Machine Learning for Information Extraction from Online Documents

Dayne Freitag
dayne@cs.cmu.edu
School of Computer Science
Carnegie Mellon University

Abstract

The field of *information extraction* (IE) is concerned with applying natural language processing (NLP) to extract essential details from text documents automatically. Recent results have demonstrated the viability of this idea for collections of journalistic prose in narrow domains. The idea's appeal and applicability, however, is much broader than work to date seems to imply. In particular, the online environment is a rich source of domains in which an IE system would prove useful. Examples include collections of medical diagnoses that have been transcribed for machine storage, Internet recipe collections, official university Web sites, and "bboard" seminar announcements. If IE is to have an impact, however, it must be possible to adapt to new domains with a minimum of manual tuning. Furthermore, traditional IE regards a document only as a grammatical object; this is too limiting and may lead to less accurate extraction than is possible when other kinds of information, such as mark-up or layout, are available.

The ultimate goal of this research is a system of IE tools which is retargetable. Rather than manually optimize an extraction system for each new domain, it should be possible to bootstrap it by demonstrating extraction on sample documents. I intend to show that such a system can be built using machine learning methods. A major theme will be the use of as much document information as possible, linguistic and non-linguistic. I will investigate how multiple learners can be employed and their various biases exploited to maximize system performance.

1 Introduction

In addition to its other uses, the computer has become increasingly important as a communication device. In this role, the machine is less a tool than a medium for information exchange. As such, the computer offers distinct advantages over older technologies, including permanence of information, convenience in manipulating information, and, of course, sheer volume. This last advantage is also a curse, known in its negative form as "information glut."

Much of the information communicated by computer takes the form of text. Rather than replace the printed word, the computer has, if anything, increased its importance. Unfortunately, text is not as easily manipulated by machine as are more machine-oriented information formats. Thus, while the computer exposes the user to larger quantities of textual information, to date it has done little to mitigate the attendant overload.

Various fields of study have been conceived to address this problem. In particular, the fields of *information retrieval* and *information filtering* attempt, through relatively simple statistical techniques, to buffer the user at the document level, fetching or screening off whole documents based on some notion of the user's needs. A somewhat more recent approach to the problem attempts to summarize documents in terms of the essential details they contain, allowing the user to survey document collections more easily. This last approach is known as *information extraction* (IE).

This thesis explores the possibility of information extraction in the kind of domain that arises naturally wherever humans use computers to store or transmit textual information. In particular, we are interested here in domains where there is no guarantee of grammaticality. We concentrate on extraction from electronic natural language as it occurs in the wild. We claim that machine learning techniques can be applied successfully to such domains.

1.1 Applications

Anyone with much experience in an online environment will need little convincing of the potential usefulness of an IE system for the typical online document collection. To ground the discussion, and to give an idea of the variety of problems I am considering, I list four such domains, along with possible applications.

Patient records Transcribed doctor's reports of patients admitted to hospitals find their way into large text databases. Such records detail things like patient symptoms, diagnosis, treatment, and result. Processing large collections of such records would enable important statistical analyses.

Online recipes Huge collections of recipes are available online, most of them inaccessible beyond the keyword-search level. Reliable extraction would permit a frame-based representation of recipes and facilitate access at a more conceptual level.

University Web sites While the arrangement of pages differ, official university Web sites all have certain stereotypic components, such as listings of servers or pointers to personal pages. Identifying such typical links can be viewed as an extraction problem; solving it would facilitate autonomous computer navigation of such sites.

Seminar posts Seminar announcements at CMU are frequently made via electronic "bboards." The ability to extract details, such as title, date, time, location, etc., would make possible the efficient monitoring of bboard posts for meetings of interest. A personal calendar agent might use such a facility.

1.2 Information Extraction

Although the idea of automatically capturing the gist of text is not new, the field of information extraction is relatively young. It was conceived in the late 1980's out of a desire to put existing NLP technologies to practical use. In terms of linguistic sophistication, the quintessential information extraction task, as defined for this young field, lies somewhere between information retrieval and full text understanding.

