Robust Subgraph Generation Improves Abstract Meaning Representation Parsing [Werling et al.]

2015/8/24
ACL読み会@すずかけ台

東京大学 鶴岡研
D1 江里口瑛子
Abstract Meaning Representation; AMR [Banarescu et al., 2013]

“He gleefully ran to his dog Rover.”

AMR Parsing

- 自然言語の意味をグラフで記述
- based on neo-Davidsonian semantics
- AMRの構造
  - node: concept
  - edge: relation
- 機械翻訳, QAなどへ応用
**AMR parsing process**

**STEP 1:** concept identification (NER++)

**STEP 2:** relation identification (SRL++)

*Text*

**AMR Parsing**
Figure 2: A graphical explanation of our method. We represent the derivation process for a sentence. First, the tokens in the sentence are labeled with derivation actions, then these actions are used to generate AMR subgraphs, which are then stitched together to form a coherent whole.

The primary contribution of this paper is a novel approach to the NER++ task, illustrated in Figure 2. We notice that the subgraphs aligned to lexical items can often be generated from a small set of generative actions which generalize across tokens. For example, most verbs generate an AMR node corresponding to the verb sense of the appropriate PropBank frame—e.g., `run` generates `run-01` in Figure 1. This allows us to frame the NER++ task as the task of classifying one of a small number of actions for each token, rather than choosing a specific AMR subgraph for every token in the sentence.

Our approach to the end-to-end AMR parsing task is therefore as follows: we define an action space for generating AMR concepts, and create a classifier for classifying lexical items into one of these actions (Section 4). This classifier is trained from automatically generated alignments between the gold AMR trees and their associated sentences (Section 5), using an objective which favors alignment mistakes which are least harmful to the NER++ component. Finally, the concept subgraphs are combined into a coherent AMR parse using the maximum spanning connected subgraph algorithm of Flanigan et al. (2014).

We show that our approach provides a large boost to recall over previous approaches, and that end-to-end performance is improved from 59 to 62 smatch (an F1 measure of correct AMR arcs; see Cai and Knight (2013)) when incorporated into the SRL++ parser of Flanigan et al. (2014). When evaluating the performance of our action classifier in isolation, we obtain an action classification accuracy of 84.1%.

The AMR Formalism

AMR is a language for expressing semantics as a rooted, directed, and potentially cyclic graph, where nodes represent concepts and arcs are relationships between concepts. AMR is based on neo-Davidsonian semantics, (Davidson, 1967; Parsons, 1990). The nodes (concepts) in an AMR graph do not have to be explicitly grounded in the source sentence, and while such an alignment is often generated to train AMR parsers, it is not provided in the training corpora. The semantics of nodes can represent lexical items (e.g., `dog`), sense tagged lexical items (e.g., `run-01`), type markers (e.g., `date-entity`), and a host of other phenomena.

The edges (relationships) in AMR describe one of a number of semantic relationships between concepts. The most salient of these is semantic role labels, such as the `ARG0` and `destination` arcs in Figure 2. However, often these arcs define other phenomena.

STEP1: concept identification (NER++)

STEP2: relation identification (SRL++)

Text

AMR Parsing
AMR parsing process

**STEP 1:**
concept identification (NER++)

**STEP 2:**
relation identification (SRL++)

Text

He gleefully ran to his dog Rover.

NER++ 1/2
IDENT LEMMA VERB NONE NONE IDENT NAME

NER++ 2/2
he glee run-01 dog

SRL++

:ARG0 :mod :dest :name :op1

AMR Parsing

構築したsubgraphに対して言語制約を満たしながら全域化
Figure 1: A graphical explanation of our method. We represent the derivation process for the sentence “He gleefully ran to his dog Rover.” First, the tokens in the sentence are labeled with derivation actions, then these actions are used to generate AMR subgraphs, which are then stitched together to form a coherent whole.

The primary contribution of this paper is a novel approach to the NER++ task, illustrated in Figure 2. We notice that the subgraphs aligned to lexical items can often be generated from a small set of generative actions which generalize across tokens. For example, most verbs generate an AMR node corresponding to the verb sense of the appropriate PropBank frame – e.g., “run” generates “run-01” in Figure 1. This allows us to frame the NER++ task as the task of classifying one of a small number of actions for each token, rather than choosing a specific AMR subgraph for every token in the sentence.

