Considering Automatic Aids to Corpus Annotation

David Day and Benjamin Wellner
MITRE Corporation
Mail Stop K329
202 Burlington Road
Bedford, MA 01730, USA
day@mitre.org, wellner@mitre.org

Abstract

In this paper we view mixed-initiative corpus annotation from the perspective of knowledge engineering, and discuss some of the opportunities, challenges and dangers that are presented by using mixed-initiative annotation tools. We begin this discussion by describing an existing mixed-initiative annotation tool for open-ended phrase-level annotation, the Alembic Workbench. We discuss how this tool currently operates, the nature of its skill acquisition component, and our plans to extend it in a number of ways, including incorporating an active learning capability. Having set the stage with a concrete example, we identify a number of opportunities and challenges that are presented by the mixed-initiative approach to corpus annotation, including the benefits that might accrue when supporting "layered" annotation environments, the adoption of intensional/procedural annotation paradigms, the inclusion of lexical resource construction interleaved with corpus annotation, and other topics.

1 Introduction

One way of viewing corpus annotation is as a form of "knowledge engineering," where the *annotator* intends to enable a machine to reproduce the behavior being performed. A motivation for adopting such a view is that there is a practical interest in having machines be able to automatically perform some types of annotation. For example, "named entity" tagging, the ability to identify proper names that refer to entities of a particular restricted set of semantic classes (e.g., person, location, organization) was initially developed merely as a means to measure the contribution of this stage of linguistic analysis to a set of more complex domain-specific information extraction tasks. In recent years this capability has been shown to be valuable as a constituent to quite different information processing tasks, including topic detection and tracking, information retrieval, and others.

Another motivation for automating the annotation process is simply to increase the productivity of the corpus annotation process itself. Even if the ultimate goal of a particular annotation process is to build a static repository of annotated data to support fundamental linguistic research and analysis, there is a great benefit in producing as much of a given type of annotation as possible within restricted schedules and budgets. In general, the greater the size of the corpus, the more informed and statistically well-founded are the conclusions that can be drawn. From the point of view of knowledge engineering, most forms of corpus annotation involve a model of "learning by example," where some number of positive examples are meant to drive the skill acquisition component. In practice this skill acquisition is often carried out by a mix of human engineering (e.g., programming), machine-aided analysis, and machine learning techniques when possible. In this paper we want to expand on this skill acquisition model in a number of ways:

- Argue how these techniques can and should be applied across the full range of linguistic annotation tasks.
- Expand the notion of "mixed initiative" (or "incremental bootstrapping") annotation to incorporate not just learning by example, but other methods that increase the

expressive power of the "annotator" to influence skill acquisition.

- Encourage the use of "earlier" language processing stages in the annotation of later stages.
- Focussed corpus selection and annotation through "active learning" (or "sample selection").
- Common annotation frameworks and tools can help to increase these bootstrapping capabilities.

2 Mixed initiative corpus development

The notion of using partially machine-annotated data to "bootstrap" the human annotation process dates back at least to Brill's Ph.D. thesis (Brill, 1993), and probably earlier. The bootrapping procedure operates on the observation that there are many data points in some annotation tasks that are quite easily performed computationally. Even relatively poor performing procedures can prove effective for increasing productivity if there is a sufficiently large amount of data that is annotated correctly *and* if the labor required to fix the remaining bootstrapping errors is relatively small compared to the baseline manual tagging effort. In such a situation the bootstrapping procedure will have increased the effective productivity of the human annotator by the degree of the bootstrapping procedure's accuracy. For large corpus collections, this can represent a sizable savings in human labor.

