Toward General-Purpose Learning for Information Extraction

Dayne Freitag
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dayne@cs.cmu.edu

Abstract

Two trends are evident in the recent evolution of the field of information extraction: a preference for simple, often corpus-driven techniques over linguistically sophisticated ones; and a broadening of the central problem definition to include many non-traditional text domains. This development calls for information extraction systems which are as *retargetable* and *general* as possible. Here, we describe SRV, a learning architecture for information extraction which is designed for maximum generality and flexibility. SRV can exploit domain-specific information, including linguistic syntax and lexical information, in the form of features provided to the system explicitly as input for training. This process is illustrated using a domain created from Reuters corporate acquisitions articles. Features are derived from two general-purpose NLP systems, Sleator and Temperly's link grammar parser and Wordnet. Experiments compare the learner's performance with and without such linguistic information. Surprisingly, in many cases, the system performs as well without this information as with it.

1 Introduction

The field of *information extraction* (IE) is concerned with using natural language processing (NLP) to extract essential details from text documents automatically. While the problems of retrieval, routing, and filtering have received considerable attention through the years, IE is only now coming into its own as an information management sub-discipline.

Progress in the field of IE has been away from general NLP systems, that must be tuned to work in a particular domain, toward faster systems that perform less linguistic processing of documents and can be more readily targeted at novel domains (e.g., (Appelt et al., 1993)). A natural part of this development has been the introduction of machine learning techniques to facilitate the domain engineering effort (Riloff, 1996; Soderland and Lehnert, 1994).

Several researchers have reported IE systems which use machine learning at their core (Soderland, 1996; Califf and Mooney, 1997). Rather than spend human effort tuning a system for an IE domain, it becomes possible to conceive of *training* it on a document sample. Aside from the obvious savings in human development effort, this has significant implications for information extraction as a discipline:

Retargetability Moving to a novel domain should no longer be a question of code modification; at most some feature engineering should be required.

Generality It should be possible to handle a much wider range of domains than previously. In addition to domains characterized by grammatical prose, we should be able to perform information extraction in domains involving less traditional structure, such as netnews articles and Web pages.

In this paper we describe a learning algorithm similar in spirit to FOIL (Quinlan, 1990), which takes as input a set of tagged documents, and a set of features that control generalization, and produces rules that describe how to extract information from novel documents. For this system, introducing linguistic or any other information particular to a domain is an exercise in feature definition, separate from the central algorithm, which is constant. We describe a set of experiments, involving a document collection of newswire articles, in which this learner is compared with simpler learning algorithms.

2 SRV

In order to be suitable for the widest possible variety of textual domains, including collections made up of informal E-mail messages, World Wide Web pages, or netnews posts, a learner must avoid any assumptions about the structure of documents that might be invalidated by new domains. It is not safe to assume, for example, that text will be grammatical, or that all tokens encountered will have entries in a lexicon available to the system. Fundamentally, a document is simply a sequence of terms. Beyond this, it becomes difficult to make assumptions that are not violated by some common and important domain of interest.

At the same time, however, when structural assumptions are justified, they may be critical to the success of the system. It should be possible, therefore, to make structural information available to the learner as input for training. The machine learning method with which we experiment here, SRV, was designed with these considerations in mind. In experiments reported elsewhere, we have applied SRV to collections of electronic seminar announcements and World Wide Web pages (Freitag, 1998). Readers interested in a more thorough description of SRV are referred to (Freitag, 1998). Here, we list its most salient characteristics:

- **Lack of structural assumptions.** SRV assumes nothing about the structure of a field instance[1] or the text in which it is embedded—only that an instance is an unbroken fragment of text. During learning and prediction, SRV inspects *every* fragment of appropriate size.
- **Token-oriented features.** Learning is guided by a feature set which is separate from the core algorithm. Features describe aspects of individual tokens, such as capitalized, numeric, noun. Rules can posit feature values for individual tokens, or for all tokens in a fragment, and can constrain the ordering and positioning of tokens.
- **Relational features.** SRV also includes a notion of *relational* features, such as next-token, which map a given token to another token in its environment. SRV uses such features to explore the context of fragments under investigation.
- **Top-down greedy rule search.** SRV constructs rules from general to specific, as in FOIL (Quinlan, 1990). Top-down search is more sensitive to patterns in the data, and less dependent on heuristics, than the bottom-up search used by similar systems (Soderland, 1996; Califf and Mooney, 1997).
- **Rule validation.** Training is followed by validation, in which individual rules are tested on a reserved portion of the training documents. Statistics collected in this way are used to associate a confidence with each prediction, which are used to manipulate the accuracy-coverage trade-off.

