Pattern-based Aggregation of Named Entity Extractors

T. Lemmond[1], P. Kidwell[1], K. Boakye[1], N. Perry[2], J. Guensche[1], J. Nitao[1], W. Hanley[1], R. Prenger[1], and R. Glaser[1]
[1]Lawrence Livermore National Laboratory, Livermore, CA, USA
[2]Mathematics Department, Brigham Young University, Provo, UT, USA

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. 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. 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 conditions extractors produce a range of interdependent errors 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 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 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.

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 describes the 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 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 that is associated with a *location* 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. 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

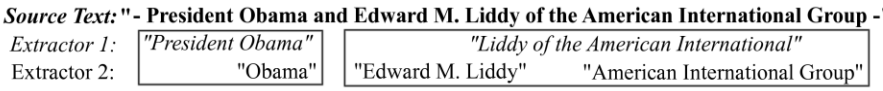

Fig. 1. Meta-entities formed from extracted data: "President Obama", "Edward M. Liddy of the American International Group".

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*.

2.1 The pattern representation

Named entity extractors leverage a variety of different methodologies to correctly extract fragments from text that represent real-world entities, such as people, organizations, or locations. Many 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. The PME utilizes an encoding of the combined base extractor output, **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 the combined entity output of the base extractors at a given location in the corpus encapsulates all available information regarding the ground truth. 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"

Let D_{mk} denote the entity output of base extractor *k* used to form meta-entity m, and let $\mathbf{D}_m = \{\mathbf{D}_{m1},...,\mathbf{D}_{mK}\}$. 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 1-way or *simple* 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 $\mathbf{d}_m = \{d_{m1},...,d_{mK}\}$.

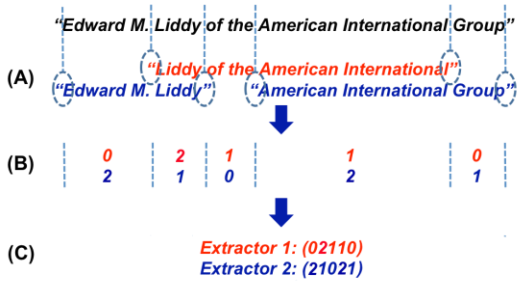

Fig. 2. The pattern-based encoding associated with extracted data relative to a meta-entity.

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. Empirically, we have found that the performance of the PME appears to benefit from this assumption.

When the ground truth, G_m, associated with a meta-entity *m* is known and the above assumption is made, 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 $g_m = (21021)$.

2.2 The pattern dictionary

The pattern-based encoding described in the previous section relies solely on the joint *structure* of the entity data being encoded relative to a given segmented meta-entity. 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. However, despite the similar encoding of the extracted data, their associated ground truths differ. In particular, the ground truth in Figure 3 is given by G_m ={"Joe Biden", "Delaware"}, with the associated pattern $g_m = (02002)$. Hence, 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 $a_0 = 1, a_1 = 2, a_s = 3a_{s-1} - a_{s-2}$. 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*). This process involves forming a collection of meta-entities from the extractor output, $\mathbf{D}$, and for each meta-entity *m*, determining the ground truth hypothesis 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 $D(\alpha_1,...,\alpha_n)$. For our application, we used a noninformative Dirichlet prior, $D(\alpha_1 = ... = \alpha_n = 1 \setminus n)$, 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 $A_j = 1/n + f_j$ and $B_j = 1 + N - (1/n + f_j)$. 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

$$\tilde{\theta}_j = E(\theta_j \mid f_1,..., f_n) = \frac{1}{1+N}\frac{1}{n} + \frac{N}{1+N}\frac{f_j}{N},$$

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}.

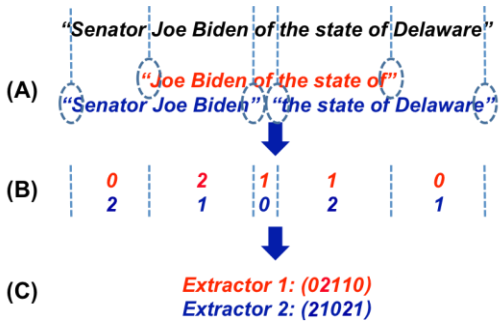

Fig. 3. The joint pattern representation for a different collection of extracted data, identical to that in Fig. 2.

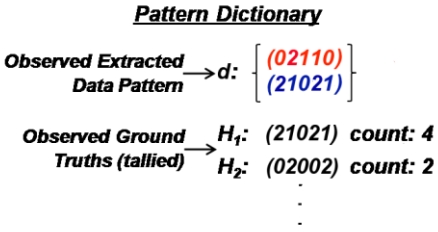

Fig. 4. Example pattern dictionary entry.

