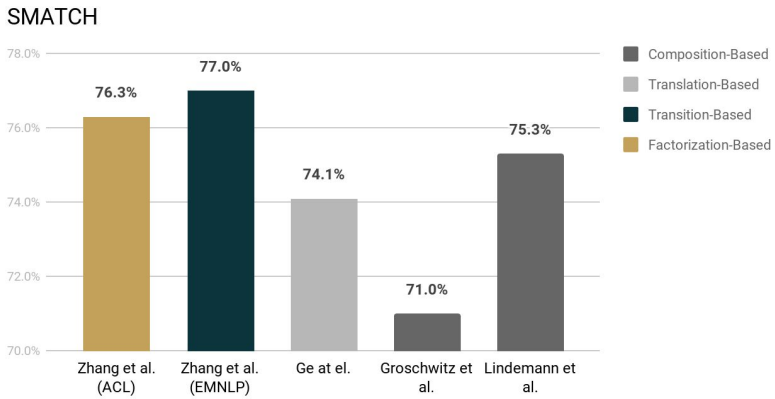
Named Entities AMR 2.0:
F-score on the named entities.



Performance

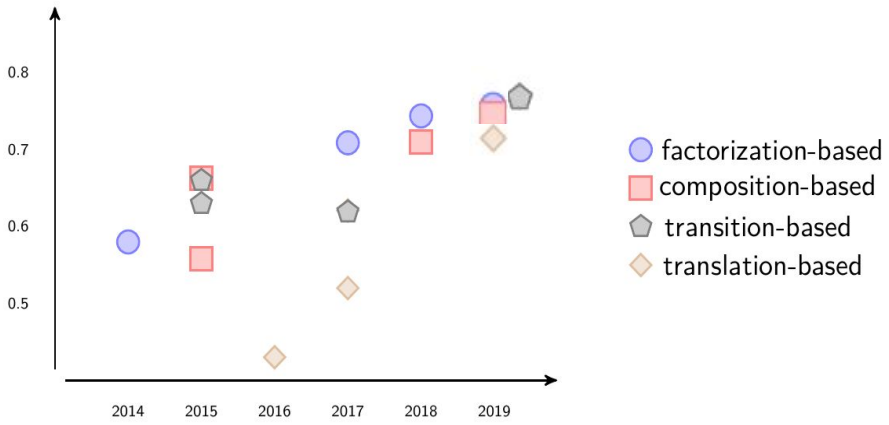
Negation AMR 2.0:
F-score on all negated concepts by looking for the :polarity role.





Conclusions & Remarks

- Improvements are shown in all the groups during years and their performances are currently in the same ballpark.
- Explicitly modeling specific components of AMR, i.e, Reentrancy, NER, SRL or might yield significant improvements.
- Focusing on separate sub tasks of parsing, i.e., concept identification, relation identification and training them jointly has shown to be effective.
- Seq2Seq approaches suffer from data sparsity due to the size of annotated training data, therefore silver-standard data augmentation has shown to be effective.



THANK YOU

Stay Safe :)

References

Factorization - Based

1. Flanigan, Jeffrey, Sam Thomson, Jaime G. Carbonell, Chris Dyer and Noah A. Smith. “A Discriminative Graph-Based Parser for the Abstract Meaning Representation.” *ACL* (2014).
2. Foland, William and James H. Martin. “CU-NLP at SemEval-2016 Task 8: AMR Parsing using LSTM-based Recurrent Neural Networks.” *SemEval@NAACL-HLT* (2016).
3. Lyu, Chunchuan and Ivan Titov. “AMR Parsing as Graph Prediction with Latent Alignment.” *ACL* (2018).
4. Zhang, Sheng, Xutai Ma, Kevin Duh and Benjamin Van Durme. “AMR Parsing as Sequence-to-Graph Transduction.” *ACL* (2019).

Transition - Based

1. Wang, Chuan, Nianwen Xue and Sameer Pradhan. “A Transition-based Algorithm for AMR Parsing.” *HLT-NAACL* (2015).
2. Damonte, Marco, Shay B. Cohen and Giorgio Satta. “An Incremental Parser for Abstract Meaning Representation.” *EACL* (2016).
3. Liu, Yijia, Wanxiang Che, Bo Zheng, Bing Qin and Ting Liu. “An AMR Aligner Tuned by Transition-based Parser.” *EMNLP* (2018).
4. Zhang, Sheng, Xutai Ma, Kevin Duh and Benjamin Van Durme. “Broad-Coverage Semantic Parsing as Transduction.” *EMNLP/IJCNLP* (2019).

References

Translation - Based

1. Peng, Xiaochang, Chuan Wang, Daniel Gildea and Nianwen Xue. “Addressing the Data Sparsity Issue in Neural AMR Parsing.” *EACL* (2017).
2. Konstas, Ioannis, Srini Iyer, Mark Yatskar, Yejin Choi and Luke Zettlemoyer. “Neural AMR: Sequence-to-Sequence Models for Parsing and Generation.” *ACL* (2017).
3. Ge, DongLai, Junhui Li, Muhua Zhu and Shoushan Li. “Modeling Source Syntax and Semantics for Neural AMR Parsing.” *IJCAI* (2019).

Composition - Based

1. Artzi, Yoav, Kenton Lee and Luke Zettlemoyer. “Broad-coverage CCG Semantic Parsing with AMR.” *EMNLP* (2015).
2. Groschwitz, Jonas, Matthias Lindemann, Meaghan Fowlie, Mark Johnson and Alexander Koller. “AMR Dependency Parsing with a Typed Semantic Algebra.” *ACL* (2018).
3. Lindemann, Matthias, Jonas Groschwitz and Alexander Koller. “Compositional Semantic Parsing Across Graphbanks.” *ACL* (2019).

Further reading: Groschwitz, Jonas, Meaghan Fowlie, Mark Johnson and Alexander Koller. “A constrained graph algebra for semantic parsing with AMRs.” *IWCS* (2017).