Abstract—Automatic information extraction (IE) has emerged as a critical tool for commercial, industrial, and governmental applications that are confronted with an explosive growth of digital information. Within the framework of information extraction a hierarchy of objectives exists, many of which are heavily dependent upon the automatic recognition of people, places, and organizations—or, more specifically, named entities—in text documents. In this paper, we present a probabilistic approach to aggregating the results of multiple existing entity extraction technologies. The key achievements presented include: (i) the ability to quantify uncertainty in individual extractors and their parameter estimates, and (ii) increased robustness to the over-fitting commonly observed when individual extractors are trained and evaluated on data from different sources. By utilizing Bayesian Model Averaging (BMA), we develop a coherent, data-driven approach for estimating posterior distributions over extracted entities. We demonstrate our approach on several data sets widely used in named entity extraction. The results compare favorably to existing off-the-shelf approaches in traditional settings, as well as in settings where training data are not representative of data encountered under operational conditions.

Keywords: Entity extraction, bayesian model averaging

1. Introduction

The explosion in the number of electronic documents (e.g., news articles, blogs, and emails) brought about by the advent of the internet and related technologies has made the automatic processing of text increasingly critical. In particular, systems that perform knowledge discovery based on information extracted from text are of growing interest to commercial, industrial, and governmental organizations, as they support analysis, decision making, and the development of strategies and policies. Since named entities (e.g., persons, places, and organizations) and their relationships often constitute a significant portion of the information content within source text, named entity extraction (NEE) has emerged as a key component of these systems.

The purpose of NEE is to automatically identify references to real-world named entities within structured or unstructured text documents, often as part of a more extensive information extraction and analysis effort. Success in this task depends upon accuracy in both the segmentation of text into entity and non-entity regions, as well as the classification of entity regions according to a prescribed (and often hierarchical) collection of entity types. NEE has received considerable attention from the natural language processing (NLP) and, more specifically, information extraction (IE) communities, as evidenced by competitive evaluation tasks such as the Message Understanding Conference (MUC) [1] and the Conference on Computational Natural Language Learning (CoNLL) [2]. Numerous algorithms have been proposed for NEE and have been incorporated into knowledge systems in both research and operational settings.

In an effort to improve upon these systems, some researchers have investigated techniques for combining multiple “base” extraction algorithms into an “aggregate” extraction algorithm. These include methods such as voting [3], [4], stacking [5], or using classifiers for combination [6]. Results from these efforts have demonstrated that further gains can indeed be obtained by leveraging the respective strengths of different extractors.

In this paper, we introduce an aggregation technique based on the principle of Bayesian Model Averaging (BMA). Using the framework discussed in [7], our BMA-based approach estimates a posterior probability distribution over ground-truth hypotheses (i.e. possible segment label assignments) for a “meta-entity”—a region of text defined by the union over individual extractor entity segmentations. This is accomplished as follows: 1) a meta-entity is constructed from the joint output of the constituent base extractors; 2) a “hypothesis space” consisting of possible label assignments to the meta-entity segments is formed; 3) each extraction (i.e. base or aggregate) algorithm produces a distribution over the hypothesis space; and, finally 4) BMA is used to combine the hypothesis probability estimates produced by each of the algorithms based on the respective model posteriors.

This methodology aims to improve on existing extraction techniques in two respects: 1) reducing the variability in performance by accounting for uncertainty associated with individual model estimates, and 2) increasing robustness to the over-fitting frequently associated with training and evaluating on data from different sources. Moreover, unlike many existing aggregation methods, this approach produces a true posterior distribution over possible “hypotheses”, thereby enabling the confidence in the extracted data to be quantified.

To present a comprehensive background, Sections 2 and 3 provide a description of the major categories of entity extraction algorithms and common combination techniques, respectively. Section 4 describes the BMA approach and its application to NEE, followed by a discussion of model...
estimation and implementation in Section 5. Experimental results are presented and discussed in Section 6, with conclusions and future directions following in Section 7.

2. Entity Extraction Algorithms

Although the substantial investments made by the NLP and IE communities in NEE have generated numerous approaches for solving this problem, these diverse methods can be roughly grouped into a few major categories. These categories include rule-based approaches as well as supervised, semi-supervised, and unsupervised learning methods. In this section, we provide a brief overview of their respective characteristics.

