MEANING REPRESENTATION PARSING

Avikalp Kumar Gupta & Anshul Goyal

SemEval 2016: Task 8 (http://alt.qcri.org/semeval2016/task8/)

AMR (Abstract Meaning Representation)

- Who is doing what to whom in a sentence
  Semantic parsing is necessary for AMR
- Different from a parse tree, it is abstract

AMR (Abstract Meaning Representation)

eg: "The London emergency services said that altogether 11 people had been sent to hospital for treatment due to minor wounds."

AMR:

```
(s / say-01
      :ARG0 (s2 / service
            :mod (e / emergency)
            :location (c / city :wiki ``London''
                  :name (n / name :op1 ``London'')))
      :ARG1 (s3 / send-01
            :ARG1 (p / person :quant 11)
            :ARG2 (h / hospital)
            :mod (a / altogether)
            :purpose (t / treat-03
                  :ARG1 p
                  :ARG2 (w / wound-01
                        :ARG1 p
                        :mod (m / minor)))))
```

AMR (Abstract Meaning Representation)

A single AMR can be expressed in various ways in English:

eg.

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

Can be expressed in the following ways:

- The boy wants the girls to believe him
- The boy desires the girl to believe him.
- The boy desires to be believed by the girl.
- The boy has a desire to be believed by the girl.
- The boy’s desire is for the girl to believe him.
- The boy is desirous of the girl believing him.

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

AMR

- AMR is represented using a rooted, directed acyclic graph with labels on edges (relations) and leaves (concepts)
- Hence:
  1. Task1: Find concepts
  2. Task2: Find relations

EARLIER WORK (JAMR)

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A Smith. A discriminative graph-based parser for the abstract meaning representation. 2014.

Concept Identification

- A sequence of words is fed into a tool called *clex*, which on the basis of some rules, returns a concept fragment from the training data.

Relation Identification

- We build a connected graph using the concepts as nodes
- The following constraints are applied on the graph:
  1. Preserving (concept)
  2. Simple
  3. Connected
  4. Deterministic
- Given constraints, we seek the maximum scoring subgraph based on the feature set in appendix 1

Results

concepts	Train P	Train R	Train F_1	Test P	Test R	Test F_1
gold	.85	.95	.90	.76	.84	.80
automatic	.69	.78	.73	.52	.66	.58

Table 5: Parser performance.

Implementation of JAMR parser

The London emergency services said that altogether 11 people had been sent to hospital for treatment due to minor wounds."

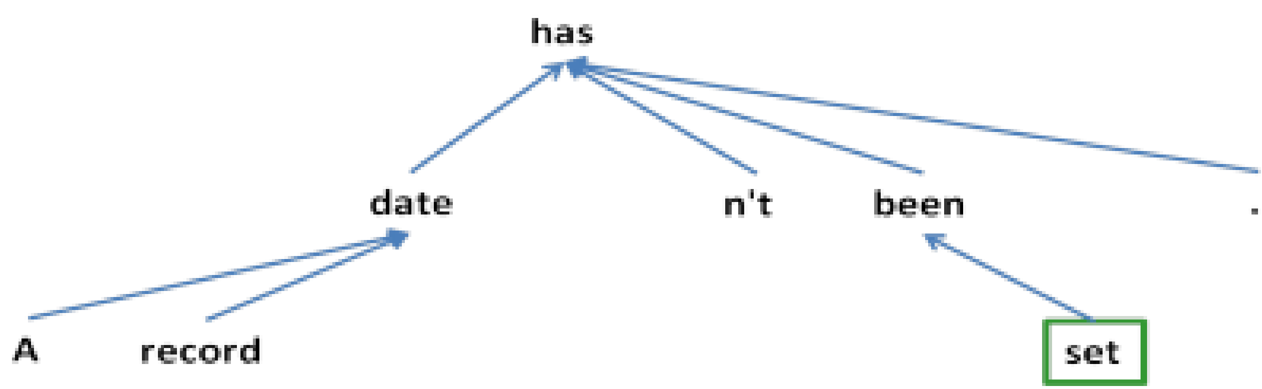
```
(s2 / say-01
      :ARG0 (s3 / service
            :mod (e / emergency
                  :mod (c / country
                        :name (n / name
                              :op1 "London"
                              :op2 "The"))))
      :ARG1 (s / send-01
            :ARG2 (h / hospital)
            :mod (a / altogether)
            :purpose (t / treat-03
                  :ARG1 11
                  :ARG2 (w / wound
                        :mod (m / minor))))))

```

Challenges

- Capturing the semantics of the sentence
- Making the algorithm compatible all languages (since the baseline used a lot of English-specific rules)

Approach

- We are using Deep Bidirectional LSTM for semantic role labeling as proposed by Wei Xu and Jie Zhou in the below sited paper
- Using this we get a semantically parsed graph, but with unlabelled edges.



Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. 2015.

Approach

- Now to label these edges, initially, we plan to
  - use the *clex* tool over each entity to recognize the concept it belongs to
  - and then use the relation identification function over these concepts, giving excessive weights to the edges found in the semantic parsed graph

Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. 2015.

QUESTIONS?

APPENDIX I

Name	Description
Label	For each $\ell \in L_E$, 1 if the edge has that label
Self edge	1 if the edge is between two nodes in the same fragment
Tail fragment root	1 if the edge's tail is the root of its graph fragment
Head fragment root	1 if the edge's head is the root of its graph fragment
Path	Dependency edge labels and parts of speech on the shortest syntactic path between any two words in the two spans
Distance	Number of tokens (plus one) between the two concepts' spans (zero if the same)
Distance indicators	A feature for each distance value, that is 1 if the spans are of that distance
Log distance	Logarithm of the distance feature plus one.
Bias	1 for any edge.

Table 1: Features used in relation identification. In addition to the features above, the following conjunctions are used (Tail and Head concepts are elements of L_V): Tail concept $\wedge$ Label, Head concept $\wedge$ Label, Path $\wedge$ Label, Path $\wedge$ Head concept, Path $\wedge$ Tail concept, Path $\wedge$ Head concept $\wedge$ Label, Path $\wedge$ Tail concept $\wedge$ Label, Path $\wedge$ Head word, Path $\wedge$ Tail word, Path $\wedge$ Head word $\wedge$ Label, Path $\wedge$ Tail word $\wedge$ Label, Distance $\wedge$ Label, Distance $\wedge$ Path, and Distance $\wedge$ Path $\wedge$ Label. To conjoin the distance feature with anything else, we multiply by the distance.