Extracting laboratory test information from biomedical text

Yanna Shen Kang, Mehmet Kayaalp

Center for Devices and Radiological Health, U.S. Food and Drug Administration, Silver Spring, 1The Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institutes of Health, Bethesda, Maryland, USA

E-mail: *Yanna Shen Kang ‑ yanna.kang@fda.hhs.gov
*Corresponding author

Received: 05 April 13  Accepted: 03 July 2013  Published: 31 August 13

Abstract

Background: No previous study reported the efficacy of current natural language processing (NLP) methods for extracting laboratory test information from narrative documents. This study investigates the pathology informatics question of how accurately such information can be extracted from text with the current tools and techniques, especially machine learning and symbolic NLP methods. The study data came from a text corpus maintained by the U.S. Food and Drug Administration, containing a rich set of information on laboratory tests and test devices. Methods: The authors developed a symbolic information extraction (SIE) system to extract device and test specific information about four types of laboratory test entities: Specimens, analytes, units of measures and detection limits. They compared the performance of SIE and three prominent machine learning based NLP systems, LingPipe, GATE and BANNER, each implementing a distinct supervised machine learning method, hidden Markov models, support vector machines and conditional random fields, respectively. Results: Machine learning systems recognized laboratory test entities with moderately high recall, but low precision rates. Their recall rates were relatively higher when the number of distinct entity values (e.g., the spectrum of specimens) was very limited or when lexical morphology of the entity was distinctive (as in units of measures), yet SIE outperformed them with statistically significant margins on extracting specimen, analyte and detection limit information in both precision and F-measure. Its high recall performance was statistically significant on analyte information extraction. Conclusions: Despite its shortcomings against machine learning methods, a well-tailored symbolic system may better discern relevancy among a pile of information of the same type and may outperform a machine learning system by tapping into lexically non-local contextual information such as the document structure.

Key words: Biomedical information extraction, extraction of laboratory test information, information extraction, machine learning, symbolic named-entity recognition

INTRODUCTION

With the increasing volume of online biomedical research publications and electronic medical records, reading and manually processing each and every text document for research is no longer feasible. In the era of big data, algorithmically extracting necessary information from large corpora of scientific articles and clinical reports...
becomes a sine qua non for coping with the exponential growth of scientific knowledge. To that end, we need intelligent and effective text mining techniques to extract relevant information from unstructured data and make them available to health-care researchers. In most biomedical studies, the data comprise measurements of analytes in biological specimens; hence, the algorithms of interest should be capable of extracting not only analyte and specimen information, but also the pertinent numeric quantities of analytes along with their units of measures. The question is how well can we do that using the modern tools and techniques?

Biomedical information extraction (BMIE) has captured the interests of researchers in recent years. The literature on BMIE has been primarily focused on the detection of biomedical topics[1-3] and extraction of gene and protein information from text.[4,6] Although the volume of publications on disease and chemical compound detection in unstructured text has been steadily growing,[7] there are very few studies published on detecting diagnostic laboratory information and related biomedical named entities.

This study tackles this neglected area of BMIE research and attempts to gauge how well existing methods perform to extract diagnostic laboratory information from narrative text. More specifically, the focus of the study is how to distinguish a particular specimen or a particular measurement value of interest (e.g., 2.5 mg/dL) among a number of other, less relevant specimens and measurement values that are also discussed in the same text. The existing machine learning methods, which are quite successful in detecting well-defined classes of named entities such as units of measures (e.g., kg, mg, dl., mg/dL), are not as capable of deciding on which numeric measure is of interest. This study demonstrates that symbolic approaches may have an edge over learning methods in distinguishing relevancy among named entity values (e.g., a particular analyte detection limit of a diagnostic lab device), which requires differentiation of entities and their values at a very fine level of granularity.

Although the majority of terms used in our symbolic methods are domain specific, these methods are in general, readily transferable to different named entity recognition and information extraction problems in other domains.

BACKGROUND

Text mining is a major research area of biomedical informatics.[8] We need reliable methods for extracting scientific facts and experimental results from biomedical literature as well as extracting clinical information from clinical narrative reports. Current methods and tools have been tested on various biomedical entities.[1-7] Even though laboratory test information are crucial components of any clinical narrative, the authors could not find in the literature a single pathology informatics study on detecting and extracting laboratory test information from narrative text.

