DICTIONARY CONSTRUCTION BY DOMAIN EXPERTS

Ellen Riloff and Wendy G. Lehnert

Department of Computer Science
University of Massachusetts
Amherst MA 01003

Sites participating in the recent message understanding conferences have increasingly focused their research on developing methods for automated knowledge acquisition and tools for human-assisted knowledge engineering. However, it is important to remember that the ultimate users of these tools will be domain experts, not natural language processing researchers. Domain experts have extensive knowledge about the task and the domain, but will have little or no background in linguistics or text processing. Tools that assume familiarity with computational linguistics will be of limited use in practical development scenarios.

To investigate practical dictionary construction, we conducted an experiment with government analysts. We wanted to demonstrate that domain experts with no background in text processing could successfully use the AutoSlog dictionary construction tool [Riloff 1993]. We compared the dictionaries constructed by the government analysts with a dictionary constructed by a UMass researcher. The results of the experiment suggest that domain experts can successfully use AutoSlog with only minimal training and achieve performance levels comparable to NLP researchers.

AutoSlog is a system that automatically constructs a dictionary for information extraction tasks. Given a training corpus, AutoSlog proposes domain-specific *concept node* definitions that CIRCUS [Lehnert 1991] uses to extract information from text. However, many of the definitions proposed by AutoSlog should not be retained in the permanent dictionary because they are useless or too risky. We therefore rely on a human-in-the-loop to manually skim the definitions proposed by AutoSlog and separate the good ones from the bad ones. Figure 1 shows a snapshot of the AutoSlog interface used to review potential dictionary entries.

Two government analysts agreed to be the subjects of our experiment. Both analysts had generated templates for the joint ventures domain, so they were experts with the EJV domain and the template-filling task. Neither analyst had any background in linguistics or text processing and had no previous experience with our system. Before they began using the AutoSlog interface, we gave them a 1.5 hour tutorial to explain how AutoSlog works and how to use the interface. The tutorial included some examples to highlight important issues and general decision-making advice. Finally, we gave each analyst a set of 1575 concept node definitions to review. These included definitions to extract 8 types of information: jv-entities, facilities, person names, product/service descriptions, ownership percentages, total revenue amounts, revenue rate amounts, and ownership capitalization amounts.

We did not give the analysts all of the concept node definitions proposed by AutoSlog for the EJV domain. AutoSlog actually proposed 3167 concept node definitions, but the analysts were only available for two days and we did not expect them to be able to review 3167 definitions in this limited time frame. So we created an "abridged" version of the dictionary by eliminating jv-entity and product/service patterns that appeared only infrequently in the corpus.[1] The resulting "abridged" dictionary contained 1575 concept node definitions.

We compared the analysts' dictionaries with the dictionary generated by UMass for the final Tipster evaluation. However, the official UMass dictionary was based on the complete set of 3167 definitions originally proposed by AutoSlog as well as definitions that were spawned by AutoSlog's optional generalization modules. We did not use the generalization modules in this experiment, due to time constraints. To create a comparable UMass dictionary, we removed all of the "generalized" definitions from the UMass dictionary as well as the definitions that were not among the 1575 given to the analysts. The resulting UMass dictionary was a much smaller subset of the official UMass dictionary.

Analyst A took approximately 12.0 hours and Analyst B took approximately 10.6 hours to filter their respective dictionaries. Figure 2 shows the number of definitions that each analyst kept, separated by types. For comparison's sake, we also show the breakdown for the smaller UMass dictionary.

---

[1] While processing the training corpus, AutoSlog keeps track of the number of times that it proposes each definition (it may propose a definition more than once if the same pattern appears multiple times in the corpus). We removed all jv-entity definitions that were proposed < 2 times and all product/service definitions that were proposed < 3 times. We eliminated jv-entity and product/service definitions only because the sheer number of these definitions overwhelmed the other types.