The bootstrapping procedure can take many forms, and it can be arrived at in many ways, either through annotator-derived heuristics, systematic analysis of the corpus annotated so far, or through more automatic means utilizing machine/statistical learning techniques. We use the term *mixed initiative* annotation to refer to an environment in which (a) the bootstrapping procedure is derived automatically and (b) it can be invoked at arbitrary points during the course of annotation. (The alternative term "incremental bootstrapping" has also been suggested.) Subsequent invocations of the bootstrapping procedure can perform better than earlier invocations as a function of new

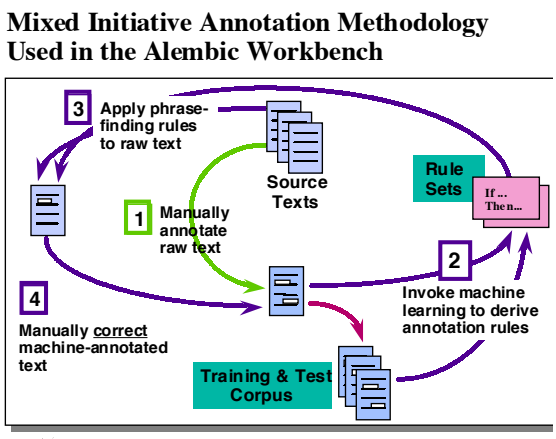

Figure 1: The mixed-initiative spiral model of corpus annotation.

evidence—usually in the form of a larger annotated corpus, since more examples are available to drive the bootstrapping procedure. Since first proposed, a number of tools have been built that provide mixed-initiative environments for a variety of annotation tasks.(Brants et al., 1997; Bennett et al., 1997)

3 Mixed-initiative annotation in Alembic Workbench

We have developed a mixed-initiative annotation tool for building phrase-level annotated corpora; we will describe it briefly here in order to place in context our more recent work as well as to ground our subsequent discussion of general issues regarding the opportunities, challenges and dangers presented by mixed-initiative approaches to enhancing annotation productivity. This tool, the Alembic Workbench (Day et al., 1997) (or simply Workbench in this article), induces a finite state transducer in the form of a sequence of transformation rules, which can then be used to bootstrap (annotate) similar textual data. The transformation-based learner (TBL) and the transformational rule sequence interpreter are both provided by the Alembic multi-lingual natural language processing system (Aberdeen et al., 1995; Vilain and Day, 1996). A graphical depiction of the mixed-initiative methodology adopted by the Workbench is shown in Figure 1.

The class of annotations susceptible to mixed-initiative annotation in the Workbench

might be best described as "phrasal"—any contiguous character sequence or multi-word sequence may be annotated by the user and associated with some "tag." The Workbench has been extended for a number of special purpose and general purpose annotation tasks, such as MUC6-style co-reference tagging, and general "relational" annotations. Neither of these tagging enhancements have yet been closely integrated with our machine-learning techniques—even though we are presently working on machine-learning approaches to both of these problems outside the specific scope of mixed-initiative annotation. These tags may be defined using the SGML/XML mechanisms for specifying generic identifiers of annotation elements and their associated attribute/value pairs. While this notational scheme allows for the expression of complex relationships among particular tags, the machine learning component currently used to support mixed-initiative annotation in the Workbench ignores these subtleties and reduces all SGML/XML elements with distinct structures into what are essentially unique, un-interpreted symbols.

The transformation-based rule sequence learning technique used in Alembic has proved very effective for deriving accurate annotation performance on the basis of exceedingly impoverished amounts of training data. We have observed F-measures in the range of 55-75 within 15-20 minutes of beginning a new annotation task in both English and Spanish texts, and lower but still helpful values in the same amount of annotation of Chinese and Japanese texts. (See section 3.1 for more background on the phrase-rule learning behavior of Alembic.) This is important in producing enhanced productivity through bootstrapping an automatic tagging procedure. The earlier the machine can provide pre-annotated data at a reasonable level of performance, the faster the combined activities of human and machine can build up a large corpus. In addition, the precision and recall of the automatically derived rule sequences increase as the size of the available training corpus increases. The shape of this learning curve is invariably asymptotic. The asymptote peaks at different levels (on blind test data) for different tasks and/or for different parameters of the learning environment, for reasons that are important, but not always easy to determine. The two main reasons appear to be the size and typicality of the corpus, and the representational power of the rule patterns available to the rule learner.