Data

The study data came from a document corpus maintained by the U.S. Food and Drug Administration (FDA), containing a rich set of information on laboratory tests and test devices. Section 510(K) of the Food, Drug and Cosmetic Act[9] requires medical device manufacturers to notify the U.S. FDA of their intent to market a medical device at least 90 days in advance. In general, every premarket notification (PMN) needs to demonstrate that the device to be marketed is substantially equivalent to (i.e., at least as safe and effective as) a legally marketed device (referred to as the predicate).[10] For each in-vitro diagnostic device, FDA reviews its application and generates a decision summary, determining whether the proposed device is substantially equivalent to the predicate device. FDA has made these decision summaries publically available in a downloadable PDF format. We obtained these decision summaries by querying the 510(K) PMN database[11] using 510(K) numbers. Decision summaries provides this study a rich and valuable set of laboratory test information, such as specimens and analytes that were used in and measured by the corresponding medical devices.

Laboratory test information can be found in a wide range of biomedical articles – from scientific text describing results of clinical trials to clinical reports and from drug inserts to consumer health news. Mining any of these corpora would provide a different set of facts to build our knowledge base. Although the corpus of FDA decision summaries is not identical to any other information sources, the problems of laboratory information extraction from this corpus and others are expected to be similar. Given FDA decision summaries are free from copyright and protected health information, yet abundant with valuable laboratory test information, they comprise an almost ideal text corpus for this study.

Once laboratory information is extracted, they can be utilized in various ways depending upon the information source. For example, it can be fed into Logical Observation Identifiers Names and Codes (LOINC),[12] which is a universal code system for identifying laboratory and clinical observations. Medical device manufacturers can benefit from such structured information. If the information source is about a clinical trial, the information can be input into the clinical trials registry.[13] If the source is a journal article, the information can be used to enrich the corresponding PubMed/MEDLINE[14] entry as well as various biomedical databases.[15]

Natural Language Processing Methods

Information extraction methods can be viewed in
at least two distinct sets of approaches: Symbolic NLP and (machine) learning approaches. Hybrid approaches combining symbolic and learning approaches have also been used. An NLP system is qualified as a symbolic system when its inference is based mainly on linguistic (syntactic and semantic) information. In this study, we developed a symbolic information extraction (SIE) method to detect four types of diagnostic laboratory entities (specimens, analytes, units of measures and detection limits) and extract their values from unstructured text. We evaluated SIE along with three types of learning methods, namely conditional random fields (CRFs), hidden Markov models (HMMs) and support vector machines (SVMs) and compared their performances on recall, precision and F-measure.

**Symbolic Approaches**
Symbolic approaches use manually constructed linguistic rules and regular expressions to detect entities and to extract their values from text. Symbolic methods, which usually are domain specific, enable experts to code their domain knowledge into algorithms; hence, the inference of entities of interest is driven by knowledge (i.e., not by data). Well-designed symbolic methods may perform with high precision and low error rates. They may be capable of detecting lexically complex entities (e.g., those represented in long noun phrases), which are challenging for current learning models in general. On the other hand, they may also be brittle due to narrowness of represented knowledge and their performance degradation may be drastic when the domain characteristics change slightly.

Some symbolic systems were designed to extract information about people, organizations and locations, while others to extract biomedical entities such as genes, proteins or organisms. For example, Crowley et al., developed a Cancer Tissue Information Extraction system that performed concept coding of cancer tissue specimens using regular expression matching, gazetteer lookup, concept filters and other symbolic methods. Fukuda et al., constructed rules for recognizing proteins, genes and chemical compounds in the text using linguistic constraints as well as lexical and contextual information. Natayanaswamy et al., developed a BMIE system that detects protein names based on surface information on character strings without requiring any specific term dictionary.

**Machine Learning Approaches**
Machine learning approaches contain rule learning and model learning methods. Rule learning methods inductively construct a set of classification rules that maximize a target performance metric based on correctly labeled entities in a training dataset. Model learning on the other hand, involves learning parameters of a predefined probabilistic structure such as HMM and CRF, and in rare occasions, it may also involve learning model structures in tandem. Probabilistic learning methods have been widely used in BMIE, especially since the announcement of GENIA corpus. To the best of our knowledge, there is no existing probabilistic learning system that is capable of distinguishing relevancy among various values of the same biomedical entity; e.g., distinguishing the 2-mg/dL detection limit of a diagnostic laboratory device for a particular analyte from ten other impertinent measurements between 0.5 and 20 mg/dL.

HMMs have often been used for sequential data labeling such as named entity recognition and information extraction in natural language text. In the token-based NLP tasks, an HMM maximizes the joint probability \( p(x, y) \), where \( x \) and \( y \) are sequences of input tokens (and token features) and the output labels, respectively. Token features typically include orthographical and morphological characteristics. Standard HMMs assume that input features are conditionally independent given their associated label. However, the issue related to this assumption is one of the most fundamental problems in sequential data labeling task such as the BMIE task described in this paper.

