The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

意味解析システム ccg2lambda による金融ドキュメント処理

Semantic Parsing in ccg2lambda and its Application to Financial Document Processing

外園 康智 *1 Yasunori Hokazono
長谷川 貴博 *2 Takahiro Hasegawa
渡邉 知樹 *3 Kazuki Watanabe
馬目 華奈 *4 Kana Manome
簗 有紀子 *4 Yukiko Yana
谷中 瞳 *3 Hitomi Yanaka
田中 リベカ *4 Ribeka Tanaka
Pascual Martínez-Gómez*5 Pascual Martínez-Gómez
峯島 宏次 *4 Koji Mineshima
戸次 大介 *4 Daisuke Bekki

*1野村総合研究所 *2オメガ・パートナーズ *3東京大学 *4お茶の水女子大学
*5産業技術総合研究所

In this paper we study the capabilities of ccg2lambda to perform semantic parsing and natural language inferences in the financial domain. We observe that the system is conveniently modular and highly interpretable for humans since it includes graphical visualizations of semantic representations; these visualizations allow us to trace the process of semantic composition and easily identify points of failure. Moreover, it produces results in XML which make it easy to integrate in larger, in-house systems. The system separates the compositional mechanism from the specification of semantic theories, allowing extensions to handle new linguistic phenomena with a relatively small effort. Finally, we present our approach to adapt the system to the specialized vocabulary that arises in the financial domain by using axioms and ontologies.

1. はじめに

本稿では、日本語・英語のテキストを論理式に変換し、自動推論を行う意味解析システム ccg2lambda[*1] の基本的な機能を解説し、特にこのシステムを金融ドキュメントの処理へと応用する試みについて紹介する。

金融は情報産業とも言われるように、契約書、商品説明書、当局への届出書など、多くのドキュメントが生成され、関係者間でやり取りされる。その多くは、金商法などを元に、投資判断を誤解させないためのコンプライアンスや記載ルールが設定されている。商品や契約の細かな差異により単語や表現のバリエーションは多いが、ルール自体は適用範囲を広くするために抽象化された表現になっている。そのため、チェック担当者は、金融知識に加え、文の意味内容の理解と正確な推論が要求される。本研究では、このような専門性と正確性が要求されるチェックを自動で行うシステムの構築を試みる。

システムの対象業務は、上場申請書類の内容チェック、投資信託の商品概要書の記載チェック、取引契約書の不備チェックである。これらは、用語や表現が似ているので共通としたが、システムの応用範囲は、金融に限らず、法務関連や特許、契約書類など幅広く想定される。

深い意味解析と推論が要求されるテキスト処理のため、本研究では、形式意味論に基づく open domain の意味解析・推論システム ccg2lambda [Martínez-Gómez 16] を用いる。これは、組合せ範疇文法 (Combinatory Categorial Grammar, CCG) [Steedman 00, 戸次 10] に基づく統語・意味解析と自然言語推論を扱う高階論理の定理証明系を組み合わせたシステムであり、含意関係認識 [Mineshima 15, Martínez-Gómez 17] や文類似度計算 [Yanaka 17] に応用されている。

ccg2lambda では、統語・意味解析から推論までの各モジュールが明確に区別されており、統語情報・意味合成・意味表現を CCG 導出木としてグラフィカルに表現する機能をサポートしている。このため、処理プロセスのどの部分で解析エラーが起こったのかを容易に同定することが可能である。また、ccg2lambda は、特定の意味論からは独立な、ラムダ計算による一般的な意味合成メカニズムを用いているため、解析の対象となるドメインに特有の構文や言語現象に応じて、低コストで意味テンプレートを拡張することができる。ccg2lambda 自体は open domain の意味解析システムであるが、独自辞書や推論に必要となる公理を追加することで、特定ドメインに適合したシステムを構築・改良することが可能となっている。

