Automatic Generation of Wide-Coverage Semantic Representations in NLTK

Jake Vasilakes

August 18, 2015

Semantic Parsing: Mapping natural language (NL) sentences to a machine readable meaning representation (MR).

Semantic Parsing: Mapping natural language (NL) sentences to a machine readable meaning representation (MR).

John admires Mary. $\Rightarrow$ $admires(Mary, John)$

Supervised approaches

(Ge & Mooney, 2005; Wong & Mooney, 2006; Kate & Mooney, 2006)

- Learn a mapping from NL to constituent predicates in the target meaning representation language (MRL).
- Requires annotated training data, e.g. NL-MR pairs.

Supervised approaches

(Ge & Mooney, 2005; Wong & Mooney, 2006; Kate & Mooney, 2006)

- Learn a mapping from NL to constituent predicates in the target meaning representation language (MRL).
- Requires annotated training data, e.g. NL-MR pairs.
- Issues: Limited domain/methods generalize poorly to other MRLs. Require gold-standard MRs for each NL input sentence.

Minimally-supervised approaches

(Clarke, Goldwasser, Chang, & Roth, 2010; Goldwasser, Reichart, Clarke, & Roth, 2011)

- Training data is NL query and gold-standard answer.
- Treat MR as a latent variable. Find best MR out of possible MRs for an input sentence using a weight vector learned via a feedback signal (e.g. +1 if MR gets correct answer, -1 if incorrect).

Minimally-supervised approaches

(Clarke et al., 2010; Goldwasser et al., 2011)

- Training data is NL query and gold-standard answer.
- Treat MR as a latent variable. Find best MR out of possible MRs for an input sentence using a weight vector learned via a feedback signal (e.g. +1 if MR gets correct answer, -1 if incorrect).
- Issue: Reliance on training data still limits breadth of domain.

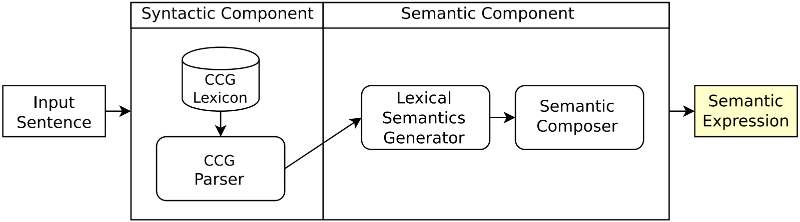
Goal: automatic generation of ungrounded meaning representations without hand-annotated data.

Goal: automatic generation of ungrounded meaning representations without hand-annotated data.

Requirements:

1. Wide-coverage: determine semantic representation for input sentence in any domain.
2. Compositionality: semantic representation of a NL sentence determined by the semantic representations of its constituent words and how they combine.

System Structure



Two components:

1. Syntactic Component: syntactic parse of the input sentence using CCG.
2. Semantic Component: generation of lexical semantics and composition.

Combinatory Categorial Grammar (CCG)

$$\cfrac{\cfrac{John}{NP} \qquad \cfrac{\cfrac{admires}{(S\backslash NP)/NP} \qquad \cfrac{Mary}{NP}}{S\backslash NP}\,>}{S}\,<$$

Combinatory Categorial Grammar (CCG)

$$\cfrac{\cfrac{\textit{John}}{NP} \qquad \cfrac{\cfrac{\textit{admires}}{(S\backslash NP)/NP} \qquad \cfrac{\textit{Mary}}{NP}}{S\backslash NP}{>}}{S}{<}$$

Information provided by the CCG parse

- Syntactic category of constituents, e.g. admires :: $(S \setminus NP)/NP$.
- Combinatory rules, e.g. forward application ($>$).

Lexical semantics

- Translate syntactic category into an expression in Neo-Davidsonian event semantics.
- This process is *automatic* and *deterministic*.
- Possible because CCG syntactic categories specify functions.

Lexical semantics

- Translate syntactic category into an expression in Neo-Davidsonian event semantics.
- This process is *automatic* and *deterministic*.
- Possible because CCG syntactic categories specify functions.

admires :: $(S \setminus NP)/NP \quad \Rightarrow$

Lexical semantics

- Translate syntactic category into an expression in Neo-Davidsonian event semantics.
- This process is *automatic* and *deterministic*.
- Possible because CCG syntactic categories specify functions.

admires :: $(S \setminus NP)/NP \quad \Rightarrow$
$\lambda z \lambda y. \exists e.[admires.agent(e, y) \wedge admires.patient(e, z)]$

Lexical semantics

- Translate syntactic category into an expression in Neo-Davidsonian event semantics.
- This process is *automatic* and *deterministic*.
- Possible because CCG syntactic categories specify functions.

admires :: $(S \setminus NP)/NP \quad \Rightarrow$
$\lambda z \lambda y . \exists e . [admires.agent(e, y) \wedge admires.patient(e, z)]$

Approach adopted from GraphParser developed by (Reddy et al., 2014).

Composition

- Compose constituent semantic expressions guided by the syntactic parse tree.
- Recursively build subexpressions according to the structure of the parse tree and the CCG combinatory rules used.
- E.g. CCG application rule $\mapsto$ functional application.

