SemEval-2016 Task 8: Meaning Representation Parsing

Jonathan May
Information Sciences Institute
University of Southern California
jonmay@isi.edu

Abstract

In this report we summarize the results of the SemEval 2016 Task 8: Meaning Representation Parsing. Participants were asked to generate Abstract Meaning Representation (AMR) (Banarescu et al., 2013) graphs for a set of English sentences in the news and discussion forum domains. Eleven sites submitted valid systems. The availability of state-of-the-art baseline systems was a key factor in lowering the bar to entry; many submissions relied on CAMR (Wang et al., 2015b; Wang et al., 2015a) as a baseline system and added extensions to it to improve scores. The evaluation set was quite difficult to parse, particularly due to creative approaches to word representation in the web forum portion. The top scoring systems scored 0.62 F1 according to the Smatch (Cai and Knight, 2013) evaluation heuristic. We show some sample sentences along with a comparison of system parses and perform quantitative ablative studies.

1 Introduction

Abstract Meaning Representation (AMR) is a compact, readable, whole-sentence semantic annotation (Banarescu et al., 2013). It includes entity identification and typing, PropBank semantic roles (Kingsbury and Palmer, 2002), individual entities playing multiple roles, as well as treatments of modality, negation, etc. AMR abstracts in numerous ways, e.g., by assigning the same conceptual structure to *fear* (v), *fear* (n), and *afraid* (adj). Figure 1 gives an example.

With the recent public release of a sizeable corpus of English/AMR pairs (LDC2014T12), there has been substantial interest in creating parsers to recover this formalism from plain text. Several parsers were released in the past couple of years (Flanigan et al., 2014; Wang et al., 2015b; Werling et al., 2015; Wang et al., 2015a; Artzi et al., 2015; Pust et al., 2015). This body of work constitutes many diverse and interesting scientific contributions, but it is difficult to adequately determine which parser is numerically superior, due to heterogeneous evaluation decisions and the lack of a controlled blind evaluation. The purpose of this task, therefore, was to provide a competitive environment in which to determine one winner and award a trophy to said winner.

```
(f / fear-01
  :polarity "-"
  :ARG0 ( s / soldier )
  :ARG1 ( d / die-01
          :ARG1 s ))
```

The soldier was not afraid of dying.
The soldier was not afraid to die.
The soldier did not fear death.

Figure 1: An Abstract Meaning Representation (AMR) with several English renderings. Example borrowed from Pust et al. (2015).

2 Training Data

LDC released a new corpus of AMRs (LDC2015E86), created as part of the DARPA DEFT program, in August of 2015. The new corpus, which was annotated by teams at SDL, LDC, and the University of Colorado, and supervised by Ulf Hermjakob at USC/ISI, is an extension of pre-

Proceedings of SemEval-2016, pages 988–997,
San Diego, California, June 16-17, 2016.

vious releases (LDC2014E41 and LDC2014T12). It contains 19,572 sentences (subsuming, in turn, the 18,779 AMRs from LDC2014E41 and the 13,051 AMRs from LDC2014T12), partitioned into training, development, and test splits, from a variety of news and discussion forum sources.

The AMRs in this corpus have changed somewhat from their counterparts in LDC2014E41, consistent with the evolution of the AMR standard. They now contain wikification via the `:wiki` attribute, they use new (as of July 2015) PropBank framesets that are unified across parts of speech, they have been deepened in a number of ways, and various corrections have been applied.

3 Other Resources

We made the following resources available to participants:

- The aforementioned AMR corpus (LDC2015E86), which included automatically generated AMR-English alignments over tokenized sentences.
- The tokenizer (from Ulf Hermjakob) used to produce the tokenized sentences in the training corpus.
- The AMR specification, used by annotators in producing the AMRs.[1]
- A deterministic, input-agnostic trivial baseline 'parser' courtesy of Ulf Hermjakob.
- The JAMR parser (Flanigan et al., 2014) as a strong baseline. We provided setup scripts to process the released training data but otherwise provided the parser as is.
- An unsupervised AMR-to-English aligner (Pourdamghani et al., 2014).
- The same Smatch (Cai and Knight, 2013) scoring script used in the evaluation.
- A Python AMR manipulation library, from Nathan Schneider.

