Research in Information Extraction: 1996-98

Ralph Grishman
Department of Computer Science
New York University
New York, NY 10003
grishman@cs.nyu.edu

Definitions and Goals

Information extraction involves picking out specified types of information from natural language text. Recent Message Understanding Conferences [1,2,3] have developed a spectrum of such tasks, and we have worked on two of them, at opposite ends of the spectrum: the *named entity* task, which involves identifying and classifying names, and the *scenario template* task, which involves extracting critical information (participants, location, date, etc.) about specified classes of events.

We have of course been concerned about performance: trying to build systems which come close to human accuracy, or at least perform with sufficient accuracy to be of practical value. In addition, we have long been concerned with portability: the ability to adapt our systems to new classes of events, to new domains, and even to new languages. We want to create systems which can be ported easily and, if possible, by people who don't know the internal workings of the system. Only in this way can systems for new tasks be created cheaply enough to be widely used.

Earlier systems were based entirely on hand-crafted rules. As larger annotated training corpora have become available and methods for learning from corpora have become better understood, more researchers have focused on corpus-trained systems, avoiding separate hand-coded knowledge or rules. The approach we have taken has been more eclectic and opportunistic: to use corpus-driven methods, but also to employ separate "world knowledge", hand-coded rules, and user interaction in rule acquisition where appropriate. In some cases this has allowed us to achieve greater performance or to learn rules from far fewer examples.

Named Entity

The named entity task involves identifying and classifying several types of names -- people, organizations, and locations -- as well as some other phrases, such as dates and times. Achieving fairly good performance on this task is easy, but approaching human performance is difficult because the hard cases must be resolved based on many different types of evidence: the words starting or ending a name ("Mr.", "Corp."), the context of a name ("... died"); other mentions of a name in a text ("Mr. Smith ... Smith reported..."). Corpus-based learning may be helpful in gathering and balancing these types of evidence. Fortunately, names are very frequent in many types of text, so it is easy to get a substantial training set for this task.

We explored two learning methods, decision trees and maximum entropy. In both cases, we sought to combine criteria which could be gathered from the training corpus with generalizations which could be obtained from external sources and rules developed by hand.

Decision Tree

Our decision tree method is described in detail in [4,5]. The internal nodes of the tree test various properties of a token; based on these properties, the leaves of the tree specify the probability that a given token starts, continues, or ends a name of a given type (person, organization, ...). The tree is built automatically from a training corpus annotated with the various types of names. In tagging new text, we first use the decision tree to determine these probabilities; we then use a Viterbi algorithm to find the most likely consistent tagging (e.g., one in which a 'start person' is followed by an 'end person' and not an 'end organization').

This approach was applied to the Japanese named entity task. The decisions in the decision tree are based on the character type of the token, the part-of-speech (as determined by Juman), and various word lists. These word lists, which include common titles, common company suffixes, major company names, etc., were gathered from the training corpus and the WWW.

Maximum Entropy

Our maximum entropy method is described in detail in [6,7]. Again, we are developing a function which takes as input various features of the tokens in a text, and yields the probability that a given token starts, continues, or ends a name of a given type. However, the form of the function is different: instead of being a sequence of discrete decisions (a decision tree), the probability is computed as a product of functions on the individual features, with coefficients determined from the corpus.

This approach was applied to both English and Japanese named entity tasks. For Japanese named entity, when used with the same set of features as the decision tree model, the performance was about the same. However, when the feature set was extended to include individual lexical items from the training corpus, the performance was substantially improved (from F=80.0 to 83.8[1]). The decision tree model was not able to use individual words as features as effectively, because these features fragmented the training data.

For English named entity, the performance of the system with features based on word form (e.g., capitalization), individual lexical items, and hand-collected word lists was already quite good (F=92.9 on test data from the training domain). However, there had been substantial work at NYU and elsewhere on building by hand patterns for named entity classification, and we wanted to take advantage of that work. In particular, while there might be gaps in these hand-written patterns, they did capture some situations where complex combinations of features could be used to classify names with high precision ... complex combinations that were not likely to be learned automatically. We utilized this work by treating the output of the hand-coded named entity rules as another set of features to be considered by the maximum entropy method. Adding our hand-coded rules (which, by themselves, performed at F=92.2) yielded a system with F=95.7; adding the rules from two other sites brought the performance to F=97.4. We thus demonstrated how the combination of hand-coded and corpus-acquired rules could be more effective than either alone.

