LLNL-CONF-455911

Pattern-based Aggregation of Named Entity Extractors

R. E. Glaser, P. B. Kidwell, K. A. Boakye, N. C. Perry, J. W. Guensche, J. J. Nitao, W. G. Hanley, T. D. Lemmond

September 20, 2010

SIAM
Mesa, AZ, United States
April 28, 2011 through April 30, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Pattern-based Aggregation of Named Entity Extractors

Ronald E. Glaser[1] Paul B. Kidwell[1] Kofi A. Boakye[1] Nathan C. Perry[2] Joseph W. Guensche[1]
John J. Nitao[1] William G. Hanley[1] Tracy D. Lemmond[1]

1. Lawrence Livermore National Laboratory
2. Brigham Young University

Abstract

Despite significant advances in named entity extraction technologies, state-of-the-art extraction tools achieve insufficient accuracy rates for practical use in many operational settings. However, they are not all prone to the same types of error, suggesting that substantial improvements may be achieved via appropriate combinations of existing tools, provided their behavior can be accurately characterized and quantified. Previously developed aggregation methodologies primarily focus upon naïve voting methods, which implicitly assign equal weight to their constituent extractors. Other more sophisticated approaches that utilize a fine-grained error space to characterize entity extractor performance do not lend themselves naturally to a model of joint extractor behaviors. In this paper, we present an inference framework that leverages the joint characteristics of their error processes via a pattern-based representation of extracted entity data. This approach has been shown to produce statistically significant improvements in entity extraction relative to standard performance metrics and to mitigate the weak performance of entity extractors operating under suboptimal conditions. Moreover, this aggregation methodology provides a framework for quantifying uncertainty in extracted entity output, and it can readily adapt to sparse data conditions.

Keywords: Knowledge discovery, text mining, named entity extraction, probabilistic aggregation, ensemble learning

1 Introduction

Since the 1980s, the sophistication of machine learning and computer technologies has increased dramatically, enabling the development of solutions to a wide variety of challenges facing the Natural Language Processing (NLP) community. These problems range from the development of search engines that can interpret simple natural language queries to the construction of knowledge discovery systems predicated upon reliable information extraction from heterogeneous data sources. Knowledge discovery systems are of particular interest to commercial, industrial and government organizations that rely upon computer processing to perform transactions; evaluate consumer demands; and, in general, to draw conclusions and make decisions that depend upon an extensive and reliable knowledge base. Often, the construction of such a knowledge base depends to a large degree upon the automatic recognition and extraction of complex relational information and, more fundamentally, related named entities (e.g., people, organizations) from a collection, or *corpus*, of text documents (e.g., e-mail, news articles, medical records, weblogs, intelligence reports). Consequently, the fidelity of knowledge discovery systems is particularly susceptible to errors introduced during the automatic extraction process.

However, even state-of-the-art entity extraction tools are vulnerable to variations in (1) the source and domain of a corpus and its adherence to conventional lexical, syntactical, and grammatical rules; (2) the availability and reliability of manually annotated data; and (3) the complexity of entity types targeted for extraction. Under these and other challenging conditions, extractors produce a range of interdependent errors that mangle entity output and often fail to achieve high accuracy rates in operational settings. However, many extraction technologies, distinguished by the nature of their underlying algorithms, possess complementary characteristics that may be combined to selectively amplify their most attractive attributes (e.g., low miss or false alarm rates) and mitigate their respective weaknesses.

Many extractor combination methods that aim to leverage these characteristics have relied upon variations of a “voting” mechanism (e.g., majority vote [1]). In practice, such approaches often fall short, as they depend heavily upon the number and type of extractors chosen, and they do not account for variations in the underlying extraction algorithms and the differing characteristics of their errors. Moreover, such systems tend to be limited in their ability to assess uncertainty, a critical capability for evaluating reliability in downstream analysis and decision-making. Proposed enhancements to the basic voting mechanism include, but are not limited to, weighting of the constituent (i.e., *base*) extractors' output [2]; stacking of entity extractors [3]-[5]; establishing a vote "threshold" [6]; and bagging of entity data [7].

Even more sophisticated combination techniques, such as that described in [8], fail to adequately account for text within a local neighborhood of a word of interest. Indeed, a method based on the Conditional Random Field (CRF) model presented by [9] demonstrated that performance may be enhanced by incorporating the classification structure of nearby words. More recently, Lemmond, et al. [10] utilized a fine-grained hierarchical error space to characterize named entity extractors' error processes and aggregate their output entity data accordingly.

The aggregation methodology described in this paper, called the *pattern-based meta-extractor (PME)*, utilizes a pattern-based representation of named entity data to evaluate the joint performance characteristics of its base entity extractors. The resulting characterization is utilized to determine the most likely truth, given base extractor output.

Section 2 of this paper describes this novel pattern representation, along with its use in characterizing base extractor performance and aggregating entity output. In Section 3, we discuss enhancements that enable the PME to adapt to sparse data conditions. Finally, experimental results are presented and discussed in Section 4, with conclusions and future research given in Section 5.

2 Extractor Characterization

In the following discussion, we assume that an entity can be expressed as a text string (i.e., *name*) that is associated with a *location*[1] in the source text. To enable the characterization of base extractor performance, we assume an annotated set of documents is available (distinct from those used for training) to serve as an "evaluation corpus" for the base extractors[2]. The *ground truth* entity data, G, consists of the true (i.e., manually annotated) entities identified in the evaluation corpus. The meta-extractor aggregates the output of $K > 1$ base entity extractors, where D_k denotes the output of extractor k relative to a corpus. When the locations of a ground truth entity and an extracted entity intersect, we say that the entities *overlap*[3].