In a rule-based NEE system, entities are identified via a set of rules typically triggered by lexical, syntactical and grammatical cues. These rules are often hand-crafted using linguistic or corpus-based knowledge, and the triggering process is modeled as a finite-state transducer. A simple example of this approach is template matching via regular expressions. While such an approach can be effective and robust to shifting operational conditions, in cases where sufficient representative data exist, rule-based systems are typically outperformed by statistical learning approaches.

Supervised learning—the current state-of-the-art paradigm for NEE—utilizes features derived from text to infer decision rules that attempt to correctly identify and classify entities. Positive and negative examples of entities used to train the algorithm are obtained from a large collection of manually annotated documents. The particular learning algorithm employed varies based upon application-specific limitations and/or specifications, but the most widely accepted include support vector machines (SVMs), decision trees (DTs), hidden Markov models (HMMs), maximum entropy models (MEMs), and conditional random fields (CRFs).

While supervised learning methodologies generally perform quite well in an ideal operating environment (i.e., having plentiful representative data for training), they tend to be highly vulnerable to evolving or sparse data conditions. Semi-supervised (or “weakly supervised”) and unsupervised methods attempt to address these issues by circumventing the need for extensive manual annotation.

Specifically, semi-supervised learning is generally an iterative procedure in which a small number of labeled “seed” examples are used to initiate the learning process. The algorithm subsequently generates new training examples by applying the learning from the previous step to unannotated data. The process is repeated until no new examples are generated. One typical approach involves identifying contextual clues from the seed examples and attempting to find new examples that appear in similar contexts. New context information and additional examples are then obtained in an iterative fashion.

Unsupervised learning algorithms, on the other hand, require no annotated data for training. Generally, they rely on clustering methods to group named entities based upon similarity of context. Alternative approaches rely on external lexical resources, lexical patterns, and on statistics computed over a large unannotated corpus.

3. Combination techniques

With the variety of extraction algorithms available, a natural extension to traditional NEE approaches is to combine these algorithms—and, consequently, their underlying models—in an attempt to achieve improved performance. The expectation is that these algorithms will collectively use rich and diverse feature representations and will possess complementary characteristics that can be leveraged to enhance positive attributes (e.g., low false alarm or miss rates) while mitigating their individual weaknesses. The most straightforward and intuitive of such approaches utilizes a voting mechanism. Voting techniques examine the outputs of the various models and select the classification with a weight exceeding some threshold. Variations in the voting mechanism employed typically differ in regard to their weighting scheme for individual models. Example voting methods include at-least-N “minority” voting [4] and weighted voting via SVMs [3].

A more sophisticated combination scheme discussed in [?] interpolates a word-conditional class probability distribution across the base extractors $BE_i^n = BE_1, BE_2, \ldots, BE_n$, where the class, $C$, corresponds to a word’s position relative to a named entity (start/within/end/outside). This distribution, $P(C|w; BE_i^n)$, is interpolated using weights estimated from training data.

One limitation common to many of these methods is their failure to account for the local context of a word or entity of interest. A CRF model, as proposed by [6], addresses this shortcoming and was shown to yield enhanced performance.

An alternative to the parallel combination techniques described above is the serial process of stacking [5]. In stacking, two or more classifiers are trained in sequence such that each successive classifier incorporates the results of those preceding it. Of course, the above combination approaches can themselves be combined to produce a new methodology, as demonstrated in [8].

Recently, [Lemmond11] proposed a new parallel combination technique based on a “pattern” representation of base extractor output. Specifically, this pattern-based meta-extractor (PME) utilizes a pattern that encodes the joint characteristics of the combined extractor output, $D$, and (implicitly) their associated errors. The union of overlapping base extractor output regions—the “meta-entity”—provides the textual extent over which a pattern is encoded.

By observing the frequency of these patterns jointly with similar encodings of ground-truth labels for an annotated “evaluation” set, we can compute an estimate of the probability of a hypothesized ground-truth, $h$, given an observed joint
extractor output \(d\). We then select the hypothesis according to
\[
h' = \arg\max_{h \in \Omega} p(h|d, \tilde{h}, \tilde{d})
\]  
(1)

where \(\Omega\) is the set of possible hypotheses for a given meta-entity \(p(h|d, \tilde{h}, \tilde{d})\) is the estimated probability of hypothesis \(h\) given an observed output \(d\) and the evaluation set \((\tilde{h}, \tilde{d})\).