Scenario Template

The general goals for the scenario template (event extraction) task were the same as those for the named entity task: improve performance and portability. However, the approach was somewhat different because we expected that the environment would be different. The named entity task is applicable across a range of domains, and so we can justify preparing a substantial number of training examples; furthermore, such data is relatively easy to prepare because the task is simple and the phenomenon frequent.

In contrast, scenario template is really a large collection of very diverse tasks (one task for each type of event), and each instance is more complex than the named entity task. We expect that "real life" training sets will be quite small -- even smaller than the 100-article sets which characterized the last two MUCs. Accordingly, while the process of building a system for a new class of events is example driven, there is much more emphasis on having a person in the loop to generalize and adjust the patterns and rules as they are being created.

Proteus Extraction Tool

Over the last two years we have built an increasingly rich interface for the customization of event extraction systems. Such a system in driven by a number of "knowledge bases", including a lexicon, a concept hierarchy, a set of task-specific templates (frames), and a set of patterns to be matched against the text. Our interface is able to inspect and modify all these knowledge bases, as well as manipulate documents and observe the results and intermediate stages of extraction on these documents. At the heart of this interface is a capability for taking a sample sentence along with its mapping into templates and produce an extraction pattern which is suitably generalized syntactically and semantically to operate on new text. The syntactic generalization is done fully automatically, while the semantic generalization is done in interaction with the user. This system is described more fully in [8,9,10].

Multilinguality

Another dimension of portability is portability to new *languages*. We noted earlier our work on named entity systems which could operate in both English and Japanese. Porting event extraction systems is more complex, because there are more components to the system. In particular, the English system uses almost entirely locally-written software coded in Lisp, and operates as a single process. In moving to other languages, we found that we wanted to make use of pre-existing software for such tasks as tokenization, part-of-speech tagging, and name recognition. We therefore moved towards a multi-

[1] The F measure is a combination of the recall and precision measures.

process structure in which information is communicated to the extraction engine in the form of SGML annotations on the document. Pre-existing components are embedded in "wrappers" so that they may communicate with the main extraction engine using SGML mark-up.

We have ported the entire extraction system (including the customization interface) to Japanese. The external components for this system are the Juman tokenizer/tagger and the Japanese named entity tagger described above. We have implemented the management succession scenario in Japanese using this system; the system is further described in [11] in this volume.

We have also ported the core extraction system to Swedish. For the Swedish system, the texts were first processed by SweCG, the Swedish Constraint Grammar developed at Helsinki University and commercialized by Lingsoft. The SweCG does lexical lookup, two-level morphological analysis and disambiguation. The analysis consists of part-of-speech tags, morphological features, and some semantic information. In the next module, the SweCG output was transformed into the SGML format required by the core extraction system. Already at this stage, some information (semantic tags, knowledge bases, capitalization and other heuristics) was used for name recognition. The remainder of the text analysis was performed by syntactic and semantic patterns within the core system. Since Swedish has a richer morphology than English, the pattern formalism was slightly extended to allow for more of the morphological information from the Swedish tagger to be used in the patterns.

Performance Enhancements

The final aspect of our work on information extraction has been an effort to improve the level of performance on the scenario template task. Specifically, we have studied the management succession task of MUC-6 [1], which requires the system to determine who has started or left which management position at which company. We tried to improve our performance on this task above the MUC-6 level (57% recall, 70% precision, F=62.82 on training corpus; 47% recall, 70% precision, F=56.39 on test corpus).

We made a large number of system changes in 1997, some general, others specific to the management succession scenario, including

- improvements in name recognition
- improvements in reference resolution, including handling of conjoined antecedents, coreference from copula clauses, and headless anaphors
- analysis of verb tense (which was then used to fill the "on the job" slot in this scenario)
- some additional noun phrase and event patterns ("the late ...", "... was laid off")

These changes raised performance on the training corpus to 68% recall, 75% precision, F=71.34, and 55% recall, 74% precision, F=63.11 on the test corpus.