本稿では、関連研究を紹介した上で (§2)、ccg2lambda の意味解析 (§3) と推論システム (§4) について解説する。特に金融ドキュメントを対象とした含意関係認識と矛盾検知について、具体例に基づいて説明する。金融ドキュメントチェックシステムの全体像、および、RTE テストセットに基づくシステムの評価については、[馬目 18] を参照してほしい。

2. 関連研究

論理式に基づく深い意味解析の手法は、英語 CCGBank [Hockenmaier 07]、及び、それに基づく高速な CCG パーザ [Clark 07] の出現により、大きな進展を遂げている。

CCG に基づいてテキストを論理式に変換する最初のシステムとしては、Boxer [Bos 04] がある。Boxer では、CCG の導出木は談話表示理論 (DRT) [Kamp 93] の意味表示、すなわち、談話表示構造 (Discourse Representation Structure, DRS) へと変換される。DRS は 1 階述語論理 (FOL) の論理式に変換可能であり、それを定理証明器・モデル構築器と接合することで、含意関係認識 [Bos 05] や意味類似度計算 [Bjerva 14] へと応用されている。他にテキストから DRS への変換を行う open domain の意味解析器としては、日本語と英語を対象にしたスコープ制御理論に基づく Treebank Semantics [Butler 12] や、フランス語を対象にした Type-Logical Grammar に基づく Grail [Moot 15] がある。

CCG に基づく意味解析・推論システムの最近の研究としては、CCG パーザの出力を、文の表層形に類似した意味表現である Lambda Logical Form (LLF) へと変換する LangPro [Abzianidze 17] がある。LangPro は、推論システムとし

*1 https://github.com/mynlp/ccg2lambda

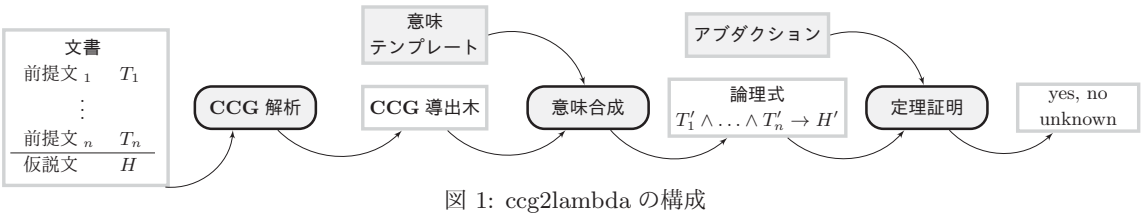

図 1: ccg2lambda の構成

て LLF に基づくタブローシステムを実装し、論理推論を含む含意関係認識のテストセットである SICK RTE データセットで高精度を達成している [Abzianidze 15]。

本研究が依拠する ccg2lambda は、FOL を部分系として含む高階論理に基づく推論システムを採用している。また、辞書項目の記述に比較的シンプルな設計に基づくテンプレートを用いているため、意味論を柔軟に拡張することが可能であり、多様なユーザーが比較的容易に扱えるような意味解析パイプラインを提供している。これは既存の論理ベースの意味解析システムと異なる点のひとつである。

3. 意味解析

含意関係認識とは、テキスト（前提文）$T_1, \ldots, T_n$ と仮説文 H との間に含意関係が成立するか否かを判定する課題である。ccg2lambda は、$T_1, \ldots, T_n$ と H を入力として、CCG パーザによる構文解析・意味解析により論理式 $T'_1, \ldots, T'_n, H'$ を得る。その上で、必要なドメイン知識を表現した知識ベースのもとで、$T'_1 \wedge \cdots \wedge T'_n \rightarrow H'$ が証明可能であるかどうかの判定を行う。図 1 に ccg2lambda の各モジュールの関係を示す。

以下では、より具体的な文の解析プロセスを順に説明する。

3.1 CCG パーザ

現在 ccg2lambda がサポートしている CCG パーザは、英語では C&C [Clark 07]、EasyCCG [Lewis 14]、depccg [Yoshikawa 17] の三つである。日本語 CCG パーザは、Jigg [Noji 16] と depccg の二つをサポートしている。複数のパーザの出力を用いることにより、パーザに由来するエラーを軽減することが可能となっている。