One notable property of the PME methodology is that it models the joint characteristics of base extractors and the errors they are likely to produce without knowledge of the underlying algorithms or their individual error processes. As such, each base extractor can be regarded as a “black box” whose output alone is necessary for aggregation. This distinctive characteristic of the PME enables it to address practical issues such as language independence and proprietary restrictions of base extractors. Another notable property is that this method yields a probability estimate for each possible ground-truth hypothesis, facilitating the use of BMA, which is discussed in Section 4.

4. Bayesian Model Averaging

Bayesian Model Averaging (BMA) is a statistical technique designed to account for the uncertainty inherent in the model selection process [9]. This is sharply contrasted with the typical statistical approach in which a single model is selected from a class of models, and fitting proceeds as if this model had generated the data at hand. In NEE, it is common for a single extraction algorithm to be selected a priori and its parameters estimated, or for a collection of algorithms to be combined according to a single aggregation rule. Consequently, NEE represents an appropriate domain for the application of this model averaging technique.

BMA is used to estimate a posterior probability distribution, \(\pi\), over the value of interest, \(\Delta\), given the available data, \(D\), by integrating over a class of models, \(M\), and the model parameters. This can be expressed as
\[
\pi(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)\pi(\Delta|M, D)
\]

where \(P(M|D)\) is the model posterior and \(\pi(\Delta|M, D)\) is the posterior distribution of the value of interest produced by the model \(M\). Thus, BMA provides a principled mechanism for combining the posterior distributions produced by the individual models by weighting each model in proportion to its posterior probability. Using Bayes’ rule, this model posterior can be calculated as
\[
P(M|D) = \frac{P(M)P(D|M)}{\sum_{M \in \mathcal{M}} P(M)P(D|M)}.
\]  
(2)

Furthermore, the posterior expectation and variance can be computed as a function of the individual model estimates of the respective quantities. Specifically,
\[
E(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)E(\Delta|M, D)
\]

and
\[
\text{var}(\Delta|D) = \sum_{M \in \mathcal{M}} P(M|D)(\text{var}(\Delta|M, D) + E(\Delta|M, D)^2) - E(\Delta|D)^2.
\]

5. Models, Estimation, and Implementation

As previously mentioned, the general NEE task consists of both the segmentation of text into entity and non-entity regions and the classification of entity regions according to entity type. Within the meta-entity framework, however, this task reduces to a modified classification problem. More formally, the classification consists of identifying the correct hypothesis \(h'\) from the set of possible hypotheses \(h \in \Omega\) given the observed output \(d\) and the training data \((\tilde{h}, \tilde{d})\). In our case, we use a maximum a posteriori (MAP) decision rule:
\[
h' = \arg\max_{h \in \Omega} p(h|d, \tilde{h}, \tilde{d}).
\]

This hypothesis probability estimate is model-dependent. To address the uncertainty inherent in model selection, we can reformulate the estimate within the context of model averaging as
\[
p(h|d, \tilde{h}, \tilde{d}) = \sum_{M \in \mathcal{M}} P(M|h, \tilde{h}, \tilde{d})p(h|d, \tilde{h}, \tilde{d}, M) \quad (3)
\]

where the model posterior does not depend upon the newly observed output—i.e., \(P(M|h, \tilde{h}, \tilde{d}) = P(M|d, \tilde{h}, \tilde{d})\)—and the posterior distribution of \(h\) produced by algorithm \(M\) is weighted based on the training data.

Aggregating the output of base and/or aggregate extraction algorithms via BMA requires that we specify a model to describe the relationship between extractor output and the underlying truth entity. We begin by assuming that ground truth is generated by the extractor output (by a fixed conditional distribution) where meta-entities are exchangeable within the corpus. This assumption allows a “bag-of-meta-entities” approach similar to the bag-of-words approach of [10] to be employed, with the distinction that, a bag is formed with respect to the corpus rather than an individual document.