2.1 The Pattern Representation

Named entity extractors leverage different methodologies that can be coarsely partitioned into three fundamental types: rule-based, statistical and heuristic. Despite their algorithmic differences, their common objective is to correctly extract fragments from text that represent real-world entities, such as people, organizations, or locations. At a high level, this task may be regarded as a three-stage process in which an extractor (in some prescribed order) must *detect* a reference to an entity in a document, *identify* the offsets that delineate the name of a detected entity, and *classify* it as to its type. We focus chiefly on the first two stages in this paper.

Many of the most effective extractors are proprietary, and hence, direct analysis of the characteristic error processes of their underlying algorithms is often infeasible. Therefore, we choose to treat each extractor as a "black box". However, when the base entity extractors are applied to a corpus for which the ground truth, G, is known, mistakes in their output, D_k, represent an observable transformation of the truth that is driven by their underlying error processes. In [10], this transformation was described in terms of a hierarchical error space relative to which the behaviors of each base extractor could be explicitly quantified. Despite the independence assumptions used in that study, the resulting meta-extractor achieved significant improvements over the performance of its base entity extractors. The PME methodology aims to further enhance these performance gains by relaxing those assumptions when sufficient data are available. Specifically, the PME utilizes an encoding of the combined base extractor output, $\mathbf{D}$, that encodes the joint characteristics of the extractors' output and resultant errors.

To lay a foundation for this encoding, we revisit a construct originally proposed in [10] called the *meta-entity*. This meta-extraction methodology assumed that when the base extractors are applied to a corpus for which ground truth is unknown, their combined entity output at a given location in the corpus encapsulates all available information regarding the corresponding underlying ground truth.

[1] We express the location of a text string in terms of its start and end offsets relative to the first character in the corpus.

[2] Three distinct corpora are required for: (1) training the base extractors; (2) evaluating their performance, and thereby training the meta-extractor; (3) testing the meta-extractor.

[3] We generally assume that ground truth entities do not overlap and that the entities extracted by extractor k do not overlap.

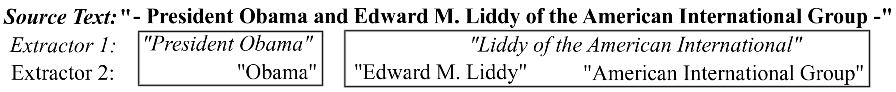
Source Text: "- President Obama and Edward M. Liddy of the American International Group -"
Extractor 1:
"President Obama"
"Liddy of the American International"
Extractor 2:
"Obama"
"Edward M. Liddy"
"American International Group"

Fig. 1. Meta-entities formed from extracted data: “President Obama”, “Edward M. Liddy of the American International Group”.

Hence, to facilitate discovery of the truth, mutually overlapping entities output by the K base extractors may be concatenated to form a *meta-entity*, which in turn can be used to generate a space of hypotheses over the ground truth. For example, in Figure 1, the extracted data within each rectangle can be concatenated to form two distinct meta-entities consisting of the following fragments of text:

(i) “President Obama”
(ii) “Edward M. Liddy of the American International Group”

This meta-entity concept, as summarized above, forms the basis for the PME encoding.

Let D_{mk} denote the entity output of base extractor k used to form meta-entity m, and let $\mathbf{D}_m = \{D_{m1},\ldots,D_{mK}\}$. Note that $\mathbf{D}_m$ consists of the K-way joint entity output of the K base extractors and possesses a distinctive structure that can be characterized by the boundaries of its individual entities. Specifically, the locations of its entity boundaries collectively define a *K-way pattern*, $\mathbf{d}_m$, relative to m that can be encoded numerically via the following process (illustrated in Figure 2):

(A) Meta-entity m is partitioned into s segments terminating at the $s+1$ unique entity boundaries in $\mathbf{D}_m$.
(B) For each extractor k, a string of length s (a *simple*[4] pattern denoted d_{mk}) is constructed, in which “2” indicates the beginning of an entity, “1” represents the middle or end of an entity, and “0” indicates that the segment was not extracted by extractor k.
(C) We represent the K-way pattern corresponding to the segmented meta-entity m by $\mathbf{d}_m = \{d_{m1},\ldots,d_{mK}\}$.

Note that this segmentation strategy is motivated by the assumption that, if two words in the meta-entity remain “unbroken” by the base extractors (e.g., “American International” in Figure 2), then they most likely remain unbroken in ground truth. That is, either they appear together within a single ground truth entity, or they do not appear in the ground truth at all. When this assumption holds, the likelihood that the truth will be discovered may increase, due to a simplification of the patterns involved. There is an implicit tradeoff, however, in that when this condition fails to hold, the truth can never be determined. Empirically, we have found that such cases are fairly rare, and on balance, the performance of the PME appears to benefit from this assumption.

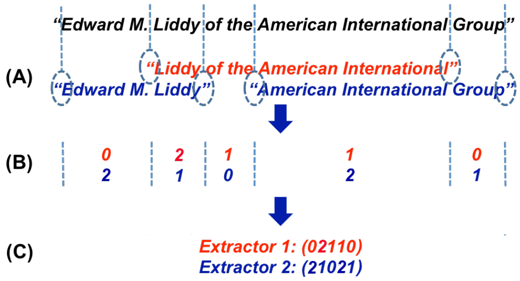
“Edward M. Liddy of the American International Group”
(A)
“Liddy of the American International”
“Edward M. Liddy”
“American International Group”
(B)
0
2
2
1
1
0
1
2
0
1
(C)
Extractor 1: (02110)
Extractor 2: (21021)

Fig. 2. The pattern-based encoding associated with extracted data relative to a meta-entity.