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.

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 ($N^{(K)} = 0$). 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 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 and LBM.

3.1.1 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 $\mathbf{d}_{mt}$, 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 $\cup_t \mathbf{d}_{mt}$, the resulting pattern dictionary counts, i.e. the observed frequencies $f_1, \ldots, f_n$ ($\sum f_j = N$) 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.

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.

3.1.2 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 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 $x = l^{(i)}(H_{mj})$, is the solution to

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

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 suggest 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, 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.

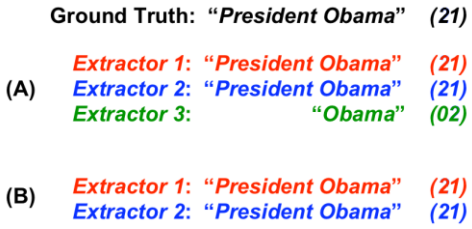

Fig. 5. The 2-way pattern representation formed by Extractors 1 and 2 (B), as well as that of its associated ground truth, maintains the sementation of the original 3-way pattern (A), despite the lack of disagreement between the two extractors.

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 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 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. For example, let us consider a 3-way pattern $\mathbf{d}_m$, together with a hypothesis H_{mj}, as a sequence of columns as shown in Table 1.

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

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

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

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

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

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. We have generally found this small window size to be fairly effective, requiring the least amount of data to obtain reliable probability estimates.

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, $n = 0,...,3$ (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 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 (Figures 6 and 7).

The second scenario involved more challenging conditions in which the base extractors were not trained using representative data. 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 (Figure 8).

4.1 Results

In the following figures, we have presented our results in terms of *F Measure*, where 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 their combined impact.

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 ≤ 0.002 in

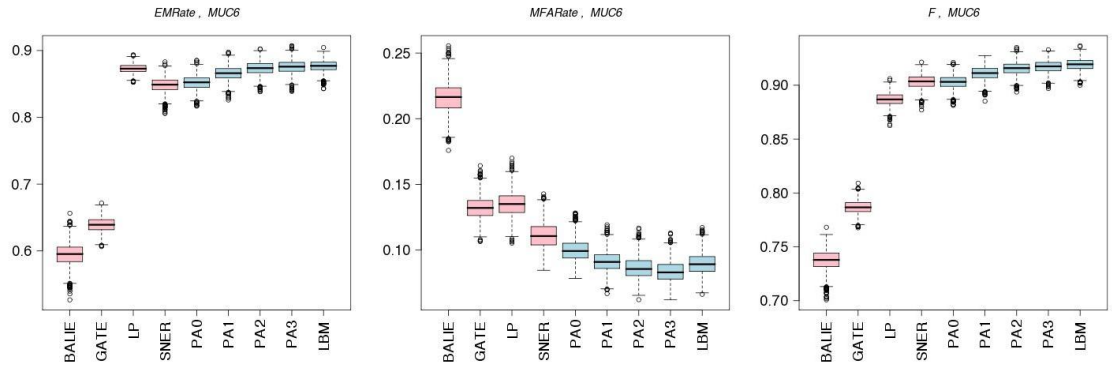

Fig. 6. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC6 for the first experimental scenario. "PA*n*" represents the pattern algorithm, using the simple *k*-way decision stepping down process to step down up to *n* levels. "LBM" presents results from the LBM method, alpha = 0.3. Patterns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

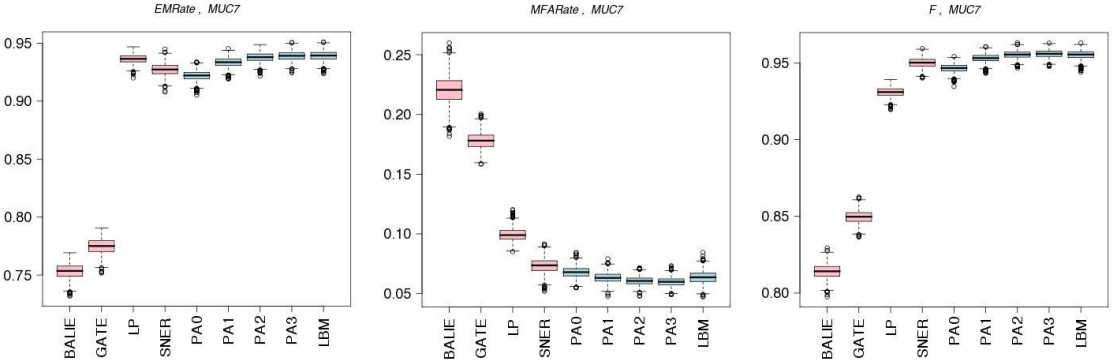

Fig. 7. Left to Right: Exact match rates, Miss + FA rates, F measure on MUC7 for the first experimental scenario. BALIE and GATE performed poorly relative to LP and SNER, much like MUC6. The LBM again uses alpha = 0.3.