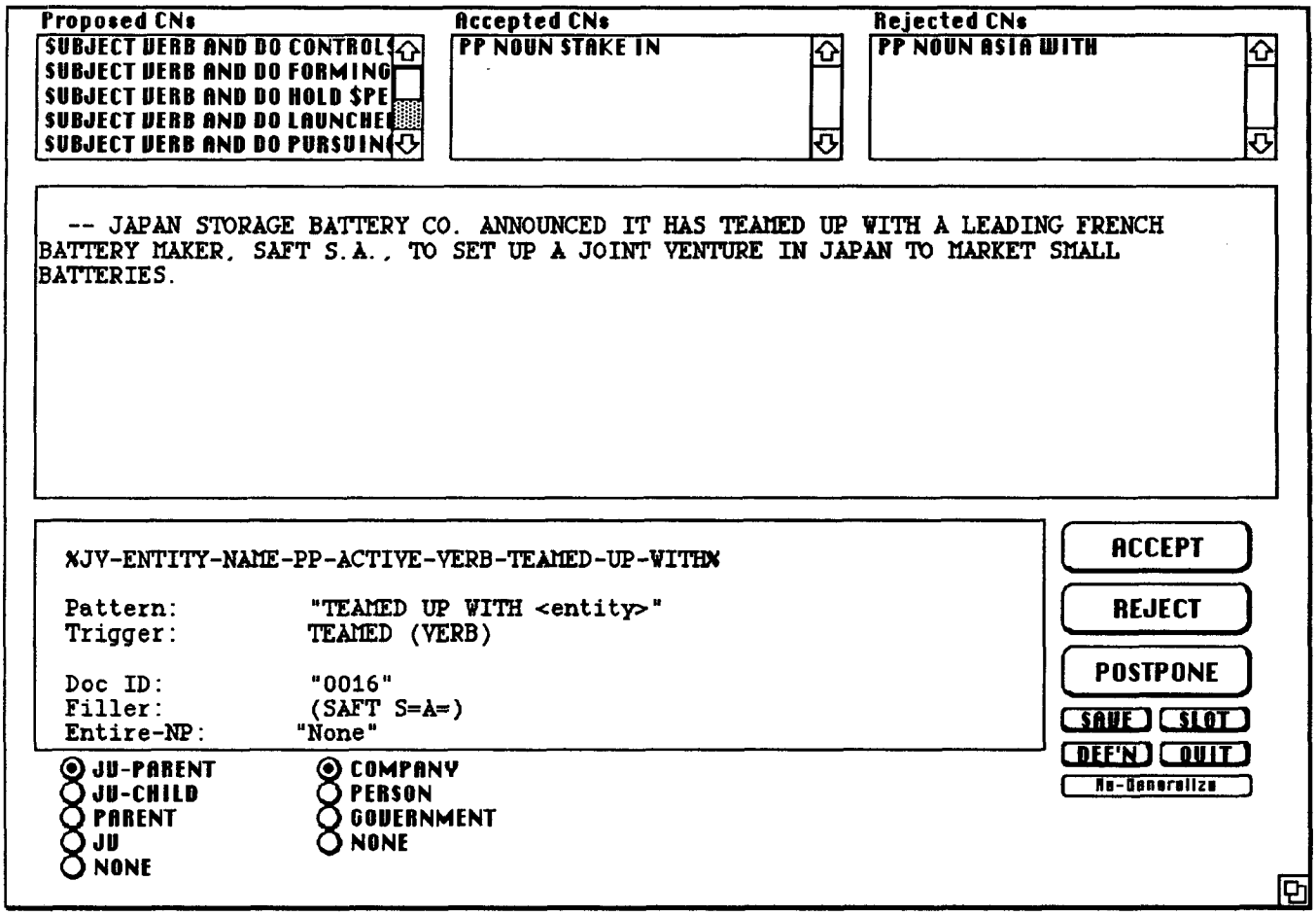

Figure 1: The AutoSlog Interface Tool

CN Type	# proposed by AutoSlog	# kept (UMass)	# kept (Analyst A)	# kept (Analyst B)
entity	688	311	357	423
facility	80	20	16	55
ownership-percent	174	91	117	91
person	243	119	149	52
prod_serv	316	76	152	44
revenue-rate	19	14	12	16
revenue-total	30	22	15	26
total-capitalization	25	14	13	22
TOTAL	1575	667	831	729