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

Description	Code	Sents
Agence France-Presse news	afp	23
Associated Press news	apw	52
BOLT discussion forum	bolt	257
New York Times news	nyt	471
Weblog	web1	232
Xinhua news	xin	18

Table 1: Split by domain of evaluation data.

System	Precision	Recall	F1
Brandeis/cx/RPI	0.57	0.67	**0.62**
CLIP@UMD	0.40	0.48	0.44
CMU	0.53	0.61	0.56
CU-NLP	0.53	0.58	0.56
DynamicPower	0.34	0.40	0.37
ICL-HD	0.54	0.67	0.60
M2L	0.54	0.66	0.60
RIGA	0.57	0.68	**0.62**
Meaning Factory	0.46	0.48	0.47
UCL+Sheffield	0.56	0.65	0.60
UofR	0.49	0.51	0.50
Determ. baseline	0.23	0.26	0.24
JAMR baseline	0.43	0.58	0.50

Table 2: Main Results: Mean of five runs of Smatch 2.0.2 with five restarts per run is shown; Standard deviation of F1 was about 0.0002 per system. See Table 5 for slight correction.

4 Evaluation Data

For the specific purposes of this task, DEFT commissioned and LDC released an additional set of English sentences along with AMR annotations[2] that had not been previously seen. This blind evaluation set consists of 1,053 sentences in a roughly 50/50 discussion forum/newswire split. The distribution of sentences by source is shown in Table 1.

5 Task Definition

We deliberately chose a single, simple task. Participants were given English sentences and had to return an AMR graph (henceforth, 'an AMR') for each sentence. AMRs were scored against a gold AMR with the Smatch heuristic F1-derived tool and

[2] LDC2015R33 for just the sentences, and LDC2015R36 for sentences with their AMRs.

metric. Smatch (Cai and Knight, 2013) is calculated by matching instance, attribute, and relation tuples to a reference AMR (See Section 7.2). Since variable naming need not be globally consistent, heuristic hill-climbing is done to search for the best match in sub-exponential time. A trophy was given to the team with the highest Smatch score under consistent heuristic conditions.[3]

6 Participants and Results

11 teams participated in the task.[4] Their systems and scores are shown in Table 2. Below are brief descriptions of each of the various systems, based on summaries provided by the system authors. Readers are encouraged to consult individual system description papers for more details.

6.1 CAMR-based systems

A number of teams made use of the CAMR system from Wang et al. (2015a). These systems proved among the highest-scoring and had little variance from each other in terms of system score.

6.1.1 Brandeis / cemantix.org / RPI (Wang et al., 2016)

This team, the originators of CAMR, started with their existing AMR parser and experimented with three sets of new features: 1) rich named entities, 2) a verbalization list, and 3) semantic role labels. They also used the RPI Wikifier to wikify the concepts in the AMR graph.

6.1.2 ICL-HD (Brandt et al., 2016)

This team attempted to improve AMR parsing by exploiting preposition semantic role labeling information retrieved from a multi-layer feed-forward neural network. Prepositional semantics was included as features into CAMR. The inclusion of the features modified the behavior of CAMR when creating meaning representations triggered by prepositional semantics.

6.1.3 RIGA (Barzdins and Gosko, 2016)

Besides developing a novel character-level neural translation based AMR parser, this team also extended the Smatch scoring tool with the C6.0 rule-based classifier to produce a human-readable report on the error patterns frequency observed in the scored AMR graphs. They improved CAMR by adding to it a manually crafted wrapper fixing the identified CAMR parser errors. A small further gain was achieved by combining the neural and CAMR+wrapper parsers in an ensemble.