In building our current transformation-based learning (TBL) system we have made a number of design decisions that have had an impact on training speed. These decision have to do with the need to actually apply newly induced rules to the training data and subsequently recompute the corpus-wide statistics that drive the next rule induction cycle. While the system is able to achieve very fast learning times (5-30 minutes) on small to moderate amounts of training data (5,000 to 75,000 exemplars), much longer training times result when the training corpus reaches hundreds of thousands or even millions of exemplars.

Of course, there is no need to iterate the learning algorithm over all the training data after each document. Indeed, it seems reasonable and practical to invoke the learning algorithm less and less frequently as the size of the corpus increases and the performance of the automatically derived rule sequences differ less and less from each other with each incremental addition of a human-annotated file. In order to avoid having to restart the learning algorithm from an initial null state after each mixed-initiative invocation of the learning algorithm, it is straightforward within a TBL model to begin learning *on top of* of an existing level of competence (a previously derived rule sequence). In this case the training corpus might consist of only those few files that have been annotated since the last learning procedure was called, and the newly derived rules are concatenated onto the end of the existing rule sequence. However, eventually it is desirable to start fresh, since it becomes more and more likely that new opportunities for generalization can be found in a larger training set, leading to increases in the ability of the new rule sequence to apply successfully to unseen data.

Nonetheless, we are also interested addressing the problem of learning performance directly. One approach we intend to pursue is the incorporation of MITRE's HMM-based "Phrag" (Palmer et al., 1999) phrase-parsing

learner within the Workbench's mixed-initiative repertoire, which we imagine could be increasingly relied upon as the size of the training set reaches very large proportions.

Currently the default "granularity" of mixed-initiative annotation within the Workbench is that of a document or file. As long as a single file is fully annotated, it can be used as the basis of phrase-rule learning, either alone or in combination with a corpus consisting of other annotated documents/files. Of course, documents can be arbitrarily reduced to smaller chunks if there is a strong need for this. Ideally one would like the granularity used in mixed-initiative annotation (1) to be identified and adopted directly by the system itself, rather than relying on the annotator to make such decisions; and (2) to be a function of the annotation task being performed. For example, phrase tagging (and many other annotation tasks such as sentence parsing) could be segmented at the sentence level. Updating the existing pre-annotation procedure could be invoked based on the amount of performance improvement achieved in the previous two invocations, as well as other heuristics that designers might identify. Other annotation tasks, such as co-reference annotation, discourse structure and entity and relation extraction, etc., might *require* segmentation at the document level.

3.1 Why Alembic phrase rule learning appears to work

In the past few years we have often been surprised at the ability of Alembic's phrase-rule learning apparatus to create quite reasonable tagging performance with only meager amounts of data annotated to the user's specifications. We have often had cause to wonder: Was our learning algorithm and associated Alembic infrastructure really so good? How did it squeeze out such good performance (e.g., around 70-75 F-measure) on such a paltry example base as 1,500 words of annotated Spanish newswire text? We would like to perform a detailed analysis, but our informal conclusions are already leading us to establish new priorities in our attempts to build rapidly portable natural language processing capabilities. We devote a subsection to each of these conclusions below.

3.1.1 The "right" level of analysis

Like many other systems designed in the course of the last five years, Alembic has been built using a number of important natural language processing components, each of which had become newly available in the previous years. These components include tokenization (word segmentation), sentence tagging, and morphological analysis (part-of-speech tagging). Empirically it has become clear that many useful types of general-purpose and specialized phrase tagging tasks (from named entity tagging to sentence chunking) can find all of the information they need from this mix of information made available by pre-processing. In other words, this is due to a successful application of the "divide and conquer" principle adopted by the computational linguistics community as a whole over the past ten years.

3.1.2 Locality of influence

In a similar vein, these same tagging tasks (perhaps best exemplified by "named entity tagging") have adopted a decision environment in which a fairly strict locality of influence is respected, and this locality has been sufficient for addressing the phrasal phenomena of interest. Not only has this seemed to be true for the rule schemata used in rule-based tagging systems, but also for the modeling techniques adopted in Hidden-Markov Model approaches to phrase tagging as well.