[1] We use the terms *field* and *field instance* for the rather generic IE concepts of *slot* and *slot filler*. For a newswire article about a corporate acquisition, for example, a field instance might be the text fragment listing the amount paid as part of the deal.

3 Case Study

SRV's default feature set, designed for informal domains where parsing is difficult, includes no features more sophisticated than those immediately computable from a cursory inspection of tokens. The experiments described here were an exercise in the design of features to capture syntactic and lexical information.

3.1 Domain

As part of these experiments we defined an information extraction problem using a publicly available corpus. 600 articles were sampled from the "acquisition" set in the Reuters corpus (Lewis, 1992) and tagged to identify instances of nine fields. Fields include those for the official names of the parties to an acquisition (acquired, purchaser, seller), as well as their short names (acqabr, purchabr, sellerabr), the location of the purchased company or resource (acqloc), the price paid (dlramt), and any short phrases summarizing the progress of negotiations (status). The fields vary widely in length and frequency of occurrence, both of which have a significant impact on the difficulty they present for learners.

3.2 Feature Set Design

We augmented SRV's default feature set with features derived using two publicly available

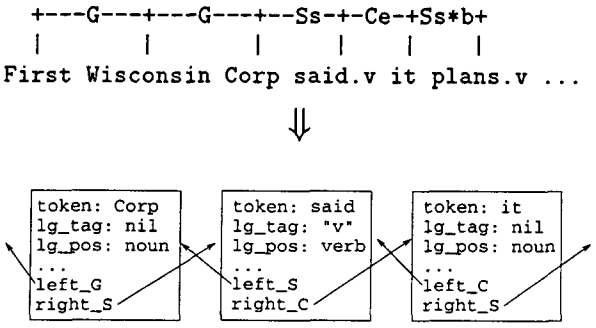

Figure 1: An example of link grammar feature derivation.

NLP tools, the link grammar parser and Wordnet.

The link grammar parser takes a sentence as input and returns a complete parse in which terms are connected in typed binary relations ("links") which represent syntactic relationships (Sleator and Temperley, 1993). We mapped these links to relational features: A token on the right side of a link of type X has a corresponding relational feature called left_X that maps to the token on the left side of the link. In addition, several non-relational features, such as part of speech, are derived from parser output. Figure 1 shows part of a link grammar parse and its translation into features.

Our object in using Wordnet (Miller, 1995) is to enable SRV to recognize that the phrases, "A bought B," and, "X acquired Y," are instantiations of the same underlying pattern. Although "bought" and "acquired" do not belong to the same "synset" in Wordnet, they are nevertheless closely related in Wordnet by means of the "hypernym" (or "is-a") relation. To exploit such semantic relationships we created a single token feature, called wn_word. In contrast with features already outlined, which are mostly boolean, this feature is set-valued. For nouns and verbs, its value is a set of identifiers representing all synsets in the hypernym path to the root of the hypernym tree in which a word occurs. For adjectives and adverbs, these synset identifiers were drawn from the cluster of closely related synsets. In the case of multiple Wordnet senses, we used the most common sense of a word, according to Wordnet, to construct this set.

3.3 Competing Learners

We compare the performance of SRV with that of two simple learning approaches, which make predictions based on raw term statistics. Rote (see (Freitag, 1998)), memorizes field instances seen during training and only makes predictions when the same fragments are encountered in novel documents. Bayes is a statistical approach based on the "Naive Bayes" algorithm (Mitchell, 1997). Our implementation is described in (Freitag, 1997). Note that although these learners are "simple," they are not necessarily ineffective. We have experimented with them in several domains and have been surprised by their level of performance in some cases.