When the ground truth, G_m, associated with a meta-entity m is known and the above assumption is satisfied, an analogous simple pattern representation of ground truth can be derived from the meta-entity segmentation. For example, in Figure 2, the ground truth is given by G_m = {“Edward M. Liddy”, “American International Group”}, and its associated pattern is given by $g_m = (21021)$. Note that, relative to a given segmented meta-entity m, any simple pattern having the same number of segments as m is invertible; that is, it can be readily decoded to reflect the underlying entity data that it represents.

2.2 The Pattern Dictionary

The pattern-based encoding described in the previous section, by definition, relies solely on the joint *structure* of the entity data being encoded relative to a given segmented meta-entity. This is by design; many application domains require language independent extraction tools. Consequently, a particular K-way pattern of extracted data may be repeatedly observed in a corpus regardless of the actual text involved in the associated meta-entities. For example, in Figure 3, the extracted data are associated with a joint pattern identical to that shown in Figure 2. Notice, however, that although the extracted data for these two examples give rise to the same encoding, their associated ground truths differ. In particular, the ground truth in Figure 3 is given by G_m ={“Joe Biden”, “Delaware”}, with the associ-

[4] A *1-way pattern* is also referred to as a *simple* pattern.

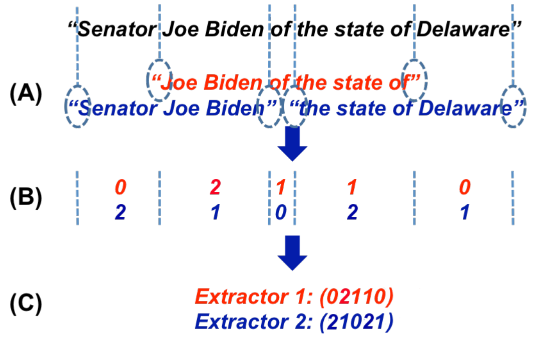

Fig. 3. The joint pattern representation for a different collection of extracted data, identical to that in Figure 2.

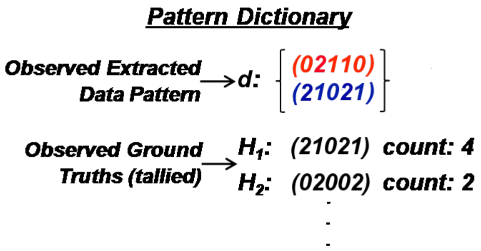

Fig. 4. Example pattern dictionary entry.

ated pattern $g_m = (02002)$. Hence, it is readily apparent that a particular pattern of extracted data, $\mathbf{d}_m$, may be associated with many different ground truth patterns; in fact, the total number a_s of unique ground truth hypotheses that may be encoded for a meta-entity of length s segments is given by $a_0 = 1, a_1 = 2, a_s = 3a_{s-1} - a_{s-2}$. This leads to $a_2 = 5$, $a_3 = 13$, $a_4 = 34$, $a_5 = 89$, and so on. Clearly, only a subspace of the possible encodings will be observed in the training data for long patterns. Indeed, in practice, as pattern length increases, the relative size of this observed subspace shrinks rapidly. Some implications of this behavior will be discussed in later sections.

In an operational setting, the base entity extractors are applied to a corpus for which ground truth is unknown. With access to *only* the extracted entity output of its K extractors, the PME must determine the most likely ground truth (i.e., the set of *true* named entities, G)[5]. This process involves (1) forming a collection of meta-entities from the extractor output, $\mathbf{D}$, and (2) for each meta-entity m, determining the ground truth hypothesis (i.e., pattern) that is most plausible in a Bayesian sense among the a_s possible hypotheses. We will show that the optimal ground truth hypothesis H_m^*, given $\mathbf{D}_m$, is that most frequently associated with the K-way pattern $\mathbf{d}_m$ in the evaluation data set.

Evaluation of base extractor performance relative to an annotated data set consists of constructing a database, or *pattern dictionary*, from the evaluation data that stores counts of observed ground truth patterns for each K-way pattern derived from the extracted data. For example, a final entry in the pattern dictionary might resemble that shown in Figure 4 for the 2-way pattern presented in Figures 2 and 3.

Consider a particular meta-entity m of size s having the K-way pattern $\mathbf{d}_m$ and unknown ground truth. Let $\theta_1,...,\theta_n\,(\sum\theta_j = 1)$ denote the respective probabilities of the $n = a_s$ hypothesized ground truths, $H_{m1},...,H_{mn}$. Suppose there are a total of $N = N^{(K)} \geq 1$ occurrences in the pattern dictionary of the pattern $\mathbf{d}_m$. Since the corresponding collection of N meta-entities may be regarded as a random sample from the population which generates the pattern $\mathbf{d}_m$, the resulting pattern dictionary counts, i.e., the observed frequencies $f_1,...,f_n\,(\sum f_j = N)$ of the set of possible ground truths, may be modeled as following a multinomial distribution. The frequency f_j may be viewed as the number of "votes" for the ground truth hypothesis H_{mj}.

The conjugate prior for the multinomial distribution is the Dirichlet distribution, $D(\alpha_1,...,\alpha_n)$, where the parameter α_j is essentially the number of *a priori* votes for hypothesis H_{mj}. For our application, we have used a noninformative Dirichlet prior, i.e., $\alpha_1 = \cdots = \alpha_n = 1/n$, which, in effect, splits a single *a priori* vote evenly among the candidate ground truths.