Figure 2: Comparative Dictionary Sizes

We compared the dictionaries constructed by the analysts with the UMass dictionary in the following manner. We took the official UMass/Hughes system, removed the official UMass dictionary, and replaced it with a new dictionary (the smaller UMass dictionary or an analysts' dictionary). One complication is that the UMass/Hughes system includes two modules, TTG and MayTag, that use the concept node dictionary during training. In a clean experimental design, we should ideally retrain these components for each new dictionary. We did retrain the template generator (TTG), but we did not retrain MayTag. We expect that this should not have a significant impact on the relative performances of the dictionaries, but we are not certain of its exact impact. Finally, we scored

each new version of the UMass/Hughes system on the Tips3 test set. Figure 3 shows the results for each dictionary.

The F-measures (P&R) were extremely close across all 3 dictionaries. In fact, both analysts' dictionaries achieved slightly higher F-measures than the UMass dictionary. The error rates (ERR) for all three dictionaries were identical. But we do see some variation in the recall and precision scores. We also see variations when we score the three parts of Tips3 separately (see Figure 4).

In general, the analysts' dictionaries achieved slightly higher recall but lower precision than the UMass dictionary. We hypothesize that this is because the UMass researcher was not very familiar with the corpus and was therefore somewhat conservative about keeping definitions. The analysts were much more familiar with the corpus and were probably more willing to keep definitions for patterns that they had seen before. There is usually a trade-off involved in making these decisions: a liberal strategy will often result in higher recall but lower precision whereas a conservative strategy may result in lower recall but higher precision.

It is interesting to note that even though there was great variation across the individual dictionaries (see Figure 2), the resulting scores were very similar. This may be because some definitions can contribute a disproportionate amount of performance if they are frequently triggered by a given test set. If the three dictionaries were in agreement on that subset of the dictionary that is most heavily used, those definitions could dominate overall system performance. Some dictionary definitions are more important than others.

To summarize, this experiment suggests that domain experts can successfully use AutoSlog to build domain-specific dictionaries for information extraction. With only 1.5. hours of training, two domain experts constructed dictionaries that achieved performance comparable to a dictionary constructed by a UMass researcher. Although this was only one small experiment, the results lend credibility to the claim that domain experts can build effective dictionaries for information extraction.

BIBLIOGRAPHY

Lehnert, W. (1991). Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. *Advances in Connectionist and Neural Computation Theory. Vol. I.* (ed: J. Pollack and J. Barnden) Ablex Publishing, Norwood, New Jersey. pp. 135-164.

Riloff, E. "Automatically Constructing a Dictionary for Information Extraction Tasks". *Proceedings of the Eleventh National Conference on Artificial Intelligence. 1993.* pp. 811-816.

TIPS3	Recall	Precision	P&R	ERR
UMass/Hughes	18	51	27.06	83
Analyst A	19	47	27.39	83
Analyst B	20	47	27.89	83

Figure 3: Comparative Scores for Tips3

TIPS3/Part1	Recall	Precision	P&R	ERR
UMass/Hughes	18	51	27.04	83
Analyst A	20	48	28.00	82
Analyst B	22	47	29.69	81

TIPS3/Part2	Recall	Precision	P&R	ERR
UMass/Hughes	17	52	26.03	84
Analyst A	18	48	25.92	84
Analyst B	20	47	27.75	83

TIPS3/Part3	Recall	Precision	P&R	ERR
UMass/Hughes	20	50	28.12	82
Analyst A	20	46	27.96	82
Analyst B	17	48	25.25	84

Figure 4: Comparative Scores for Part1, Part2, and Part3