

構文情報を用いた Abstract Meaning Representation の 高精度アラインメント

扇本岳大

鶴岡慶雅

東京大学 工学系研究科 東京大学 工学系研究科

{ougimoto,tsuruoka}@logos.t.u-tokyo.ac.jp

1 はじめに

自然言語処理において、意味解析は固有表現抽出、意味役割付与、共参照解析、固有表現抽出など多くの分野にまたがっている。しかしながら、それぞれの解析に関する研究は独立して行われており、別々の分野として発展してきた経緯がある。そのため、分野ごとにデータを学習する手法も、評価に用いるデータも、結果を評価する手法も異なっている。したがって、分野間の関連性や相互作用を捉えた解析を行うことは難しくなっており、複数の分野で解析されてきた意味表現を一括して表現することができる形式が長い間望まれてきた。

それを受けて新しく考案された意味表現方法が抽象的意味表現 (Abstract Meaning Representation, AMR) [2] である。AMR は文中から意味を持った概念のみを取り出し、それをグラフ構造で表現したものである。このグラフは根を持ちサイクル構造を内部に含まない有向グラフとなっている。例として、“The boy wants for the girl to believe him.” に対する AMR を図 1 と図 2 にて示す。図 1 は AMR を視覚的にわかりやすいグラフ構造によって図示したもの、図 2 はデータ中での AMR を示したもので両者が表現していることは同じである。

AMR にはその中の各部分が文中のどの単語に対応しているかという対応関係がデータ中で示されておらず、明らかではない。こうした対応関係を特定するのがアラインメント処理である。文から AMR を生成するパーザーのほとんどはアラインメント結果を利用して機械学習を行っており、精度が高いアラインメント処理は AMR の応用上重要になっている。アラインメント処理の例として “The SMH and Jennings were then sued over 3 articles published in the lead-up to the 2000 Olympics .” の文をアラインメントした結果を図 3 にて示す。対象となる原文が上部に、その AMR

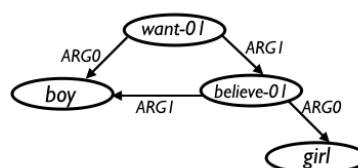


図 1: AMR の例 (グラフによる表現)

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

図 2: AMR の例 (データ中での表現)

が下部に示されており、対応した部分同士が同じ色によって示されている。

現在、最も使用されているアラインメントのアルゴリズムとして JAMR aligner [3] がある。このアルゴリズムは基本的にノードと単語の文字的な対応を調べるのみであるため、同じ文中に同じ単語が複数回出現するなど複雑な場合にはあまりうまく対応できない。そこで、本研究ではそのような場合にも対応するため、文の構文情報を使用したアルゴリズムを提案する。実験において、JAMR aligner との性能比較を行ったところ、提案手法の精度の方が 7.6% 上回り、その有効性が検証された。

2 関連研究

2.1 JAMR aligner

JAMR aligner はデータを用いて機械学習を行うのではなく、人手で定めた規則を用いて処理を行うルー

The SMH and Jennings were then sued over 3 articles published in the lead-up to the 2000 Olympics .

```
( s / sue-01
:ARG1 ( a / and
:op1 ( c / company : name ( n / name : op1 "The" : op2 "SMH" )
:op2 ( p / person : name ( n2 / name : op1 "Jennings" ) )
:ARG2 ( a2 / article : quant 3
:ARG1-of ( p2 / publish-01
:time ( l / lead-05
:ARG2 ( g / game : name ( n3 / name : op1 "Olympics" )
:time ( d / date-entity : year 2000 ) ) ) )
:mod ( t / then ) )
```

図 3: AMR のアラインメント処理の例

ルベースのアラインメント処理を行う。また、複数の AMR のノードに対して複数の単語を割り当てる多対多のアラインメントがなされる。エッジはアラインメントの対象外としており、ノードに対してのみアラインメントを行う。F 値で 90% の精度が報告されている。

手法としては基本的には文中の単語と AMR のノードのラベルとの文字上のマッチングを取り、一致したものを対応付けていくというのが基本的な方針になる。しかしながら、それだけでは図 3 の “company”、 “person”、 “name”、 “game”、 “date-entity” といった文中には出現しないノードに対しては対応できなくなる。こうしたノードに対応するため、JAMR aligner では人手で定めたルールを 14 用意し、その各ルールによって様々な場合に対応できるようにしている。そしてこれらのルールを人手で決めた順番に適用していき、すべてのルールを適用し終わった段階でアラインメントは終了し、その時までに対応づけられたすべてのノードと単語の対応を出力する。