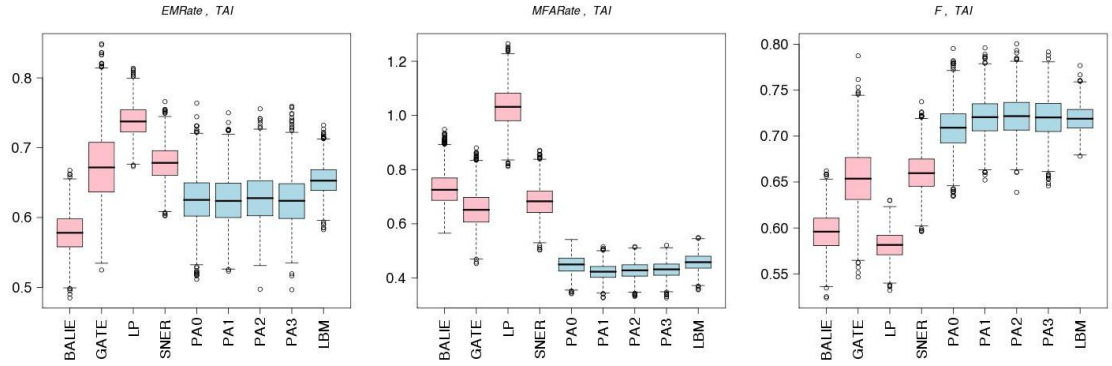

Fig. 8. Left to Right: Exact match rates, Miss + FA rates, F measure on TAI for the second experimental scenario. The performances of BALIE and GATE were more robust relative to LP and SNER. The LBM again uses alpha = 0.3.

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, we have observed that it is sometimes advantageous to interrupt the stepping down process and defer the decision to the sequential method, particularly when the data are sparse.

In the second experimental scenario, we examine results from the TAI data set. 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. As it turns out, this annotation was poorly performed with many underlying true entities unidentified. Hence, this situation mimics those of the second 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.

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. The performance of LingPipe dropped precipitously, largely because its error rate increased by nearly an order of magnitude. Indeed, it produced 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.

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. The simple *k*-way decision, though generally effective, 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. 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.

Both of the methods 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 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 often the most effective.

In text applications, a wide variety of meta-entities is 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 efforts will investigate systems that can assign meta-entities to the most favorable given specific operational conditions and meta-entity features.

Acknowledgements

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

6 References

[1] Kozareva, Z., Ferrández, O., et al. 2007. Combining data-driven systems for improving named entity recognition. *Data & Knowledge Engineering*, 61-3 (Jun. 2007), 449-466. doi:10.1016/j.datak.2006.06.014.

[2] Duong, D., Goertzel, B., et al. 2006. 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. doi:10.1109/IJCNN.2006.246831.

[3] Wang, H. and Zhao, T. 2008. Identifying named entities in biomedical text based on stacked generalization. In *Proc. 7th World Congress on Intelligent Control and Automation* (Chongqing, China, 2008), 160-164. doi:10.1109/WCICA.2008.4592917.

[4] Wu, D., Ngai, G. and Carpuat, M. 2003. A stacked, voted, stacked model for named entity recognition. In *Proc. CoNLL-2003*, 4 (Edmonton, Canada, 2003), 200-203. doi:10.3115/1119176.1119209.

[5] Florian, R. 2002. Named entity recognition as a house of cards: classifier stacking. In *Proc. 6th Conference on Natural Language Learning*, 20 (Taipei, Taiwan, 2002), 1-4. doi:10.3115/1118853.1118863.

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

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

[8] Florian, R., Ittycheriah, A., et al. 2003. Named entity recognition through classifier combination. In *Proc. CoNLL-2003*, 4 (Edmonton, Canada, 2003), 168-171. doi:10.3115/1119176.1119201.

[9] Si, L., Kanungo, T. and Huang, X. 2005. Boosting performance of bio-entity recognition by combining results from multiple systems. In *Proc. 5th International Workshop on Bioinformatics* (Chicago, IL, 2005), 76-83. doi:10.1145/1134030.1134044.

[10] Lemmond, T., et al. 2010. Enhanced Named Entity Extraction via Error-Driven Aggregation. In *Proc. Intl. Conference on Data Mining* (Las Vegas, NV, Jul., 2010), 31-37.

[11] Cunningham, H., Maynard, D., et al. 2002. 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/.