6.1.4 M2L (Puzikov et al., 2016)

This team attempted to improve upon CAMR by using a feed-forward neural network classification algorithm. They also experimented with various ways of enriching CAMR's feature set. Unlike ICL-HD and RIGA they were not able to benefit from feed-forward neural networks, but were able to benefit from feature enhancements.

6.2 Other Approaches

The other teams either improved upon their existing AMR parsers, converted existing semantic parsing tools and pipelines into AMR, or constructed AMR parsers from scratch with novel techniques.

6.2.1 CLIP@UMD (Rao et al., 2016)

This team developed a novel technique for AMR parsing that uses the Learning to Search (L2S) algorithm. They decomposed the AMR prediction problem into three problems—that of predicting the concepts, predicting the root, and predicting the relations between the predicted concepts. Using L2S allowed them to model the learning of concepts and relations in a unified framework which aims to minimize the loss over the entire predicted structure, as opposed to minimizing the loss over concepts and relations in two separate stages.

6.2.2 CMU (Flanigan et al., 2016)

This team's entry is a set of improvements to JAMR (Flanigan et al., 2014). The improvements are: a novel training loss function for structured prediction, new sources for concepts, improved features, and improvements to the rule-based aligner

[3] Four random restarts.

[4] A twelfth team, CUCLEAR, participated but produced invalid AMRs that could not be scored.

	Full AMR	Instances	Attributes	Relations
Brandeis/cx/RPI	**0.6195**	0.7433	0.6043	0.5494
CLIP@UMD	0.4370	0.6097	0.4013	0.3712
CMU	0.5636	0.7288	0.5433	0.4960
CU-NLP	0.5566	0.7338	0.2837	0.5338
DynamicPower	0.3706	0.4088	0.3560	0.3955
ICL-HD	0.6005	0.7161	0.5361	**0.5517**
M2L	0.5952	0.7245	0.5099	0.5378
RIGA	**0.6196**	0.7298	**0.6288**	0.5507
Meaning Factory	0.4702	0.5596	0.5400	0.4120
UCL+Sheffield	0.5983	**0.7545**	0.5914	0.5155
UofR	0.4985	0.7054	0.5586	0.4203
Determ. baseline	0.2440	0.2269	0.0014	0.3556
JAMR baseline	0.4965	0.6970	0.3089	0.4562

Table 3: Ablation of instances, attributes, and relations.

in Flanigan et al. (2014). The overall architecture of the system and the decoding algorithms for concept identification and relation identification are unchanged from Flanigan et al. (2014).

6.2.3 Dynamic Power (Butler, 2016)

No use was made of the training data provided by the task. Instead, existing components were combined to form a pipeline able to take raw sentences as input and output meaning representations. The components are a part-of-speech tagger and parser trained on the Penn Parsed Corpus of Modern British English to produce syntactic parse trees, a semantic role labeler, and a named entity recognizer to supplement obtained parse trees with word sense, functional and named entity information. This information is passed into an adapted Tarskian satisfaction relation for a Dynamic Semantics that is used to transform a syntactic parse into a predicate logic based meaning representation, followed by conversion to the required Penman notation.

6.2.4 The Meaning Factory (Bjerva et al., 2016)

This team employed an existing open-domain semantic parser, Boxer (Curran et al., 2007), which produces semantic representations based on Discourse Representation Theory. As the meaning representations produced by Boxer are considerably different from AMRs, the team used a hybrid conversion method to map Boxer's output to AMRs. This process involves lexical adaptation, a conversion from DRT-representations to AMR, as well as post-processing of the output.

6.2.5 UCL+Sheffield (Goodman et al., 2016)

This team developed a novel transition-based parsing algorithm using exact imitation learning, in which the parser learns a statistical model by imitating the actions of an expert on the training data. They used the imitation learning algorithm DAGGER to improve the performance, and applied an alpha-bound as a simple noise reduction technique.