The posterior distribution of $\theta_1,...,\theta_n$ then, given the observed frequencies $f_1,...,f_n$, is $D(1/n + f_1,...,1/n + f_n)$. These frequencies have the effect of updating the number of votes for hypothesis H_{mj} to $1/n + f_j$. Hence, the marginal posterior distribution of θ_j is the beta distribution with parameters $A_j = \alpha_j + f_j = 1/n + f_j$ and $B_j = \sum\alpha_i + \sum f_i - (\alpha_j + f_j) = 1 + N - (1/n + f_j)$. It is this distribution that should be used to model the credibility of the hypothesized ground truth H_{mj}. In particular, the posterior mean for θ_j is given by

$$\tilde{\theta}_j = E\left(\theta_j \middle| f_1,\ldots,f_n\right) = \frac{1/n + f_j}{1+N}$$
$$= \frac{1}{1+N}\frac{1}{n} + \frac{N}{1+N}\frac{f_j}{N},$$

[5] To address the efficiency requirements of many real-world (e.g., streaming text) applications, we assume that the source text cannot directly be accessed in this task.

which is a weighted average of the prior mean, $1/n$, for θ_j and the sample proportion, $\hat{\theta}_j = f_j / N$, of observed patterns associated with H_{mj}. The weight $1/(1 + N)$ represents the fraction of evidence coming from the prior.

The Bayesian optimum ground truth hypothesis H_m^* is the H_{mj} that maximizes the posterior mean $\tilde{\theta}_j$. Moreover, it is apparent from the formulation that it is equivalent to maximize $\hat{\theta}_j$. Hence, the optimal hypothesis is simply that most frequently associated with the K-way pattern $\mathbf{d}_m$ in the evaluation data set, easily determined via the pattern dictionary.

In some applications of this technology, analysts may wish to consider some sub-optimal hypotheses having relatively high measures of plausibility. Candidate hypotheses H_{mj} may be ranked equivalently by $\tilde{\theta}_j$ or $\hat{\theta}_j$, although as a point estimate of credibility, $\tilde{\theta}_j$ serves as the preferred figure of merit in the Bayesian paradigm. In addition, estimate uncertainty may be quantified by means of a Bayesian interval for θ_j based upon its beta posterior (easily constructed from the inverse cumulative beta distribution).

The Bayesian interval, by capturing a specified portion of the posterior distribution, provides a range of plausible values. For example, an 80% Bayesian interval can be defined to capture the central 80% of the distribution by extending from the 10th to the 90th percentile. In instances of sparseness of relevant pattern data (small N), in order to get reasonably short ranges, lower probability Bayesian intervals (e.g., 50%) may be used. A useful list for an analyst would display the posterior mean and associated Bayesian interval for the top hypotheses.

Since a K-way extracted pattern may be associated with many different ground truths, it is natural at this point to question the use of *structure* alone in attempting to discover the truth. Indeed, the use of gazetteers, lexicons, stop-word lists and other commonly employed language-specific tools would undoubtedly enhance performance in some cases. However, since we are motivated by a practical need for language-independent systems, our goal in this work is to optimize performance in the absence of linguistic and semantic knowledge.

3 Unprecedented Patterns

When new extractor output $\mathbf{D}_m$ is encountered in the field, it may happen that the associated K-way pattern, $\mathbf{d}_m$, was not observed in the evaluation data set and, consequently, cannot be found in the pattern dictionary ($N^{(K)} = 0$). Under conditions in which (1) the evaluation data set is of large enough size; (2) there are few base extractors under consideration; and/or (3) the base extractors exhibit similar behaviors with regard to extraction errors, this phenomenon is not frequently observed. Unfortunately, in practice, these conditions often do not hold, and hence, we present two enhancements of the PME that enable it to adapt to these challenging conditions.

3.1 Stepping Down

The K-way pattern described above is essentially a joint model over the K extractors and their corresponding behavior with respect to a given meta-entity. It is reasonable to assume that the pattern algorithm, if necessary, can utilize progressively weaker marginal models in an effort to capture some patterns that would not otherwise be observed. We call this process "stepping down".

Stepping down involves reducing the number of extractors represented by the patterns in the dictionary in an effort to increase the likelihood that a given joint pattern will have been observed. This means that, in building the pattern dictionary, we must additionally store counts of observed ground truth patterns for each k-way pattern derived from the extracted data, $1 \leq k \leq K - 1$. During operation of the PME, when a K-way pattern cannot be found in the dictionary, frequencies of these smaller k-way patterns, $k < K$, are used to determine plausible ground truth. The particular value of k employed will be referred to as the stepping down *level*.

Here, we focus chiefly upon two approaches to implementing this stepping down procedure.

Simple k-way decision

A straightforward implementation of stepping down involves querying the dictionary for all possible k-way patterns, for successively smaller k, $k < K$, until one or more patterns is found. A K-way pattern $\mathbf{d}_m$ induces $T = \binom{K}{k}$ k-way patterns $\mathbf{d}_{mt}$, $t = 1, \ldots, T$, according to the combination of extractors represented. As shown in Figure 5, each k-way pattern $\mathbf{d}_{mt}$ and its associated ground truth patterns are reconfigured, if necessary, to comply with the segmentation induced by the s-segment K-way pattern $\mathbf{d}_m$. Again, let $\theta_1, \ldots, \theta_n$ denote the respective probabilities of the $n = a_s$ possible ground truths, $H_{m1}, \ldots, H_{mn}$. Suppose there are a total of $N_t \geq 0$ occurrences in the pattern dictionary of the pattern $\mathbf{d}_{mt}$,

Ground Truth: "*President Obama*" (*21*)

(A) **Extractor 1: "*President Obama*" (*21*)**
Extractor 2: "*President Obama*" (*21*)
Extractor 3: "*Obama*" (*02*)

(B) **Extractor 1: "*President Obama*" (*21*)**
Extractor 2: "*President Obama*" (*21*)

Fig. 5. The 2-way pattern representation formed by Extractors 1 and 2 (B), as well as that of its associated ground truth, maintains the sementation of the original 3-way pattern (A), despite the lack of disagreement between the two extractors.

