NYU: Description of the MENE Named Entity System as Used in MUC-7

Andrew Borthwick, John Sterling, Eugene Agichtein and Ralph Grishman
Computer Science Department
New York University
715 Broadway, 7th floor
New York, NY 10003, USA
{borthwic,sterling,agichtn,grishman}@cs.nyu.edu

INTRODUCTION

This paper describes a new system called "Maximum Entropy Named Entity" or "MENE" (pronounced "meanie") which was NYU's entrant in the MUC-7 named entity evaluation. By working within the framework of maximum entropy theory and utilizing a flexible object-based architecture, the system is able to make use of an extraordinarily diverse range of knowledge sources in making its tagging decisions. These knowledge sources include capitalization features, lexical features and features indicating the current type of text (i.e. headline or main body). It makes use of a broad array of dictionaries of useful single or multi-word terms such as first names, company names, and corporate suffixes. These dictionaries required no manual editing and were either downloaded from the web or were simply "obvious" lists entered by hand.

This system, built from off-the-shelf knowledge sources, contained no hand-generated patterns and achieved a result on dry run data which is comparable with that of the best statistical systems. Further experiments showed that when combined with handcoded systems from NYU, the University of Manitoba, and IsoQuest, Inc., MENE was able to generate scores which exceeded the highest scores thus-far reported by any system on a MUC evaluation.

Given appropriate training data, we believe that this system is highly portable to other domains and languages and have already achieved state-of-the-art results on upper-case English. We also feel that there are plenty of avenues to explore in enhancing the system's performance on English-language newspaper text.

Although the system was ranked fourth out of the 14 entries in the N.E. evaluation, we were diappointed with our performance on the formal evaluation in which we got an F-measure of 88.80. We believe that the deterioration in performance was mostly due to the shift in domains caused by training the system on airline disaster articles and testing it on rocket and missile launch articles.

MAXIMUM ENTROPY

Given a tokenization of a test corpus and a set of n (for MUC-7, $n = 7$) tags which define the name categories of the task at hand, the problem of named entity recognition can be reduced to the problem of assigning one of $4n + 1$ tags to each token. For any particular tag x from the set of n tags, we could be in one of 4 states: x_start, x_continue, x_end, and x_unique. In addition, a token could be tagged as "other" to indicate that it is not part of a named entity. For instance, we would tag the phrase [Jerry Lee Lewis flew to Paris] as [person_start, person_continue, person_end, other, other, location_unique]. This approach is essentially the same as [7].

The 29 tags of MUC-7 form the space of "futures" for a maximum entropy formulation of our N.E. problem. A maximum entropy solution to this, or any other similar problem allows the computation of

$p(f|h)$ for any f from the space of possible futures, F, for every h from the space of possible histories, H. A "history" in maximum entropy is all of the conditioning data which enables you to make a decision among the space of futures. In the named entity problem, this could be broadly viewed as all information derivable from the test corpus relative to the current token (i.e. the token whose tag you are trying to determine).

The computation of $p(f|h)$ in M.E. is dependent on a set of binary-valued "features" which, hopefully, are helpful in making a prediction about the future. For instance, one of our features is

$$g(h, f) = \left\{ \begin{array}{lll} 1 & : & \text{if current\_token\_capitalized(h)} = \\ & & \text{true and } f = \text{location\_start} \\ 0 & : & \text{else} \end{array} \right\} \tag{1}$$

Given a set of features and some training data, the maximum entropy estimation process produces a model in which every feature g_i has associated with it a parameter α_i. This allows us to compute the conditional probability by combining the parameters multiplicatively as follows:

$$P(f|h) = \frac{\prod_i \alpha_i^{g_i(h,f)}}{Z_\alpha(h)} \tag{2}$$

$$Z_\alpha(h) = \sum_f \prod_i \alpha_i^{g_i(h,f)} \tag{3}$$

The maximum entropy estimation technique guarantees that for every feature g_i, the expected value of g_i according to the M.E. model will equal the empirical expectation of g_i in the training corpus.

More complete discussions of M.E., including a description of the M.E. estimation procedure and references to some of the many new computational linguistics systems which are successfully using M.E. can be found in the following useful introduction: [5]. As many authors have remarked, though, the key thing about M.E. is that it allows the modeler to concentrate on finding the features that characterize the problem while letting the M.E. estimation routine worry about assigning relative weights to the features.