6.2.6 UofR (Peng and Gildea, 2016)

This team applied a synchronous-graph-grammar-based approach for string-to-AMR parsing. They applied Markov Chain Monte Carlo (MCMC) algorithms to learn Synchronous Hyperedge Replacement Grammar (SHRG) rules from a forest that represents likely derivations consistent with a fixed string-to-graph alignment (extracted using an automatic aligner). They make an analogy of string-to-AMR parsing to the task of phrase-based machine translation and came up with an efficient algorithm to learn graph grammars from string-graph pairs. They proposed an effective approximation strategy to resolve the complexity issue of graph compositions. Then they used the

Earley algorithm with cube-pruning for AMR parsing given new sentences and the learned SHRG.

6.2.7 CU-NLP (Foland and Martin, 2016)

This parser does not rely on a syntactic pre-parse, or heavily engineered features, and uses five recurrent neural networks as the key architectural components for estimating AMR graph structure.

7 Result Ablations

We conduct several ablations to attempt to empirically determine what aspects of the AMR parsing task were more or less difficult for the various systems.

7.1 Impact of Wikification

The AMR standard has recently been expanded to include wikification and the data used in this task reflected that expansion. Since this is a rather recent change to the standard and requires some kind of global external knowledge of, at a minimum, Wikipedia's ontology, we suspected performance on `:wiki` attributes would suffer. To measure the effect of wikification, we performed two ablation experiments, the results of which are in Figure 2. In the first ("no wiki"), we removed `:wiki` attributes and their values from reference and system sets before scoring. In the second ("bad wiki"), we replaced the value of all `:wiki` attributes with a dummy entry to artificially create systems that did not get any wikification correct.

The "no wiki" ablations show that the inclusion of wikification into the AMR standard had a very small impact on overall system scores. No system's score changed by more than 0.01 when wikification was removed, indicating that systems appear to wikify about as well as they handle the rest of AMR's attributes. The "bad wiki" ablations show performance drop when wikification is corrupted of around 0.02 to 0.03 for six of the systems, and a negligible performance drop for the remaining systems. This result indicates that the systems with a performance drop are doing a fairly good job at wikification.

7.2 Performance on different parts of the AMR

In this set of ablations we examine systems' relative performance on correctly identifying *instances*, *attributes*, and *relations* of the AMRs. Instances are the labeled nodes of the AMR. In the example AMR of Figure 1, the instances are `fear-01`, `soldier`, and `die-01`. To match an instance one must simply match the instance's label.[5]

Attributes are labeled string properties of nodes. In the example AMR, there is a **polarity** attribute attached to the `fear-01` instance with a value of "-." There is also an implicit attribute of "TOP" attached to the root node of the graph, with the node's instance as the attribute value. To match an attribute one must match the attribute's label and value, and the attribute's instance must be aligned with the corresponding instance in the reference graph.

Relations are labeled edges between two instances. In the example AMR, the relations `(f, s, ARG0)`, `(f, d, ARG1)`, and `(d, s, ARG1)` exist. To match a relation, the labeled edge between two nodes of the hypothesis must match the label of the edge between the correspondingly aligned nodes of the reference graph.

It should not be surprising that systems tend to perform best at instance matching and worst at relation matching. Note, however, that the best performing systems on instances and relations were not the overall best performing systems. Ablation results can be seen in Table 3.

7.3 Performance on different data sources

As discussed in Section 8, less formal sentences, sentences with misspellings, and sentences with non-standard representations of meaning were the hardest to parse. We ablate the results by domain of origin in Table 4. While the strongest-performing systems tended to perform best across ablations, we note that the machine-translated and informal corpora were overall the hardest sections to parse.

[5] That is, correctly generating a multi-set of instances with the same labels as those in the reference is sufficient for a perfect score. The task of correctly generating instances that are in proper relation to each other is handled by the relations.

8 Qualitative Comparison

In this section we examine some of the sentences that the systems found particularly easy or difficult to parse.

8.1 Easiest Sentences

The easiest sentence to parse in the eval corpus was the sentence "I was tempted."[6]

It has a gold AMR of:

```
(t / tempt-01
      :ARG1 (i / i))
```

The mean score for this sentence was 0.977. All submitted systems except one parsed it perfectly.