1.2.1 MUC

Under ARPA's leadership the IE community has lately come of age, convening almost annually as the Message Understanding Conference (MUC) [1] [2]. At this conference, participants' systems are subjected to rigorous blind tests on text domains drawn from the real world. The performance of tested systems has shown enough promise, while leaving plenty of room for improvement, to excite considerable interest in the field. A study involving the data from one domain used in MUC-5 suggested the gap between machine and human performance was about 20-30%; humans achieved 82% precision with 79% recall, while the best automatic systems got 57% precision with 53% recall [42]. These figures suggest that this domain is a challenging one for current automatic extraction methods.[1] Although there is variation in performance across MUC domains, which range from newswire articles on terrorism and corporate mergers to technical texts on microelectronics, in no case can the extraction problem be considered to have been solved by any system.

In fact, MUC domains are selected with an eye to pushing the NLP envelope. Although there is considerable diversity of approach, as revealed by MUC proceedings, most IE researchers are interested in exercising their NLP technology. Lately, this has changed slightly, as systems employing "quick and dirty" approaches and machine learning components have yielded results competetive with the more NLP-oriented systems.

1.2.2 Current approaches

According to an active member of the MUC community, the "generic" information extraction system is a pipeline of ten components, ranging from preprocessing modules and filters, to linguistic components for syntactic and semantic analysis, to post-processing modules which construct a final answer [14]. More interesting for this thesis than the identification of these components in MUC systems is their use of machine learning methods. In the paragraphs that

[1] They also imply that it is hard to define the information extraction task in a domain-independent way. The relatively low human perforance points to variations in interpretation and the difficulty of rigorous definition. In general, therefore, it seems desirable either to use such figures for strictly comparative purposes, or to interpret them with some application in mind.

follow therefore, I sample some of the systems described in [1] and [2] with this in mind.

A survey of the field reveals strong systems which, to judge from their descriptions, contain no significant machine learning components. Among the competitive systems in this class are those which rely on semantic analysis, as in standard NLP [17] [16], and those which attempt a cheaper approach but still require considerable manual engineering [15]. The widest use of machine learning in the field appears to be in filtering, deciding which portions of a document to subject to expensive further analysis, and which to discard as irrelevant [20]. This problem can be tackled, for example, at the paragraph level, and standard statistical approaches can be applied [12]. Another common application of learning is language modeling (e.g., [41]) for the construction of probabilistic parsers.

The most interesting learning developments, from the perspective of this thesis, involve the acquisition of probabilistic extraction patterns. This first appears in the Hughes Trainable Text Skimmer, in which both a k-nearest neighbor and a Bayesian classifier are applied to the problem of estimating the likelihood that a text fragment is an instantiation of a target field [6]. This approach sees further development in the CIRCUS system [18], where textual *triggers*, as introduced in the FASTUS system [4], are generalized into *concept nodes*: syntactic extraction patterns centered on individual words which are acquired through statistical analysis of a training corpus.

This approach has proven successful and has lead to further interesting developments. In the original Hughes/Umass system, the concept node dictionary was "learned" using a corpus annotated with field labels. For each field AutoSlog (the pattern acquiring component) took statistics on words and phrases occurring in syntactic patterns involving the field. From these words and phrases a raw collection of concept nodes was accumulated. Those patterns which indicated the presence of a field with high probability were retained. This collection was then manually reviewed to build the dictionary.

This approach subsequently evolved in two distinct directions. AutoSlog became AutoSlog-TS, doing away with the need for an annotated corpus, relying instead on two document sets, relevant and irrelevant [30]. In parallel work, AutoSlog became CRYSTAL, which searches the space of candidate concept nodes more thoroughly, relying on a specific-to-general ordering in the spirit of the Version Space algorithm [36] [22]. Other interesting work on the same system involves using information extraction as a basis for document classification and using machine learning for co-reference resolution [29] [37].

1.2.3 Departures from traditional IE

Although much of the work described above serves as an inspiration for this proposal, our focus is different from that of "traditional" IE. All MUC domains require, and most systems fielded at MUC emphasize, a NLP-intensive approach to extraction. In order for such systems to work to their full potential, documents in a collection must consist of formal prose written to a fairly high standard.