with $N = N^{(k)} = \sum N_t \geq 1$. Since we regard the corresponding collection of N meta-entities as a random sample from the population which generates patterns from $\bigcup_t \mathbf{d}_{mt}$, the resulting pattern dictionary counts, i.e. the observed frequencies $f_1, \ldots, f_n$ ($\Sigma f_j = N$) of the set of possible ground truths, may again be modeled as following a multinomial distribution. Here the frequencies are pooled over the T k-way pattern dictionaries. Bayesian inferences proceed as in the full K-way case, with the same expressions for $\tilde{\theta}_j$ and $\hat{\theta}_j$. Analogous Bayesian intervals may be constructed. It is interesting to note that the sample proportion $\hat{\theta}_j$ may be expressed as $\hat{\theta}_j = \sum_{t:N_t>0} \frac{N_t}{N}\frac{f_{jt}}{N_t}$, where f_{jt} denotes the frequency of ground truth H_{mj} occurring in the t^{th} k-way dictionary, and $\sum_t f_{jt} = f_j$. Hence $\hat{\theta}_j$ is a weighted average of the k-way sample proportions $\hat{r}_{jt} = \frac{f_{jt}}{N_t}$, weighted by the relative sample sizes.

While this approach has been shown to be reasonably effective, it does not explore and compare probability estimates for all extractor combinations at all values of k. To this end, we have developed an alternative approach that does so.

Lower Bound Maximization (LBM)

The essence of the LBM method consists of stepping down to the "best" combination of extractors, subject to a constraint on the reliability of the estimated probability of the top-ranking hypothesis associated with each combination. The LBM accounts for the fact that some combinations may exhibit better performance than others and leverages this fact given a pre-specified level of confidence.

The LBM method uses the lower Bayesian bound as a metric to compare hypotheses' probability estimates. Specifically, for each combination of base extractors i, the lower bound on the estimated probability of hypothesis H_{mj}, denoted by $x = l^{(i)}\left(H_{mj}\right)$, is the solution to

$$I_x\left(A_j^{(i)}, B_j^{(i)}\right) = \alpha,$$

where I_x denotes the incomplete beta function, and the parameters of the corresponding beta distribution are computed in a fashion similar to that described in the preceding section.

The parameter α < 0.5 is pre-specified such that $1-\alpha$ indicates the desired degree of confidence in a bound. Since higher bounds indicate greater plausibility, by comparing the bounds over all levels and hypotheses, we effectively are able to rank the ground truth probabilities. The LBM optimum ground truth hypothesis, H_m^*, achieves the largest bound, i.e.

$$H_m^* = \arg\max_{H_{mj}} \left(\max_i l^{(i)}(H_{mj})\right).$$

Empirically, we have found the LBM method to be fairly insensitive to the choice of α.

In a similar fashion as stepping down, the LBM simultaneously addresses both the quality and uncertainty of estimates by assigning heavier weights to hypotheses associated with more observations $N^{(i)}$. Moreover, by introducing a confidence metric, it provides an avenue for directly comparing the estimates arising from the totality of possible extractor combinations.

Note that, although the simple k-way decision method described above aggregates the votes within each level $k \leq K$, the LBM methodology detailed here takes a different approach. Indeed, one could imagine implementing the LBM over all levels $k \leq K$, rather than over all combinations of base extractors. We have, in fact, investigated this approach and have found in empirical tests that when data are plentiful, it performs equally well. However, when data are sparse, the former implementation appears to be highly susceptible to the influence of weak base extractors. We conjecture that the latter approach, as described, has the advantage of disregarding weak base extractors, thereby improving performance under sparse data conditions.

3.2 A Sequential Meta-Entity Model

Although the marginal models utilized in Section 3.1 enhance the PME's ability to make decisions under sparse data conditions, there certainly remain cases in which even these techniques are unsuccessful.

Recall from our previous discussion that the *K*-way pattern encodes joint information among the errors (implicitly, via text structure) as well as among the base extractors. In many cases, the rarest of meta-entities consist of lengthy patterns, which represent a complex sequence of errors and disagreement among the extractors. Moreover, the underlying dependencies among extractors and among these implicit errors is unknown. Thus, it is reasonable to incrementally break down a *K*-way pattern across errors, rather than across extractors, so that the patterns arising from a single meta-entity are represented by progressively fewer segments. We can address this approach via a sequential modeling technique that is often used in other language-based applications[6]. For example, let us consider a *3*-way pattern $\mathbf{d}_{\mathrm{m}}$, together with a hypothesis H_{mj}, as a sequence of columns as shown in Table 1.

Table 1: Columnwise representation of a pattern and corresponding hypothesis.

	c_1	c_2	c_3	c_4
d_{m1}	2	1	2	1
d_{m2}	2	1	0	2
d_{m3}	2	0	0	2
H_{mj}	2	1	0	2

We can decompose the joint probability of the pattern ($\mathbf{d}_m$, H_{mj}) in Table 1 as follows:

$$P(\mathbf{d}_m, H_{mj}) = P(c_1, c_2, c_3, c_4)$$
$$= P(c_1)\prod_{t=2}^{4} P(c_t \mid c_{t-1}, \ldots, c_1),$$

where each column pattern is dependent upon those that precede it. Hence, when a complex pattern is encountered that cannot be handled by the previously described methods, we make the assumption that each column pattern is dependent only upon the preceding *n* columns, with $n < s-1$, giving

$$P(\mathbf{d}_m, H_{mj}) = P(c_1)\prod_{t=2}^{s} P(c_t \mid c_{t-1}, \ldots, c_{t-n}).$$

Under this framework, we select the hypothesis H_m^* that satisfies

$$H_m^* = \arg\max_{H_{mj}} P(\mathbf{d}_m, H_{mj}).$$

Note that taking $n = 1$ in this sequential modeling approach yields a standard Markov model, which is commonly employed in natural language applications. We have generally found this small window size to be fairly effective, requiring the least amount of data to obtain reliable probability estimates. Additionally, this approach can be applied to meta-entities segmented as described in Section 2, or meta-entities segmented by their individual tokens. Both approaches have performed well empirically.