Another sentence that was quite easy to parse was the sentence "David Cameron is the prime minister of the United Kingdom."[7] Two systems parsed it perfectly and a third omitted wikification but was otherwise perfect. Figure 3 shows a detailed comparison of each system's performance on the sentence. In general we see that shorter sentences from the familiar and formal news domain are parsed best by the submitted systems.

8.2 Hardest Sentences

Five sentences were unable to be parsed in any way by any system.[8] They are shown below, along with their AMRs:

```
E-mail: mward(at)statesman.com.
(e / email-address-entity
  :value "mward@statesman.com")
x
(s / string-entity :value "x")
*sigh*
(s / sigh-01)
Yes_it_is.
(y / yes)
M E D I A A D V I S O R Y
(a / advise-01
      :ARG1 (m / media))
```

Data noise was another confounding factor. In the next example,[9] which had an average score of 0.17, parsers were confused both by the misspelling ("lie" for "like") and by the quoted title, which all systems except UCL+Sheffield, tried to parse for meaning.

```
Why not a title lie "School Officials Screw over Rape Victim?"
(t / title-01 :polarity -
   :ARG1-of (r / resemble-01
         :ARG2 (t2 / title-01 :wiki "A_Rape_on_Campus"
               :name (n2 / name :op1 "School" :op2 "Officials"
                                :op3 "Screw" :op4 "Over"
                                :op5 "Rape" :op6 "Victim")))
   :ARG1-of (c / cause-01
         :ARG0 (a / amr-unknown)))
```

We note that all of these difficult sentences are not conceptually hard for humans to parse. Humans have far less difficulty in resolving errors or processing non-standard tokenization than do computers.

9 There Can Be Only One?

We intended to award a single trophy to the single best system, according to the narrow evaluation conditions (balanced F1 via Smatch 2.0.2 with 5 restarts, to two decimal places). However, the top two systems, Brandeis/cemantix.org/RPI and RIGA, scored identically according to that metric. Hoping to elicit some consistent difference between the systems, we ran Smatch with 20 restarts, looked at four decimal places, and re-ran five times. Each system scored a mean of 0.6214 with standard deviation of 0.00013. We thus capitulate in the face of overwhelming statistics and award the inaugural trophy to both teams, equally.[10]

10 Conclusion

The results of this competition and the interest in participation in it demonstrate that AMR parsing is a difficult, competitive task. The large number of systems using released code lowered the bar to entry significantly but may have led to a narrowing of diversity in approaches. Low-level irregularities such as creative tokenization and misspellings befuddled the systems. We hope to conduct another AMR parsing competition in the future, in the biomedical domain, and also conduct a generation competition.

[6] nyt_eng_20130426_0143.23.

[7] bolt-eng-DF-200-192446-3811676_0094.5.

[8] nyt_eng_20131029_0042.18, web1-eng-DF-225-195996-5376307_0002.3, web1-eng-DF-233-195474-1207335_0002.1, web1-eng-DF-233-195474-1207335_0010.2, and web1-eng-DF-183-195729-5441907_0001_3.

[9] bolt-eng-DF-170-181122-8787556_0049_6.

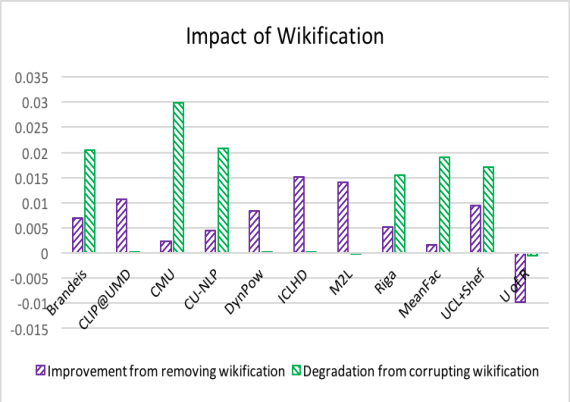
[10] Funding for trophies graciously provided by the Jelinek-Mercer Institute for Semantic Translation.

