Extracting Procedures from Text

Engin Cinar Sahin
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213
csahin@cmu.edu

ABSTRACT
We present our progress on a fully automated system that extracts procedures from text. The system extracts a structured, plan-like graph that represents the procedure described in the text it was given. We hope such a system can be used to populate knowledge bases with real-world procedures that will fuel procedure representation and reasoning research geared towards everyday applications. The results obtained on the training data indicate that the task, and the method presented are feasible and should be investigated further.

Categories and Subject Descriptors
I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms
Algorithms

Keywords
knowledge acquisition from text, knowledge extraction

1. INTRODUCTION
Information extraction is the field of extracting structured data from unstructured or partially structured text. The ultimate goal is to store the extracted information in a knowledge base which will hopefully fuel further applications and research. We can see this trend in named entity extraction [1, 5] and POS tagging [6]. In this paper, we outline a method of extracting procedures from text and present our preliminary experiments on cooking recipes.

A procedure is a manner of executing a set of actions to accomplish a greater task. We will refer to any text describing a procedure as *procedural text*. There has been a wealth of research in representing, reasoning and planning procedures for many years [4, 7]. Procedures may have complex structure, and it is most natural for us to communicate them in natural language. We hope that with the recent progress in natural language processing and information extraction we can start to extract procedures from text. Our hope is to make possible to easily populate knowledge bases capable of doing deep reasoning about procedures and time so that computers may one day assist regular people in their daily lives.

1.1 Procedures and Procedural Text
The structure of procedures presented here is a simplification of the action/event representation in the Scone Knowledge Base System[3]. Procedures are composed of three kinds of actions: 1) *elementary actions*, actions that the agent is assumed to know ("cut", "slice"); 2) *abstract actions*, actions that are constructed through logical relations between other actions ("cut or slice"); 3) *sub-procedures*, actions that have defined procedures for themselves ("To make the sauce, ...") In this simplified representation, there can be six relations between actions. Abstract actions are constructed through 'or', 'and' and 'equals' relations between other actions. The actions in a sub-procedure have part-of relations with the sub-procedure action. Finally, any action may have 'do after' or 'do during' relations with other actions.

The structure of procedures can be complex, and to limit confusion, people have developed a strict style in procedural text. Furthermore, elementary actions are assumed to be known by the reader, which limits the vocabulary of mentionable actions. These make automated extraction both feasible, and relatively easy. Procedural text is usually in the imperative mood, so most action words are in the root form ("Stir in the sauce."). However, the progressive aspect may be used when there is a 'do during' relation ("While baking, ..."), and the infinitive may be used to signal an upcoming sub-procedure ("To make the sauce ..."). Conjunctions and disjunctions of actions are usually made explicit through connectives ("and", "or"), whereas equivalences between actions are usually left to the reader to identify. Equivalences can usually be inferred from ac-

K-CAP'07, October 28–31, 2007, Whistler, British Columbia, Canada.
ACM 978-1-59593-643-1/07/0010

Table 1: Recipe extraction results

Method	% correct
gold standard tokens	53.5%
gold standard tokens + rules	55.2%
CMM-SVM tokens	45.5%
CMM-SVM tokens + rules	47.1%

tion words that are repeated or with hyper/hyponymy relations ("Simmer the chicken for 40 minutes. While cooking ..."). The order which the action tokens appears is usually congruent to the temporal sequence of actions in the procedure, however sometimes temporal adverbs can signal an incongruity. Not every possible way to convey the relations between actions are listed here, but these are how most people communicate them.

2. METHOD

We present a simple yet effective method for extracting procedures from text. First, action tokens are identified. A sequential procedure is constructed using these tokens and their order of appearance in the text. Then, a set of rules identify parallel actions, abstract actions and sub-procedures in the sequence and modify the sequence accordingly. Finally, the sequence may be reordered according to any temporal adverbs or modifiers that alter the sequence of actions in the procedure.

We experimented with various sequential taggers using the MinorThird[2] toolkit to identify action tokens and found the Conditional Markov Model with Support Vector Machine inner learners to be the most effective. This model would usually need a lot of training data, but the strict style and limited vocabulary in procedural text makes the learning problem much simpler. We use a simple regular expression based rule system and at the moment, only parallel action rules may be defined.

3. DATASET AND EXPERIMENTS

More than 15,000 recipes were collected from a public recipe web site. Out of the 15,000 recipes, 1,000 recipes were randomly chosen as the data set and were tagged with the procedure they described. We have split the data into a 650 training and 350 testing set. The results presented here are only on the training set. We are reserving the test set for the fully implemented system.

We used a CMM with SVM inner learners to identify action tokens. The learner achieved a 96.8% F-1 score on the training set (5-fold cross validation) on identifying action tokens. As of now, only a few parallel action rules are written. Results are shown in Table 1. A procedure is correctly extracted, if all actions and relations between them were correctly extracted. The table shows results for using gold standard action tokens versus using CMM-SVM tokens, and using the parallel rules versus not using the rules for each case.

4. CONCLUSIONS AND FURTHER WORK

Given this first working prototype with only a few rules written, half of the recipes could be extracted correctly. We find these results very encouraging, and we hope that once the system is fully implemented more than 70% of the procedures can be extracted correctly.

The goal of this system is to extract plan-like graphs for the procedures described in text. This involves finding what actions are involved in the procedure, and what relations between them exist. More complete understanding and extraction of procedures require background knowledge and reasoning to resolve ambiguities, to fill in unstated assumptions and to guide a more complex grammar or rule-set. This system by itself is a start for such understanding and could be useful in building large knowledge bases with some human effort.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the Defense Advance Research Projects Agency (DARPA) under Contract No. NBCHD030010 and by a generous gift from Cisco Systems, Inc.

6. REFERENCES

[1] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[2] W. W. Cohen. Minorthird: Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data, 2004.

[3] S. E. Fahlman. Marker-passing inference in the scone knowledge-base system. In *Proc. of the First International Conference on Knowledge Science, Engineering and Management*, pages 114–126. Springer-Verlag, 2006.

[4] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *Proc. of the 2nd IJCAI*, pages 608–620, London, UK, 1971.

[5] D. Klein, J. Smarr, H. Nguyen, and C. Manning. Named entity recognition with character-level models, 2003.

[6] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In E. Brill and K. Church, editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, Somerset, New Jersey, 1996.

[7] M. Veloso, J. Carbonell, A. Pérez, D. Borrajo, E. Fink, and J. Blythe. Integrating planning and learning: The PRODIGY architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 7(1):81–120.