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

<table>
<thead>
<tr>
<th>concepts</th>
<th>Train</th>
<th></th>
<th>Test</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>$F_1$</td>
<td>P</td>
</tr>
<tr>
<td>gold</td>
<td>.85</td>
<td>.95</td>
<td>.90</td>
<td>.76</td>
</tr>
<tr>
<td>automatic</td>
<td>.69</td>
<td>.78</td>
<td>.73</td>
<td>.52</td>
</tr>
</tbody>
</table>

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.”
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.

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

QUESTIONS?
## Appendix I

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

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 $\land$ Label, Head concept $\land$ Label, Path $\land$ Label, Path $\land$ Head concept, Path $\land$ Tail concept, Path $\land$ Head concept $\land$ Label, Path $\land$ Tail concept $\land$ Label, Path $\land$ Head word, Path $\land$ Tail word, Path $\land$ Head word $\land$ Label, Path $\land$ Tail word $\land$ Label, Distance $\land$ Label, Distance $\land$ Path, and Distance $\land$ Path $\land$ Label. To conjoin the distance feature with anything else, we multiply by the distance.