	afp	apw	bolt	nyt	web1	xin	all
# Sentences	23	52	257	471	232	18	1053
Brandeis/cx/RPI	0.6287	**0.6829**	**0.6052**	0.6285	0.5933	0.5703	**0.6195**
CLIP@UMD	0.4334	0.4723	0.4211	0.446	0.4223	0.3879	0.437
CMU	0.6303	0.6747	0.5954	0.5354	0.572	0.57	0.5636
CU-NLP	0.5602	0.5949	0.5621	0.5536	0.5496	0.5479	0.5566
DynamicPower	0.3249	0.3847	0.3765	0.3715	0.3702	0.3366	0.3706
ICL-HD	0.6136	0.6572	0.581	0.6111	0.573	0.5615	0.6005
M2L	0.5987	0.6409	0.5788	0.603	0.5789	0.5456	0.5952
RIGA	**0.6611**	0.6715	0.6004	**0.6292**	**0.598**	0.5603	**0.6196**
The Meaning Factory	0.5019	0.5542	0.4679	0.4611	0.4566	0.5281	0.4702
UCL+Sheffield	0.611	0.6672	0.5942	0.596	0.5891	**0.5936**	0.5983
UofR	0.5248	0.5475	0.5022	0.4938	0.4908	0.5039	0.4985
mean	0.56552	0.60757	0.54637	0.54832	0.53715	0.53178	0.54926
Determ. baseline	0.2791	0.2799	0.2095	0.2589	0.2279	0.2562	0.2440
JAMR baseline	0.5422	0.5714	0.5302	0.4722	0.5027	0.5049	0.4965

Table 4: Ablation of Smatch scores by text source. AP wire ('apw') data was the easiest to parse, Web forum ('web1') and Xinhua ('xin') were the hardest.

	With wiki	No wiki	Bad wiki
Brandeis/cx/RPI	0.6195	0.6264	0.5991
CLIP@UMD	0.4370	0.4477	0.4369
CMU	0.5636	0.5660	0.5337
CU-NLP	0.5566	0.5611	0.5358
DynamicPower	0.3706	0.3790	0.3704
ICL-HD	0.6005	0.6157	0.6004
M2L	0.5952	0.6092	0.5954
RIGA	0.6196	0.6247	0.6042
Meaning Factory	0.4702	0.4718	0.4512
UCL+Sheffield	0.5983	0.6077	0.5812
UofR	0.4985	0.4887	0.4990

(a) Comparison of regular systems ('With wiki'), systems and references with all wikification removed ('No wiki'), and systems with wikification corrupted ('Bad wiki').



(b) Removing wikification from hypothesis and reference raises scores by less than 0.01 Smatch in eight systems. Corrupting wikification in the hypothesis lowers scores by 0.015 or more in six systems.

Figure 2: Ablations of `:wiki` attribute.

```
(h / have-org-role-91
   :ARG0 (p / person
      :wiki "David_Cameron"
      :name (n / name :op1 "David"
                      :op2 "Cameron"))
   :ARG1 (c / country
      :wiki "United_Kingdom"
      :name (n2 / name :op1 "United"
                       :op2 "Kingdom"))
   :ARG2 (m / minister
      :mod (p2 / prime)))
```

(a) Gold / CMU / RIGA (1.0)
(M2L = 0.95; missing wiki)

```
(x6 / have-org-role-91
  :ARG0 (x1 / person
     :wiki "David_Cameron"
     :name (n / name :op1 "David"
                     :op2 "Cameron"))
  :ARG1 (x9 / country
     :wiki "United_Kingdom"
     :name (n1 / name :op1 "United"
                      :op2 "Kingdom"))
  :ARG2 (m / minister)
  :mod (x5 / prime))
```

(b) Brandeis / cemantix.org / RPI (0.95)
(prime not mod of minister)

```
(c3 / minister
   :mod (c1 / prime)
   :ARG0 (c2 / have-org-role-91
           :ARG2 c3)
   :mod (c0 / person
           :name (n0 / name :op1 "David"
                            :op2 "Cameron"))
   :ARG1 (c4 / country
           :name (n1 / name :op1 "United"
                            :op2 "Kingdom")))
```