LingPipe is a platform for various text processing tasks, including named entity recognition, word sense disambiguation, spelling correction and topic classification. LingPipe’s trainable probabilistic named entity recognizer is based on a first-order HMM and can be used to detect news entities (e.g., people, locations and organizations) or biomedical entities (e.g., genes, organisms, malignancies and chemicals). CRFs are another type of graphical models, which are often used for labeling or parsing natural language text or biological sequences. In contrast to HMMs, CRF models are trained to maximize the conditional probability \( p(y|x) \), where \( x \) and \( y \) are sequences of input tokens along with token features and the output labels, respectively. This conditional nature relaxes the independence assumption required by HMMs while provides tractability for inference.

**BANNER** is an open-source biomedical named entity recognition system. It incorporates a number of token features including lemmatization, part-of-speech tags, numeric normalization and shallow syntax features. For sequential labeling, BANNER employs Mallet version 0.4, which implements a second-order CRF. Another learning algorithm that is frequently used in natural language text labeling is SVMs. Like CRF models, SVMs may incorporate dependent input features. We used an SVM model implemented in GATE, which is an NLP framework for software development. This particular implementation of SVM introduces the
so-called uneven margins parameter to handle unbalanced training sets like ours, where only a small number of mentions of the laboratory test entities of interest exist in the text.

In contrast to the symbolic knowledge-based methods, where knowledge is encoded manually, machine learning methods are data-driven, where knowledge is implicitly encoded in numeric parameters. Depending on the size and quality of the training corpus, algorithmic knowledge acquisition processes may enable machine learning systems to avert the brittleness problem, which generally afflicts symbolic systems due to their narrowly focused knowledge. Machine learning methods are effective in detecting entities as long as their features (entities of interest or lexical pointers to them) come from a limited set of lexical terms represented fully in the training dataset.

Training Corpus
The size and quality of the training (data) are important criteria that affect the performance of a machine learning system. The performance of a learning system usually improves asymptotically as the training size increases. It is difficult to determine a priori what the optimum training size would be because it depends on a number of factors such as the number of features that the learning system uses, the prevalence and distributions of the classes to be predicted in the population and the complexity of the problem. In reality, the training size is determined usually by the constraints of the real world – the availability of the data and the cost of annotation that the organization could afford.

Testing the optimum training size is usually a separate research question by itself and is rarely performed when different learning systems are compared; however, Tawari[49] studied the effect of various training sizes on an information extraction problem that was very similar to ours. He extracted from web pages three different types of information, product names, product prices and product images, using a CRF and two different SVM methods. The training sizes were varied between 40 and 130 cases. In all experiments, the F-measures had improved slightly by 5.25 ± 3.25% as the training size increased from 40 to 130. Given the amount of variance overlaps between the results of the smallest and largest training datasets, it is likely that the performance improvements were not statistically significant.

METHODS

We constructed a number of rules to detect the named entities of interest from unstructured text. In the first phase, they yielded a large set of potential named entities (candidates), which we later pruned into a smaller set using a likelihood ranking metric.

Document Preprocessing
FDa decision summaries usually contain a document title and multiple sections dedicated to the intended use and descriptions of technological characteristics of the device. This observation can easily be translated into a rule of symbolic methods, but it is significantly harder for a generic machine learning system to identify such relations. Our system associates each token with sentence and document section information and uses them in inference. To utilize such document structure information, we process the document through a pipeline of preprocessors: Document section detection, sentence boundary detection and sentence tokenization (into “words”).

We performed document structure analysis to detect document sections. Recall that we need to convert the PDF documents downloaded from the FDA website to plain text documents. Unfortunately, surface characteristics of section headers such as increased font size, boldness and italicization were lost during the conversion from PDF format to plain text. As most section headers are labeled using all uppercase or first-letter uppercase letters and enumerated with Arabic or Roman numerals, we used regular expressions to match section headers and their numbers. We also used rules such as the maximum number of tokens allowed in a header and the punctuation-character rule to differentiate a section header from regular sentences in the text body. Subsections were detected analogously within each section of the document.

We used regular expressions to detect sentence[49] and token boundaries. At the end of the preprocessing step, we attained a hierarchical document structure of sections, sentences and token sequences.

Extracting Diagnostic Lab Data
We extracted information for four types of named entities: Analytes, specimens, units of measures and detection limits [Table 1]. Detection limit defines the sensitivity of the diagnostic device for a particular analyte measured in a particular specimen. It is expressed as a number associated with a unit of measure. Obviously, detecting the proper unit of measure is a prerequisite for extracting detection limits. Analytes and specimens are closely related as well. Due to these dependencies, we searched these four entities in the following order: (1) Analyte names, (2) specimen names, (3) units of measures and (4) detection limits.