First, consider the ground truth \(h_i\) and extractor output \(d_i\) associated with the \(i\)-th of \(n\) meta-entities extracted, and denote the evaluation set as \(\tilde{h} = (h_1, \ldots, h_n)\) and \(\tilde{d} = (d_1, \ldots, d_n)\). A generative process producing \(h_i\) under model \(M\) is given by
\[
l_i|M \sim \text{Poisson}(\gamma_M)\]
\[
d_i|M, l_i \sim \text{Multinomial}(\beta_{Ml_i})
\]
where $l_i$ is the length of the $i$-th meta-entity. That is, a new pair $h_i, d_i$ can be generated by (1) drawing the meta-entity length $l_i$ from a Poisson distribution, (2) drawing the joint extractor output $d_i$ from a multinomial over all joint outputs of a given length, and finally (3) drawing the ground truth $h_i$ from a multinomial conditioned upon $d_i$. The dimension of the multinomial distribution over ground truth—and, consequently, the parameter vector $\theta$—depends upon $d_i$—specifically, length $l_i$ of the meta-entity determined by $d_i$. The number of possible representations of the truth under the well-known BIO (begin/inside/outside) model is equal to all sequences of B-I-O where an O can not immediately precede an I. The rate of growth can therefore be described by the recursive formula $a_l = 3a_{l-1} - a_{l-2}$ based upon [11], where $l$ is the length of the meta-entity and $(a_0, a_1) = (1, 2)$.

The overall likelihood for the data $(\hat{h}, \hat{d})$ under model $M$ can be computed according to

$$p(\hat{h}, \hat{d}|\theta, \beta, \gamma) = \prod_{i=1}^{n} p(h_i|\theta_{Md_i}) p(d_i|\beta_{Md_i}) p(l_i|\gamma_M)$$

(4)

where $p(d_i|M, l_i)$ and $p(l_i|M)$ can either be modeled or taken as exogenous in which case they do not contribute to the likelihood. Ultimately, a total of $\sum_{M \in \mathcal{M}} |\mathcal{D}_M|$ multinomial models for $h'd$ must be estimated, where $\mathcal{D}_M$ is the collection of multinomials whose size grows at a rate of $a_l^b$ with $b$ representing the number of constituent extractors whose output is modeled. In practice, meta-entities of length greater than 5 are rarely observed limiting the actual number of fitted models.

Traditionally, there are two primary challenges encountered when implementing model averaging: (1) summing over the possibly large class of models, $\mathcal{M}$; and (2) computing the model likelihood, $P(D|M)$, which involves integrating over all possible model parameter values. In the case of extraction algorithms, however, we only address the latter, as the classes of models considered are small and efficient enough to be readily enumerated and evaluated.

The model likelihood is determined by integrating over all possible parameter values by

$$P(\hat{h}, \hat{d}; M) = \int p(\hat{h}, \hat{d}; M, \theta, \beta, \gamma) P(\theta, \beta, \gamma|M) d\theta d\beta d\gamma.$$

(5)

Rather than attempt to evaluate this integral directly, we approximate it by evaluating the likelihood given a point estimate in place of the integral—not an uncommon practice [12]. For example, when $h_i$ is taken as the sole random component then $P(\hat{h}, \hat{d}; M) \approx P(\hat{h}, \hat{d}; M, \hat{\theta})$. One complication of this approach is the potentially varying amount of evaluation data available for estimating the different multinomial models. A simple model likelihood (or log-likelihood) calculation would have the undesirable effect of penalizing models with more evaluation data. Additionally, the exponential dependence of the likelihood on the proportion of correctly classified samples potentially places almost all of the probability mass on a single model [12]. To address this issue, we choose, instead, to compute the mean log-likelihood of the model.

The practical issues of BMA for NEE are not limited to those mentioned above. Additional considerations include parameter estimation, the form of the output of the extraction algorithms, the class of models, and the model priors. These are discussed below.

### 5.1 Parameter Estimation

Recall from Section 5 that, under the meta-entity framework, a model $M$ consists of a set of multinomial models $\mathcal{D}_M$, each of which has a set of parameters $\theta$ that must be estimated. A reasonable approach is to perform maximum likelihood parameter estimation, but difficulties arise when faced with potentially small amounts of training data. To address this, we employ a Bayesian estimate using a non-informative Dirichlet prior $D(\alpha, \ldots, \alpha)$. Using the posterior expectation as the parameter estimate yields

$$\hat{\theta}_{Mdh} = \frac{n_{Mdh} + \alpha}{n_{Md} + aL\alpha}$$

where $n_{Mdh}$ denotes the number of training examples of under model $M$, extractor output $d$, and ground-truth hypothesis $h$, and $n_{Md} = \sum_{h \in \Omega} n_{Mdh}$. The estimates of $\beta$ are similarly obtained.