(c) CLIP@UMD (0.77)
(minister is root; wrong ARG0 direction, no wiki)

```
( EVENT-4 / copula
   :ARG0 ( PERSON-david_cameron / PERSON
           :name ( n-1 / name :op1 "david"
                              :op2 "cameron"))
   :ARG1 ( ENTITY-1 / minister
           :mod ( ATTRIB-3 / prime)
           :poss ( LOCATION-united_kingdom / LOCATION
           :name ( n-2 / name :op1 "united"
                              :op2 "kingdom"))))
```

(d) DynamicPower (.68)
(wrong frame, capitalization, no wiki)

```
(x1 / person
  :name (n / name :op1 "David"
                  :op2 "Cameron")
  :ARG0-of (x6 / have-org-role-91
       :ARG2 (m / minister
            :mod (x5 / prime))
       :ARG1 (x9 / country
            :name (n1 / name :op1 "United"
                             :op2 "Kingdom"))))
```

(e) ICL-HD (.89)
(no wiki, wrong root)

```
(c / country
  :ARG0-of (h / have-org-role-91
            :ARG1 (p / person
                     :name (n / name :op1 "David"
                                     :op2 "Cameron")
                     :wiki "David_Cameron")
            :ARG2 (m / minister
                     :mod (p2 / prime)))
  :name (n2 / name :op1 "United"
                   :op2 "Kingdom")
  :wiki "United_Kingdom")
```

(f) UofR (.85)
(ARG swap, wrong root)

```
(e1 / thing
  :name (n5 / name :op1 "prime"
                   :op2 "minister")
  :wiki "Prime_Minister_of_the_United_Kingdom"
  :domain (x1 / person
            :name (n3 / name :op1 "david"
                             :op2 "cameron")
            :wiki "David_Cameron" )
  :poss (x2 / country
          :name (n7 / name :op1 "united"
                           :op2 "kingdom")
          :wiki "United_Kingdom" ))
```

(g) The Meaning Factory (.59)
(wrong frame, capitalization, errant wiki)

```
( n / person
  :wiki "-"
  :name ( m / name :op1 "David"
                   :op2 "Cameron")
  :UNKNOWN ( f / minister
             :mod ( e / prime )
             :mod ( l / country
                    :wiki "United_Kingdom"
                    :name ( k / name :op1 "United"
                                     :op2 "Kingdom"))))
```

(h) UCL+Sheffield (.74)
(no frame, wiki missing, wrong relations)

```
(n1.2 / have-org-role-91
  :ARG0 (n1.1000 / person
     :wiki "David_Cameron"
     :name (n1.0 / name :op1 david
                        :op2 cameron  ) )
  :ARG2 (n1.5 / minister
     :mod (n1.4 / prime  )
     :ARG2-of (n1.6 / have-org-role-91
           :ARG1 (n1.1008 / country
              :wiki "United_Kingdom"
              :name (n1.8 / name :op1 united
                                 :op2 kingdom )))))
```

(i) CU-NLP (.71)
(capitalization, extra role, attachment)

Figure 3: A comparison of parser performance on the sentence “David Cameron is the Prime Minister of the United Kingdom.”

System	Prec.	Rec.	F1	Δ F1
Brandeis/cx/RPI	0.57	0.68	0.62	0.00
CLIPUMD	0.40	0.49	0.44	0.00
CMU	0.53	0.61	0.56	0.00
CU-NLP	0.58	0.63	0.61	0.05
DynamicPower	0.34	0.40	0.37	0.00
ICLHD	0.54	0.67	0.60	0.00
M2L	0.54	0.66	0.60	0.00
Riga	0.57	0.68	0.62	0.00
TheMeaningFactory	0.46	0.49	0.47	0.00
UCL+Sheffield	0.56	0.65	0.60	0.00
UofR	0.49	0.51	0.50	0.00

Table 5: Corected Results: Double quote independence restored. Mean of five runs of corrected Smatch 2.0.2 with five restarts per run is shown; The CU-NLP team improves; no other system changed within .02 F1.