4 Results

The results presented here represent average performances over several separate experiments. In each experiment, the 600 documents in the collection were randomly partitioned into two sets of 300 documents each. One of the two subsets was then used to train each of the learners, the other to measure the performance of the learned extractors.

We compared four learners: each of the two simple learners, Bayes and Rote, and SRV with two different feature sets, its default feature set, which contains no "sophisticated" features, and the default set augmented with the features derived from the link grammar parser and Wordnet. We will refer to the latter as SRV+ling.

Results are reported in terms of two metrics closely related to *precision* and *recall*, as seen in information retrieval: *Accuracy*, the percentage of documents for which a learner predicted correctly (extracted the field in question) over all documents for which the learner predicted; and *coverage*, the percentage of documents having the field in question for which a learner made *some* prediction.

4.1 Performance

Table 1 shows the results of a ten-fold experiment comparing all four learners on all nine fields. Note that accuracy and coverage must be considered together when comparing learners. For example, Rote often achieves reasonable accuracy at very low coverage.

Table 2 shows the results of a three-fold experiment, comparing all learners at fixed cover-

Alg	*Acc*	*Cov*	*Acc*	*Cov*	*Acc*	*Cov*
	acquired		purchaser		seller	
Rote	59.6	18.5	43.2	23.2	38.5	15.2
Bayes	19.8	100	36.9	100	15.6	100
SRV	38.4	96.6	42.9	97.9	16.3	86.4
SRVlng	38.0	95.6	42.4	96.3	16.4	82.7
	acqabr		purchabr		sellerabr	
Rote	16.1	42.5	3.6	41.9	2.7	27.3
Bayes	23.2	100	39.6	100	16.0	100
SRV	31.8	99.8	41.4	99.6	14.3	95.1
SRVlng	35.5	99.2	43.2	99.3	14.7	91.8
	acqloc		status		dlramt	
Rote	6.4	63.1	42.0	94.5	63.2	48.5
Bayes	7.0	100	33.3	100	24.1	100
SRV	12.7	83.7	39.1	89.8	50.5	91.0
SRVlng	15.4	80.2	41.5	87.9	52.1	89.4

Table 1: Accuracy and coverage for all four learners on the acquisitions fields.

age levels, 20% and 80%, on four fields which we considered representative of the wide range of behavior we observed. In addition, in order to assess the contribution of each kind of linguistic information (syntactic and lexical) to SRV's performance, we ran experiments in which its basic feature set was augmented with only one type or the other.

4.2 Discussion

Perhaps surprisingly, but consistent with results we have obtained in other domains, there is no one algorithm which outperforms the others on all fields. Rather than the absolute difficulty of a field, we speak of the suitability of a learner's *inductive bias* for a field (Mitchell, 1997). Bayes is clearly better than SRV on the seller and sellerabr fields at all points on the accuracy-coverage curve. We suspect this may be due, in part, to the relative infrequency of these fields in the data.

The one field for which the linguistic features offer benefit at all points along the accuracy-coverage curve is acqabr.[2] We surmise that two factors contribute to this success: a high frequency of occurrence for this field (2.42 times per document on average), and consistent occurrence in a linguistically rich context.

[2]The acqabr differences in Table 2 (a 3-split experiment) are *not* significant at the 95% confidence level. However, the full 10-split averages, with 95% error margins, are: at 20% coverage, 61.5±4.4 for SRV and 68.5±4.2 for SRV+ling; at 80% coverage, 37.1±2.0 for SRV and 42.4±2.1 for SRV+ling.

Field	80%	20%	80%	20%	80%	20%
	Rote		Bayes		SRV	
purchaser	—	50.3	40.6	55.9	45.3	55.7
acqabr	—	24.4	29.3	50.6	40.0	63.4
dlramt	—	69.5	45.9	71.4	57.1	66.7
status	46.7	65.3	39.4	62.1	43.8	72.5
	SRV+ling		srv+lg		srv+wn	
purchaser	48.5	56.3	46.3	63.5	46.7	58.1
acqabr	44.3	75.4	40.4	71.4	41.9	72.5
dlramt	57.1	61.9	55.4	67.3	52.6	67.4
status	43.3	72.6	38.8	74.8	42.2	74.1

Table 2: Accuracy from a three-split experiment at fixed coverage levels.