SYSTEM ARCHITECTURE

MENE consists of a set of C++ and Perl modules which forms a wrapper around an M.E. toolkit [6] which computes the values of the alpha parameters of equation 2 from a pair of training files created by MENE. MENE's flexibility is due to the fact that it can incorporate just about any binary-valued feature which is a function of the history and future of the current token. In the following sections, we will discuss each of MENE's feature classes in turn.

Binary Features

While all of MENE's features have binary-valued output, the "binary" features are features whose "history" can be considered to be either on or off for a given token. Examples are "the token begins with a capitalized letter" or "the token is a four-digit number". The binary features which MENE uses are very similar to those used in BBN's Nymble system [1]. Figure 1 gives an example of a binary feature.

Lexical Features

To create a lexical history, the tokens at $w_{-2} \ldots w_2$ (where the current token is denoted as w_0) are compared with the vocabulary and their vocabulary indices are recorded.

$$g(h, f) = \left\{ \begin{array}{lll} 1 & : & \text{if Lexical\_View}(token_{-1}\text{(h))} = \text{“Mr” and } f = \text{per-} \\ & & \text{son\_unique} \\ 0 & : & \text{else} \end{array} \right\} \tag{4}$$

- Correctly predicts: Mr **Jones**

A more subtle feature picked up by MENE: preceding word is "to" and future is "location_unique". Given the domain of the MUC-7 training data, "to" is a weak indicator, but a real one. This is an example of a feature which MENE can make use of but which the constructor of a hand-coded system would probably regard as too risky to incorporate. This feature, in conjunction with other weak features, can allow MENE to pick up names that other systems might miss.

The bulk of MENE's power comes from these lexical features. A version of the system which stripped out all features other than section and lexical features achieved a dry run F-score of 88.13. This is very encouraging because these features are completely portable to new domains since they are acquired with absolutely no human intervention or reference to external knowledge sources.

Section Features

MENE has features which make predictions based on the current section of the article, like "Date", "Preamble", and "Text". Since section features fire on every token in a given section, they have very low precision, but they play a key role by establishing the background probability of the occurrence of the different futures. For instance, in NYU's evaluation system, the alpha value assigned to the feature which predicts "other" given a current section of "main body of text" is 7.9 times stronger than the feature which predicts "person_unique" in the same section. Thus the system predicts "other" by default.

Dictionary Features

Multi-word dictionaries are an important element of MENE. A pre-processing step summarizes the information in the dictionary on a token-by-token basis by assigning to every token one of the following five tags for each dictionary: start, continue, end, unique, other. I.e. if "British Airways" was in our dictionary, a dictionary feature would see the phrase "on British Airways Flight 962" as "other, start, end, other, other". The following table lists the dictionaries used by MENE in the MUC-7 evaluation:

Dictionary	Number of Entries	Data Source	Examples
first names	1245	www.babyname.com	John, Julie, April
corporate names	10300	www.marketguide.com	Exxon Corporation
corporate names without suffixes	10300	"corporate names" processed through a perl script	Exxon
colleges and universities	1225	http://www.utexas.edu/world/univ/alpha/	New York University; Oberlin College
Corporate Suffixes	244	Tipster resource	Inc.; Incorporated; AG
Dates and times	51	Hand Entered	Wednesday, April, EST, a.m.
2-letter State Abbreviations	50	www.usps.gov	NY, CA
World Regions	14	www.yahoo.com	Africa, Pacific Rim

Table 1: Dictionaries used in MENE

Note that we don't have to worry about words appearing in the dictionary which are commonly used in another sense. I.e. we can leave dangerous-looking names like "Storm" in the first-name dictionary because whenever the first-name feature fires on Storm, the lexical feature for Storm will also fire and, assuming that the use of Storm as "other" exceeded the use of Storm as person_start, we can expect that the lexical feature will have a high enough alpha value to outweigh the dictionary feature.

External Systems Features

For NYU's official entry in the MUC-7 evaluation, MENE took in the output of a significantly enhanced version of the traditional, hand-coded "Proteus" named-entity tagger which we entered in MUC-6 [2]. In addition, subsequent to the evaluation, the University of Manitoba [4] and IsoQuest, Inc. [3] shared with us the outputs of their systems on our training corpora as well as on various test corpora. The output sent to us was the standard MUC-7 output, so our collaborators didn't have to do any special processing for us. These systems were incorporated into MENE by a fairly simple process of token alignment which resulted in the "futures" produced by the three external systems become three different "histories" for MENE. The external system features can query this data in a window of $w_{-1} \ldots w_1$ around the current token.

$$g(h, f) = \left\{ \begin{array}{lcl} 1 & : & \text{if Proteus\_System\_Future}(token_0\text{(h)}) = \text{"person\_start" and } f = \text{person\_start} \\ 0 & : & \text{else} \end{array} \right\} \quad (5)$$

- Correctly predicts: **Richard** M. Nixon, in a case where Proteus has correctly tagged "Richard".