3.1.3 The "right" lexical resources and built-in predicates

We believe that the most specific reason that the Alembic phrase rule learner has managed to perform so well with very limited amounts of training data has been the considerable lexical resources that we have made available to the learner. It so happens that when Alembic has exhibited these surprisingly good learning behaviors it is often the case that the resulting rules include a liberal mixture of references to one or another of the special-purpose word lists that we have developed in the course of manually building various natural language processing capabilities. (Other frequently occurring rule patterns exhibited in successful rule

sequences are those making use of the part-of-speech of lexical items. In CJK languages particular character prefixes and suffixes are also highly represented.)

These word lists are derived in a wide variety of ways: names extracted from the US Census; hand-coded lists expanded from core words easily predicted to be contextually important markers; expansions of words using thesauri, dictionaries or similar resources; words found from an analysis of the internal and external contexts of annotated phrases in manually tagged training data (supervised context analysis); and sometimes words found in these same contexts but from large collections of automatically tagged data (unsupervised context analysis). Regardless of the particulars of how they are derived, these resources allow for a boost in the generality of rules learned from a small corpus. While the learner might happen to pick references to these word lists for purely local (and perhaps almost arbitrary) reasons in the context of some very small annotated corpus, this serendipity will lead to many more correct applications when different word choices are encountered in previously unseen data.

Our current presumption is that replicating efficient mixed-initiative successes for other tasks and in other arenas of language processing will rely heavily on providing similar advantages as those identified above. For example, in order to support the rapid mixed-initiative annotation of certain types of relation/event data (e.g., the "template relation" and "scenario template" tasks of the various MUC evaluations (Def, 1995; Grishman and Sundheim, 1996)), one must make available to the learning component the same notion of "locality" as is warranted for such distinct phenomena. This kind of locality might be exemplified by an intermediate "SVO" (Subject/Verb/Object/modifier) representation of a given sentence, which could be derived in a variety of ways, either *via* treebank-style parses, or from dependency-like syntactic models such as "grammatical relations."). We are particularly interested in merging the mixed-initiative development of lexical resources with the mixed-initiative development of annotated corpora. (Previous work of others in this area includes (Riloff and Jones, 1999; Blum and Mitchell, 1998).) We anticipate that a host of unsupervised learning techniques will be especially useful in helping to quickly bootstrap the acquisition of useful word lists.

4 Active learning

One way of increasing the effective productivity of the human annotator while holding the capabilities of the skill acquisition component constant is by increasing the utility of the annotated data being supplied to the skill acquisition component. In the event that bootstrapping is being performed manually through heuristic insights, the annotator may try to tune the corpus sampling mechanism to favor sentences, paragraphs or documents that would seem to provide the greatest opportunity for instructing and testing the emerging automated annotation component. It is also possible to perform this sample selection of raw data through automatic means. This interplay between learner and example selection is sometimes referred to as "active learning" (or sample selection).(Lewis and Catlett, 1994)

Engelson and Dagan (Engelson and Dagan, 1996) demonstrated an automatic method for selecting part-of-speech training sentences using a votes from a set of automated annotation "experts." This and other work prompted us to look at how such techniques could be incorporated into the Workbench's mixed-initiative model. Alembic's phrase-rule learner contains a number of parameters that are well suited to construct such a family of experts.

The basic insight of active learning is that not all training data are equally informative, and that the "confidence" of the induced decision system in classifying (tagging) some particular exemplar is inversely proportional to the likely utility of that exemplar, were it to be correctly classified. If a particular annotation decision is made very confidently, it is likely due to the fact that many exemplars have informed the decision rule, and so increased the associated level of confidence. But how is "confidence" expressed in transformational rule sequences? In most cases, there is no analog to confidence in transformation rule sequences. However, if one can build a mixture of experts, then one analog to confidence in such systems is the number of experts that voted for the same tagging

1. Induce N different decision criteria by using varying parameter values.
2. Apply N decision criteria to unseen data.
3. Select for manual annotation those sentences for which there are sufficiently divergent classifications.
4. Annotate manually (with or without pre-tagging).