A fragment is a acqabr, if:
- it contains exactly one token;
- the token (T) is capitalized;
- T is followed by a lower-case token;
- T is preceded by a lower-case token;
- T has a right AN-link to a token (U) with wn_word value "possession";
- U is preceded by a token with wn_word value "stock";
- and the token two tokens before T is not a two-character token.

to purchase 4.5 mln Trilogy common shares at
acquire another 2.4 mln Roach treasury shares

Figure 2: A learned rule for acqabr using linguistic features, along with two fragments of matching text. The AN-link connects a noun modifier to the noun it modifies (to "shares" in both examples).

Figure 2 shows a SRV+ling rule that is able to exploit both types of linguistic information. The Wordnet synsets for "possession" and "stock" come from the same branch in a hypernym tree—"possession" is a generalization of "stock"[3]—and both match the collocations "common shares" and "treasury shares." That the paths [right_AN] and [right_AN prev_tok] both connect to the same synset indicates the presence of a two-word Wordnet collocation.

It is natural to ask why SRV+ling does not

[3]SRV, with its general-to-specific search bias, often employs Wordnet this way—first more general synsets, followed by specializations of the same concept.

outperform SRV more consistently. After all, the features available to SRV+ling are a superset of those available to SRV. As we see it, there are two basic explanations:

- **Noise.** Heuristic choices made in handling syntactically intractable sentences and in disambiguating Wordnet word senses introduced noise into the linguistic features. The combination of noisy features and a very flexible learner may have led to overfitting that offset any advantages the linguistic features provided.
- **Cheap features equally effective.** The simple features may have provided most of the necessary information. For example, generalizing "acquired" and "bought" is only useful in the absence of enough data to form rules for each verb separately.

4.3 Conclusion

More than similar systems, SRV satisfies the criteria of *generality* and *retargetability*. The separation of domain-specific information from the central algorithm, in the form of an extensible feature set, allows quick porting to novel domains.

Here, we have sketched this porting process. Surprisingly, although there is preliminary evidence that general-purpose linguistic information can provide benefit in some cases, most of the extraction performance can be achieved with only the simplest of information.

Obviously, the learners described here are not intended to solve the information extraction problem outright, but to serve as a source of information for a post-processing component that will reconcile all of the predictions for a document, hopefully filling whole templates more accurately than is possible with any single learner. How this might be accomplished is one theme of our future work in this area.

Acknowledgments

Part of this research was conducted as part of a summer internship at Just Research. And it was supported in part by the Darpa HPKB program under contract F30602-97-1-0215.

References

Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson. 1993. FASTUS: a finite-state processor for information extraction from real-world text. *Proceedings of IJCAI-93*, pages 1172–1178.

M. E. Califf and R. J. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *Working Papers of ACL-97 Workshop on Natural Language Learning.*

D. Freitag. 1997. Using grammatical inference to improve precision in information extraction. In *Notes of the ICML-97 Workshop on Automata Induction, Grammatical Inference, and Language Acquisition.* http://www.cs.cmu.edu/~pdupont/ml97p/ml97_GI_wkshp.tar.

Dayne Freitag. 1998. Information extraction from HTML: Application of a general machine learning approach. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).*

D. Lewis. 1992. *Representation and Learning in Information Retrieval.* Ph.D. thesis, Univ. of Massachusetts. CS Tech. Report 91–93.

G.A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, pages 39–41, November.

Tom M. Mitchell. 1997. *Machine Learning.* The McGraw-Hill Companies, Inc.

J. R. Quinlan. 1990. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266.

E. Riloff. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 1044–1049.

Daniel Sleator and Davy Temperley. 1993. Parsing English with a link grammar. *Third International Workshop on Parsing Technologies.*

Stephen Soderland and Wendy Lehnert. 1994. Wrap-Up: a trainable discourse module for information extraction. *Journal of Artificial Intelligence Research*, 2:131–158.

S. Soderland. 1996. *Learning Text Analysis Rules for Domain-specific Natural Language Processing.* Ph.D. thesis, University of Massachusetts. CS Tech. Report 96-087.