### 5.2 Classification

Frequently, the task of classification is separated into two paradigms: 1) Hard classification; and 2) Soft classification. Here we focus on hard classification. Referring to equations 2 and 3, there are two places which require attention in the implementation for BMA: 1) the computation of the model posteriors via model likelihood $P(\hat{h}, \hat{d}; M)$; and 2) the posterior predictive distribution $p(h|d, \hat{h}, \hat{d}; M)$. The latter is easily resolved, as under a hard classification $p(h = h'|d, \hat{h}, \hat{d}; M) = 1$ for the assigned class $h'$ and 0 for all others.

The computation of likelihood $P(\hat{h}|\hat{d}, M, \hat{\theta})$ is almost as easily handled. The likelihood is computed by taking a product of the probability that each training observation is classified correctly. That is,

$$P(\hat{h}|\hat{d}; M, \hat{\theta}) = \prod_{h \in \Omega} \theta_{Mdh} \frac{n_{Mdh}}{n_{Md} - n_{Mdh}},$$

where $\theta_{Mdh} = 1 - \epsilon$ and $\epsilon$ is an error rate associated with the algorithm [13].

### 5.3 Model Priors

Two types of priors figure into the BMA framework: 1) $p(\theta|M)$, a prior on the parameters given the model; and 2) $p(M)$, a prior distribution over the possible models.
Although non-informative priors are typically desirable for the parameters of a given model, these distributions have been shown to be somewhat less effective when specified over a class of models [14]. As noted in Section 5.1, we use a Dirichlet prior for the multinomial distribution parameters. With regard to the prior distribution over models, we consider several options, discussed below.

1) Uniform A uniform prior, \( P(M) = 1/|\mathcal{M}| \), over the class of models results in a probability distribution which tends to place more weight on simple models. This is a result of the composition of the model classes and the fact the joint output of more complex models has a higher dimensionality decreasing likelihood.

2) Complexity based A prior that places proportionally more weight on the more complicated models can be used to produce a model posterior that more evenly distributes probability over the class of models. In our case, if the joint output space of \( k \) extractors grows at a rate of \( a^k \) then consider \( p \propto a^k \).

3) Exact Match Rate An empirical or subjective prior based on the overall performance of a given model can also be used. One reasonable option is

\[
P(M) \propto E_M
\]

where \( E_M \) is the exact match rate, or frequency which the extractor output is identical to the ground truth, associated with model \( M \).

5.4 Model Classes

The class of models \( \mathcal{M} \) may be formed in a number of ways, although some of the most interesting focus on addressing a paucity of training data. Typically more complicated aggregation models that account for joint behavior of the constituent extractors require estimating many parameters leading to less reliable estimates than those obtained under simpler frameworks.

1) Off-the-shelf algorithms The output of any collection of existing entity-extraction algorithms can be easily handled within the model averaging framework. First, a meta-entity is constructed relative to the joint output of the collection. Second, the output of each algorithm is recorded relative to the joint and the error probabilities are calculated. Finally, a prediction is made on the newly extracted data by evaluating the model posteriors relative to the joint output.

2) Pattern and likelihood algorithms The pattern algorithm and likelihood algorithms developed by [7] both use the meta-entity construct and are thus naturally suited to determine the class of models. The performance of these algorithms can vary substantially based upon characteristics of the joint extraction. For example, under the pattern algorithm there are fewer training examples for longer joint outputs, resulting in parameter estimates with higher variability. Within the pattern algorithm framework these problems can be addressed by considering subsets of extractors, or by making independence assumptions. By considering various subsets of extractors a model posterior probability can be calculated which reflects the relative confidence in a specific subset, and similarly for the independence-based approach.

3) Unions In general, any combination of model classes can be combined for BMA, provided that the constituent outputs are represented under the meta-entity framework, thereby transforming the problem into one of classification.

6. Experiments

We investigate the performance of BMA from several directions. First, we examine the impact of the tuning parameters and model classes. Second, the performance is compared with other state-of-the-art extraction technologies in a number of different operational settings.

Each experiment was performed utilizing annotated data sets. These datasets include those widely used by the NEE community MUC6, MUC7, and CONLL which are comprised of 7617, 11969, and 26872 entities respectively. It should be noted that although these data sets were manually annotated using a common set of guidelines inter-annotator disagreement produced F-measures of 0.96 to 0.97.

![Fig. 1: A comparison of complexity based prior weighting across independence (left) and subset (right) based model classes. The best performance was attained by balancing the prior weights over the range of models. The independence based model class performance was less dependent upon the choice of prior weights, while the the subset class produced the highest f-measure over all priors.](image)
to the constituent base extraction algorithms. One of two paradigms was used depending upon the nature of the underlying algorithm: 1) Base extraction algorithms whose output was used directly were fit using the standard paradigm; and 2) Aggregation-based algorithms employed a training-training-test split of 45-45-10.