Such collections are the exception in the online environment. Real online language not only tends to play fast and loose with grammar, but it also abounds in extra-grammatical features, such as tables and ad hoc layouts. This is not to say that online text is chaotic and unpredictable. Rather, grammatical prose plays only a partial role in the universe of online communication, the size of its role depending on the domain. Moreover, an assumption of this thesis is that both grammatical prose and extra-grammatical devices are used *consistently* within a particular domain. The degree of consistency in part determines the appropriateness of the methods this thesis will explore.

Required for robust information extraction, therefore, is a system that attends not only to syntax and semantics, but also whatever other textual devices may be adopted by humans for communication via online text. Furthermore, because we cannot say in advance how all these devices will be applied in a particular domain, we want this system to be adaptive.

1.3 Thesis

The thesis of this research is that it is possible to construct such a system using machine learning methods. Retargeting this system for a new domain should involve as little as identifying and labeling relevant textual fields in a sample with a convenient interface. In this project, the validity of this thesis will be tested along several dimensions:

Generality What machine learning methods are applicable, and how can they be employed so as to be easily adapted to new domains?

Applicability What are the characteristics of domains for which these methods are appropriate, and how prevalent are they?

Efficiency How does retargeting of these methods compare with manual configuration for a new domain?

In this research several machine learning approaches will be implemented and their usefulness for extraction demonstrated in several real-world domains. The ultimate goal of this research is the construction of a machine learning architecture which can be easily retargeted to novel domains and which requires as little supervision as possible. In order for such a system to be realized, several questions must be answered, including:

- How do we exploit extra-grammatical features such as those describing document layout?
- How do we combine the predictions of multiple learners with differing biases for improved extraction performance?

- How do we best take advantage of the most expensive resource, the human trainer? What strategies can we find for maximizing the power of machine learning while minimizing the need for human intervention?

In addition to implementing and testing several machine learning approaches to extraction, this research will address some or all of these questions.

2 Techniques

This thesis can be regarded as an investigation into the automatic learning of textual formatting conventions, where "formatting" is understood in a broad sense. Stereotypic language in grammatical text, for example, can also be considered formatting to the extent that it doesn't require understanding for useful extraction. The author of a series of seminar announcements may always introduce the seminar location with the phrase, "We will meet in. . . ." I hope to show that such patterns can be acquired readily.

Formatting can occur at different levels in an electronic document. An important goal of this thesis is to exploit information from all such levels. Rather than focus on linguistically intensive techniques, as in traditional IE, I hope to make use of multiple techniques, from simple to sophisticated, and to combine them as intelligently and seemlessly as possible. In this section, after providing a formal statement of the central problem, I list some of the levels of abstraction, or document *views*, where we might look for formatting. Each of these views constitutes a different search space; the bias of a particular learner will be more or less appropriate depending on the view. Thus, I end the section with a listing of learners I consider appropriate to this problem with a brief discussion of how each might be applied.

2.1 Formal Problem Statement

A document D is the primary unit with which the system will be trained and its performance evaluated. At its most primitive level, D is simply a block of ASCII.

The *terms* mapping, $\mathcal{T}(D) = t_1, \ldots, t_n$, maps a document to a sequence of terms, where a term is a humanly meaningful unit of text, either a word or a unit of punctuation. I include punctuation in this mapping, because $\mathcal{T}$ is the mapping in terms of which the main problem is defined, and because extracted text may be less intelligible without punctuation.

A *field* from a document D, $f = t_i, \ldots, t_j$, is a contiguous subsequence of $\mathcal{T}(D)$. A *field extractor*, $\mathcal{F}(D) = f$, maps from an document D to a field from D, or to the null field. A meaningful field extractor presents a consistent semantic picture over a set of documents, always extracting from seminar announcements, for example, the field describing the location of the seminar. Of course, for this to happen, the document set itself must have some semantic integrity.

Note that we can implement a field extractor $\mathcal{F}$ in terms of an auxiliary function, $\mathcal{G}(D, f) = \mathbf{R}$, which maps from a field to a real number representing the system's confidence that f is the field $\mathcal{F}$ should return. Given such a function $\mathcal{G}$, implementing $\mathcal{F}$ may involve as little as iterating over a document, presenting $\mathcal{G}$ with fields of appropriate size, and picking the field on which $\mathcal{G}$'s output is highest. The function $\mathcal{G}$ is similar in form to functions learned in document classification. Extraction, therefore, can be viewed as a kind of text classification. Consequently, traditional induction algorithms can be adapted to the problem of IE.