3.2 意味テンプレート

意味テンプレートは CCG の導出木の各ノードに対してラムダ項を割り当てる規則を記したものである。ラムダ項の形式は NLTK [Garrette 09] を用いている。意味テンプレートは、YAML[*2] ファイルに記述する。

意味テンプレートに記述される各規則には必ず `semantics` と `category` という属性が記述されなければならない。例えば、名詞に対応した規則として以下がある。

```
- category : N
  semantics : λE.λx.E(x)
```

この規則が適用されるノードとして、例えば、統語範疇 N をもつ「少年」という語のノードがあるとき、そこで得られる意味表現は、$(\lambda E.\lambda x.E(x))(\text{少年})$ となり、β 簡約が行われ、最終的な意味表現は $\lambda x.\text{少年}(x)$ となる。

名詞一般ではなく、「固有名詞」という品詞タグに適用するテンプレートを用意するには、`pos` という品詞に対応した属性を用いて、固有名詞のための規則を新しく作る。ノードに対して複数の規則が適用できる場合は、YAML ファイルの下に書かれた規則が優先して適用される。上記の規則は固有名詞に対しても適用可能であるので、以下の固有名詞だけを対象とする規則を新たに追加すればよい。

```
- category : NP
  semantics : λE.E
  pos1 : 固有名詞
```

形態素解析による品詞タグ付けには、kuromoji[*3] を用いている。また、CCG パーザから出力される統語範疇のより詳細な条件を属性に追加することができる。例えば、NP の統語素性 $[case = to]$ を以下のように指定することができる。

```
- category : S\NP[case = to]
```

Coq による自動推論に用いる意味表示の型は、自動で設定されるが、明示的に意味テンプレートで指定することもできる。例えば、自動詞の意味表示は `entity` を項にとって命題を返す関数であり、`coq_type` 属性で以下のように型を指定できる。

```
- category : S\NP
  semantics : λE.λx.E(x)
  coq_type : Entity → Prop
```

子を持つノードに対して適用される規則に対しては、`rule` という属性を用いることで、CCG の組合せ規則を参照する形で意味割り当てを行うことが可能である。また、子ノードを対象とした属性もある。子ノードに対して、左から順に 0、1 と順序を定めると、それぞれ child0、child1 として属性の対象にすることができる。例えば、右の子ノードに対して統語範疇を制限したいときは、`child1_category` 属性を用いる。子ノードのさらに子ノードに遡ることも可能であり、例えば、右の子ノードの左の子ノードの品詞を指定したいときは `child1_child0_pos` 属性を用いる。具体例として、$S\backslash NP$ を $NP\backslash NP$ へと変換する unary rule（いわゆる内の関係の連体修飾節を扱うための規則）は、次のように指定することができる。

```
- category : NP\NP
  rule : ANDint
  child0_category : S\NP
```

3.3 出力形式

CCG 導出木から意味表示を得るためには、semparse.py を用いる。

```
# python semparse.py ccgtrees.xml
templates.yaml semantics.xml
```

ここで、ccgtrees.xml は Jigg [Noji 16] の出力形式で記述された CCG 導出木の XML ファイルであり、templates.yaml は

*2 http://www.yaml.org/spec/

*3 https://www.atilika.com/ja/kuromoji/

```
<root>
  <sentences>
    <sentence>
      <tokens>
        <token base="tea" surf="tea" pos="NN"/>
        <token ...>
      </tokens>
      <ccg>
        <span id="s1" child="s2" category="N"
              rule="lex"/>
        <span ...>
      </ccg>
      <semantics>
        <span id="s1" child="s2" sem="\y._tea(y)"
              type="_tea : Entity -> Prop"/>
        <span ...>
      </semantics>
    </sentence>
  </sentences>
</root>
```