Filters and Trigger Events
The types of both specimens and units of measures are finite. It is unlikely that a new specimen type or a new unit of measure would be introduced in a decision summary. We built filters for our system based on our background knowledge along with external knowledge sources. Using a large collection of de-identified clinical laboratory
test records, we compiled a list of analytes and a list of units of measures. We also extracted a list of specimen names and their variants from LOINC. Although not all-encompassing, these two lists greatly helped us recognize required information and filter out the noise.

As we process new documents, we may find a few new named entities and numerous new variants of the known ones that are not on the lists. We thus devised lexical rules, which we call trigger events, to detect such new forms. Certain words such as sample, specimen, detection limit and limit of detection (LOD) are usually present within multiword noun phrases of interest and alert the algorithm to check the words surrounding them. These trigger events enable us to detect multiword named entities that are not in our specimen, analyte, or unit of measure list (e.g., head hair sample); thus, they may not be identified through simple dictionary lookups. For example, trigger events may yield false positive results in cases such as patient sample or 20 specimens but the algorithm would filter out such false positive results by verifying that the word patient and the number 20 were not in our list of specimen types.

**Contextualization**

Besides filters and trigger events, contextualization is also an effective tool to improve accuracy. In this study, context is signified by sections, which have been identified at the document preprocessing stage as referred above. In most cases, a decision summary contains a section titled as analyte or Measurand, where analytes measured by the laboratory test are discussed; thus, the algorithm should focus on this section to detect the analyte of interest.

Contextualization also helps us decide on what to look for and select the appropriate pattern recognition method accordingly. For example, intended use sections frequently discuss about analytes in conjunction with the corresponding specimens in the same sentence, which we can detect both using a regular expression as described below.

| Entity type         | Definition                                                                 | Examples of extracted information | Methods                                      | Ranked by             |
|---------------------|---------------------------------------------------------------------------|-----------------------------------|----------------------------------------------|-----------------------|
| Analyte             | Substance or chemical constituent that is analyzed                        | Glucose, methadone                | Pattern recognition, filtering, contextualization | Frequency             |
| Specimen            | Sample of bodily substance collected for examination                      | Blood, urine, serum               | Pattern recognition, filtering, contextualization | Frequency, distance   |
| Unit of measure     | Unit of measurement for the analyte                                       | mg/dL, ng/mL, U/mL                | Filtering, trigger events                    | Frequency, distance, context, domain-knowledge |
| Detection limit     | Lowest analyte quantity detectable by the device in specimen              | 10 mg/dL, 300 ng/mL, 1 U/mL       | Pattern recognition, trigger events, contextualization | Frequency, distance, context |

**Pattern Recognition in the Sentence**

When filters and trigger events fail to recognize an entity of interest, our system may still capture it using patterns that are spread throughout a sentence in the Intended Use section. This technique is effective when the analyte and the specimen of interest are expressed in the same sentence. Figure 1 shows a deterministic finite state automaton (DFSA) with three distinct paths, each representing a pattern of collocations of analytes and specimens in the same sentence [Table 2 refers examples].

Pattern 1: @detection of $analyte in $specimen
  @delimiter/connector

Pattern 2: @detect $analyte in $specimen @delimiter/connector

Pattern 3: Use $specimen to @detect $analyte @delimiter/connector.

The variables in the DFSA are:
- @detection, which is a noun (e.g., detection, characterization, identification, determination, measurement, assessment, or evaluation) playing a trigger event role for detecting analytes
- @detect, a verb (e.g., detect, characterize, identify, determine, measure, assess, or evaluate) triggering analyte recognition
- @delimiter/connector, either a punctuation character (e.g., period or semicolon) specifying a context boundary or a connecting word (e.g., from, with, who, whose, which, or that); and
- $analyte and $specimen, which are target entities to be detected.

Handling of unknown words and long compounding word sequences are challenges in BMIE. Pattern recognition methods, such as the ones described above, can help us detect long noun phrases or even unknown words that are not contained in our dictionaries or thesauri. As shown in Table 2, long noun phrases such as clinician-collected female endovaginal and male urethral swab specimens, patient-collected vaginal swab specimens (in a clinical setting), and male and female urine specimens were
correctly identified as specimen names.

While pattern recognition described in this section is the major technique for detecting analytes and specimens, filters and trigger events as described above and contextualization stage as described above play important roles to reduce the number of false positive labels produced by the pattern recognition techniques.

For recognizing detection limit quantities, we use regular expressions and rely on trigger events and contextualization to filter out false positives. For recognizing units of measures, filtering and trigger events play a major role. Table 1 shows a summary of techniques used for detecting these laboratory test entities.

