# Open-Domain Semantic Parsing

Shay Cohen
ILCC, School of Informatics
University of Edinburgh

July 19, 2019

# NLP - Then and Now...

A few years ago:

- Some "core" problems
- A specialized system:
  - Optimization algorithm
  - Feature engineering
  - Inference
  - Often, a specialized learning algorithm for training

Now:

- A proliferation of problems
- A generic system:
  - Optimization is a hyperparameter
  - Architecture engineering
  - Backprop as a generic solution for training

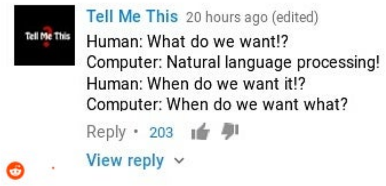
One of the great advantages of current state: we specify models easily in terms of a computation graph (neural network) and not worry about the "details." Heavy hyperparameter tuning

# What's Next?

For problems where the training data mostly covers our bases (and we have enough of it), neural networks can greatly help. We just specify a model and estimate it...

But what do we do with the rest? When there is not enough data?

- Dialogues
- Summarization
- Question answering




When reasoning and background knowledge is required, current ML just by itself is not going to cut it... Need an intermediate representation

# What is Semantic Parsing?

The common slogan for semantic parsing:

> *Find who did what to whom in a sentence.*

This can actually be solved using predicate-argument structure, syntax is sufficient.

This is also typical for semantic role labeling, which is a form of semantic parsing (Gildea and Jurafsky, 2002)

# What is Semantic Parsing?

The common slogan for semantic parsing:

> *Find who did what to whom in a sentence.*

This can actually be solved using predicate-argument structure, syntax is sufficient.

This is also typical for semantic role labeling, which is a form of semantic parsing (Gildea and Jurafsky, 2002)

Semantic parsing is now more like:

> *Find who did what to whom, and where, and when, and how, and why... and using what... in a sentence, or in a paragraph, or in a whole document.*

# What is Semantic Parsing?

The common slogan for semantic parsing:

> *Find who did what to whom in a sentence.*

This can actually be solved using predicate-argument structure, syntax is sufficient.

This is also typical for semantic role labeling, which is a form of semantic parsing (Gildea and Jurafsky, 2002)

Semantic parsing is now more like:

> *Find who did what to whom, and where, and when, and how, and why... and using what... in a sentence, or in a paragraph, or in a whole document.*

How do we even represent such information?

# Two Representations

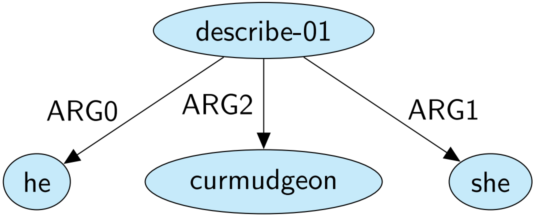
Abstract Meaning Representation (Banarescu et al. 2013):

- Light annotation guidelines
- Much information is left underspecified
- Canonicalizes language

On the other end of the spectrum, Discourse Representation Theory (Kemp and Reyle, 1993):

- Heavily influenced by logical-style semantics
- Designed to represent a set of sentences (a document)
- Canonicalizes language

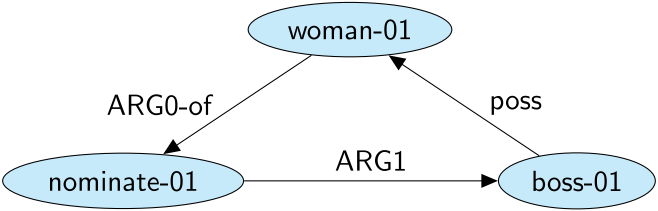
# Abstract Meaning Representation




1. He described her as a curmudgeon,
2. His description of her: curmudgeon,
3. She was a curmudgeon, according to his description.

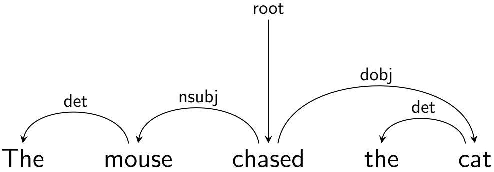
# Challenges with AMR

- Can have “re-entrancies” – nodes with multiple parents. Not amenable to a nice generative story or easy inference. (Maybe even loops!)