図 2: 意味合成結果の XML 形式

3.2 節で述べた意味テンプレートである。出力結果は semantics.xml に保存される。

出力形式は Stanford CoreNLP に従う。具体例を図 2 に示す。各ノードの属性の情報は ⟨*token*⟩ タグに記載されている。⟨*semantic*⟩ タグの内部にある ⟨*span*⟩ タグは sem 属性と type 属性をもつ。sem 属性には対象となるノードの意味表示が、type 属性には対象のノードの `coq_type` 属性が記述される。

visualize.py を用いて見やすい出力を得ることができる。

```
# python visualize.py semantics.xml
> semantics.html
```

このコマンドは、semantics.xml に保存されている意味表示の情報を HTML 形式の CCG 導出木に変換し、semantics.html に出力する。semantics.html の CCG 導出木では、各ノードを合成する組合せ規則が記載されており、各ノードに意味表示が付加されている。ルートノードの意味表示が入力文の意味表示（論理式）を表している。

4. 推論

4.1 含意関係認識と矛盾検知

得られた意味表示間の含意関係・矛盾関係の証明は、意味表示の XML ファイルを引数として、prove.py を実行することで実行可能である。次のコマンドを実行すると、semantics.xml に記述された意味表示間の含意関係・矛盾関係の証明が行われ、含意 (yes) か、矛盾 (no) か、不明 (unknown) かという証明の結果が出力される。

```
# python prove.py semantics.xml
```

prove.py では主に `--graph_out`、`--ncores`、`--proof` という三つのオプションがある。`--graph_out` オプションは、HTML 出力を指定するオプションである。出力 HTML ファイル名を引数に指定すると、CCG 導出木、意味表示、定理証明に要したスクリプトを表示した HTML ファイルを出力できる。`--ncores` オプションは、証明の同時実行数を指定するオプションである。大量の文について証明を行いたい場合に、証明の同時実行数を増やすことで効率的に処理することができる。`--proof` オプションは、XML 出力を指定するオプションである。ccg2lambda から得られた統語解析・意味解析・推論の結果を他のアプリケーションで利用したい場合に便利なオプションである。

(1) 本社債は、劣後特約が付されているため、発行者が発行する劣後特約が付されていない社債と比較して、元利金の弁済順位が低い劣後社債です。
(2) 本債券を含む劣後社債は、活発な流通市場が形成されていないため、一般の社債に比べて流動性が低くなります。
(3) 本社債は、売却できない、または希望する条件では売却できず、金利水準や発行者の経営状況または財務状況、格付けの状況等により元本を割り込む可能性があります。
(4) 劣後社債は、一般に流動性が高くなります。
(5) 本社債は、弁財順位が、一般の社債より高くなります。
(6) 本債券は、上場株式等の譲渡所得等として申告分離税の対象となります。

図 3: 矛盾検知の具体例

4.2 含意関係認識と矛盾検知の具体例

意味表示間の含意関係・矛盾関係の両方の証明を試みることにより、1 つのドキュメント内に互いに矛盾するような文が含まれていた場合に、その矛盾を検知することができる。図 3 の金融ドキュメントの具体例に基づいて説明しよう。このうち、(2) と (4)、(1) と (5) は互いに整合的でない。前者の矛盾は、記載内容のチェック項目として、例えば、「劣後社債は流動性が低い」という内容に注目したときに検知可能である。この項目の記載有無を (2) に対して調べると、(2) を前提文、上記チェック項目を仮説文とする入力に対して、出力結果は含意 (yes) となる。一方、同じ項目の記載を (4) に対して同様に調べた場合、(4) を前提文 T、上記チェック項目を仮説文 H とする入力に対して、出力結果は矛盾 (no) となる。これは形容詞「低い」「高い」が反義語であることによる。

T: 劣後社債は、一般に流動性が高くなります。
H: 劣後社債は流動性が低い。

このように、1 つの仮説文に対して含意 (yes) と矛盾 (no) の両方の結果が得られたことにより、それぞれの推論の前提文であった (2) と (4) の記述が矛盾していると判定できる。

4.3 アブダクション

自然言語の推論において、内容語間の意味的関係を語彙知識から補完する必要があるケースはしばしば存在する。たとえば、次の金融テキスト T にチェック項目 H の内容が含まれているかを含意関係の証明から示すことを考える。