Our approach to the end-to-end AMR parsing task is therefore as follows: we define an action space for generating AMR concepts, and create a classifier for classifying lexical items into one of these actions (Section 4). This classifier is trained from automatically generated alignments between the gold AMR trees and their associated sentences (Section 5), using an objective which favors alignment mistakes which are least harmful to the NER++ component. Finally, the concept subgraphs are combined into a coherent AMR parse using the maximum spanning connected subgraph algorithm of Flanigan et al. (2014).

We show that our approach provides a large boost to recall over previous approaches, and that end-to-end performance is improved from 59 to 62 smatch (an F1 measure of correct AMR arcs; see Cai and Knight (2013)) when incorporated into the SRL++ parser of Flanigan et al. (2014). When evaluating the performance of our action classifier in isolation, we obtain an action classification accuracy of 84.1%.

2 The AMR Formalism

AMR is a language for expressing semantics as a rooted, directed, and potentially cyclic graph, where nodes represent concepts and arcs are relationships between concepts. AMR is based on neo-Davidsonian semantics, (Davidson, 1967; Parsons, 1990). The nodes (concepts) in an AMR graph do not have to be explicitly grounded in the source sentence, and while such an alignment is often generated to train AMR parsers, it is not provided in the training corpora. The semantics of nodes can represent lexical items (e.g., “dog”), sense tagged lexical items (e.g., “run-01”), type markers (e.g., “date-entity”), and a host of other phenomena.

The edges (relationships) in AMR describe one of a number of semantic relationships between concepts. The most salient of these is semantic role labels, such as the ARG0 and destination arcs in Figure 2. However, often these arcs define other phenomena.
AMR parsing process

STEP1: concept identification

Text

STEP2: relation identification

AMR subgraphを作成
AMR subgraph

• トークンとAMRのnodeは非一対一対応
• 1つのトークン、または複数のトークンから、nodeを複数含むAMRのsubgraphが生成
• subgraph=1つのconceptの表現

Figure 3: AMR representation of the word sailor, which is notable for breaking the word up into a self-contained multi-node unit unpacking the derivational morphology of the word.

3. Task Decomposition

To the best of our knowledge, the JAMR parser is the only published end-to-end AMR parser at the time of publication. An important insight in JAMR is that AMR parsing can be broken into two distinct tasks: (1) NER++ (concept identification): the task of interpreting what entities are being referred to in the text, realized by generating the best AMR subgraphs for a given set of tokens, and (2) SRL++ (relation identification): the task of discovering what relationships exist between entities, realized by taking the disjoint subgraphs generated by NER++ and creating a fully-connected graph. We describe both tasks in more detail...
A Novel NER+++ Method

- 現状:
  - AMRの訓練データは少
  - トークンとsubgraphをマッピング対応
- 提案手法; action-based NER+++  
  - subgraphを生成する9つのactionを定義
  - 各トークンのactionを決定する分類器を作成
<table>
<thead>
<tr>
<th>ACTION</th>
<th>内容</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTITY</td>
<td>nodeのタイトルがトークンと一致</td>
</tr>
<tr>
<td>NONE</td>
<td>トークンはnodeに割り当てない</td>
</tr>
<tr>
<td>VERB</td>
<td>ProbBankから最類似動詞を得る</td>
</tr>
<tr>
<td>VALUE</td>
<td>数値変換</td>
</tr>
<tr>
<td>LEMMA</td>
<td>Lemmatize</td>
</tr>
<tr>
<td>NAME</td>
<td>name nodeを付与</td>
</tr>
<tr>
<td>PERSON</td>
<td>person node, name nodeを追加</td>
</tr>
<tr>
<td>DATE</td>
<td>SUTimeの出力を利用[Chang and Manning, 2012]</td>
</tr>
<tr>
<td></td>
<td>date-entityに変換</td>
</tr>
<tr>
<td>DICT</td>
<td>他のactionへのback off</td>
</tr>
</tbody>
</table>
Action Reliability