An *information extraction function*, $\mathcal{E}(D) = \langle f_1, \ldots, f_m \rangle$, defines an information extraction problem. Such a function can be realized as a tuple of field extractors.

Given a set of documents $\mathbf{D}$, then, which is a subset of some semantically consistent universe of documents, and mappings for an extraction function $\mathcal{E}$, the goal of learning is to find a function $\widehat{\mathcal{E}}$ that best approximates $\mathcal{E}$.

2.2 Document Views

Before learning can take place, a commitment must be made to a particular representation of the problem at hand, the so-called instance language. In the case of IE or information retrieval, we are concerned with ways to represent documents or pieces of documents, *document views*. A "bag-of-words" representation, for example, effectively sees a document as a set of alphabetic tokens. For this thesis, we want to allow for multiple views in the hope that each view will contribute information useful for extraction.

2.2.1 The ASCII view

A document D is first and foremost a block of ASCII data. Other views can be seen as mappings over the ASCII document object.

2.2.2 The terms view

D is a sequence of terms. The bag-of-words model, which ignores ordering, is basically a weakening of this view.

This view is the primary one for the problem at hand, since we define our task in terms of it. We must be able to take this view of documents in a collection in order to perform extraction, so a collection of uuencoded documents, for instance, would not form the basis of an appropriate domain.

2.2.3 The mark-up view

D is a sequence of terms interleaved with meta-terms, which provide role information about the terms. HTML contains explicit meta-terms (tags), but even ASCII contains "control" characters, such as tabs and carriage returns, the purpose of which is to partition terms. While the terms view is a uni-dimensional interpretation of a document, mark-up is multi-dimensional. Each tag represents an orthogonal dimension along which tokens can be described.

2.2.4 The layout view

D is a two-dimensional arrangement and sizing of terms intended for human consumption. This view is an interpretation of the mark-up view by some application.

In general, many important textual objects can be discerned only at this level, objects such as paragraphs, headlines, tables, mail headers, signatures, etc. Such objects are frequently employed to separate a field from surrounding text, or to associate it with surrounding text in a special way. Textual matrices, for example, imply something about the text fragments they comprise; columns typically define an attribute over multiple objects, while rows associate the various attributes of a single object.

2.2.5 The syntactic view

D is a linguistically structured object. The terms of D participate in a set of relations that bind them together in graphical structures.

Clearly, this view is the most powerful source of information for extraction in many cases—if we can make effective use of it. Unfortunately, the very existence of IE as a discipline implies shortcomings in current NLP methods. On the one hand, syntactic parsing is slow and its best use by no means clear. On the other, much real-world text is in whole or in part syntactically ill-formed.

2.3 Learning Methods

If we believe that information useful for extraction is in general available under multiple views, then it is not hard to justify the use of multiple learning approaches. The biases of a learner may or may not be suitable for particular extraction tasks or for information available under particular document views. In this section, I distinguish four classes of learner. I do not consider this list exhaustive.

2.3.1 Dictionary Construction

The system obviously must accumulate legal field values and use those values to identify novel instances. A field extraction problem may require as little as this, depending on the domain, or the system may treat the output of such a learner as evidence to be combined with the output of other learners. If the system is learning how to extract ingredients from recipes, for example, it may be fairly confident that "watermelon" refers unambiguously to one of the ingredients in a recipe under consideration. On the other hand, an extractor for addresses should not handle the token "1996" similarly.

Known variously as a keyword "dictionary" or "lexicon," the notion of a simple component that holds domain-specific values is widespread in the MUC community. Entries in such a database, which is often manually constructed, frequently contain semantic information which is used in further analysis [24] [41], or they provide authoritative forms for entities which have varying textual realizations [25]. Less typically, entries are patterns which are used directly for extraction [25] [6] [18]. The dictionary constructed by a learner of the type we are considering here can be viewed as a very simple database of this kind.