In general, using only one of these techniques can detect named entities, which however might not be the laboratory test entities of interest. It is a combination of these techniques that together reduce the number of false positives and improve accuracy.

### Candidate Ranking

The methods discussed above yield a list of potential entities that require a post-processing step to pinpoint the most plausible candidates. For example, specimen filtering may detect the term blood, which however, may not be the specimen of interest. Identifying the relevant specimen may require the consideration of all pertinent factors, namely whether the specimen name is nearby a trigger event (e.g., blood sample), whether it is part of a known pattern [Figure 1], or whether it is mentioned in a section such as intended use, where specimens of interest are discussed frequently. All these factors contribute to the likelihood of relevancy. After candidates of analyte, specimen and unit of measure and detection limit are recognized from the text, we rank them using a metric based on the following four scoring models: Frequency, distance, domain knowledge and context. For the rest of this section, let \( E \) represent a specific candidate entity instance (e.g., blood) and \( T_E \) represent the entity type of \( E \) (i.e., analyte, specimen, unit of measure, or detection limit).

#### Frequency

The relevancy of a named entity \( E \) is proportional to its relative frequency in the document, given all named entities of the same type. For example, if blood is mentioned in a document more frequently than any other specimen, it probably is the specimen of interest. The Frequency score for a specific entity \( E \) is calculated as defined in Equation (1):

\[
\text{Freq}_{\text{score}}(E) = \frac{\text{# occurrence of } E \text{ in the document}}{\text{total # entities in the document that are } T_E}
\]  

Since we normalize the raw entity frequency by the total frequency of all entities of the same type, the resulting score always lies between 0 and 1. Note that this frequency-based ranking method applies to all four entity types, i.e., analyte, specimen, unit of measure, or detection limit.

#### Distance

The relevancy of a non-analyte named entity \( E \) increases as it gets closer to the analyte term \( R \) in the document.
Its ranking score is inversely proportional to its distance to \( R \). Since there can be multiple instances for each analyte term \( R \), let \( S^8 \) denote the set of instances of \( R \). For each instance \( R_j \) in set \( S^8 \), let \( d(E, R_j) \) be the number of sentences (within the same section) from \( E \) to \( R_j \). If \( E \) and \( R_j \) are mentioned in the same sentence then \( d = 0 \), but if they are located in two subsequent sentences then \( d = 1 \). The distance score of entity \( E \) is inversely proportional to the Shortest Distance between \( E \) and \( R_j \) over all \( R_j \in S^8 \), which takes the form in Equation (2):

\[
\text{Distance}_{\text{score}}(E) = \frac{1}{\min_j(d(E, R_j)) + 1} \tag{2}
\]

Where, \( R_j \in S^8 \)

**Domain Knowledge**

The name space of units of measures is well-constrained. Any scientist or clinician can easily distinguish most units of measures from other arbitrary character strings based on their domain knowledge. We captured that particular knowledge from 2.7 million structured clinical laboratory records by associating each distinct unit of measure \( E \) with its prevalence \( P(E) \) that is the probability of its occurrence in the laboratory records (Equation (3)).

\[
\text{Domain}_{\text{score}}(E) = P(E) \tag{3}
\]

**Context**

The context score applies to the numerical quantity of a detection limit and its associated unit of measure instance. We assign \( 1 \) to their context scores when such a numerical quantity is followed by a known unit of measure (Equation (4)).

\[
\text{Context}_{\text{score}}(E) = \begin{cases} 1 & \text{if a quality is followed by a unit of measure} \\ 0 & \text{otherwise} \end{cases} \tag{4}
\]

We obtain the final score of an entity by averaging these four scoring models. The inclusion of models toward the final score depends on the entity type as presented in the Ranked-by column of Table 1. For example, for analytes, the frequency score is equivalent to the final score; whereas for specimens, the equally weighted linear combination of frequency and distance scores determines the final score. The algorithm outputs the highest ranked entity for every entity type and filters out the rest.

**EVALUATION**

We developed a prototype SIE system, based on methods described in the “Methods” as above on a set of FDA decision summaries and tested its performance on a randomly selected, separate (disjoint) set of decision summaries. We also chose three software packages BANNER, LingPipe and GATE, which are implementations of CRF, HMM and SVM, respectively (Section “Machine Learning Approaches”). We evaluated the results of SIE, CRF, HMM and SVM comparatively on the same test set.