4 Empirical Studies

In this section, we present results from three aggregation experiments using the output of (1) GATE, a rule-based extraction tool [11]; (2) LingPipe, an extraction tool based on Hidden Markov Models (HMMs) [12]; (3) Stanford Named Entity Recognizer (SNER), based on CRFs [13]; and (4) BALIE, an extraction tool that utilizes unsupervised learning [14]. These experiments were carried out using two publicly available annotated data sets, MUC6 (Wall Street Journal) and MUC7 (New York Times), as well as a small operational data set called TAI consisting of 40 annotated documents (containing approximately 700 ground truth entities).

The following studies compare the performance of the PME where stepping down is implemented up to *n* levels, $n = 0,\ldots,3$ (i.e., "PA*n*"), together with the LBM method ("LBM"). In all cases, when a pattern could not be found in the pattern dictionary after stepping down or LBM was employed, we utilized the Sequential Modeling algorithm to determine a winning hypothesis.

We focused on two relevant real-world scenarios. The first two involved a test in which the base extractors and the PME used identical training data. The PME, which requires annotated data for evaluation, necessarily used base extractors trained on less data, thus pitting these weak learners against their stronger, standalone versions. To this end, MUC6 and MUC7 were used in a 10-fold cross-validation procedure where, for each fold, 10% of the corpus was set aside for testing, and the remaining 90% was used to train and evaluate the base extractors (via 9-fold cross-validation). The resulting ten performance estimates were bootstrapped (2000 samples) and presented in box plots (Figs. 6 and 7).

The second scenario involved more challenging conditions in which the base extractors were not trained using representative data. These may include cases where (1) proprietary or rule-based extraction tools cannot be (re)trained, (2) source text is evolving over time and continual retraining of the base

[6] Natural language applications lend themselves to such models; there is inherent meaning in the *order* of words / characters, and dependencies are often localized.

extractors is computationally infeasible, or (3) the annotations used to train the base extractors are flawed. We simulated these conditions by training the base extractors on MUC6 and then evaluating their performance and aggregating their output on TAI. As in the first scenario, we performed 10-fold cross-validation, and the resulting estimates were bootstrapped and plotted (Fig. 8).

4.1 Results

In the following figures, we have presented our results in terms of *F Measure*, computed in the usual fashion, i.e.,

$$F = \frac{2PR}{P+R},$$

with the Precision, *P*, and Recall, *R*, given by

$$P = \frac{c + 0.5 * p_E}{E},\ R = \frac{c + 0.5 * p_G}{G},$$

where *G* and *E* are the number of ground truth and extracted entities, respectively; p_G and p_E are partial matches of the ground truth and extracted entities, respectively, and *c* is the number of correct extractions (i.e., true positives). This formulation for Precision and Recall is motivated by an interest in quantifying the usability of extracted data, under the assumption that a partially correct extraction is more valuable than a miss, but less valuable than a correctly extracted entity.

In addition to *F* Measure, we have presented our results in terms of Exact Match (EM) rates, and the combined Miss and False Alarm rates for each base extractor and the PME variants. These error types are often traded off to address operational requirements, but here we focus on the combined impact of both.

We also assessed statistical significance relative to F Measure via a nonparametric pairwise test performed on the results from the original ten folds.

Figures 6 and 7 present the results generated for the first experimental scenario. For both MUC6 and MUC7, the base extractors founded upon statistical methodologies, LingPipe and SNER, produced *F* measures that significantly exceeded those of GATE and BALIE (p = 0.001). In general, we expected this behavior, since statistical methodologies often excel when they are trained on representative data. However, the performance of GATE greatly exceeded that of BALIE. BALIE was trained on a set of prepackaged untagged websites, negatively impacting its performance in our experiments.

Note that, although the EM rate of the LBM method was roughly equivalent to the EM rate of LingPipe for the MUC6 experiment, LBM produced a lower error rate than SNER and, consequently a significantly higher F measure (for MUC6, p = 0.001; for MUC7, p = 0.005).

Note that for both the MUC6 and MUC7 experiments, stepping down with respect to the number of base extractors results in a significantly improved *F* measure (with a p-value $\leq$ 0.002 in each case). These results suggest that the PME benefits from stepping down as far as possible before reverting to the Sequential Modeling method (i.e., when a pattern cannot be found at the lowest level). However, other empirical tests (not shown here) have demonstrated that this is not always the case. Indeed, particularly when data are sparse, we have observed that it is sometimes advantageous to interrupt the stepping down process and defer the decision to the sequential method.

Our conjecture regarding this observation is that the optimal approach is somewhat dependent upon the underlying “true” model; i.e., the dependencies among the extractors and among the implicit errors (the text structure itself). In truth, each approach has its strengths and weaknesses, and will perform differently under varying conditions. Hence, to remedy this situation, one might imagine a technique that learns these behaviors and selects the optimal aggregation methodology.

In the second experimental scenario, we examine results from the TAI data set. To provide some background, the TAI data set was roughly one-tenth the size of MUC6 (which is roughly half the size of MUC7), and was annotated according to MUC6 guidelines[7]. As it turns out, this annotation was poorly performed with many underlying true entities unidentified. Hence, this situation mimics those of the third condition described above (i.e., annotations used to train the base extractors are flawed). Specifically, we may regard the incomplete TAI annotations as a relevance-based annotation, in which only entities of interest have been identified relative to some operational need. In such a case, MUC6 turns out to be nonrepresentative, and base extractors trained on MUC6 are poorly equipped to perform effectively when applied to TAI.