T: 本債券は、上場株式等の譲渡所得等として申告分離税の対象となります。
H: 本債券は、分離課税の対象となる。

このとき、純粋な論理推論では「申告分離税」と「分離課税」は異なる述語として扱われるため、「分離課税」が「申告分離税」の上位概念であるという知識を推論に補完しなければ、T, H 間の含意関係を示すことができない。すなわち、語彙知識を用いて「申告分離税」と「分離課税」間の語彙関係を公理として推論に補完する（アブダクション）仕組みが必要である。ccg2lambda では、次のように prove.py のオプション `--abduction` に `spsa` を設定することで、アブダクション機能 [Martínez-Gómez 17] が利用できる。

```
# python prove.py --abduction spsa
```

アブダクションではまず、通常の証明を試みた結果、証明できずに残っている結論中の論理式に対して、共通の自由変数を持つ前提中の論理式を特定する。これは前提と結論の意味表示において単語間のアライメントを行うことと同じ意味合いを持ち、これによって効率的な前提探索を実現している。次に、特定された前提と結論の論理式間で意味的関係が成り立つか否かを語彙知識を用いてチェックする。語彙知識から意味的関係が確認できた場合、公理を補完して再度証明を実行する。

アブダクションに用いる語彙知識は、デフォルトでは英語 WordNet [Miller 95], VerbOcean [Chklovski 04] が語彙知識として登録されている。ccg2lambda に搭載されているアブダクション機能は、使用する語彙知識を限定しない。そのため、タスク固有のオントロジー知識など、他の語彙知識をユーザが追加することも容易である。

5. まとめと展望

本稿では、意味解析・推論システム ccg2lambda を用いて、金融ドキュメント解析を行う手法について解説した。含意関係認識・矛盾検知にこの種のシステムを応用する際のボトルネックの一つは、CCG パーザの精度にある [馬目 18]。この問題に対処するため、[Yoshikawa 18] では、前提文と結論文の文間整合性を考慮した CCG パージングの手法が提案されており、含意関係認識タスクでの精度向上が報告されている。

この他にも、ccg2lambda の拡張として、文類似度計算 [Yanaka 17] への応用や、アブダクション・システムを単語間の語彙知識だけでなく、フレーズ間の語彙知識を補完できるよう拡張する試みがある [Yanaka 18]。アブダクションを知識ベース補完 (Knowledge Base Completion) と組み合わせることで、自然言語推論のための定理証明を高速化する手法も提案されている [吉川 18]。こうした基本技術の発展により、個別ドメインへの応用はさらに進展するものと期待される。

参考文献

[Abzianidze 15] Abzianidze, L.: A Tableau Prover for Natural Logic and Language, in *Proceedings of EMNLP2015*, pp. 2492–2502 (2015)

[Abzianidze 17] Abzianidze, L.: LangPro: Natural Language Theorem Prover, in *Proceedings of EMNLP2017: System Demonstrations*, pp. 115–120 (2017)

[Bjerva 14] Bjerva, J., Bos, J., Goot, van der R., and Nissim, M.: The Meaning Factory: Formal semantics for recognizing textual entailment and determining semantic similarity, in *Proceedings of SemEval2014*, pp. 642–646 (2014)

[Bos 04] Bos, J., Clark, S., Steedman, M., Curran, J. R., and Hockenmaier, J.: Wide-coverage semantic representations from a CCG parser, in *Proceedings of COLING2014*, pp. 1240–1246 (2004)

[Bos 05] Bos, J. and Markert, K.: Recognising textual entailment with logical inference, in *Proceedings of HLT/EMNLP2005*, pp. 628–635 (2005)

[Butler 12] Butler, A. and Yoshimoto, K.: Banking meaning representations from treebanks, *Linguistic Issues in Language Technology*, Vol. 7, No. 1 (2012)

[Chklovski 04] Chklovski, T. and Pantel, P.: VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations, in *Proceedings of EMNLP2004*, pp. 33–40 (2004)