We continued making changes in 1998, although primarily of a scenario-specific nature. In particular, we added several rules for suppressing spurious events, and rules for jobs which can be held concurrently (e.g., CEO and president of the same firm). Altogether these changes yielded an improvement of 2.3 F on the training corpus (69% recall, 79% precision); on the test corpus, however, the F was almost unchanged, with 0.4% gain in precision but 0.6% loss of recall.

At the 24-month Tipster meeting, colleagues from SRI International reported similar problems with improving MUC-6 performance. They noted that the official scoring procedure uses very liberal criteria for matching a system response to the key, seeking to maximize the score, and will give some credit even for wildly incorrect responses (e.g., for a hiring event where both the name of the person and the name of the company are wrong, but it is correctly reported that someone was hired and is now on the job). "Precision improving" modifications may reduce these near-random matches, thus reducing recall while raising precision. SRI reported that a metric which required a closer correspondence between key and response in order to get any credit (specifically, requiring that the person, position, and organization name for an event *all* be correct) indicated substantial gains as they improved their system, even though the official F measure was largely unchanged.

In a similar vein, we have examined a measure based on only three slots from the template: the person's name, the organization name, and the management post. We reasoned that "junk" (erroneously generated) templates are less likely to have correct values for these slots than for slots with binary values, such as the IN_AND_OUT slot (which indicates whether someone was starting or leaving a job). We used the official scorer to align the system response to the key. We then computed recall, precision, and F measure for the sum of these three slots, and found steady improvement, even when the F score over all the slots showed a slight dip. For the test corpus, our MUC-6 official run had recall 55.2%, precision 75.4%, F=63.7 for these three slots; at the end of 1997, we had recall 60.0%, precision 75.1%, F=66.7; after the changes described above for 1998, we had recall 61.1%, precision 75.3%, F=67.4.

Acknowledgements

The porting of the event extraction system to Swedish was done by Kristofer Franzén with funding from the Swedish Foundation for International Cooperation in Research and Higher Education, the Swedish Institute, and the Swedish Council for Research in the Humanities and Social Sciences.

References

[1] *Proceedings of the Sixth Message Understanding Conference (MUC-6).* Columbia, Maryland, November, 1995, Morgan Kaufmann.

[2] Ralph Grishman, Beth Sundheim. Message Understanding Conference - 6: A Brief History. *Proceedings of the 16th International Conference on Computational Linguistics*, Copenhagen, Denmark, 1996.

[3] *Proceedings of the Seventh Message Understanding Conference (MUC-7),* Fairfax, Virginia, April 29 - May 1, 1998. Available through the S.A.I.C. MUC web site, `http://www.muc.saic.com/`.

[4] Satoshi Sekine. NYU: Description of the Japanese NE System used for MET-2. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, Virginia, April 29 - May 1, 1998.

[5] Satoshi Sekine, Ralph Grishman and Hiroyuki Shinnou. A Decision Tree Method for Finding and Classifying Names in Japanese Texts. *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Canada, August 1998.

[6] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Description of the MENE Named Entity System as used in MUC-7. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, Virginia, April 29 - May 1, 1998.

[7] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Canada, August 1998.

[8] Roman Yangarber and Ralph Grishman. Customization of Information Extraction Systems. *Proceedings of International Workshop on Lexically Driven Information Extraction*, Frascati, Italy, July 16, 1997.

[9] Roman Yangarber and Ralph Grishman. NYU: Description of the Proteus/PET System as used for MUC-7 ST. *Proceedings of the Seventh Message Understanding Conference (MUC-7),* Fairfax, Virginia, April 29 - May 1, 1998.

[10] Roman Yangarber and Ralph Grishman, Transforming Examples into Patterns for Information Extraction, this volume.

[11] Chikashi Nobata, Satoshi Sekine, and Roman Yangarber, Japanese IE System and Customization Tool, this volume.