It is important to note that MENE has features which predict a different future than the future predicted by the external system. This can be seen as the process by which MENE learns the errors which the external system is likely to make. An example of this is that on the evaluation system the feature which predicted person_unique given a tag of person_unique by Proteus had only a 76% higher weight than the feature which predicted person_start given person_unique. In other words, Proteus had a tendency to chop off multi-word names at the first word. MENE learned this and made it easy to override Proteus in this way. Given proper training data, MENE can pinpoint and selectively correct the weaknesses of a handcoded system.

FEATURE SELECTION

Features are chosen by a very simple method. All possible features from the classes we want included in our model are put into a "feature pool". For instance, if we want lexical features in our model which activate on a range of $token_{-2} \ldots token_2$, our vocabulary has a size of V, and we have 29 futures, we will add $(5 \cdot (V + 1) \cdot 29)$ lexical features to the pool. The $V + 1$ term comes from the fact that we include all words in the vocabulary plus the unknown word. From this pool, we then select all features which fire at least three times on the training corpus. Note that this algorithm is entirely free of human intervention. Once the modeler has selected the classes of features, MENE will both select all the relevant features and train the features to have the proper weightings.

DECODING

After having trained the features of an M.E. model and assigned the proper weight (alpha values) to each of the features, decoding (i.e. "marking up") a new piece of text is a fairly simple process of tokenizing the text and doing various preprocessing steps like looking up words in the dictionaries. Then for each token we check each feature to whether if fires and combine the alpha values of the firing features according to equation 2. Finally, we run a viterbi search to find the highest probability path through the lattice of conditional probabilities which doesn't produce any invalid tag sequences (for instance we can't produce the sequence [person_start, location_end]). Further details on the viterbi search can be found in [7].

RESULTS

MENE's maximum entropy training algorithm gives it reasonable performance with moderate-sized training corpora or few information sources, while allowing it to really shine when more training data and information sources are added. Table 2 shows MENE's performance on the within-domain corpus from MUC-7's dry run as well as the out-of-domain data from MUC-7's formal run. All systems shown were trained on 350 aviation disaster articles (this training corpus consisted of about 270,000 words, which our system turned into 321,000 tokens).

Systems	Dry Run F-Measure	Dry Run Precision	Dry run Recall	Formal F-Measure	Formal Precision	Formal Recall
MENE (ME)	92.20	96	89	84.22	91	78
IsoQuest (IQ)	96.27	98	94	91.60	93	90
Manitoba (Ma)	93.32	94	92	86.37	87	85
Proteus (Pr)	92.24	95	90	86.21	93	85
MENE + IsoQuest	96.55	98	95	91.53	94	89
MENE + Proteus	95.61	97	94	88.80	93	85
MENE + Manitoba	95.49	97	94	88.91	92	86
ME + Ma + IQ	96.81	98	95	91.84	95	89
ME + Pr + IQ	96.78	98	96	92.05	95	89
ME + Pr + Ma	96.48	97	95	90.34	93	88
ME + Pr + Ma + IQ	97.12	98	96	92.00	95	89

Table 2: System combinations on unseen data from the MUC-7 dry-run and formal test sets

Note the smooth progression of the dry run scores as more information is added to the system. Also note that, when combined under MENE, the three weakest systems, MENE, Proteus, and Manitoba outperform the strongest single system of the group, IsoQuest's. Finally, the top dry-run score of 97.12 from combining all three systems seems to be competitive with human performance. According to results published elsewhere in this volume, human performance on the MUC-7 formal run data was in a range of 96.95 to 97.60. Even better is the score of 97.38 shown in table 3 below which we achieved by adding an additional 75 articles from the formal-run test corpus into our training data. In addition to being an outstanding result, this figure shows MENE's responsiveness to good training material.