### 6.1 Model prior and class comparisons

As mentioned in Section 5, the choice of prior distribution and the set of model classes are two practical issues of the BMA approach. Here we explore the implications of these choices with regard to performance in the NEE task. Three model priors and three model classes were analyzed. The model priors were 1) Uniform; 2) Complexity-based; and 3) Empirical based on exact match rate. The model classes included 1) Off-the-shelf (or “base” extractors); 2) Subsets of base extractors; and 3) Different base extractor independence assumptions.

A first step in understanding the behavior of this approach is to study the sensitivity of model averaging to the prior distributions over model class and model parameters. We begin by examining these relationships and inferring the optimal choices for future predictions by training our model on MUC6 and evaluating its performance on MUC7. Figure 1 plots F-measure on the y-axis versus the mean complexity of a given combination of extractors on the x-axis where the mean complexity is computed as \( \sum_M P(M)c_M/|M| \) with \( c_M \) denoting the number of base extractors jointly modeled. Instead of exploring all possible forms of \( P(M) \) only unimodal functions were considered on the premise that in general either the more complicated or simpler models need to be weighted more heavily. Each line in the two panels represents a combination of model prior \( \alpha \) and a model class, independence or subset.

The plots in figure 1 show that the optimal weighting scheme emphasizes the importance of more complicated joint models. When the individual model posterior expectations were relatively even, more complicated models dominated, deferring to the less complicated extractors when insufficient training data were available or the complicated models were poor predictors. Interestingly, the choice of \( \alpha \) for the non-informative Dirichlet prior had a very limited effect on f-measure while a highest f-measure was attained at \( \alpha = 1.5 \), decreasing the alpha to .1 or .01 did correspond to a pronounced decrease in performance. This result may stem from the importance placed on accurate estimation of the most likely hypothesis by the MAP decision rule, i.e. accurate estimation of the probability of the most likely hypothesis is aided by larger values of \( \alpha \) although this potentially causes the probabilities of the less likely hypotheses to be overestimated which is of no consequence under this paradigm.

Figure 2 presents boxplots of performance of the various model class and prior combinations as determined by F-score. The boxplots were generated using 1000 bootstrap samples of the weighted F-score from the 10 cross-validation folds. Weighting was determined by the number of ground-truth entities within each fold. Figure 2(a) shows results in which the training and test data originated from the MUC6 data set (the “matched” data condition) and Figure (b) shows results in which training data originated from MUC6 and test data from CoNLL (the “mismatched” data condition).

The first noteworthy result is that a substantial performance difference exists between the base extractor model class and those of extractor subsets and extractor independence. Between the subsets and independence, no notable difference exists. The base extractor models fail to leverage joint information from any of the extractors, and poorer performance results. Also of note, is the performance difference between the matched and mismatched conditions. A moderate degradation is observed when testing on a data set differing from the training data set, though the pairing of the subsets model class and the complexity-based prior seems particularly robust. Lastly, we see that, regarding the model...
prior, the complexity-based prior appears to perform as well or better than the other two across both model classes and evaluation conditions.

6.2 Alternate algorithm comparisons

In addition to analyzing differences among the various extraction systems within the BMA framework, analyzing BMA relative to alternate extraction algorithms is naturally of interest. Here we compare a single BMA system—the subsets model class with the complexity-based prior—to a majority rule approach and the basic pattern meta-entity extractor (PME) system. Figure 3 presents these results in the same fashion as above.

For the matched data condition, the performance of majority rule, PME, and BMA are rather similar. Majority rule does appear to lag behind PME and BMA performance is slightly higher than PME, but the difference across systems is small. This is in contrast to the mismatched condition, in which majority rule performance degrades dramatically, but that of PME and BMA do so only moderately.

7. Conclusions

Utilizing bayesian model averaging, we have demonstrated an approach to entity extraction which is capable of: (i) reducing the variability in performance by accounting for uncertainty associated with individual models, and (ii) increasing robustness to over-fitting associated with training on a single corpus. In practice, developing priors based on the complexity of the constituent models produced the best results in terms of F-measure. We observed that while model class and selection of a prior are separate components of the process they should be considered simultaneously. Additionally, this approach can be applied to a wide variety extractors and aggregation algorithms as they are all treated as "black boxes".

8. Acknowledgments

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