2.2 JAMR aligner の問題点

前項のように、JAMR aligner は基本的にノードのラベルと単語が一致しているかを調べることを繰り返していく手法となっている。そのため、JAMR aligner には以下に示すような問題点が存在する。

同じ単語

ノードのラベルとの一致を調べる際、同じ単語が文中に複数回出現しているとの単語とマッチングすべきかを判断できず、マッチングがランダムに行われ、それに失敗した場合間違えることになる。同じ文中に同じ単語が複数回出現する場合は多く、大きな誤りの原因となっている。

文中に出現しないノード

AMR では文中に出現した単語が同じ意味を持つ

別のノードとして表現されることがよくある。例えば、権利という意味を表す “right” という単語は AMR では “entitle-01” というノードで表現される。このように文字を用いた対応が取れないものに関してはアラインメントを行うことができない。

3 提案手法

前章で示したような問題点を克服するため、構文情報 (依存構造) を用いた新たなアルゴリズムを提案する。このアルゴリズムは大きく分けて以下のようなステップに分けることができる。

文字列マッチング

JAMR aligner と同様に、AMR 中の各ノードと対象の文中の各単語の間で文字的な対応関係があるかどうかを調べる。複数の候補が一致する場合、ランダムに一つの候補を選ぶ JAMR aligner と異なり、マッチングが取れた組み合わせをすべて記憶しておく。

構文情報を用いた候補の絞り込み

文字列マッチングで AMR のノードに対して対応する単語が複数考えられる場合や一つも対応づけが取れなかった場合などに、構文情報を用いることで正しい対応づけを予測する。

特殊なノードの処理

AMR には特殊な AMR 独自ノード (図 3 の “date-entity” など) がしばしば含まれている。前のステップでは処理しきれないこうしたノードに対応する単語を決める。

3.1 文字列マッチング

AMR のノードには文中の単語と文字的に明らかな対応関係あるものが多い。そのため、最初に AMR 中の各ノードと文中の各単語の文字的な対応関係があるかどうかをすべての組み合わせについて調べる。その際、文中の日付を表す表現や数字を表す表現をすべて算用数字に直すことや、否定を表す表現には専用のラベルを付与するなどの前処理を行う。また、4 文字以上の部分文字列が一致した場合も対応関係があるものとみなす。文字列マッチングの結果は以下のように分類することができる。

1. AMR 中のノードと文中の単語との間に一対一の対応関係がある場合
2. 一つの AMR のノードに複数の文中の単語が対応している場合
3. 複数の AMR のノードが同一の単語に対応している場合
4. 複数の AMR のノードに対応する単語がそれぞれ複数あり、それらの間に重複がある場合
5. AMR 中のノードと文字的に対応している文中の単語が存在しない場合

3.2 構文情報を用いた候補の絞り込み

3.1 節の文字列マッチングにおいて複数の対応づけの候補が考えられる場合、構文情報を用いて候補の絞り込みを行う。まず、3.1 節の分類の 1 の場合については各ノードに対応する単語の候補は一つしか存在しないため、絞り込みを行わず対応づけを確定する。その後、2 から 5 の場合について絞り込みを行う。

絞り込みには構文情報として対象とする文の依存構造木を使用する。まず、対象とする AMR のノードについて AMR のグラフ上での親ノードの集合、子のノードの集合、兄弟のノードの集合、それらの和集合を調べる。同様に対応関係を調べたい文中の単語について、依存構造木上での親ノードの集合、子のノードの集合、兄弟のノードの集合、それらの和集合を調べる。そして、AMR のグラフと依存構造木の対応した集合 (親なら親、和集合なら和集合) についてそれまでのアラインメント処理において既に確定した対応関係の個数を数える。この個数をスコアとみなし、すべてのノードと単語の候補についてスコアを算出し、最もスコアが高くなるような対応づけを求める。最もスコアが高くなるような対応づけが複数あった場合については、それぞれについて AMR のノードと対応する単語についてそれぞれ文字の出現頻度をベクトル化し、そのコサイン類似度を代わりにスコアとして用いる。

誤りが伝播する影響を少しでも抑えるため、候補が少なく絞り込みが容易なものから順に対応づけを確定していく。最初に、3.1 節の分類のうち、2、3、4 に分類されるものについて対応づけを行う。その際、候補となる AMR のノードの種類が少ない順に、その種類の数が同じ場合は候補となる単語の種類が少ない順に対応づけを確定していく。これらの場合ではそれぞれのノードについて、3.1 節でマッチングがとれた単語

