

セマンティックパーシングのためのcgc2lambdaに関する実験と考察

理情4年 05181019 鈴木陽大

1 選んだ論文とその理由

"cgc2lambda: A Compositional Semantics System (出典 : ACL Anthology 2016) (Koji Mlneshima et al)"

の論文を読んだ。この論文を読んだ動機は2つある。第一に、学科のカリキュラムで論理学に触れたことがあり、論理学や言語処理系に多少の知識があること。第二に、文から「意味」を抽出できれば、テキスト間含意関係認識や質問応答のタスクなど、様々な応用例が考えられるため、興味を持ったことである。

2 論文の要約

cgc2lambdaとは、入力文のCCG構文木をもとに、lambda式、すなわち意味表現の木を生成することである。そもそも、CCGとはCombinatory Categorical Grammar（組み合わせ範疇文法）の略であり、文の構造をS,NP,S/NPなどのカテゴリと、それらの組み合わせ規則で表したものである。C&C parserなど既存のCCGパーサを使うことで、入力文に対し、適切であろうCCG構文木が得られる。構文木は親子関係を記述するのに適しているxmlファイルで出力される。

一方、カテゴリと「意味」をつなぐ規則をyamlファイルに書き(以下、semantic templateと呼ぶ)、このファイルとCCG構文木をもとに、セマンティクスの構文木を作る。「意味」はλ式で表現されており、例えばこのように記述する。

```
- category: NP[nb=true]/N
  semantics: \E F1 F2 F3. forall x. (F1(x) -> (F2(x) -> F3(x)))
  surf: all
```

ここで、categoryがまさにCCG構文木におけるカテゴリに対応する。([nb=true]は、NPをさらに細分化するラベル。NP[nb=true]/Nは、右からNをもらい全体でNP[nb=true]になるカテゴリ。)surfは単語を意味する。上の例では、「all」という単語に対して、そのカテゴリがNP[nb=true]/Nである場合には、そのセマンティクスとして $\lambda E F1 F2 F3. \forall x. (F1(x) \rightarrow (F2(x) \rightarrow F3(x)))$ のλ式を与えることになる。このセマンティック構文木も、xml形式で出力される。

既存の類似システムでは1階述語論理式までにしか対応していなかった。しかし、この論文で紹介されているsemantic templateでは、高階述語論理式も作れるように記述されており、most, know, manageなどの単語にも対応している。（

<https://www.aclweb.org/anthology/D15-1244> 2015 Koji Mlneshima et al に詳細あり。この工夫により、後述するタスクを用いて、既存のシステムより意味表現の精度を上げることに成功したとの事。）

さて、文の意味表現が得られたら、その精度を評価したい。そのために、テキスト間含意関係認識のタスクも行い、精度を測る。すなわち、入力文がn個あるとし、各文それぞれの意味表現を構成する。その後、始めn-1個の意味表現を仮定(premise)、最後の1個を結論(conclusion)として、Coqに投げる。仮定から結論が導出できるときはyes、結論の否定が導出できるときはno、どちらも導出できないときはunknownが出力される。fracasというデー

タセットには、複数文と答のセットが346個ある。意味表現を用いて各推論を行った所、精度が69%であったとの事(2015 Koji et al)。

3 論文に対する自分の実験・意見・考察

実際に、チュートリアル(参考文献※3, ※4)に従って、ccg2lambdaのインストール、Coqのインストール、C&Cパーサのインストールを行い、含意関係認識のテストを行った。以下、自分が行なった実験を記す。

実験1：

仮定文： Nobody slept.

結論文： Ken did not sleep.