**Dataset**

We randomly selected 50 FDA decision summaries from a total of 1900 documents available at the FDA’s website. We excluded from this set three decision summaries, each reporting about multiple analytes. Exclusion of these three documents provided uniformity to the dataset and greatly simplified the analyses. In our dataset, each decision summary was associated with at most one analyte, one unit of measure and one detection limit but could contain multiple specimens.

Since the three machine learning systems that we tested in this study are generic named entity recognition systems, they output entity location and frequency information by labeling each token either as an entity or not. To generate training documents for them, we used the IOB tagging scheme and manually labeled every token as B-X, I-X, or O, where B represents the beginning, I represents the inside, O represents the outside of an entity. The suffix X denotes one of the four entity types: Analyte, Specimen, Unit, or Limit. Figure 2 illustrates an IOB tagging example where analyte and specimen names were manually annotated. To evaluate the performance of each system, we compared all predicted labels against these annotations; hence, we refer this annotation set as the gold standard. There is 1105 analyte, 877 specimen, 835 unit of measure and 70 detection limit instances/observations in this dataset. Note that information about locations of multiple instantiations of the same entity would be crucial for a named recognition study, but is out of the scope of an information extraction study.

**Experimental Method**

We tested every machine learning system through a leave-one-out cross-validation experiment on the 47 documents. That is we tested each document (without labels) by using the remaining 46 labeled documents as the training data and iterated the process 47 times to test all available documents. We repeated the cross-validation experiment for every machine learning system.

We tested the CRF model implemented on BANNER using its default input feature set that includes part-of-speech tags, lemmatization, character prefix and suffixes, character n-grams, letter-case of tokens, numeric normalization, Roman numerals and Greek letters. We...
did not alter BANNER’s default training parameters in our experiments.

We tested the JAVA version of SVM implemented on GATE. In our experiments, we decided to stick to the parameter setting as suggested in the GATE User Guide for information extraction and used its token-string, token-kind (e.g., word, number, punctuation, or space), token-orthography and part-of-speech features.\(^\text{[52]}\)

We tested HMM implemented on LingPipe using its trainable named entity recognizer, CharLmRescoringChunker, which is a first-order Markov model and claimed to be the most accurate one among the three chunkers provided by LingPipe.\(^\text{[53]}\) The parameter settings for the three machine learning systems are detailed in Appendix 1.

Recall that SIE is an information extraction system producing a set of entities per document without duplications; whereas, the three machine learning systems are generic named entity recognition systems labeling every instantiation of the same entity in various locations in the document. We mapped multiple prediction instances of the same entity into one so that we could fairly compare the information extraction results of different systems. When instances of an entity were associated with different confidence measures in a document, as in outputs of HMM and SVM based systems, we chose the greatest value among them. For example, if HMM had identified in a document two occurrences of blood as a specimen with confidence measures of 0.7 and 0.9, we would have listed blood as a recognized specimen associated with a confidence measure of 0.9. Since the CRF based system did not produce confidence measures, we used our frequency metric in Equation (1) as a proxy confidence measure for CRF.

We compared the sets of entities obtained from all systems against our gold standard. When the outputs did not exactly match to the gold standard phrases (e.g., whole blood) but contained necessary and sufficient parts of these phrases (e.g., blood sample) we evaluated them as true positives. For each of the four systems, we calculated the mean precision, recall and F-measure per entity type over all 47 runs.

There were two different modes of evaluation: Uniform and weighted. Under the uniform evaluation mode, we considered all predictions are equally valid and weighed them equally. Under the weighted evaluation mode, we weighed every prediction proportional to their corresponding confidence measures. For example, if a system labeled serum, plasma and blood as specimen names with confidence measures 0.9, 0.8 and 0.7, respectively and the gold standard specimen name was serum, the tally is 1 true positive and 2 false positives under the uniform evaluation mode and 0.9 \times 1 + 0.8 \times 1 = 1.5 false positives under the weighted evaluation mode. We computed the final precision, recall and F-measure results as the arithmetic mean of the tallies obtained from all documents.

We associated each precision, recall and F-measure with a 95% confidence interval (CI), which we computed through a non-parametric bootstrap method called bias-corrected, accelerated percentile intervals\(^\text{[54]}\) using the package boot in R.\(^\text{[55]}\) For the scores where CIs were overlapping, we computed statistical significance based on Wilcoxon paired signed test with adjustments as suggested by Pratt\(^\text{[56]}\) using the package coin in R.\(^\text{[57]}\) F-measure is the harmonic mean of recall R and precision P, i.e., \(2 \frac{RP}{R+P}\).

RESULTS

The box-and-whisker plot in Figure 3 presents the token distribution characteristics (by quartiles) of the four entity types over the 47 documents, where the black diamond in each box denotes the mean count of entity tokens per document. Note that a document in our test corpus contains a total of 2711 tokens in average.