**The woman who nominated her boss**

- Graph is grounded in the sentence, but not like a dependency tree – concepts are introduced as abstractive nodes



# Data Source for Abstract Meaning Representation

- Data source: transcripts and English translations of Mandarin Chinese broadcast news programming from China Central TV, Wall Street Journal, Xinhua news texts (translated), other newswire data
- Total number of sentences: around 39,000
- (Unfortunately?) Continues the old tradition of the NLP community in parsing newswire text...

# Our Parser (Damonte et al. 2017)

A left-to-right incremental parser that scans a sentence and adds concepts and edges between them

Maintains a stack that keeps track of the current state of the parser

Incremental, but not in the traditional sense: might have a disconnected structure (cognitively plausible?)

Uses a high-coverage "trick" to handle re-entrancies

Very fast (in practice, linear in the length of the sentence)

The actions: LARC, RARC, REENTRANCY, REDUCE, SHIFT

# Transition System

*~~The~~ ~~boy~~ wants to believe the girl*

STACK

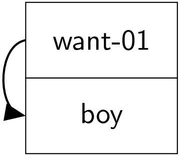
GRAPH

| |
|---|
| boy |

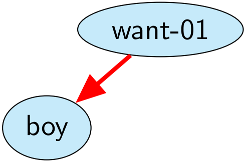
# Transition System

*~~The~~ ~~boy~~ ~~wants~~ to believe the girl*

STACK




GRAPH

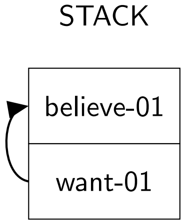


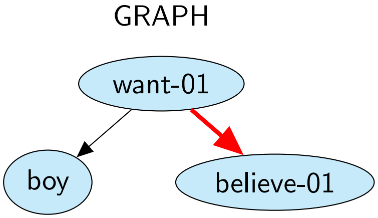
# Transition System

*~~The boy wants~~ to believe the girl*

# Transition System

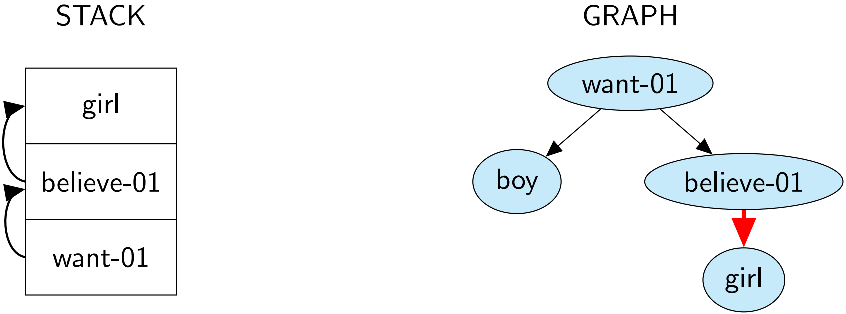
*~~The boy wants to believe~~ the girl*



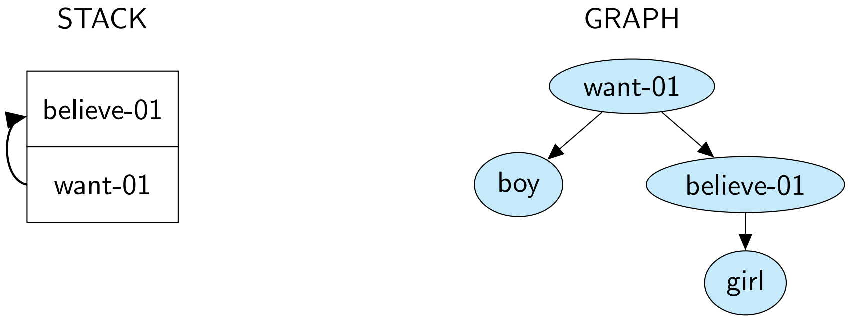
# Transition System

*~~The boy wants to believe the girl~~*



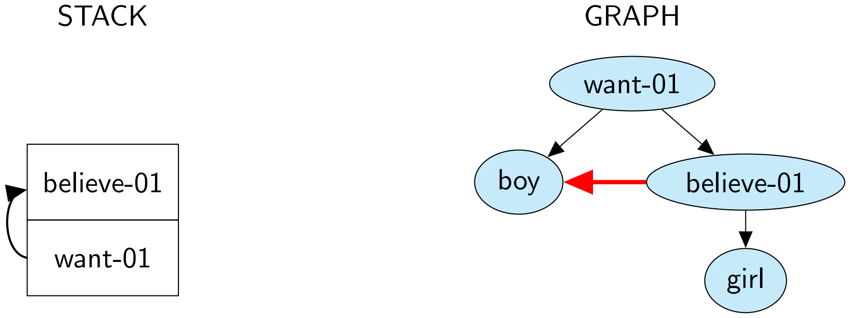
# Transition System

~~*The boy wants to believe the girl*~~



# Transition System

~~*The boy wants to believe the girl*~~



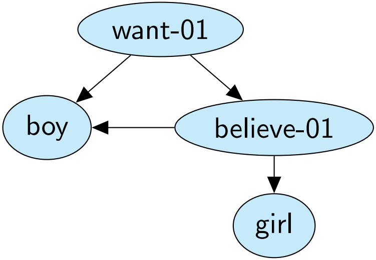
# Transition System

~~*The boy wants to believe the girl*~~

STACK

| |
|---|
| want-01 |

GRAPH

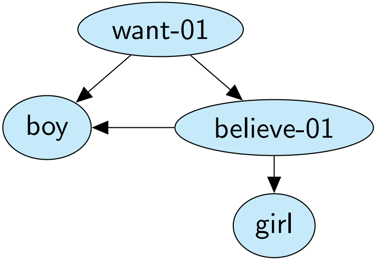