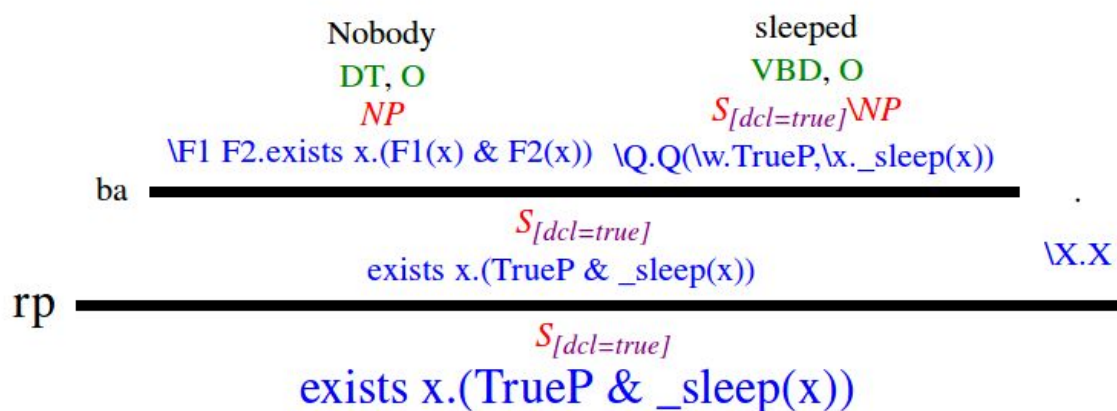
結果： Unknown

期待される結果はyesであったが、unknownと出力された。得られた構文木を解析したところ、

以下の、Nobodyの部分のλ式が、 $\lambda F1. \lambda F2. \text{exists } x. (F1(x) \ \& \ F2(x))$ 、となっているのが明らかにおかしい。これは、yamlファイルで、単語「nobody」に対する意味表現が書かれておらず、したがって、NPのデフォルトの意味表現になってしまったことが原因である。

i

Premise 0, tree s0_ccg0: Nobody slept .



そこで、

- category: NP
 - semantics: $\lambda E \ F1 \ F2. \text{exist } x. (_people(x) \rightarrow (F1(x) \rightarrow F2(x)))$
 - surf: someone
- category: NP
 - semantics: $\lambda E \ F1 \ F2. \text{exist } x. (_people(x) \rightarrow (F1(x) \rightarrow F2(x)))$
 - surf: somebody
- category: NP
 - semantics: $\lambda E \ F1 \ F2. - \text{exist } x. (_people(x) \rightarrow (F1(x) \rightarrow F2(x)))$
 - surf: nobody

のように、いくつかの単語に関してSemanticsを追加した。これにより、実験 1 を再び行なったところ、yesの結果が正しく得られた。このように、デフォルトと違う意味を持つ単語について、その意味表現をそれぞれ追加していけば、より正しいセマンティック構文木が得られるようになる。（ただ、全ての単語について記述するのは、現実的に不可能なので、どうするか。）

実験2：

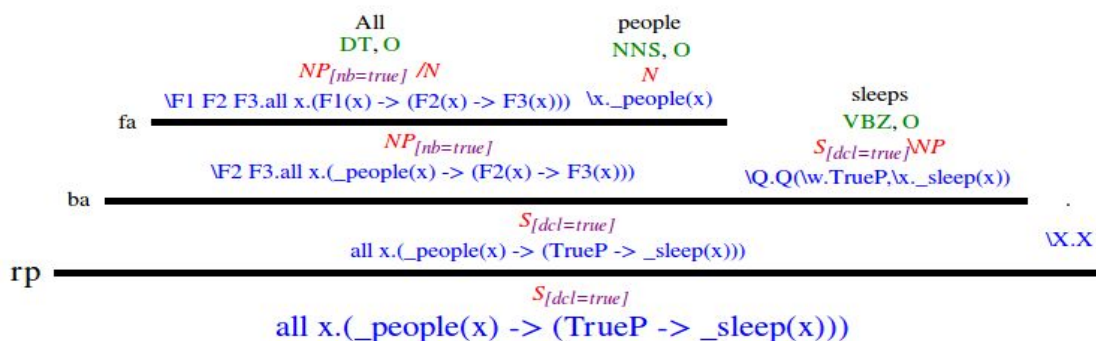
仮定文： All people sleeps.

結論文： Ken sleeps.

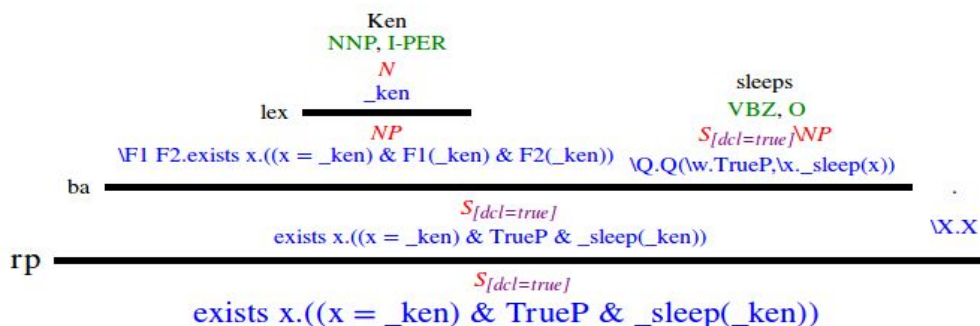
結果： Unknown

期待される結果はyesであったが、unknownと出力された。得られた構文木は以下のように、全く問題なさそうに見える。

Premise 0, tree s0_ccg0: All people sleeps .