Figure 2: Active learning algorithm used in Alembic Workbench experiments

decision—independent of the nature of the decision mechanisms used in the constituent decision systems. The basic active learning algorithm used in our recent experiments with the Workbench is presented in Figure 2.

The Alembic transformation-based rule learning algorithm selects a rule at each epoch of the learning algorithm. We have experimented with a number of evaluation criteria for this step of the process: "yield minus sacrifice" (count the number of new, correct annotations created by applying a rule, then subtracting from this value the number of incorrect annotations created by applying this same rule); "log likelihood;" and "F-measure" (harmonic mean of the recall and precision measures for this rule), parameterized by *beta*, which indicates the relative weight given to the recall measure compared to the precision measure. We eventually adopted the F-measure approach, not only because it tended to give us the best empirical results on the problems we are addressing at that time, but also because it provided us the opportunity to transparently weight the performance more towards recall or more towards precision, which can be an important practical difference in various real world application contexts.

Varying the decision criterion by varying the *beta* value of the objective function allows us to easily define sets of experts from which "confidence" measures can be induced through their level of agreement. Indeed, the F-measure metric alone offers the opportunity for deriving a family of decision experts simply by modifying the single *beta* parameter. We would also like to use the Phrag HMM-based tagger on the same data to create an expert with a quite different bias. We are in the early stages of experimenting with this form of active learning, selecting sentences and/or documents on the basis of the degree to which multiple separately derived rule sequence "experts" agree on annotation assignments. These early results are encouraging. From an initial training set of 442 sentences (containing 705 target phrases), a subsequent unannotated corpus of 1,462 words was used as the universe of possible sentences for subsequent manual annotation. Approximately 10% were down-selected based on two different criteria: random selection or using low confidence measures as derived from voting as described above. Following the manual annotation of these two incremental additions to the training set, we observed that the performance of separately trained automatic taggers differed on test data by about 5% .[1]

5 Discussion

Corpus annotation has implications not just for providing productivity enhancements for linguists (computational and otherwise), but also as a model for how useful information extraction systems (an important class of intelligent agents) can be derived through a largely example-based knowledge engineering/acquisition process. With both of these contexts in mind, it is useful to reflect on some of the outstanding opportunities in mixed-initiative annotation, as well as the difficulties and dangers that accompany them.

5.1 Layered annotations for multi-staged mixed-initiative corpus development

Some annotation tasks depend so strongly on "earlier" annotations that it will become important to build annotation environments in which these earlier annotation "layers" are made apparent to the annotator. For example, when annotating *grammatical relations* (Ferro et al., 1999; Ferro, 1998), the job of the annotator is to establish various pre-specified types of relationships among sentence "chunks," where these chunks consist of simple phrases such as "noun group," "verb group," "preposition group," and

[1] We are in the midst of our explorations of this task; we hope to be able to report the results of more robust experiments soon.

the like. Thus, instead of being presented with a text in a standard Workbench textual display (and being able to draw relationships between arbitrary pairs of words), it is important that these sub-groupings are *already* visually apparent and made to control the interface so that asserting only group-level relationships is possible.[2]

Other opportunities for such interdependence of annotation tasks can be seen when annotating discourse-level relations and events (e.g., MUC-style "template relations" and "scenario templates"). While particular relationships may be asserted in a variety of ways, the ability to view and operate directly on, for example, an "SVO" (subject-verb-object-modifier) representation of a set of sentences might enhance not only the productivity of the annotator, but also build in important links across processing levels that are important to one's method of attacking a given computational linguistics problem. This ability to build upon the layers of annotation derived previously will become an increasingly important technique for building mixed-initiative annotation tools. It could prove especially fruitful in the support for a richer language by which the human annotator can directly influence the mixed-initiative process, as discussed in the next section.

