User-Oriented Machine Learning Strategies for Information Extraction: Putting the Human Back in the Loop

David Pierce and Claire Cardie
Department of Computer Science
Cornell University
Ithaca NY 14853
pierce, cardie@cs.cornell.edu

Efforts in information extraction (IE) have concentrated on fundamental issues concerning the viability of the technology. Two of these important issues are scalability and portability. To illustrate, consider the AutoSlog system for learning a dictionary of extraction patterns [Riloff, 1993]. AutoSlog automatically acquires extraction patterns based on training documents annotated with extraction templates, thus eliminating the need for a human programmer to encode patterns or rules for extraction. Even further, more recent versions of AutoSlog acquire these patterns without explicit training data, using either relevancy signatures [Riloff, 1996] or mutual bootstrapping with a semantic lexicon [Riloff and Jones, 1999], eliminating the need for a human annotator to produce training data from raw documents. In short, to address the issues of scalability and portability, it has been important to find ways to get the human out of the loop, both as a programmer encoding patterns, and as an annotator producing training data.

Scalability and portability have been useful and important issues for driving research in information extraction. However, as IE technology matures, it becomes increasingly relevant to consider issues concerning the deployment of IE applications for real users. Real users will not be experts in machine learning or text processing, nor will they care how extraction patterns are encoded or acquired. They will, however, be uniquely qualified experts in the identification of the specific information they wish to locate and extract. Real IE applications must allow users to specify the nature of the information structures they desire while shielding them from the details of how the system locates new structures. Assuming users specify their information needs by providing examples, there is a natural tension between two criteria of *coverage* and *responsiveness*. Coverage is the system's ability to extract all desired information for the user, i.e. to completely cover the task; coverage encourages the system to demand training examples from the user. In contrast, responsiveness is the system's ability to achieve a reasonable level of performance without undue burden upon the user; responsiveness thus discourages the system from demanding training examples.

This position paper outlines a *user-oriented* meta-learning strategy that attempts to balance coverage and responsiveness when training examples are solicited from a real user. Following a high-level description of the strategy, we discuss some related experiments that indicate the promise of this learning paradigm.

The key idea behind user-oriented learning is the recognition of the complementary strengths of the human user and the IE system. On one hand, the human is proficient at judging an information structure as desirable or undesirable; on the other, the machine is proficient at rapidly locating similar examples from large quantities of text. Both human and machine are allowed to exercise their strengths in a scenario where the human begins by providing training examples, the machine attempts to locate additional instances based on the training examples, the human responds by confirming the desirability (or undesirability) of the new instances, the machine adds them to the training examples and continues to search for more instances, and so on. Figure 1 sketches this process as three interleaving procedures. The procedures access a collection of documents D, and sets of positive, negative, and unconfirmed examples E^+, E^-, and $E^?$, respectively. In the ANNOTATE procedure, the user explicitly provides training data, while in the CONFIRM procedure, the user judges the desirability of instances located by the system.[1] Finally, in the LOCATE procedure, the system finds new instances on the basis of all confirmed training data.

From the point of view of the user, user-oriented learning manages the tension between accuracy, coverage, and responsiveness very naturally. By interleaving the ANNOTATE and CONFIRM procedures, we allow the user to decide what priority is given to either criterion. If better coverage is desired, the user may wish to annotate more examples. If better accuracy is desired, the user may wish to review and confirm more instances. The perceived responsiveness, of course, depends upon the patience of the user, but we note that user-oriented learning—having the system locate candidate instances and require only a simple yes/no response from the user—is a vast improvement over annotating all the training data by hand.

[1] If the learner uses exclusively positive examples, the user might either correct, or simply discard, negative examples during the confirmation phase.

ANNOTATE
allow the user to annotate an example x from D
add x to E^+

LOCATE
retrain on E^+ and E^-
locate new candidate examples X from D
add X to $E^?$

CONFIRM
allow the user to select x from $E^?$
if the user marks x correct
add x to E^+
else
add x to E^-

Figure 1: User-Oriented Learning

Before concluding our brief description of user-oriented learning, it is interesting to note some parallels between user-oriented learning, active learning, and weakly supervised learning. Active learning [Cohn *et al.*, 1994] is a fully supervised strategy whose goal to process training examples in their most useful or informative order in the hope of reducing the total number of training examples required to reach a given level of performance. Useful examples are presented to the user for annotation, then added to the growing body of training data. Weakly supervised learning strategies (e.g. cotraining [Blum and Mitchell, 1998]) also seek to reduce the required amount of training data, in a more drastic fashion, by using additional unlabeled data to improve performance. This is done by allowing the learner to label instances from the unlabeled for subsequent use as its own training data. User-oriented learning can be viewed as a hybrid of active learning and weakly supervised learning, in that on one hand the user is requested to process certain examples, and on the other, the system also locates and labels examples for itself. However, user-oriented learning has partial advantages over both of these other strategies. In contrast to active learning, the user is only required to confirm the desirability of new examples (in the confirmation phase)—a yes/no decision—rather than locate and annotate them from scratch. And in contrast to weakly supervised learning, the system does not rely on its own classification of newly located instances, instead receiving definitive judgments from the user. Thus, due to its use of a combination of annotation and confirmation, we might call user-oriented learning a *moderately supervised* method.

Next we briefly touch upon experimental results that indicate the promise of user-oriented and similarly moderately supervised approaches to learning. Our empirical evidence derives from a study of base noun phrase identification [Pierce and Cardie, 2001]. Here we sketch the results of the study without delving into the details of base noun phrase identification. In this study, a number of experiments investigated the use of the incremental cotraining algorithm of Blum and Mitchell [1998]. In an initial experiment, we found that the performance of cotraining was dulled due to mistakes made by the learner in labeling its own training data. To resolve this problem, we tested a moderately supervised variant of cotraining in which the instances selected and labeled were subsequently corrected by a human annotator. The moderately supervised variant was more successful than its counterpart, but it exhibited a slight deficiency in task coverage. We believe that this deficiency can be eliminated using a hybrid algorithm combining cotraining with active learning. The resulting algorithm would be quite similar to user-oriented as described above.

To summarize, we have presented a user-oriented, moderately supervised approach to learning motivated by the prospect of building information extraction systems that can be configured by real users by providing examples that indicate their information need. User-oriented learning puts the human back in the loop to leverage his or her ability to provide feedback on the progress of the learning process. Experiments with related learning processes suggest that moderately supervised methods similar to user-oriented learning will address the problems of task coverage and training data degradation suffered by incremental weakly supervised algorithms such as cotraining.

References

[Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*, 1998.

[Cohn *et al.*, 1994] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[Pierce and Cardie, 2001] D. Pierce and C. Cardie. Co-training for large text processing tasks: A case study using base noun phrase identification. 2001. Submitted for publication.

[Riloff and Jones, 1999] E. Riloff and R. Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 474–479, 1999.

[Riloff, 1993] E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816. American Association for Artificial Intelligence, 1993.

[Riloff, 1996] E. Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049, 1996.