[7] The annotation was performed by a third party, and was apparently poor because the annotator ran out of time. Hence, the annotated data were not deliberately flawed.

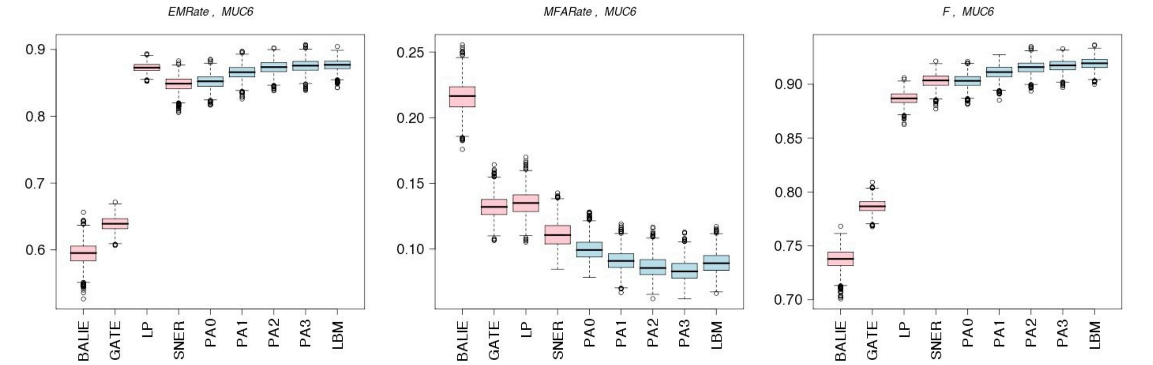

Fig. 6. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC6 for the first experimental scenario. "PA*n*" represents the pattern algorithm, using the simple *k*-way decision stepping down process to step down up to *n* levels. "LBM" presents results from the LBM method, alpha = 0.3. Patterns not found are processed using the Sequential Modeling method.

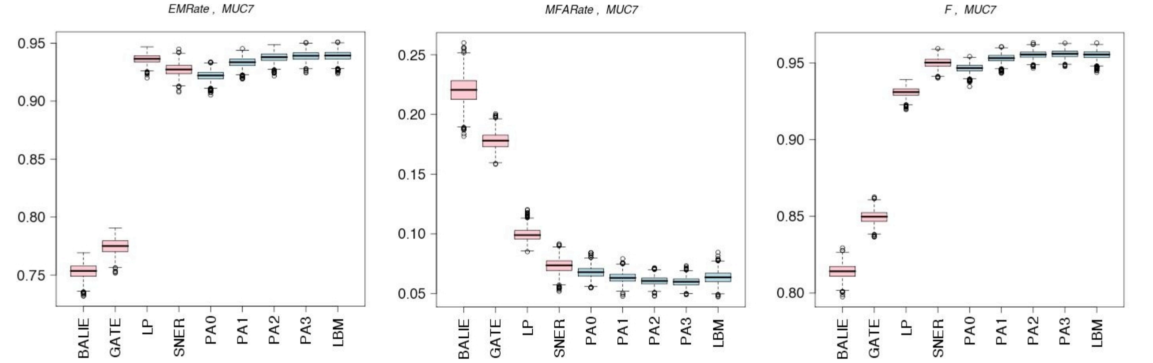

Fig. 7. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC7 for the first experimental scenario. BALIE and GATE performed poorly relative to LP and SNER, much like MUC6. The LBM again uses alpha = 0.3.

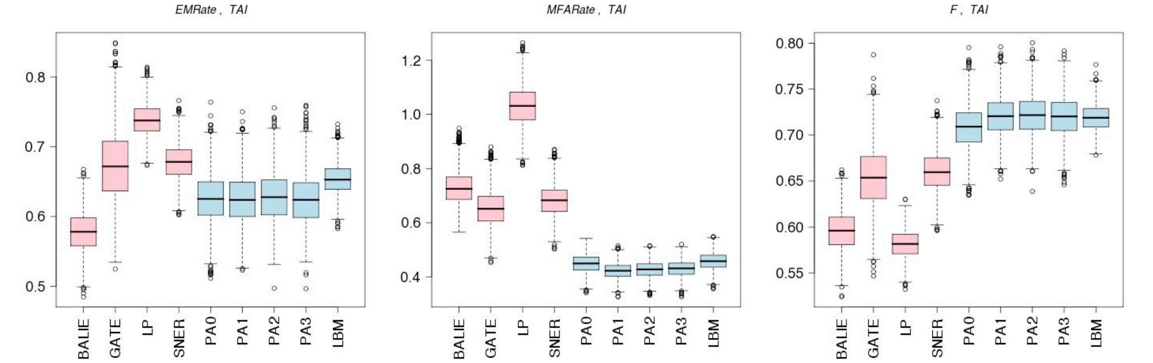

Fig. 8. Left to Right: Exact match rates, Miss + FA rates, F measure on TAI for the second experimental scenario. The performances of BALIE and GATE were more robust relative to LP and SNER. The LBM again uses alpha = 0.3.

The results for TAI are presented in Figure 8. It is clear from the plot that, as in other experiments, the PME successfully mitigated the decreased performance of the base extractors. Note that GATE's performance remained relatively robust, as it does not require training and, hence, is not susceptible to the flaws in the training data set. SNER's performance degraded significantly, but at least produced results comparable to GATE. On the other hand, the performance of LingPipe dropped precipitously, largely because its error rate increased by nearly an order of magnitude. Indeed, its false alarm rate was approximately equal to the number of ground truth entities (i.e., there was roughly one false alarm per ground truth entity). We have observed that our version of LingPipe tends, in general, to produce more false alarms than other methods. This propensity was amplified under these conditions.