Conclusion, tree s1_ccg0: Ken sleeps .



何が原因なのだろうか。考えられるのは、結局、「Kenは人である」という人間にとっての暗黙知が、機械には必要だったということ。「Kenはpersonである」ということも明示的に書かないといけない。（C&Cパーサだとそれが検証しづらかったので、別の類似文で検証したところ、

仮定1: All men sleeps.

仮定2: Ken is a man.

結論： Ken sleeps.

において、仮定 2 がない場合はUnknown, 仮定 2 がある場合はyesが出力された。）

したがって、「暗黙知のデータベース」みたいなものがあれば、実際Coqに入れる前に、単語間の関係（今回だと、Ken is a person）を導き出してそのラムダ式も仮定に加えることで、ただし推論ができそう。ただ、単語数やその関係はもはや無限にあるため、汎化的なものを作るかと言われたらかなり難しそうである。

とすると、実験1でも、Nobodyとkenの関係は機械にはわからないはずであるから、正解が出力されたのはむしろ違和感。そこで意味表現を見直したところ、おかしい事が判明した。具体的には、Nobody sleptの意味が

-exists x.(_people(x) -> (TrueP -> _sleep(x)))

になっていたが、これは偽で、正しくは

-exists x.(_people(x) \wedge (TrueP -> _sleep(x)))

である。よくよく考えてみたら、自分で追加実装した λ 式が間違っていたようだ。論理式は難しい。

正しくは、

- category: NP

 semantics: $\lambda E F1 F2. \text{exist } x. (_people(x) \ \& \ F1(x) \ \& \ F2(x))$

 surf: someone

- category: NP

 semantics: $\lambda E F1 F2. \text{exist } x. (_people(x) \ \& \ F1(x) \ \& \ F2(x))$

 surf: somebody

- category: NP

 semantics: $\lambda E F1 F2. \neg \text{exist } x. (_people(x) \ \& \ F1(x) \ \& \ F2(x))$

 surf: nobody

であるはずで、これを用いると

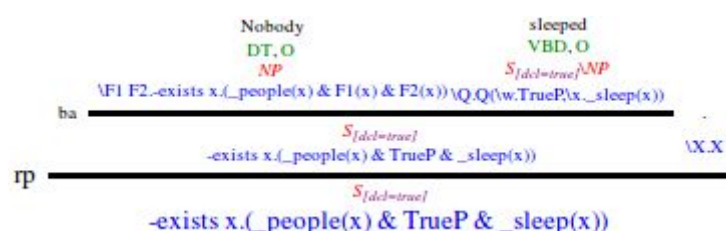
実験3 :

仮定文 : Nobody slept.

結論文 : Ken did not sleep.

結果 : Unknown

となり正しく、セマンティック木も以下のように正しくなった。



このように、今回扱ったsemantic template (semantic_templates_en_emnlp2015.yamlのこと)を拡張していくことで、より広範囲に及ぶ意味表現を扱えるようになることがわかった。

ただ、セマンティックで使われるラベル間の関係（例えば、MikeとPeopleや、PeopleとPersonなど）は機械は把握しておらず、ただのラベルとでしか見ていないので、その問題をどう解消していくかが今後の課題であると思う。

4 参考文献

- 1 Koji Mineshima et al. 2016. ccg2lambda: A Compositional Semantics System. ACL Anthology
- 2 Koji Mineshima et al. 2015. Higher-order logical inference with compositional semantics. ACL Anthology
- 3 ccg2lambda、Coq、パーサのインストールについて
<http://abelard.flet.keio.ac.jp/person/minesima/c2l/doc.html>
- 4 semantic templateの書き方の詳細
<http://abelard.flet.keio.ac.jp/person/minesima/c2l/semantics.html>