2.3.2 Vector-space learning

Although the tokens which constitute an address are not in general unique to addresses, there are tokens which are typical of addresses and which are infrequent in general text, tokens such as "Ave." and "PA". Ideas from the field of information retrieval (IR) can be applied to determine how probably a text segment belongs to a particular class. A Bayesian approach, as investigated in IR, can return class estimates given a set of words. Alternatively, the text segments can be treated as small documents for the purpose of TFIDF weighting and Rocchio-style prototyping.

It is longstanding practice within IR to represent documents for retrieval as vectors of real numbers where each number corresponds to a document term [39]. TFIDF [33] [34], the standard weighting scheme, has also been used in document *classification* (or *categorization* [19]). The accepted approach to vector-space document classification is based on work done originally by Rocchio [31]. This work is relevant to the extent that an extraction problem can be cast as a classification problem. If we regard the text fragments we get by sliding a fixed-size window over a document themselves as documents, we can apply these methods directly.

Related to TFIDF-based methods are probabilistic techniques. A common approach is one involving Bayes Rule, which uses term occurrence information directly and assumes statistical independence among terms [21]. In this approach, hypotheses are document class assignments and the frequency of individual terms evidence for or against hypotheses.

2.3.3 Grammar Inference

Many fields obey strict syntactic rules. For example, seminar times, expressed numerically (e.g., "4:30 p.m."), follow a very predictable pattern. Where such a pattern can be expressed in terms of a finite state automaton, FSA learning methods are applicable. In some cases, several syntactically rigid patterns are possible for a particular class of item. For example, a date can be expressed as

3-4-97, 3/4/1997, or March 4, 1997. In such cases it is an interesting problem to learn a disjunctive set of FSA recognizers.

Given an abstract alphabet that expresses things like whitespace usage, line justification, and capitalization, we might regard layout-view structures, such as paragraphs, centered portions of text, and tables, as the result of a stochastic process obeying Markovian assumptions. Insofar as this is possible, such patterns might be learnable by a hidden Markov model.

The problem of *grammar inference* has a rich history in computer science and continues to attract much interest [40] [7]. Although the general problem of inferring languages from examples is known to be quite hard, work continues on special language classes and heuristic methods with good empirical performance. The idea of applying grammar inference to the analysis of document structure is not novel. Ahonen and Mannila, for example, consider the problem of inducing a regular grammar for document structure from examples, where examples are documents such as dictionaries, encyclopedias, and user manuals [3]. Rus and Summers [32] attempt to infer trees that represent document structure based on indentation profiles. In the same work, novel methods for detecting tables and figures from whitespace measurements are described.

Hidden Markov Models (HMM) [28] perform a kind of stochastic grammar inference. Model-merging methods, as commonly used in grammar inference, have been shown effective when used in connection with HMM's [38]. Although HMM's have been applied extensively to speech processing, they have seen little use with written text. (One exception is [23], where it is argued that HMM's can be used to model text effectively for information retrieval.) Nevertheless, HMM's are a useful technique for the construction of higher-level feature detectors. Text is sequential at multiple levels of granularity. Although language itself is characterized by super-regular structure [8], several of the MUC systems use finite-state grammar approximations based on [26], rather than general parsers. We might expect that other useful aspects of written text, such as capitalization and punctuation patterns, can be modeled effectively under Markovian assumptions.

2.3.4 Relational Rule Inference

Learners in this class, such as CN2 [10] [9] FOIL [27], and BRUTE [35], can be used both in performing sub-tasks and in the integration of predictions from other learners. At the sub-task level, they can be used effectively in conjunction with simple syntactic analysis because of their ability to form rules using multiple words and grammatical relations to express sophisticated grammatical patterns.

Their flexibility, power, and robustness in the face of noise make them attractive candidates as integrators of evidence. Particularly interesting is the apparent isomorphism between the kind of concept such a learner forms and the typical structure of the extraction problem. Rule learners induce a disjunctive set of rules. While a data collection may not follow a single format for a given information item, it typically has a limited set of formatting conventions from which to choose. In the seminar announcement domain, for example, many locations are either embedded in grammatical sentences or occur in a row of a miniature table with the string `Place:`. The two formats are mutually exclusive and are visible under different document views; a disjunctive learner is a natural fit to this class of field.