# Transition System

~~*The boy wants to believe the girl*~~

STACK

GRAPH



# Learning the Parser

- Use an "oracle" that scans a string and builds an existing graph in the training data
- Each action in this process is recorded, leading to a training set of the form $(\mathrm{context}, \mathrm{action})$
- Train a feed-forward neural network that classifies $\mathrm{context}$ into $\mathrm{action}$

We now have a full parser

# Evaluation

Recall: semantic parsing is now more like:

> *Find who did what to whom, and where, and when, and why... and using what... in a sentence, or in a paragraph, or in a whole document.*

This means we need to solve Named Entity Recognition, Semantic Role Labeling, identifying negation, Named Entity Linking...

# Evaluation

Recall: semantic parsing is now more like:

> *Find who did what to whom, and where, and when, and why... and using what... in a sentence, or in a paragraph, or in a whole document.*

This means we need to solve Named Entity Recognition, Semantic Role Labeling, identifying negation, Named Entity Linking...

NLP evaluation is uni-dimensional. For AMR parsing, we use the Smatch score

Too simplistic to unravel the behavior of a parser

Solution: a set of evaluation metrics for AMR (Damonte et al. 2017)

# Experiments

| Metric | JAMR ('14) | CAMR | JAMR ('16) | Ours (EACL '17) |
|---|---|---|---|---|
| Smatch | 58 | 63 | **67** | 64 |
| Unlabeled | 61 | **69** | **69** | **69** |
| No WSD | 58 | 64 | **68** | 65 |
| NP-only | 47 | 54 | **58** | 55 |
| Reentrancy | 38 | 41 | **42** | 41 |
| Concepts | 79 | 80 | **83** | **83** |
| Named Ent. | 75 | 75 | 79 | **83** |
| Wikification | 0 | 0 | **75** | 64 |
| Negations | 16 | 18 | 45 | **48** |
| SRL | 55 | **60** | **60** | 56 |

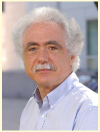
JAMR: Flanigan et al. (2014)

CAMR: Wang et al. (2015)
State of the art as of now: 72+ on Smatch (Lyu and Titov, 2018)

# Paraphrase Detection with AMR (Issa et al., 2018)

A test case for AMR parsing:

If AMR indeed canonicalizes language, then paraphrase detection, a longstanding problem in NLP is easy (check if AMRs are identical)

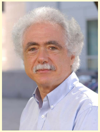


*Two sentences are paraphrases if they produce the same representation the internal formalism for meaning (Winograd, 1972)*

# Paraphrase Detection with AMR (Issa et al., 2018)

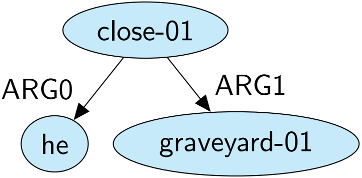
A test case for AMR parsing:

If AMR indeed canonicalizes language, then paraphrase detection, a longstanding problem in NLP is easy (check if AMRs are identical)

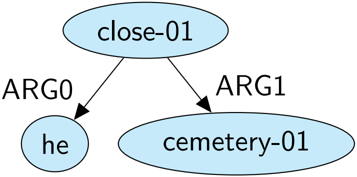


*Two sentences are paraphrases if they produce the same representation the internal formalism for meaning (Winograd, 1972)*

In practice:




He closed the graveyard




He closed the cemetery

# Paraphrase Detection with AMR

Instead, use AMR in a “soft” manner

Process to decide if two sentences are paraphrases:

- Parse the two sentences using an AMR parser
- Compute similarity between the two resulting graphs (it is a bit more complex than that)

A side note: to make sure we exploit the AMR graph and not just a syntactic structure, we also had a baseline in which dependency trees are reduced to AMR graphs

# A Bit on the Similarity Metric

- Perform Singular Value Decomposition (SVD) on a matrix $T$ of sentences by concepts such that

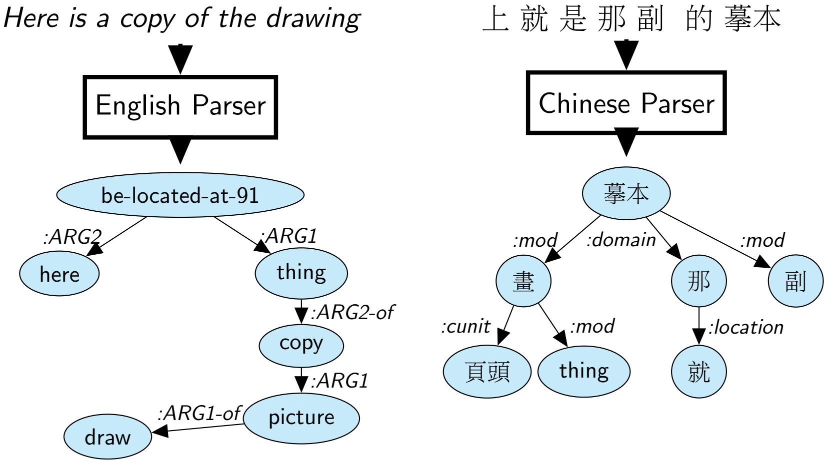
$$T_{k\ell} = \mathrm{PG}(\ell, k) \times \mathrm{count}(\ell, k)$$

  where $\mathrm{PG}$ measures the importance of the $\ell$th concept for the AMR graph for the $k$th sentence
- The output is a continuous representation for each sentence

# Paraphrase Detection with AMR: Results

| System | acc. | $F_1$ |
|---|---|---|
| Most common class | 66.5 | 79.9 |
| Mitchell and Lapata (2010) | 73.0 | 82.3 |
| Baroni and Lenci (2010) | 73.5 | 82.2 |
| Socher et al. (2011) | 76.8 | 83.6 |
| Guo and Diab (2012) | 71.5 | NR |
| Ji and Eisenstein (2013) (ind.) | 80.0 | 85.4 |
| Ji and Eisenstein (2013) (trans.) | 80.4 | 86.0 |
| Dependency (inductive) | 70.6 | 80.7 |
| Dependency (transductive) | 79.0 | 84.1 |
| AMR (inductive) | 68.7 | 80.9 |
| AMR (transductive) | 86.6 | 90.0 |

