AMR Normalization for Fairer Evaluation

Michael Wayne Goodman goodmami@uw.edu
Nanyang Technological University, Singapore

2019-09-13

Presentation agenda

Introduction: AMR, PENMAN, and Smatch

Normalization

Experiment

Conclusion

AMR

Abstract Meaning Representation

- Compact encoding of sentential semantics as a DAG
- Independent of any syntactic analyses
- Hand-annotated gold data: some free, most LDC
- The “Penn Treebank of semantics” (Banarescu et al., 2013)

Example

- “I had let my tools drop from my hands.”
  (The Little Prince Corpus, id: `lpp_1943.355`)

```
(l / let-01
   :ARG0 (i / i)
   :ARG1 (d / drop-01
            :ARG1 (t / tool
                     :poss i)
            :ARG3 (h / hand
                     :part-of i)))
```

PENMAN Notation

AMR is encoded in PENMAN notation

- `l` is node id, `let-01` is node label, `:ARG0` is edge label
- Bracketing alone forms a tree
  - Node ids allow re-entrancy
  - Inverted edges (`:part-of`) allow multiple roots

```
(l / let-01
   :ARG0 (i / i)
   :ARG1 (d / drop-01
            :ARG1 (t / tool
                     :poss i)
            :ARG3 (h / hand
                     :part-of i)))
```

Triples

PENMAN graphs translate to a conjunction of triples

```
(l / let-01                          instance(l, let-01) ^
   :ARG0 (i / i)                     ARG0(l, i) ^
                                     instance(i, i) ^
   :ARG1 (d / drop-01                ARG1(l, d) ^
                                     instance(d, drop-01)
            :ARG1 (t / tool          ARG1(d, t) ^
                                     instance(t, tool) ^
                    :poss i)         poss(t, i) ^
            :ARG3 (h / hand          ARG3(d, h) ^
                                     instance(h, hand) ^
                    :part-of i)))    part-of(h, i)
```

Back to AMR

What is AMR beyond PENMAN graphs?

- AMR is the model, PENMAN the encoding scheme
- Made up of “concepts” (nodes) and “relations” (edges)
- Verbal concepts taken from OntoNotes (Weischedel et al., 2011), others invented as necessary
- Defined by the AMR Specification[1] and annotator docs
- Mostly finite inventory of roles (except `:opN`, `:sntN`)
- Constraints (e.g., no cycles), and valid transformations (inversions, reification)

[1] https://github.com/amrisi/amr-guidelines/blob/master/amr.md

Smatch

Smatch is the prevailing evaluation metric for AMR

- For two AMR graphs, find mappings of node ids
- Choose the mapping that maximizes matching triples
- Calculate precision, recall, and F1 (the Smatch score)
- Example:

```
(s / see-01                    (s / see-01
   :ARG0 (g / girl)               :ARG0 (g / girl)
   :ARG1 (d / dog                 :ARG1 (c / cat))
            :quant 2))
```

Left: 7 triples, Right: 6, Matching: 5

Precision: 5/7 = 0.71; Recall: 5/6 = 0.83; F1 = 0.77

What's the Problem?

AMR has alternations that are *meaning-equivalent* according to the specification

- Some idiosyncratic role inversions, e.g.:
  - `:mod` <-> `:domain`
  - `:consist-of` <-> `:consist-of-of`
- Edge reifications, e.g.:

```
(a / ... :cause (b / ...)
```

  ...can reify `:cause` to...

```
(a / ...
   :ARG1-of (c / cause-01
               :ARG0 (c / ...)))
```

- These result in differences in the triples, and thus the Smatch score

What's the Problem?

There is no partial credit for almost-correct triples

```
   Gold              Hyp1              Hyp2
(c / chapter     (c / chapter     (c / chapter)
   :mod 7)          :quant 5)

   CAMR              JAMR             AMREager
(c / chapter     (c / chapter     (c / chapter
   :quant 7)        :li 7)           :op1 7)
```

- Getting the role wrong (CAMR, JAMR, AMREager) gets the same score as getting both the role and value wrong (Hyp1)
- Omitting the relation altogether (Hyp2) yields a higher score than having an incorrect relation.

What's the Problem?

Some "equivalent" alternations are invalid graphs

```
   Gold                Bad
(c / chapter      (c / chapter
   :mod 7)           :domain-of 5)
```

- If `:domain-of` is inverted, then `5` must be a node id, but it is a constant.

Presentation agenda

Introduction: AMR, PENMAN, and Smatch

Normalization

Experiment

Conclusion

Normalization

Question: Can we address these problems in evaluation by normalizing the triples?

Meaning-preserving normalization:

- Canonical Role Inversion
- Edge Reification

Meaning-augmenting normalization:

- Attribute Reification
- Structure Preservation

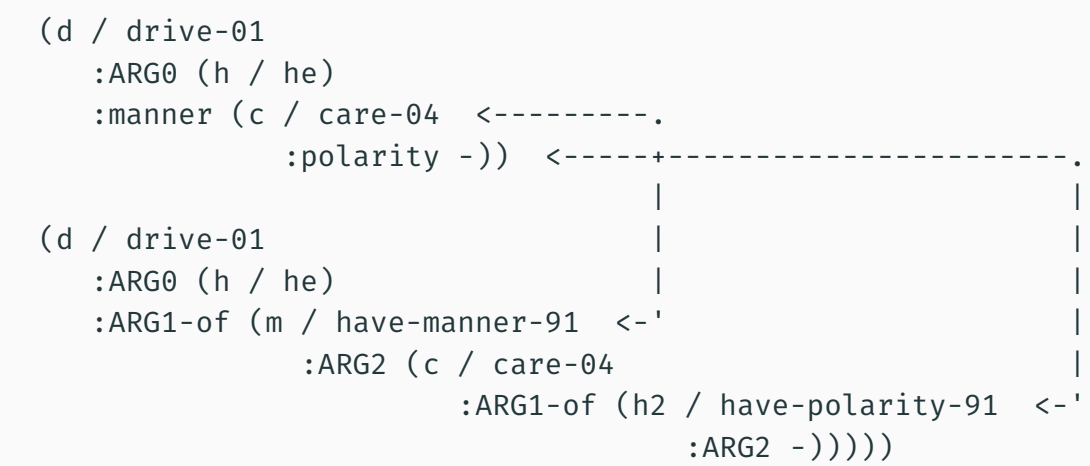
Canonical Role Inversion

Replace non-canonical role with canonical ones

- `:mod-of` -> `:domain`
- `:domain-of` -> `:mod`
- `:consist` -> `:consist-of-of`
- etc.
- (Also useful for general data cleaning)

Edge Reification

Always reify edges



Attribute Reification

Make constants into node labels

```
(c / chapter          (c / chapter
   :mod 7)     -->       :mod (_ / 7))
```

Structure Preservation

Make the tree structure evident in the triples
(using the Little Prince example, adding **TOP** relations)

```
(l / let-01
   :ARG0 (i / i :TOP l)
   :ARG1 (d / drop-01 :TOP l
            :ARG1 (t / tool :TOP d
                     :poss i)
            :ARG3 (h / hand :TOP h
                     :part-of i)))
```

Presentation agenda

Introduction: AMR, PENMAN, and Smatch

Normalization

Experiment

Conclusion

Experiment Setup

Test the relative effects of normalization on parsing evaluation for multiple parsers

- Use the Little Prince corpus with gold annotations
- Parse using JAMR (Flanigan et al., 2016)
- Parse using CAMR (Wang et al., 2016)
- Parse using AMREager (Damonte et al., 2017)
- Normalize each of the four above (various configurations)
- Compare:
  - Gold-orig $\times$ { JAMR-orig, CAMR-orig, AMREager-orig }
  - Gold-norm $\times$ { JAMR-norm, CAMR-norm, AMREager-norm }

Results

	Normalization				Score		
System	I	A	R	S	P	R	F
JAMR					0.60	0.56	0.58
	✓				0.60	0.55	0.57
		✓			0.61	0.56	0.58
			✓		0.63	0.57	0.60
				✓	0.59	0.55	0.57
CAMR					0.67	0.56	0.61
	✓				0.67	0.56	0.61
		✓			0.67	0.55	0.60
			✓		**0.70**	0.57	**0.63**
				✓	0.68	**0.58**	**0.63**
AMREager					0.57	0.52	0.55
	✓				0.57	0.52	0.55
		✓			0.57	0.53	0.55
			✓		0.61	0.57	0.59
				✓	0.59	0.54	0.56

Results

	Normalization				Score		
System	I	A	R	S	P	R	F
					0.60	0.56	0.58
	✓		✓		0.63	0.57	0.60
JAMR		✓	✓		0.64	0.57	0.60
	✓	✓	✓		0.64	0.57	0.60
	✓	✓	✓	✓	0.61	0.56	0.59
					0.67	0.56	0.61
	✓		✓		0.69	0.57	**0.63**
CAMR		✓	✓		**0.70**	0.56	0.62
	✓	✓	✓		**0.70**	0.56	0.62
	✓	✓	✓	✓	**0.70**	**0.58**	**0.63**
					0.57	0.52	0.55
	✓		✓		0.61	0.57	0.59
AMREager		✓	✓		0.60	**0.58**	0.59
	✓	✓	✓		0.60	**0.58**	0.59
	✓	✓	✓	✓	0.61	0.57	0.59

Presentation agenda

Introduction: AMR, PENMAN, and Smatch

Normalization

Experiment

Conclusion

Discussion

- Normalization slightly increases scores on this dataset
  - mainly due to partial credit
- Sometimes it does worse
  - making available previously ignored triples
  - more triples -> larger denominator in Smatch
- Effects on a single system are unimportant
- Rather, relative effects for multiple systems is interesting
- Although, relative effects on this experiment are slight
  - Role inversion harmed JAMR but not others
  - AMREager improves compared to others
- Next step: try on other corpora (Bio-AMR, LDC, ...)

Discussion

- Normalization is not promoted as a postprocessing step (in general)
- Rather as preprocessing to evaluation
- Thus it allows parser developers to take risks
- Although reduced variation may benefit sequence-based models
- Similar procedures possibly useful for non-AMR representations (e.g., EDS, DMRS)

Thanks

Thank you!

Software Available:

- Normalization
  `https://github.com/goodmami/norman`
- PENMAN graph library
  `https://github.com/goodmami/penman`

References i

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Damonte, Shay B. Cohen, and Giorgio Satta. 2017. An incremental parser for abstract meaning representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 536–546, Valencia, Spain. Association for Computational Linguistics.

References ii

Jeffrey Flanigan, Chris Dyer, Noah A Smith, and Jaime Carbonell. 2016. CMU at SemEval-2016 task 8: Graph-based AMR parsing with infinite ramp loss. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1202–1206.

Chuan Wang, Sameer Pradhan, Xiaoman Pan, Heng Ji, and Nianwen Xue. 2016. CAMR at semeval-2016 task 8: An extended transition-based AMR parser. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1173–1178, San Diego, California. Association for Computational Linguistics.

References iii

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. OntoNotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.