	精度
JAMR aligner	0.913
提案手法	0.989

表 1: 実験結果

語についてのみスコアを計算する。その上で、異なるノードは異なる単語に対応づけるという条件のもと、スコアの合計が最も高くなるような対応づけの組み合わせを求めては確定させる。また、候補となる単語の数よりも候補となる AMR のノードの数が多いときには対応づけで余ったノードができる。そうしたノードは、3.1 節の分類のうち 5 に分類されるものとして、後にそれらと一緒に単語との対応づけを行う。

最後に、3.1 節の分類のうち 5 に含まれるマッチングのうち、後のステップで処理する特殊なノードでないものについて対応づけを行う。これらについては 3.1 節のマッチングで対応する単語が見つけれなかったため、文中のすべての単語について (それまでのアラインメント処理で既に対応づけられている単語を含めて) スコアを計算し、最もスコアが高かった単語にマッチングする。

3.3 特殊なノードの処理

AMR には、それが日付を表しているということを示す “date-entity” のノード、時間を表していることを示す “temporal-quantity” のノード、金銭を表していることを示す “monetary-quantity” のノードなど独自のノードが多く存在しており、こうしたノードに対しては特殊なアラインメント処理を行う必要がある。そのため、こうしたノードに対する処理を最後に行う。JAMR aligner で定められていた 14 種類の人手で定めたルールの中からはこうしたノードに対応するためのものであり、本提案手法においても基本的にそれらのルールをそのまま使用している。ただし、JAMR aligner とは異なり、この段階で特殊なノード以外のノードへのアラインメント処理は全て終わっていることからより柔軟な処理が可能となっておりルール数は少なく、対応できる範囲はより広がっている。

4 実験

今回提案したアルゴリズムの有効性を検証するために実験を行った。今回は LDC2014T12 のコーパス

の proxy データのうち開発データとして用いていない文の中から 50 文 (合計 900 ノード) を選び、その文に対して手でアノテーションを行って正解データを作成した。既存のアラインメント手法である JAMR aligner と今回提案したアルゴリズムの両者を用いてアラインメントを行い、それぞれについて手動で行ったアノテーションに対する正解率をそれぞれ求めた。なお、依存構造木を得るために SyntaxNet [1] を使用した。また、ノードと単語を比較する際には Stanford CoreNLP toolkit [4] の lemmatizer を使用して動詞や形容詞などの語形変化を直した上で比較を行った。

結果を表 1 に示す。JAMR aligner の精度が 91.3%、提案したアルゴリズムの精度が 98.9% となっており、提案したアルゴリズムの方が 7.6% 精度が上回っている。また、JAMR aligner が正解で提案手法が誤りであった場合は評価データ中に一つもなく、提案手法が既存手法よりもうまくアラインメントを行えていることが示された。

5 おわりに

5.1 まとめ

本稿では、文の意味についての多くの側面を一括して表現できることから近年注目を集めている AMR のアラインメント処理について取り上げた。そして、その代表的な既存手法である JAMR aligner について、文中に同じ単語が複数回出現する場合や、文中の単語と対応する AMR のノードが文字的に大きく異なっている場合に正しくアラインメントが行えないという問題点を明らかにした。さらに、そうした問題点を解決するべく、アラインメントに文の構文情報を用いるアルゴリズムを提案した。実験によって検証を行った結果、提案したアルゴリズムは JAMR aligner の精度を 7.6% 上回り、その有効性が検証された。

5.2 今後の課題

今後の課題として、第一にエッジへの対応が挙げられる。現在のアルゴリズムではノードに対するアラインメントしか行えず、エッジに対しては行えない。今回提案したアルゴリズムの「構文情報を用いた候補の絞り込み」とほぼ同様の処理を行うことでエッジに対しても高い精度でアラインメントができると期待される。ただし、ほぼすべての場合に対応する単語が存在するノードと異なり、エッジには対応する単語がない

場合が多いため、どのようにそれを判別するかが課題となる。

また、今回提案したアルゴリズムによるアラインメント結果を用いて、文から AMR を生成するパーザを設計することも今後の課題として挙げられる。パーザの性能はアラインメントの性能に大きく依存するため、今回アラインメントの精度が向上した分、パーザの精度も向上することが考えられる。

謝辞

本研究は、JST、CREST の支援を受けたものである。

参考文献

- [1] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *ACL*, 2016.
- [2] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for semantic banking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.
- [3] Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the abstract meaning representation. In *ACL*, 2014.
- [4] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014.