# AMR for Other Languages (Bojar, 2014; Xue et al, 2014)



# AMR is not an Interlingua

Though it tries to abstract away from the surface form, AMR is highly biased towards English (syntax)

... Still, it is perhaps one of the closest datasets we have to represent an "interlingua"

> *"A cross-linguistic comparison of English to Chinese and Czech AMRs reveals both cases where the AMRs for the language pairs align well structurally and cases of linguistic divergence. We found that the level of compatibility of AMR between English and Chinese is higher than between English and Czech."*
> **Not an Interlingua, But Close: Comparison of English AMRs to Chinese and Czech (Xue et al., 2014)**

# Two Main Questions

Given the process of annotating AMRs is expensive:

- How do we build a cross-lingual parser *leveraging* data we already have in English?
- How do we evaluate such a cross-lingual parser when we do not have gold-standard data for it?

# Rapid Prototyping of AMR Parsers

Can we rapidly develop AMR parsers for low-resource languages?

Solution 1:

> *Translate Chinese sentence to English. Parse with an English parser*

Solution 2:

> *Train a specialized Chinese parser through "annotation projection"*

# Annotation Projection for AMR Parsing

No "interlingua" data for non-English languages

Solution:

- Train an English parser
- Parse English data that has Chinese translations
- Train Chinese parser with English AMRs and Chinese text
- We now have a Chinese parser!

# Evaluating the Parser (1)

No Chinese gold-standard data

Annotating AMR data in Chinese would require linguistic expertise

Solution:

- Use professional translators to translate the English gold-standard *sentences* to Chinese
- Use that data as gold-standard

Translated AMR sentences are soon to be available from LDC (or upon request)

# Evaluating the Parser (2)

Solution (invert the process):

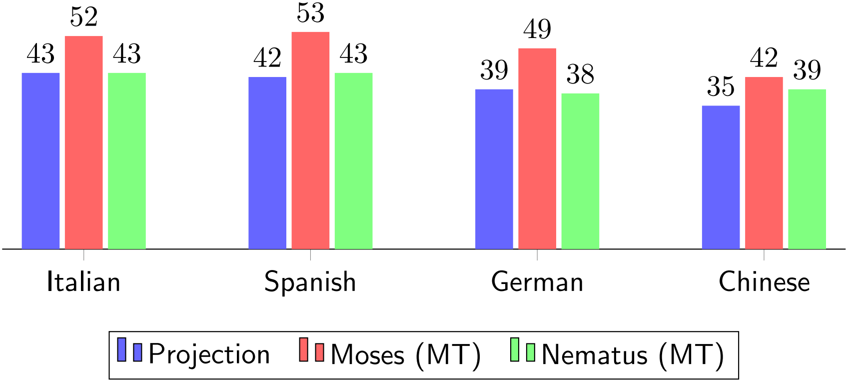
- Parse new Chinese data that has English translations
- Train an English parser with parsed-Chinese AMRs and English text
- We now have a *new* English parser!
- Test it on gold-standard data in English

Basic assumption: the quality of the original Chinese parser correlates with the quality of the new English parser on the gold-standard data

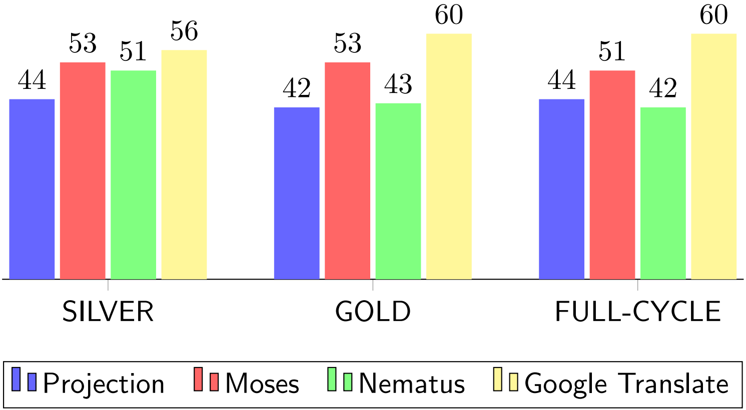
# Evaluation: Summary

- Silver: evaluate the Chinese parser on Chinese sentences from a parallel corpus to English, where the English sentences were parsed by an AMR parser
- Gold: evaluate the Chinese parser on *translation* from English of gold-standard AMR data (expensive to professionally translate)
- Full-cycle (the main "trick"): repeat the process we did for getting a Chinese parser to get an English parser, and test it on gold-standard English data (that was already available)