The formal evaluation involved a shift in topic which was not communicated to the participants beforehand– the training data focused on airline disasters while the test data was on missile and rocket launches. MENE faired much more poorly on this data than it did on the dry run data. While our performance was still reasonably good, we feel that it is necessary to view this number as a cross-domain portability result rather than an indicator of how the system can do on unseen data within its training domain. In addition, the progression of scores of the combined systems was less smooth. Although MENE improved the Manitoba and Proteus scores dramatically, it left the IsoQuest score essentially unchanged. This may have been due to the tremendous gap between the MENE- and IsoQuest-only scores. Also, there was no improvement between the MENE + Proteus + IsoQuest score and the score for all four systems. We suspect that this was due to the relatively low precision of the Manitoba system on formal-run data.

We also did a series of runs to examine how the systems performed on the dry run corpus with different amounts of training data. These experiments are summarized in table 3.

Systems	425	350	250	150	100	80	40	20	10	5
MENE	92.94	92.20	91.32	90.64	89.17	87.85	84.14	80.97	76.43	63.13
MENE + Proteus	95.73	95.61	95.56	94.46	94.30	93.44	91.69			
MENE + Manitoba	95.60	95.49	95.26	94.86	94.50	94.15	93.06			
MENE + IsoQuest	96.73	96.55	96.70	96.55	96.11					
ME + Pr + Ma + IQ	97.38	97.12								

Table 3: Systems' performances with different numbers of articles

A few conclusions can be drawn from this data. First of all, MENE needs at least 20 articles of tagged training data to get acceptable performance on its own. Secondly, there is a minimum amount of training data which is needed for MENE to improve an external system. For Proteus and the Manitoba system, this number seems to be around 80 articles. Since the IsoQuest system was stronger to start with, MENE required 150 articles to show an improvement.

MENE has also been run against all-uppercase data. On this we achieved formal run F-measures of 77.98 and 82.76 and dry run F-measures of 88.19 for the MENE-only system and 91.38 for the MENE + Proteus system. The formal run numbers suffered from the same problems as the mixed-case system, but the combined dry run number matches the best currently published result [1] on all-caps data. We have put very little effort into optimizing MENE on this type of corpus and believe that there is room for improvement here.

CONCLUSIONS AND FUTURE WORK

MENE is a very new, and, we feel, still immature system. Work started on the system in October, 1997, and the system described above was not largely in place until mid-February, 1998 (about three weeks before the evaluation). We believe that we can push the score of the MENE-only system higher by adding long-range reference-resolution features to allow MENE to profit from terms and their acronyms which it has correctly tagged elsewhere in the corpus. We would also like to explore compound features (i.e. feature A fires if features B and C both fire) and more sophisticated methods of feature selection.

Nevertheless, we believe that we have already demonstrated some very useful results. Within-domain scores for MENE-only were good and this system is highly portable as we have already demonstrated with our result on upper-case English text. Porting MENE can be done with very little effort: our result on running MENE with only lexical and section features shows that it isn't even necessary to provide it with dictionaries to generate an acceptable result. We intend to port the system to Japanese NE to further demonstrate MENE's flexibility.

However, we believe that the within-domain results on combining MENE with other systems are some of the most intriguing. We would hypothesize that, given sufficient training data, any handcoded system would benefit from having its output passed to MENE as a final step. MENE also opens up new avenues for collaboration whereby different organizations could focus on different aspects of the problem of N.E. recognition with the maximum entropy system acting as an arbitrator. MENE also offers the prospect of achieving very high performance with very little effort. Since MENE starts out with a fairly high base score just on its own, we speculate that a MENE user could then construct a hand-coded system which only focused on MENE's weaknesses, while skipping the areas in which MENE is already strong.

Finally, one can imagine a large corporation or government agency acquiring licenses to several different N.E. systems, generating some training data, and then combining it all under a MENE-like system. We have shown that this approach can yield performance which is competitive with that of a human tagger.

REFERENCES

[1] Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing* (1997).

[2] Grishman, R. The nyu system for muc-6 or where's the syntax? In *Proceedings of the Sixth Message Understanding Conference* (November 1995), Morgan Kaufmann.

[3] Krupka, G. R., and Hausman, K. Isoquest: Description of the netowl(tm) extractor system as used in muc-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* (1998).

[4] Lin, D. Using collocation statistics in information extraction. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* (1998).

[5] Ratnaparkhi, A. A simple introduction to maximum entropy models for natural language processing. Tech. Rep. 97-08, Institute for Research in Cognitive Science, University of Pennsylvania, May 1997.

[6] Ristad, E. S. Maximum entropy modeling toolkit, release 1.6 beta, February 1998. Includes documentation which has an overview of MaxEnt modeling.

[7] Sekine, S. Nyu system for japanese ne - met2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)* (1998).