[Clark 07] Clark, S. and Curran, J. R.: Wide-coverage efficient statistical parsing with CCG and log-linear models, *Computational Linguistics*, Vol. 33, No. 4, pp. 493–552 (2007)

[Garrette 09] Garrette, D. and Klein, E.: An Extensible Toolkit for Computational Semantics, in *Proceedings of IWCS2009*, pp. 116–127 (2009)

[Hockenmaier 07] Hockenmaier, J. and Steedman, M.: CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank, *Computational Linguistics*, Vol. 33, No. 3, pp. 355–396 (2007)

[Kamp 93] Kamp, H. and Reyle, U.: *From discourse to logic: An introduction to modeltheoretic semantics of natural language, formal logic and DRT*, Kluwer (1993)

[Lewis 14] Lewis, M. and Steedman, M.: A* CCG Parsing with a Supertag-factored Model, in *Proceedings of EMNLP2014*, pp. 990–1000 (2014)

[Martínez-Gómez 16] Martínez-Gómez, P., Mineshima, K., Miyao, Y., and Bekki, D.: ccg2lambda: a compositional semantics system, in *Proceedings of ACL2016 System Demonstrations*, pp. 85–90 (2016)

[Martínez-Gómez 17] Martínez-Gómez, P., Mineshima, K., Miyao, Y., and Bekki, D.: On-demand Injection of Lexical Knowledge for Recognising Textual Entailment, in *Proceedings of EACL2017*, pp. 710–720 (2017)

[Miller 95] Miller, G. A.: WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41 (1995)

[Mineshima 15] Mineshima, K., Martínez-Gómez, P., Miyao, Y., and Bekki, D.: Higher-order logical inference with compositional semantics, in *Proceedings of EMNLP2015*, pp. 2055–2061 (2015)

[Moot 15] Moot, R.: A type-logical treebank for French, *Journal of Language Modelling*, Vol. 3, No. 1, pp. 229–264 (2015)

[Noji 16] Noji, H. and Miyao, Y.: Jigg: a framework for an easy natural language processing pipeline, in *Proceedings of ACL2016 System Demonstrations*, pp. 103–108 (2016)

[Steedman 00] Steedman, M.: *The Syntactic Process*, MIT Press (2000)

[Yanaka 17] Yanaka, H., Mineshima, K., Martínez-Gómez, P., and Bekki, D.: Determining Semantic Textual Similarity using Natural Deduction Proofs, in *Proceedings of EMNLP2017*, pp. 692–702 (2017)

[Yanaka 18] Yanaka, H., Mineshima, K., Martínez-Gómez, P., and Bekki, D.: Acquisition of Phrase Correspondences using Natural Deduction Proof, in *Proceedings of NAACL2018* (2018)

[Yoshikawa 17] Yoshikawa, M., Noji, H., and Matsumoto, Y.: A* CCG Parsing with a Supertag and Dependency Factored Model, in *Proceedings of ACL2017*, pp. 277–287 (2017)

[Yoshikawa 18] Yoshikawa, M., Noji, H., Mineshima, K., and Daisuke, B.: Consistent CCG Parsing over Multiple Sentences for Improved Logical Reasoning, in *Proceedings of NAACL2018* (2018)

[吉川 18] 吉川 将司, 峯島 宏次, 能地 宏, 戸次 大介 : 知識ベース補完を用いた高階論理推論のための自動公理生成, 言語処理学会第 24 回年次大会発表論文集, pp. 113–116 (2018)

[戸次 10] 戸次 大介 : 日本語文法の形式理論: 活用体系・統語構造・意味合成, くろしお出版 (2010)

[馬目 18] 馬目 華奈, 外園 康智, 長谷川 貴博, 小西 優祐, 渡邉 知樹, 簗 有紀子, 谷中 瞳, 田中 リベカ, 峯島 宏次, 戸次 大介 : 含意関係認識による金融ドキュメントチェックへの取り組み, 言語処理学会第 24 回年次大会発表論文集, pp. 1159–1162 (2018)