Among the four entity types, the unit of measure was the highest and the detection limit was the lowest in token counts per document. Units were used to convey information about various measurements (hence, plenty of them); whereas, the detection limit was a very specific measure with very low prevalence – more than a quarter of the documents did not contain it at all. The mean detection limit token count was 2.4 per document. With the exception of three documents, which didn’t contain any unit of measure, all documents contained the remaining two entity types, namely analyte and specimen.

Tables 3 and 4 present experimental results in terms of the mean recall, precision, F-measure and their 95% CIs over 47 runs under the uniform and weighted evaluation
The highest score and the associated 95% CI boundaries in a given row were highlighted with a bold font if its differences from the scores of the other three systems in the same row were statistically significant ($P < 0.05$).

The results in the uniform evaluation mode [Table 3] show that in all information extraction tasks SIE performed better than the other three systems in precision and $F$-measure and the performance differences were statistically significant. CRF was consistently the second best performer on those measures, but the precision differences between SIE and CRF in all four tasks was on average 30.1%, ranging from 19.3% to 62.5%. The recall performance differences were not statistically significant on all tasks other than on the analyte extraction task.

We labeled the lower bound of the 95% CI of SIE’s recall for analyte extraction with N/A since we could not estimate it due to the lack of any false negative instance in this experiment. SIE clearly outperformed the other three systems on this task on all three measures [Tables 3 and 4].

Machine learning systems scored higher on specimen and units of measure extraction tasks compared to the other

### Table 3: Performance (in %) of methods under uniform evaluation mode

| Entity type      | SIE          | CRF          | HMM          | SVM          |
|------------------|--------------|--------------|--------------|--------------|
| **Analyte**      |              |              |              |              |
| R                | 100 (N/A, 100)| 36.2 (22.9, 50.0) | 42.6 (27.7, 56.5) | 59.6 (43.5, 72.3) |
| P                | 95.8 (83.0, 97.9) | 33.3 (19.6, 47.9) | 8.2 (5.3, 11.6) | 21.7 (14.3, 30.0) |
| E                | 97.9 (89.4, 98.9) | 34.7 (22.2, 48.3) | 13.7 (9.2, 19.0) | 31.8 (22.4, 41.4) |
| **Specimen**     |              |              |              |              |
| R                | 98.4 (91.8, 100) | 88.9 (79.0, 94.9) | 90.5 (80.7, 96.2) | 92.1 (81.7, 97.0) |
| P                | 96.9 (90.4, 100) | 76.7 (66.2, 85.5) | 38.8 (30.8, 46.8) | 58.0 (40.5, 67.7) |
| F                | 97.6 (93.8, 99.3) | 82.3 (73.2, 88.6) | 54.3 (45.0, 62.2) | 71.1 (56.3, 79.0) |
| **Unit of measure** |          |              |              |              |
| R                | 77.8 (61.1, 89.2) | 91.7 (76.9, 97.4) | 91.7 (77.8, 97.4) | 75.0 (58.8, 86.8) |
| P                | 59.6 (42.6, 70.2) | 37.1 (29.8, 44.9) | 21.4 (16.4, 26.4) | 17.4 (12.2, 23.1) |
| F                | 67.5 (53.2, 79.6) | 52.8 (44.3, 60.5) | 34.7 (27.9, 41.6) | 28.3 (20.5, 35.9) |
| **Detection limit** |          |              |              |              |
| R                | 45.7 (27.7, 62.9) | 14.3 (5.4, 27.8) | 68.6 (50.0, 82.4) | 0             |
| P                | 41.0 (24.4, 58.3) | 21.7 (9.1, 41.7) | 3.1 (2.3, 4.2)   | 0             |
| F                | 43.2 (27.9, 58.8) | 17.2 (7.1, 31.8) | 6.0 (4.5, 7.8)   | N/A           |

R, P, and F followed by the corresponding 95% confidence interval in parentheses. R: Recall, P: Precision, F: $F$-measure, SIE: Symbolic information extraction, CRF: Conditional random fields, HMM: Hidden Markov models, SVM: Support vector machines, N/A: not applicable.

