A Constraint Programming Approach to Probabilistic Syntactic Processing

Irene Langkilde-Geary
Independent Consultant
South Jordan, UT USA
i.l.geary@gmail.com

1 Introduction

Integer linear programming (ILP) is a framework for solving combinatorial problems with linear constraints of the form $y = c_1x_1 + c_2x_2 + ... + c_nx_n$ where the variables (ie., y and x_is) take on only integer values. ILP is a special case of a larger family of contraint-based solving techniques in which variables may take on additional types of values (eg. discrete, symbolic, real, set, and structured) or involve additional kinds of constraints (eg. logical and non-linear, such as $x \wedge y \Rightarrow z$ and $y = cx^n$). Constraint-based problem solving approaches offer a more natural way of modeling many kinds of real-world problems. Furthermore, the declarative nature of constraint-based approaches makes them versatile since the order in which the variables are solved is not predetermined. The same program can thus be reused for solving different subsets of the problem's variables. Additionally, in some cases, constraint-based approaches can solve problems more efficiently or accurately than alternative approaches.

Constraint Programming (CP) is a field of research that develops algorithms and tools for constraint-based problem solving. This abstract describes work-in-progress on a project to develop a CP-based general-purpose broad-coverage probabilistic syntactic language processing system for English. Because of its declarative nature, the system can be used for both parsing and realization as well as their subtasks (such as tagging, chunk parsing, lexical choice, or word ordering) or hybridizations (like text-to-text generation). We expect this tool to be useful for a wide range of applications from information extraction to machine translation to human-computer dialog. An ambitious project such as this poses a number of questions and difficult challenges, including: a) how to declaratively represent the syntactic structure of sentences, b) how to integrate the processing of hard constraints with soft (probabilistic) ones, c) how to overcome problems of intractibility associated with large problems and rich representations in learning, inference, as well as search.

2 Related Work

Declarative and constraint-based representations and computation mechanisms have been the subject of much research in the fields of both Linguistics and Computer Science over the last 30-40 years, at times motivating each other but also sometimes developing independently. Although there is quite a large literature on constraint-based processing in NLP, the notion of a constraint and the methods for processing them vary significantly from that in CP. See (Duchier et al., 1998; Piwek and van Deemter, 2006; Blache, 2000). The CP approach has been designed for a broader ranger of applications and rests on a stronger, more general theoretical foundation. It coherently integrates a variety of solving techniques whereas theoretical linguistic formalisms have traditionally used only a single kind of constraint solver, namely unification. In comparison, the 2009 ILPNLP workshop focuses on NLP processing using solely integer linear constraints.

Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing, pages 36–37,
Boulder, Colorado, June 2009.

3 Methodology

Three key elements of our approach are its syntactic representation, confidence-based beam search, and a novel on-demand learning and inference algorithm. The last is used to calculate probability-based feature costs and the confidences used to heuristically guide the search for the best solution. A description of the flat featurized dependency-style syntactic representation we use is available in (Langkilde-Geary and Betteridge, 2006), which describes how the entire Penn Treebank (Marcus et al., 1993) was converted to this representation. The representation has been designed to offer finer-grained declarativeness than other existing representations.

Our confidence-based search heuristic evaluates the conditional likelihood of undetermined output variables (ie., word features) at each step of search and heuristically selects the case of the mostly likely variable/value pair as the next (or only one) to explore. The likelihood is contextualized by the input variables and any output variables which have already been explored and tentatively solved. Although one theoretical advantage of CP (and ILP) is the ability to calculate an overall optimal solution through search, we unexpectedly found that our confidence-based heuristic led to the first intermediate solution typically being the optimal. This allowed us to simplify the search methodology to a one-best or threshold-based beam search without any significant loss in accuracy. The result is dramatically improved scalability.

We use the concurrent CP language Mozart/Oz to implement our approach. We previously implemented an exploratory prototype that used raw frequencies instead of smoothed probabilities for the feature costs and search heuristic confidences. (Langkilde-Geary, 2005; Langkilde-Geary, 2007). The lack of smoothing severely limited the applicability of the prototype. We are currently finishing development of the before-mentioned on-demand learning algorithm which will overcome that challenge and allow us to evaluate our approach's accuracy and efficiency on a variety of NLP tasks on common test sets. Informal preliminary results on the much-studied subtask of part-of-speech tagging indicate that our method outperforms a Naive Bayes-based baseline in terms of accuracy and within 2% of state-of-the-art single-classifier methods, while running in linear time with respect to the number of output variables or word tokens. We are not aware of any other approach that achieves this level of accuracy in comparable algorithmic time.

4 Conclusion

The versatility and potential scalability of our approach are its most noteworthy aspects. We expect it to be able to handle not only a wider variety of NLP tasks than existing approaches but also to tackle harder tasks that have been intractible before now. Although ILP has the same theoretical power as CP for efficiently solving problems, our approach takes advantage of several capabilities that CP offers that ILP doesn't, including modeling with not only linear constraints but also logical, set-based and other kinds of constraints; customized search methodology with dynamically computed costs, and conditionally applied constraints, among others.

References

P. Blache. 2000. Constraints, linguistic theories and natural language processing. *Natural Language Processing*, 1835.

D. Duchier, C. Gardent, and J. Niehren. 1998. Concurrent constraint programming in oz for natural language processing. Technical report, Universitt des Saarlandes.

I. Langkilde-Geary and J. Betteridge. 2006. A factored functional dependency transformation of the english penn treebank for probabilistic surface generation. In *Proc. LREC*.

I. Langkilde-Geary. 2005. An exploratory application of constraint optimization in mozart to probabilistic natural language processing. In H. Christiansen, P. Skadhauge, and J. Villadsen, editors, *Proceedings of the International Workshop on Constraint Solving and Language Processing (CSLP)*, volume 3438. Springer-Verlag LNAI.

I. Langkilde-Geary. 2007. Declarative syntactic processing of natural language using concurrent constraint programming and probabilistic dependency modeling. In *Proc. UCNLG*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of english: the Penn treebank. *Computational Linguistics*, 19(2).

P. Piwek and K. van Deemter. 2006. Constraint-based natural language generation: A survey. Technical report, The Open University.