- 予測したactionが誤りである場合もある
- 各actionの信頼度合い(reliability)を開発データを用いて調査

<table>
<thead>
<tr>
<th>完全に確実 ($p = 1$):</th>
<th>PERSON</th>
<th>IDENTITY</th>
<th>NAME</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>罕見の誤り ($p \approx 0.9$):</td>
<td>LEMMA</td>
<td>VALUE</td>
<td>VERB</td>
<td>DATE</td>
</tr>
<tr>
<td>誤差が期待される ($p &lt; 0.7$):</td>
<td>DICT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Action Reliability

- 開発データ中のaction分布
- 全体の74%(non-DICT)がreliability≧0.9

<table>
<thead>
<tr>
<th>Action</th>
<th># Tokens</th>
<th>% Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>41538</td>
<td>36.2</td>
</tr>
<tr>
<td>DICT</td>
<td>30027</td>
<td>26.1</td>
</tr>
<tr>
<td>IDENTITY</td>
<td>19034</td>
<td>16.6</td>
</tr>
<tr>
<td>VERB</td>
<td>11739</td>
<td>10.2</td>
</tr>
<tr>
<td>LEMMA</td>
<td>5029</td>
<td>4.5</td>
</tr>
<tr>
<td>NAME</td>
<td>4537</td>
<td>3.9</td>
</tr>
<tr>
<td>DATE</td>
<td>1418</td>
<td>1.1</td>
</tr>
<tr>
<td>PERSON</td>
<td>1336</td>
<td>1.1</td>
</tr>
<tr>
<td>VALUE</td>
<td>122</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Training the action classifier

- 訓練データは組(graph, sentence)
  - AMRのnode($n_i$)は、トークン($s_j$)と必ず対応とする
  - 全actionに対して、$s_j$から$n_i$が正しく生成されるか否かを調べる + reliabilityで判断
  - e.g. “running” -> DICT / VERB -> VERB
- 最大エントロピー分類器で学習
  - 入力(i, S); S=トークン列
  - 18素性
### Automatic Alignment of Training Data

\[
\text{max}_{Q} \sum_{i,j} Q_{i,j} \left[ \log(\text{REL}(V_{i,j})) + \alpha \varepsilon_{i,j} \right] \quad (1)
\]

\[
\text{s.t.} \quad \sum_{j} Q_{i,j} = 1 \quad \forall i \quad (2)
\]

\[
Q_{k,j} + Q_{l,j} \leq 1 \quad \forall k, l, j; n_k \leftrightarrow n_l \quad (3)
\]

- \text{actionの対数尤度最大化}
- \( Q = \{0, 1\}^{|N| \times |S|}; |N|: \text{AMRグラフ内node数}, |S|: \text{文内トークン数} \)
- \( V = T^{|N| \times |S|} \); \( V_{i,j} = \text{action} \)
- \( \varepsilon: \text{Jaro-Winkler類似度}, \alpha: \text{ハイパーパラメータ (}= 0.8) \)
実験

・データセット
  ・ニュース記事(LDC2014T12, LDC2013E117)

・end-to-endのJAMR parserを用いて比較
  ・JAMR parser (default) [Flanigan et al., 2014]
  ・JAMR SRL++ with proposed NER++ approach

・評価指標
  ・smatch =s(emantic) match [Cai and Knight, 2013]
  ・(parent, edge, child)の三つ組で評価
実験結果

- Recall値を大きく改善
- 提案NER++で、より良いsubgraphを構築
- SRL++自体の改善にまではいたっていない

<table>
<thead>
<tr>
<th>Dataset</th>
<th>System</th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014T12</td>
<td>JAMR</td>
<td>67.1</td>
<td>53.2</td>
<td>59.3</td>
</tr>
<tr>
<td></td>
<td>Our System</td>
<td>66.6</td>
<td>58.3</td>
<td>62.2</td>
</tr>
<tr>
<td>2013E117</td>
<td>JAMR</td>
<td>66.9</td>
<td>52.9</td>
<td>59.1</td>
</tr>
<tr>
<td></td>
<td>Our System</td>
<td>65.9</td>
<td>59.0</td>
<td>62.3</td>
</tr>
</tbody>
</table>
まとめ

• AMR Parsingの性能改善
• action-based NER++の提案
  • 9 actionsを定義
  • 分類器による判定→subgraphを生成
• 実験: Recall値、F1値の改善 (=NER++の改善)
• 今後の課題: actionのカバレッジや、lemmatizerの改善