Several recent papers have demonstrated the effectiveness of rule-learning systems in text classification. Apté et al [5] describe SWAP-1, a rule induction system which performs well on several standard text classification problems. Cohen [11] explores the applicability of relational learning to text classification. Because such learners express learned concepts directly in terms of features, rather than summing evidence over multiple features as, for example, a Bayesian learner, they perform well in cases where concepts can be expressed succinctly. Many information extraction problems have this quality; stereotypic language use in a particular domain permits a succinct description of extraction patterns.

3 Preliminary Results

In work described at length elsewhere [13], I implemented and tested learners from three of the four classes enumerated above: a dictionary learner, a vector-space learner, and a relational learner. Learner performance was compared on four fields from a single document collection consisting of 485 electronic seminar announcements.

This experiment was designed to answer two questions:

- Is a learning approach to extraction in a real-world domain feasible?
- Is there evidence that learners' biases cover different aspects of the extraction problem, and that by combining learners we can hope to achieve performance better than that of any individual learner?

I concluded from this experiment that a learning approach to extraction is indeed feasible in this domain. What is more, while all learners acquire useful extraction knowledge, one of the learners—the vector-space learner—does extraordinarily well on a couple of the tested fields. Moreover, there was evidence that different learners were attending to different parts of the extraction problem. For the most difficult field, the two more sophisticated learners differed considerably in behavior while achieving comparable performance.

4 Research Plan

This thesis can be decomposed into three distinct thrusts:

1. An immediate exploration into various learning methods for extraction. Several machine learning methods for various parts of the extraction problem will be implemented, tested on several domains, and refined. I expect that the bulk of this work will be completed by the end of next semester, with the early effort going into refining learners already implemented.
2. A medium-term investigation into higher-level questions. Once several learners have been tested and their strengths and weaknesses assessed, several problems remain which must be solved before a general-purpose learning architecture for extraction can be constructed. I will address some or all of these questions. I expect to be in a position to address the first of these questions by early in the coming year. In the year that follows, I will investigate at least two of the questions detailed below, and possibly all three.
3. An investigation, probably spanning the duration of this thesis, into the uses of extraction techniques for Web mining. I am fortunate to be part of a recently convened working group which is actively investigating this problem. Concurrently with work in the above two categories, therefore, I hope to explore some of the unique challenges and opportunities Web mining presents with respect to informatione extraction.

A more detailed listing of research questions in each of these three categories follows.

4.1 Near Term: Implementation

This phase of the research, which will begin immediately, will concentrate on extending and improving the learners which were implemented in the preliminary work. In particular, the vector-space learner and the relational learner show great promise. Their implementation, however, left many questions unanswered. In addition, this phase will include the implementation and testing of grammar inference methods.

4.1.1 Vector-space learner

Of the learners tested in the preliminary experiment, this approach was the strongest and most consistent. It has a number of failings, however. I will look into principled ways of applying the term occurrence statistics it takes. In addition, the current learner has difficulty identifying field boundaries; I will look for ways to overcome this problem. Finally, I will be seeking to enhance its rather impoverished feature set, introducing layout and syntactic information where useful.

4.1.2 Relational learner

The application of relational learning to information extraction is novel, and the preliminary results attest to its potential. Just as the vector-space learner will be extended with syntactic features, I will be looking for useful non-syntactic features to introduce to the relational learner, whose feature set currently represents primarily grammatical information. In addition, the current implementation includes many untested heuristics; one aim of this phase will be to refine these. One heuristic concerns the way in which fields are represented by this learner: extraction knowledge is acquired in terms of three independent learned concepts, one for each of the field boundaries and one for the syntactically main term. I will be looking for ways either to improve coordination of these three learned concepts at extraction time, or to set the problem up so that only a single concept is learned.

4.1.3 Grammar inference

Learners from the field of grammar inference seem very appropriate to this problem, both for sub-tasks of the extraction problem, such as identifying sentences and features of document layout, as well as the extraction problem itself. In this phase, I will implement learners in this class and investigate their usefulness to the problem.

4.2 Medium Term: Fundamental Questions

Problems in this category are critical to the construction of a machine learning architecture for information extraction. In this thesis I will provide the beginnings of answers for some or all of these questions.