5.2 From extensional to intensional annotation methods

We remarked earlier about how the bootstrapping of annotation can incorporate not just automatically derived annotation heuristics but also those derived from the human annotator, implemented usually as computer programs or simply regular expression macros. This has been a method frequently relied on within the computational linguistics community, since the skills for deriving the heuristics and implementing them as procedures are readily available. One of the open problems of mixed-initiative annotation environments is to provide some kind of support for more direct human intervention in the bootstrapping process other than simply adding yet another example. Of course, there a wide variety of pattern languages and annotation representations from which those inclined to write pre-annotation heuristics can choose. But are there ways in which the results of such heuristic annotation methods can be viewed and combined with example based annotations without creating confusion?

For example, if someone composes a rule and it applies to one hundred instances within a corpus, the annotator might like to view the resulting sentences directly—perhaps within a keyword-in-context type viewer—that is *also* integrated directly with the extensional annotation environment. This way exceptions to this rule can be noticed and modified easily and directly by the annotator. If this were truly a mixed-initiative environment, then such a system might on the next cycle derive a rule which *starts* with the human-authored heuristic, but derives rules (or some other representation) for capturing the exceptions identified extensionally by the annotator.

Interestingly, there has recently been a very careful empirical study (Brill and Ngai, 1999; Ngai and Yarowsky, 2000) exploring the advantages and disadvantages of extensional and intensional mixed-initiative methods for annotating a corpus. This study was carried out by Grace Ngai and David Yarowsky at Johns Hopkins University, and compared the abilities of relatively sophisticated pattern rule authors against machine learning methods for deriving tagging rules in a mixed-initiative annotation environment. The results indicate that rule-writing, while intuitively powerful, may prove difficult for supporting a mixed-initiative approach to corpus annotation. This is a provocative study, seeming at variance with our intuitions as computational linguists. The annotation community should explore these issues and discuss them fully.

5.3 The real world of task definition and collaborative development

In our own case studies and in our research focussed on mixed-initiative annotation we have often concentrated on well-defined annotation tasks and how they can most quickly be automated. In the real world, however, we know quite well from first-hand experience that the annotation process is a very long and tortuous

[2]Such an annotation tool has been developed specifically for the grammatical relations annotation task being performed internally at MITRE.

road, where many of the initial steps are concerned less with getting large amounts of annotated data quickly, but rather with exploring the very definition of the task at hand. As many of the contributors to formal language processing evaluations will tell you, much of the difficulty in starting up a new tagging task is due to the social and linguistic barriers to easy categorization. So how do the techniques we have described support and/or improve such task definition endeavors?

At the heart of any collaborative annotation effort is the detailed analysis and associated discussion of different interpretations of the linguistic phenomena, which is most often captured and brought to light through inter-annotator annotation analysis. At first this analysis is largely qualitative, and depends on *detecting* the anomalies in order to promote their discussion. Recently there has been a study of this collaborative behavior, and an associated automated method was developed that was modeled on it (Wiebe et al., 1999). Subsequently the emphasis moves towards quantitative inter-annotator analysis and the categorization of those differences. In both of these phases techniques that can boost the number and kinds of linguistic artifacts that have been annotated by one person or another can only help in the process of annotation understanding and inter-annotator reconciliation. Of course, it cannot sidestep the necessity of discussion and reflection that is necessary to come to terms with the motivations and other issues relevant to a new annotation task.

Nonetheless, there are clearly opportunities and challenges for mixed-initiative techniques that respect the collaborative nature of the annotation process. One area of interest is in building new automatic annotators by combining the existing annotation capabilities derived from separate human annotators interacting with mixed-initiative systems. For example, one could imagine new collaborative tasks could be defined through the application and analysis of distinct skills (tagging procedures, rule sequences, etc.) derived independently. This same ability may be appropriate for trying to identify and adapt to the inevitable "concept shift" that occurs with computational artifacts put to use on a daily basis.

