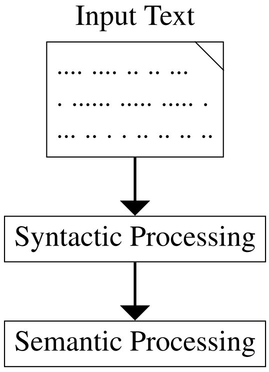
AspCcgTk: Towards Syntactic Parsing with Semantic Disambiguation by Means of Declarative Programming

Yuliya Lierler
University of Nebraska at Omaha
ylierler@unomaha.edu

Peter Schüller
Marmara University
peter.schuller@marmara.edu.tr

Natural language expressions are often ambiguous, allowing multiple interpretations. For example, the sentence – *I eat spaghetti with chopsticks* – is syntactically ambiguous. In one syntactic interpretation, the prepositional phrase "*with chopsticks*" modifies the verbal phrase "*eat spaghetti*" (which suggests that chopsticks serve as a tool for eating). In another interpretation, "*with chopsticks*" modifies the noun phrase "*spaghetti*" (which suggests that chopsticks are part of what is being eaten). The semantic constraints that arise from meanings of individual words in this sentence allow us to disambiguate the sentence and consider its former syntactic interpretation as the correct one. In this note we describe an approach that integrates syntactic analysis with semantic constraints in a system called ASPCCGTK[1]. This system is based on Answer Set Programming — a popular declarative constraint programming paradigm.

Typical natural language processing systems consist of several components including syntactic and semantic analyzers that are organized in a pipeline.



A common assumption in such a design is that these components are separate and independent: input is processed in a certain order that enriches a representation of the given text in an additive (monotonic) way. Recall the sentence

I eat spaghetti with chopsticks. (1)

Its verb phrase allows for two syntactic structures:

$$\dfrac{\dfrac{\dfrac{eat}{(VP/PP)/NP} \quad \dfrac{spaghetti}{NP}}{VP/PP} \quad \dfrac{with\ chopsticks}{PP}}{VP} \qquad \dfrac{\dfrac{eat}{VP/NP} \quad \dfrac{\dfrac{spaghetti}{NP} \quad \dfrac{with\ chopsticks}{NP\backslash NP}}{NP}}{VP} \tag{2}$$

The sentence

I eat spaghetti with meatballs (3)

is syntactically ambiguous in a similar manner. In order to assign the proper syntactic structure to each of these sentences one has to take into account *selectional restrictions*, i.e., the semantic restrictions that

[1] http://www.kr.tuwien.ac.at/staff/former_staff/ps/aspccgtk/

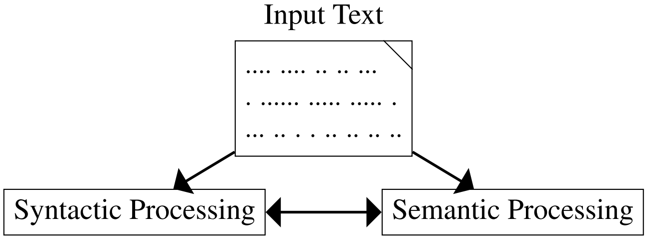
a word imposes on the environment in which it occurs. For instance, in (1) the fact that a chopstick is an instrument suggests that "*with chopsticks*" modifies "*eat spaghetti*" as a tool for eating. Thus, an approach that integrates syntactic and semantic processing is essential for proper analysis of such sentences. Common statistical methods take into account selectional restrictions *implicitly* by assigning most probable syntactic structure based on observed co-occurrences of words and structures in corpora. Yet, this is often not sufficient: for example advanced parsers, including those from Stanford (Socher et al., 2013) and Berkeley (Petrov et al., 2006), favor the same structure for both sentences (1) and (3).

The ASPCCGTK system is an English language parser based on answer set programming (Brewka et al., 2011) that provides a framework for an integration of syntax and semantic analysis. The ASPCCGTK system consists of four main components:

- a logic program *aspccg.lp* that captures the parsing problem so that extending *aspccg.lp* by a given sentence encoded as a logic program results in a program whose "answer sets" (solutions) represent syntactic parses of the sentence,
- grounder GRINGO (Gebser et al., 2010) – a popular front-end of answer set solvers,
- state-of-the-art answer set solver CLASP (Gebser et al., 2012), and
- IDPDraw (Wittocx, 2009) – a system that is used to produce a two-dimensional image for each found parse tree.