4.2.1 Exploiting layout

As noted in the introduction, I am interested in natural language as it occurs in the online environment. Thus, a part of the problem involves making effective use of common textual devices which are orthogonal to grammar. Of particular interest is *layout*, by which I understand visual devices such as tables, plan signatures, section headings, etc.

Although some research has gone into the problem of expressing document layout, little work has been done on the problem of *learning* operational aspects of layout useful for extraction. I believe that grammar inference methods will prove useful for this problem, but a relational approach may also be effective, and even ideas from machine vision could be applicable.

4.2.2 Combining learners

If it is true that the strongest automatic system for extraction from online documents must integrate information from multiple document views, then the question of how to achieve this integration is central. It is not simply a matter of finding the right heuristic. The strengths of individual learners vary with field and domain, so the problem of integration must be solved on a domain-by-domain, and even field-by-field, basis. It may be the case that general heuristics which take into account field profiles will prove adequate. For example, if we notice that a field typically occurs embedded in grammatical text, and only one of the learners is competent in such cases, we might discount the predictions of other learners. On the other hand, this may itself be a learning problem as challenging as the extraction problem itself.

4.2.3 Bootstrapping

In order to judge a machine learning approach to IE useful, we require (1) that its accuracy be comparable to that of a manual system, and (2) that adapting it to a new domain involve less labor than the configuration of a manual system. Other questions in this section concern point (1). Here, I am interested in how to achieve point (2).

It is probable that if a learning extraction system can achieve accuracy comparable with that of a manual system, adapting it to a new domain will already require less work than manual configuration. Labeling a training set, while tedious, is almost certainly less demanding than configuring a manual system, and it probably must be a part of manual configuration, anyway. Still, we can expect huge rewards from any technique that lessens the need for human intervention. In this phase I will explore the possibility of making labeling more interactive, interleaving learning with labeling, so that any early progress in system competence immediately benefits the human labeler. If the system can learn an approximately correct extractor given a few examples, then the focus of the human's labor can be shifted from labeling to reviewing the system's labels, a relatively less arduous job.

4.3 Web Mining

Not only is the World Wide Web a huge source of domains in which extraction could provide enormous benefit, its structure presents opportunities and challenges not otherwise available. A Web mining system needs an extraction component in order to recover desired information from textual Web pages, but we also can apply extraction techniques to problems not foreseen by traditional IE. One such problem is to identify hyperlinks which are likely to help an autonomous Web agent achieve some search goal. For example, we might attempt to learn an extractor for the problem, *given the home page of a faculty member, find any links to pages describing courses taught by this person.*

Note that this extractor assumes that we can identify documents belonging to the classes *home page of a faculty member* and *course page*. In general, the problem of Web mining is rife with opportunities for synergy between document classification and information extraction. If we possess a reliable classifier for pages of a particular class, then we can lessen or remove entirely the need for human labeling of hyperlinks leading to documents of that class. Similarly, reliable extractors for information typical of documents of some class can serve as a source of powerful features for a classifier. Note that both of these ideas assume *reliability*. There are many ideas to try, however, for the case in which full reliability is lacking. In the best case, the system might be able to bootstrap itself by leveraging more reliable components in the improvement of less reliable ones.

References

[1] *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, McLean, Virginia, June 1992. Morgan Kaufmann Publisher, Inc., San Francisco.

[2] *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, Baltimore, Maryland, August 1993. Morgan Kaufmann Publisher, Inc., San Francisco.

[3] Helena Ahonen and Heikki Mannila. Forming grammars for structured documents: an application of grammatical inference. In Rafael C. Carrasco and Jose Oncina, editors, *Grammatical Inference and Applications: Second International Colloquium, ICGI-94*, pages 153–167. Springer-Verlag, September 1994.

[4] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, Megumi Kameyama, and Mabry Tyson. SRI: description of the JV-FASTUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[5] Chidanand Apté, Fred Damerau, and Sholom M. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, July 1994.

[6] Stephanie E. August and Charles P. Dolan. Hughes Research Laboratories: description of the trainable text skimmer used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, June 1992.