With respect to the PME and LBM in Figure 8, their respective performance was not found to be statistically different (p = 0.55), but the results again indicate that the LBM is competitive with the PME.

5 Conclusions and Future Work

In this paper, we have presented a pattern-based aggregation methodology – the PME – that implicitly incorporates the joint behaviors of extractors and their error processes. Through the integration of marginal models and corresponding representations of extracted data, the PME has proven to be highly effective. Specifically, it has been shown to achieve statistically significant improvements in the summary metric, *F* Measure, over its base entity extractors in multiple experimental scenarios and on multiple data sets. Even under sparse data conditions, where marginal models become more critical, the PME remains highly effective.

Strategies for integrating across multiple marginal models under these conditions were also presented and their relative performance compared. One such strategy—the simple *k*-way decision—though generally effective and straightforward, makes the decision to step down based only upon the absence of a pattern in the pattern dictionary, without regard to uncertainty or accuracy across levels (i.e., different values of *k*). As a consequence, decisions may sometimes be made by few or highly variable data.

An alternative approach to the *k*-way decision, the LBM method, is able to account for the uncertainty across the various extractor combinations. Specifically, this method selects an optimum hypothesis according to a Bayesian lower bound metric appropriate and applicable across all of the combinations. As a result, it is competitive with the best-performing PA*n* algorithm in each of these empirical studies relative to *F* Measure.

Notably, both of the methods presented for stepping down require that a parameter be specified for optimal performance. Specifically, the *k*-way decision requires the selection of the minimum level *k*, while the LBM method requires that the parameter α be specified. However, our studies have generally shown that the LBM method is fairly insensitive to the choice of α, and for the *k*-way decision, the choice of $k = 1$ as the minimum level is frequently the most effective.

Although the PME is capable of adapting to sparse data conditions, maintaining high performance in the presence of such a challenge is not a simple issue to address. Future work will include sensitivity studies to evaluate the impact of data sparseness. However, this is only one of many challenging conditions that may be encountered in a real world operational setting.

In text applications, a wide variety of meta-entities are observed. These meta-entities can be distinguished by structural features derived from their underlying patterns of base extractor text. Other research we have performed has demonstrated that the effectiveness of different aggregation algorithms can be linked directly to these characteristic features. Consequently, our future focus will be the development of even more effective and robust aggregation systems. In particular, we will investigate systems that can determine, via state-of-the-art machine learning methods, the algorithms and parameters that are optimal given specific operational conditions and meta-entity features, and thus can assign meta-entities to the most favorable algorithms accordingly.

Acknowledgments

We would like to acknowledge and thank Doug Poland of Lawrence Livermore National Laboratory (LLNL) for providing the TAI data set used in this study. We also would like to thank Kyle Halliday, also of LLNL, for lending his subject matter expertise in the application space for this work.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] Z. Kozareva, O. Ferrández, et al., “Combining data-driven systems for improving named entity recognition,” *Data & Knowledge Engineering*, vol. 61-3, pp. 449-466, June 2007.

[2] D. Duong, B. Goertzel, et al., “Support vector machines to weight voters in a voting system of entity extractors,” in *Proc. IEEE World Congress on Computational Intelligence*, Vancouver, Canada, 2006, 1226-1230.

[3] H. Wang, and T. Zhao, “Identifying named entities in biomedical text based on stacked generalization,” in *Proc. 7th World Congress on Intelligent Control and Automation*, Chongqing, China, 2008, pp. 160-164.

[4] D. Wu, G. Ngai, and M. Carpuat, “A stacked, voted, stacked model for named entity recognition,” in *Proc. CoNLL-2003*, vol. 4, Edmonton, Canada, 2003, pp. 200-203.

[5] R. Florian, “Named entity recognition as a house of cards: classifier stacking,” in *Proc. 6th*

Conference on Natural Language Learning, Taipei, Taiwan, vol. 20, 2002, pp.1-4.

[6] N. Kambhatla, “Minority vote: at-least-N voting improves recall for extracting relations,” in *Proc. COLING/ACL on Main Conference Poster Sessions*, Sydney, Australia, 2006, pp. 460-466.

[7] P. Kegelmeyer and M. Goldsby, “Massive ensembles for mindlessly improving named entity recognition,” unpublished.

[8] R. Florian, A. Ittycheriah, et al., “Named entity recognition through classifier combination,” in *Proc. CoNLL-2003*, vol. 4, Edmonton, Canada, 2003, pp. 168-171.

[9] L. Si, T. Kanungo, and X. Huang, “Boosting performance of bio-entity recognition by combining results from multiple systems,” in *Proc. 5th International Workshop on Bioinformatics*, Chicago, IL, 2005, pp. 76-83.

[10] T. Lemmond, et al., “Enhanced Named Entity Extraction via Error-Driven Aggregation,” in *Proc. Intl. Conference on Data Mining*, Las Vegas, NV, July, 2010, pp. 31-37.

[11] H. Cunningham, D. Maynard, et al., “GATE: a framework and graphical development environment for robust NLP tools and applications,” in *Proc. 40th Anniversary Meeting of the Assoc. for Computational Linguistics*, Philadelphia, PA, 2002.

[12] Alias-I, LingPipe 3.8.2, 2008. http://alias-i.com/lingpipe.

[13] Stanford University, Stanford Named Entity Recognizer 1.1, 2008. http://nlp.stanford.edu/software/CRF-NER.shtml.

[14] University of Ottawa, Baseline Information Extraction (BALIE) 1.81, 2004. [Online] http://balie.sourceforge.net/.