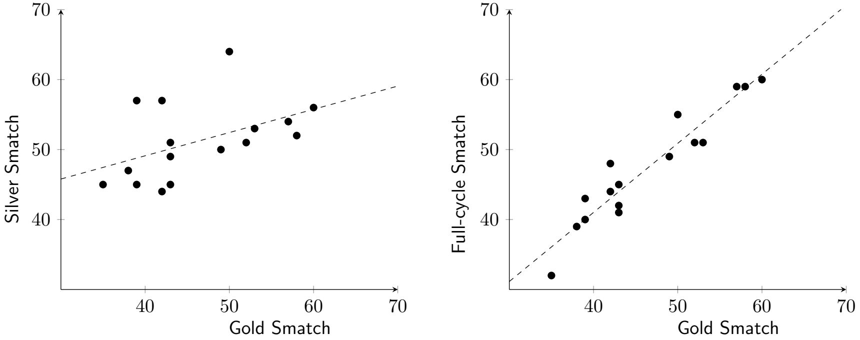
# Experiments: Gold Evaluation



# FULL-CYCLE vs SILVER vs GOLD



# Correlation between Evaluation and "Truth"



# The Inverse Problem: Generation

AMR as an intermediate representation for applications: need to be able to generate from it

- Previous work: use seq2seq model to translate an "AMR string" into a sentence (Konstas et al., 2017)
- Our research question: can we exploit better the graph structure to get a better encoding of the AMR structure (recall - reentrancies)?
- Used a model called "Graph Convolutional Neural Networks" that encodes a graph into a continuous representation

# Results for Generation

| encoder structure | BLEU |
|---|---|
| Sequence | 21.4 |
| Tree (variant 1) | 22.26 |
| Tree (variant 2) | 23.62 |
| Graph | 23.95 |

See a more detailed analysis in Damonte and Cohen (2019)

# Example of Generation

| | | |
|---|---|---|
| (1) | REF | i dont tell him but **he finds out** , |
| | SEQ | i did n't tell him but **he was out** . |
| | TREE | i do n't tell him but **found out** . |
| | GRAPH | i do n't tell him but **he found out** . |
| (2) | REF | if **you tell people** they can help you , |
| | SEQ | if **you tell him** , you can help you ! |
| | TREE | if **you tell person_name_0 you** , you can help you . |
| | GRAPH | if **you tell them** , you can help you . |
| (3) | REF | **i 'd recommend** you go and see your doctor too . |
| | SEQ | **i recommend** you go to see your doctor who is going to see your doctor . |
| | TREE | **you recommend** going to see your doctor too . |
| | GRAPH | **i recommend** you going to see your doctor too . |
| (4) | REF | (you) **tell your ex** that all communication needs to go through the lawyer . |
| | SEQ | (you) **tell** that all the communication go through lawyer . |
| | TREE | (you) **tell your ex** , tell your ex , the need for all the communication . |
| | GRAPH | (you) **tell your ex** the need to go through a lawyer . |

# Conclusion

- Language is manifested through symbols. Computational systems in general are often symbolic in nature
- Its intermediate representation, however - can be continuous or symbolic
- Symbolic: interpretable; Continuous: have a gradient
- Both have their role. Both can co-exist

*LSTMs work in practice, but can they work in theory? (Mark Steedman, 2018)*

**Code and Demos**
AMREager and multilingual parser (demo and code):
`http://cohort.inf.ed.ac.uk/amreager.html`

Generation from AMR
`http://cohort.inf.ed.ac.uk/amrgen.html`

Discourse Representation Structure parser:
`https://github.com/EdinburghNLP/EncDecDRSparsing`

**Collaborators**
Marco Damonte, Fuad Issa, Giorgio Satta

**Funding**
EU H2020 SUMMA, Bloomberg