An example parse

$$
\dfrac{\dfrac{\textit{John}}{\begin{array}{c} NP \\ john \end{array}} \quad \dfrac{\dfrac{\textit{admires}}{\begin{array}{c} (S\backslash NP)/NP \\ \lambda z \lambda y.\exists e.[admires.agent(e,y) \wedge admires.patient(e,z)] \end{array}} \quad \dfrac{\textit{Mary}}{\begin{array}{c} NP \\ mary \end{array}}}{\begin{array}{c} S\backslash NP \\ \lambda y.\exists e.[admires.agent(e,y) \wedge admires.patient(e,mary)] \end{array}}>}{\begin{array}{c} S \\ \boldsymbol{\exists e.[admires.agent(e,john) \wedge admires.patient(e,mary)]} \end{array}}<
$$

Evaluation

- Implemented as a package within the NLTK framework (`nltk.semparse`).
- Preprocessing performed using the NLTK tokenizer (`nltk.word_tokenize`) and NLTK POS tagger (`nltk.pos_tag`).
- Syntactic parsing performed by the NLTK CCG package (`nltk.ccg`).

Evaluation

- Implemented as a package within the NLTK framework (`nltk.semparse`).
- Preprocessing performed using the NLTK tokenizer (`nltk.word_tokenize`) and NLTK POS tagger (`nltk.pos_tag`).
- Syntactic parsing performed by the NLTK CCG package (`nltk.ccg`).

Coverage of syntactic and semantic components on GEOQUERY880:

Component	Questions Parsed	Coverage
Syntactic	825	93.75%
Semantic	366	41.59%
Total	**880**	

Limitations of the current system:

- **Tokenizer**: Compound nouns not treated as single entities. Cannot parse, e.g. “Great Britain”.

Limitations of the current system:

- **Tokenizer**: Compound nouns not treated as single entities. Cannot parse, e.g. "Great Britain".
- **POS tagger**: E.g. the verb "to border" assigned incorrect POS tag in nearly every case. This leads to an incorrect semantic representation. 133 queries in GEOQUERY880 contain the verb "to border". Severely impacts coverage.

Limitations of the current system:

- **Tokenizer**: Compound nouns not treated as single entities. Cannot parse, e.g. "Great Britain".
- **POS tagger**: E.g. the verb "to border" assigned incorrect POS tag in nearly every case. This leads to an incorrect semantic representation. 133 queries in GEOQUERY880 contain the verb "to border". Severely impacts coverage.
- **Syntactic component**: NLTK CCG parser is not probabilistic. Also not as expressive as current standard systems such as C&C parser, easyCCG parser.

Limitations of the current system:

- **Tokenizer**: Compound nouns not treated as single entities. Cannot parse, e.g. “Great Britain”.
- **POS tagger**: E.g. the verb “to border” assigned incorrect POS tag in nearly every case. This leads to an incorrect semantic representation. 133 queries in GEOQUERY880 contain the verb “to border”. Severely impacts coverage.
- **Syntactic component**: NLTK CCG parser is not probabilistic. Also not as expressive as current standard systems such as C&C parser, easyCCG parser.
- **Semantic component**: Some special-case linguistic phenomena have not been addressed. E.g. gerunds which act as adjectives, “the running man”.

Further tasks

Current system is preliminary work towards a semantic parser in NLTK.

- Address the limitations described above.
  - It is possible to use external tools for tokenization and POS tagging.
  - Make the NLTK CCG parser probabilistic. Develop method for reading in CCG parses from more reliable parsers.
  - Identify special cases.
- Develop methods for grounding the output expressions in a knowledge base.
  - Evaluate the system on standard question-answering and information retrieval tasks.

Thank you.

Code available at `https://github.com/jvasilakes/nltk/tree/develop/nltk/semparse`.

References I

Clarke, J., Goldwasser, D., Chang, M.-W., & Roth, D. (2010). Driving semantic parsing from the world's response. In *Proceedings of the fourteenth conference on computational natural language learning* (pp. 18–27). Stroudsburg, PA, USA: Association for Computational Linguistics.

Ge, R., & Mooney, R. J. (2005, June). A statistical semantic parser that integrates syntax and semantics. In *Proceedings of conll-2005.* Ann Arbor, Michigan.

References II

Goldwasser, D., Reichart, R., Clarke, J., & Roth, D. (2011). Confidence driven unsupervised semantic parsing. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies - volume 1* (pp. 1486–1495). Stroudsburg, PA, USA: Association for Computational Linguistics.

Kate, R. J., & Mooney, R. J. (2006). Using string-kernels for learning semantic parsers. In *In proc. of coling/acl-06* (pp. 913–920).

Reddy, S., Lapata, M., & Steedman, M. (2014). Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics (TACL)*.

References III

Wong, Y. W., & Mooney, R. J. (2006). Learning for semantic parsing with statistical machine translation. In *Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics* (pp. 439–446). Stroudsburg, PA, USA: Association for Computational Linguistics.