[7] Rafael C. Carrasco and Jose Oncina, editors. *Grammatical Inference and Applications: Second International Colloquium, ICGI-94*. Springer-Verlag, September 1994.

[8] Noam Chomsky. *Syntactic Structures.* Mouton & Co., 1957.

[9] P. Clark and R. Boswell. Rule induction with CN2: some recent improvements. In Y. Kodratoff, editor, *Machine Learning – EWSL-91*, pages 151–163. Springer-Verlag, Berlin, 1991.

[10] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–263, March 1989.

[11] William W. Cohen. Learning to classify English text with ILP methods. *Advances in Inductive Logic Programming*, 1995.

[12] Jim Cowie, Louise Guthrie, Wang Jin, Rong Wang, Takahiro Wakao, James Pustejovsky, and Scott Waterman. CRL/Brandeis: description of the Diderot system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[13] Dayne Freitag. Machine learning for information extraction from online documents: a preliminary experiment, 1996.

[14] Jerry R. Hobbs. The generic information extraction system. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[15] Jerry R. Hobbs, Douglas Appelt, Mabry Tyson, John Bear, and David Israel. SRI International: description of the FASTUS system used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, June 1992.

[16] Paul S. Jacobs, George Krupka, Lisa Rau, Michael L. Mauldin, Teruko Mitamura, Tsuyoshi Kitani, Ira Sider, and Lois Childs. GE-CMU: description of the shogun system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[17] George Krupka, Paul Jacobs, and Lisa Rau. GE NLToolset: description of the system as used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, June 1992.

[18] W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, and F. Feng. UMass/Hughes: description of the CIRCUS system used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[19] David D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *15th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, June 1992.

[20] David D. Lewis. Text filtering in MUC-3 and MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, June 1992.

[21] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, July 1994.

[22] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[23] Elke Mittendorf and Peter Schäuble. Document and passage retrieval based on hidden Markov models. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval*, July 1994.

[24] Kazunori Muraki, Shinichi Doi, and Shinich Ando. NEC: description of the VENIEX system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[25] William W. Noah and Rollin V. Weeks. TRW: description of the DEFT system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[26] Fernando Pereira. Finite-state approximations of grammars. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 20–25, Hidden Valley, Pennsylvania, 1990.

[27] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 1990.

[28] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January 1986.

[29] Ellen Riloff and Wendy Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 1994.

[30] Ellen Riloff and Jay Shoen. Automatically acquiring conceptual patterns without an annotated corpus. *Proceedings of the Third Workshop on Very Large Corpora*, 1995.

[31] J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[32] Daniela Rus and Kristen Summers. Using white space for automated document structuring. Technical Report TR94-1452, Cornell University, 1994.

[33] Gerard Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1971.

[34] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Department of Computer Science, Cornell University, 1987.

[35] R. Segal and O. Etzioni. Learning decision lists using homogeneous rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, July 1994.

[36] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[37] Stephen Soderland and Wendy Lehnert. Wrap-Up: a trainable discourse module for information extraction. *Journal of Artificial Intelligence Research*, 1994.

[38] Andreas Stolcke and Stephen M. Omohundro. Best-first model merging for hidden Markov induction. Technical Report TR-94-003, International Computer Science Institute, Berkeley, California, January 1994.

[39] C. J. van Rijsbergen. *Information Retrieval.* Butterworths, Inc., Boston, 1979.

[40] Enrique Vidal. Grammatical inference: an introductory survey. In Rafael C. Carrasco and Jose Oncina, editors, *Grammatical Inference and Applications: Second International Colloquium, ICGI-94*, pages 1–4. Springer-Verlag, September 1994.

[41] Ralph Weischedel, Damaris Ayuso, Sean Boisen, Heidi Fox, Robert Ingria, Tomoyoshi Matsukawa, Constantine Papageorgiou, Dawn MacLaughlin, Masaichiro Kitagawa, Tsutomu Sakai, June Abe, Hiroto Hosihi, Yoichi Miyamoto, and Scott Miller. BBS: description of the PLUM system as used for MUC-5. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.

[42] Craig A. Will. Comparing human and machine performance for natural language information extraction: results for English microelectronics from the MUC-5 evaluation. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, August 1993.