Acknowledgments

Many thanks to the AMR creation team: Kira Griffitt, Ulf Hermjakob, Kevin Knight, and Martha Palmer. Thanks also to the SemEval organizers: Steven Bethard, Daniel Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch. We also gratefully acknowledge the participating teams' efforts. This work was sponsored by DARPA DEFT (FA8750-13-2-0045).

Erratum

Subsequent to the camera-ready deadline for this document it was determined that changes between Smatch versions 2.0 and 2.0.2 led to quoted and non-quoted items in AMRs being judged non-identical; in previous versions they were judged identical and the change in 2.0.2 was not intended to alter this behavior. Table 5 shows the revised scores. CU-NLP saw a significant increase in overall F1 as a result of the fix, while the other systems were not affected. Thanks to William Foland for identifying the bug. This version also fixes some attribution errors.

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal, September. Association for Computational Linguistics.

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, August. Association for Computational Linguistics.

Guntis Barzdins and Didzis Gosko. 2016. RIGA at SemEval-2016 task 8: Impact of Smatch extensions and character-level neural translation on AMR parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Johannes Bjerva, Johan Bos, and Hessel Haagsma. 2016. The Meaning Factory at SemEval-2016 task 8: Producing AMRs with Boxer. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Lauritz Brandt, David Grimm, Mengfei Zhou, and Yannick Versley. 2016. ICL-HD at SemEval-2016 task 8: Meaning representation parsing - augmenting AMR parsing with a preposition semantic role labeling neural network. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Alastair Butler. 2016. DynamicPower at SemEval-2016 task 8: Processing syntactic parse trees with a dynamic semantics core. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria, August. Association for Computational Linguistics.

James Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic, June. Association for Computational Linguistics.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1426–1436, Baltimore, Maryland, June. Association for Computational Linguistics.

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)*, San Diego, California, June. Association for Computational Linguistics.

William Foland and James H. Martin. 2016. CU-NLP at SemEval-2016 task 8: AMR parsing using LSTM-based recurrent neural networks. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. UCL+Sheffield at SemEval-2016 task 8: Imitation learning for AMR parsing with an alpha-bound. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From Treebank to Propbank. In *In Language Resources and Evaluation*.

Xiaochang Peng and Daniel Gildea. 2016. UofR at SemEval-2016 task 8: Learning synchronous hyperedge replacement grammar for AMR parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Nima Pourdamghani, Yang Gao, Ulf Hermjakob, and Kevin Knight. 2014. Aligning English strings with Abstract Meaning Representation graphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 425–429, Doha, Qatar, October. Association for Computational Linguistics.

Michael Pust, Ulf Hermjakob, Kevin Knight, Daniel Marcu, and Jonathan May. 2015. Parsing English into Abstract Meaning Representation using syntax-based machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1143–1154, Lisbon, Portugal, September. Association for Computational Linguistics.

Yevgeniy Puzikov, Daisuke Kawahara, and Sadao Kurohashi. 2016. M2L at SemEval-2016 task 8: AMR parsing with neural networks. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Sudha Rao, Yogarshi Vyas, Hal Daumé III, and Philip Resnik. 2016. Clip@umd at SemEval-2016 task 8: Parser for Abstract Meaning Representation using learning to search. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015a. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 857–862, Beijing, China, July. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015b. A transition-based algorithm for AMR parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375, Denver, Colorado, May–June. Association for Computational Linguistics.

Chuan Wang, Nianwen Xue, Sameer Pradhan, Xiaoman Pan, and Heng Ji. 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)*, San Diego, California, June. Association for Computational Linguistics.

Keenon Werling, Gabor Angeli, and Christopher D. Manning. 2015. Robust subgraph generation improves Abstract Meaning Representation parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 982–991, Beijing, China, July. Association for Computational Linguistics.