Given an input of n words, ASPCCGTK instantiates the Cocke-Younger-Kasami (CYK) chart (Kasami, 1965) using the GRINGO system. The CYK chart is a $n \times n$ triangular grid. Answer set solver CLASP generates parse trees using the content of this grid. Although this answer set programming realization of CYK parsing does not outperform dedicated CYK implementations, it has the advantage of declarativity and reaches sufficient performance for being useful as a natural language processing tool (Schüller, 2013). For example, the ASPCCGTK system is used as part of the NL2KR natural language processing system (Baral et al., 2013).

The declarativity of the ASPCCGTK approach paves the way to flexible and intuitive extensions of the parser with other solution-generating and solution-constraining logic programming modules. Lierler and Schüller (2013) take advantage of this observation. In particular, they illustrate how semantic information collected in FRAMENET (Baker et al., 1998) helps to disambiguate syntactic structures for sentences (1) and (3) within ASPCCGTK. For instance, the frame *food* of FRAMENET corresponds to the word "*spaghetti*". This frame contains information that *food* only takes other *food* as constituents. Thus modifying "*spaghetti*" with "*chopsticks*" in a parse tree for (1) yields a forbidden situation. Answer set programming provides convenient, flexible means for formulating such "forbidden situations". Given FRAMENET-based information about the verb "*eat*", ASPCCGTK is able to find proper syntactic structures for sentences (1) and (3). This case study suggests that ASPCCGTK may serve as a framework for integral syntactic and semantic processing as in the following figure.



Although the ASPCCGTK system gives a promise to advance parsing accuracy, its effectiveness relies on availability of structured knowledge about natural language words and concepts. It remains to be seen whether an automatic interface between FRAMENET and ASPCCGTK can be established and to what extent such an interface improves wide coverage parsing accuracy. Moreover, besides FRAMENET there are other lexical databases such as VERBNET (Kipper-Schuler, 2005) and PROBANK (Palmer et al., 2005) which may serve as further resources for semantic disambiguation of natural language structures.

References

Baker, C. F., C. J. Fillmore, and J. B. Lowe (1998). The Berkeley FrameNet Project. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (ACL/COLING)*, San Mateo, CA, pp. 86–90.

Baral, C., J. Dzifcak, K. Kumbhare, and N. H. Vo (2013). The NL2KR system. In *Workshop on Natural Language Processing and Automated Reasoning (NLPAR)*, pp. 37–47.

Brewka, G., I. Niemelä, and M. Truszczyński (2011). Answer set programming at a glance. *Communications of the ACM 54(12)*, 92–103.

Gebser, M., R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and S. Thiele (2010). A user's guide to gringo, clasp, clingo, and iclingo. `http://sourceforge.net/projects/potassco/files/potassco_guide/2010-10-04/guide.pdf`.

Gebser, M., B. Kaufmann, and T. Schaub (2012). Conflict-driven answer set solving: From theory to practice. *Artificial Intelligence 187*, 52–89.

Kasami, T. (1965). An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, Massachusetts.

Kipper-Schuler, K. (2005). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph. D. thesis, University of Pennsylvania.

Lierler, Y. and P. Schüller (2013). Towards a tight integration of syntactic parsing with semantic disambiguation by means of declarative programming. In *International Conference on Computational Semantics (IWCS)*.

Palmer, M., D. Gildea, and P. Kingsbury (2005, March). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics 31*(1), 71–106.

Petrov, S., L. Barrett, R. Thibaux, and D. Klein (2006, July). Learning accurate, compact, and interpretable tree annotation. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (ACL/COLING)*, pp. 433–440.

Schüller, P. (2013). Flexible combinatory categorial grammar parsing using the CYK algorithm and answer set programming. In *International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR)*, pp. 499–511.

Socher, R., J. Bauer, C. D. Manning, and A. Y. Ng (2013). Parsing with compositional vector grammars. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wittocx, J. (2009). IDPDraw. Katholieke Universiteit Leuven, `http://dtai.cs.kuleuven.be/krr/software/download`.