5.4 The Tension between Naturally Occurring Phenomena and Focussed Inquiry

There is a potential danger that attends any technique that introduces labor saving methods, and mixed-initiative annotation is no exception. One of the most important problems is predicted to lie in the area of *recall*. As the automated pre-annotation process increases its capabilities, there will be a psychological tendency of human annotators to trust its guesses. And while precision errors will be fairly easy to spot (since the machine will display some text and assign a fallacious tag to it), *recall* errors—errors of omission—cannot be highlighted in principle, and so requires the human annotator to be forever vigilant and to notice "the tag that wasn't." This problem is perhaps accentuated even more with the adoption of active learning techniques. It is not known to what extent the introduction of active learning might introduce a vicious cycle of ignorance, whereby recall errors are never corrected due to tacit agreement (aligned errors) from all of the constituent decision components.

6 Conclusions

There are still opportunities for building, refining and applying mixed-initiative corpus annotation tools and environments. In this paper we have identified some of these opportunities, the challenges they pose and their potential for unintentional side effects. We grounded this discussion with a description of the Alembic Workbench tool, describing its current capabilities and the direction of our research to expand them. Successful mechanisms for quickly deriving machine-aided corpus annotation systems will have an important impact on the corpus linguistics research community. It will also lead eventually to portable, trainable language processing systems for use by non-specialists to perform customized information discovery and extraction from the glut of information available today.

References

John Aberdeen, John Burger, David Day, Lynette Hirschman, Patricia Robinson, and Marc Vilain. 1995. Description of the alembic system used for MUC-6. In *Proceedings of*

the Sixth Message Understanding Conference (MUC-6), pages 141–155, Columbia, Maryland, November.

Scott W. Bennett, Chinatsu Aone, and Craig Lovell. 1997. Learning to tag multilingual texts through observation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence, RI, USA.

A. Blum and T. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *In Proceedings of the 11th Annual Conference on Computational Learning Theory. ACM.*

Thorsten Brants, Wojciech Skut, and Brigitte Krenn. 1997. Tagging grammatical functions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, Providence, RI, USA.

Eric Brill and Grace Ngai. 1999. Man vs machine: A case study in base noun phrase learning. In *Proceedings of Association of Computational Linguistics*. Association of Computational Linguistics.

Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, University of Pennsylvania, Philadelphia, Penn.

David Day, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson, and Marc Vilain. 1997. Mixed-initiative development of language processing systems. In *Fifth Conference on Applied Natural Language Processing*, Washington, D.C., March. Association for Computational Linguistics.

Defense Advanced Research Projects Agency. 1995. *Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland, November. Morgan Kaufmann.

Sean P. Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. *Computation and Linguistic E-Print Service*, cmp-lg/9606030, June.

L. Ferro, M. Vilain, and A. Yeh. 1999. Learning transformation rules to find grammatical relations. In *Computational natural language learning (CoNLL-99)*, pages 43–52. EACL'99 workshop, `cs.CL/9906015`.

L. Ferro. 1998. Guidelines for annotating grammatical relations. Unpublished annotation guidelines.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference: A brief history. In *International Conference on Computational Linguistics*, Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

David Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning: Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156, San Francisco, CA. Morgan Kaufmann.

Grace Ngai and David Yarowsky. 2000. Rule writing or annotation: Cost-efficient resource usage for base noun phrase chunking. In *Proccedings of The 38th Annual Meeting*. Association for Computational Linguistics.

David D. Palmer, John D. Burger, and Mari Ostendorf. 1999. Information extraction from broadcast news speech data. In *Proceedings of the 1999 DARPA Broadcast News Workshop (Hub-4)*, February.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Nth National Coreference on Artificial Intelligence*. American Association for Artificial Intelligence.

Marc Vilain and David Day. 1996. Finite-state parsing by rule sequences. In *In International Conference on Computational Linguistics*, Copenhagen, Denmark, August. The International Committee on Computational Linguistics.

Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics*, pages 246–253. Association of Computational Linguistics.