### Table 4: Performance (in %) of methods under weighted evaluation mode

| Entity type      | SIE          | CRF          | HMM          | SVM          |
|------------------|--------------|--------------|--------------|--------------|
| **Analyte**      |              |              |              |              |
| R                | 100 (N/A, 100)| 33.6 (20.3, 48.3) | 39.1 (25.7, 53.4) | 48.1 (33.1, 64.1) |
| P                | 95.8 (83.0, 97.9) | 49.8 (32.2, 66.6) | 8.5 (5.6, 11.9) | 24.7 (16.0, 34.7) |
| F                | 97.9 (89.4, 98.9) | 40.1 (26.1, 55.5) | 13.9 (9.0, 19.6) | 32.7 (22.6, 44.1) |
| **Specimen**     |              |              |              |              |
| R                | 98.4 (91.8, 100) | 85.0 (72.9, 93.4) | 89.8 (79.5, 96.3) | 91.1 (80.6, 96.6) |
| P                | 96.9 (90.4, 100) | 90.0 (79.3, 95.5) | 41.9 (34.0, 50.2) | 63.1 (49.4, 71.6) |
| F                | 97.6 (93.8, 99.3) | 87.4 (76.8, 94.1) | 57.1 (48.1, 65.2) | 74.6 (63.6, 81.4) |
| **Unit of measure** |          |              |              |              |
| R                | 77.8 (61.1, 89.2) | 90.4 (74.7, 97.0) | 91.6 (77.6, 97.4) | 72.8 (55.1, 86.4) |
| P                | 59.6 (42.6, 70.2) | 67.1 (54.3, 77.7) | 22.8 (17.9, 28.4) | 23.2 (17.4, 29.1) |
| F                | 67.5 (53.2, 79.6) | 77.0 (65.0, 85.3) | 36.5 (29.8, 43.7) | 35.2 (27.5, 43.2) |
| **Detection limit** |          |              |              |              |
| R                | 45.7 (27.7, 62.9) | 11.0 (3.4, 25.8) | 66.9 (48.9, 82.2) | 0             |
| P                | 41.0 (24.4, 58.3) | 23.1 (7.7, 47.5) | 3.0 (2.3, 4.2)   | 0             |
| F                | 43.2 (27.9, 58.8) | 14.9 (5.0, 32.6) | 6.0 (4.4, 7.9)   | N/A           |

R, P, and F followed by the corresponding 95% confidence interval in parentheses. R: Recall, P: Precision, F: $F$-measure, SIE: Symbolic information extraction, CRF: Conditional random fields, HMM: Hidden Markov models, SVM: Support vector machines, N/A: not applicable.
two tasks. Specimens and units of measures are easily detectable in the text because the sets of all potential specimens and units are well-constrained. Detection limit extraction requires recognition of a numeric quantity, which is usually followed by a unit of measure. But not every measurement refers to the detection limit of the device. Although CRF and HMM found correct units of measures with high recall rates, they experienced difficulties in discerning relevant ones among a sheer number of irrelevant units of measures. Similarly, decimal numbers, each potentially a detection limit, are easily recognizable in text, but labeling every decimal numbers as detection limits were non-informative. No system was quite competent in extracting detection limit information and SVM could not identify any of those limits. SIE’s performance on this task was balanced as indicated by relatively high F-measure. Compared to SIE, none of machine learning systems did well on any of the four tasks in F-measure.

When we compare the precisions of these systems on three tasks, analytes, unit of measure and detection limit extraction, we observe that the lowest precision of SIE (41% on detection limit extraction) was higher than the highest precision of the rest (37.1% on analyte extraction by CRF). Their average precision on those three tasks was 18.2%, where CRF was the best performer with 30.7% average precision, followed by SVM (13.0%) and HMM (10.9%).

Table 4 shows the weighted mean recall, precision, F-measure and their 95% CIs. Note that the CRF-based system in this study did not produce confidence measures. We applied our frequency metric in Equation (1) to CRF results as a proxy for confidence measures. We designed SIE to produce a single prediction per entity type, unless the resulting scores of the most likely candidates were identical (i.e., were equally ranked). Even when that exception occurs, weighted average calculation would produce the same results as the uniform mode because every candidate would be weighed with the same coefficient 1; hence, its results in Tables 3 and 4 are identical.

In the uniform evaluation mode, all coefficients (weights) were 1 in counting the total of true and false positives; whereas in the weighted mode, the confidence measures of machine learning systems ranged between 0 and 1, substituting the uniform coefficients. Consequently, the recall rates were slightly lowered for the three machine learning systems as seen in Table 4. On the other hand, the weighted evaluation yielded more favorable precision and F-measure results for those systems. With the exception of one case (HMM’s detection limit performance), weighted evaluation increased the precision values of all learning systems across all tests. It also improved F-measures of all learning systems on all tasks except the detection limit extraction. These improvements were most prominent for the CRF system, where we applied our frequency metric to compensate its lack of confidence measures. The most extreme demonstrative case happened on the unit of measure task, where CRF’s recall rate decreased by 1.3% while its precision rate improved by 30%. Using our frequency metric as its confidence measures, CRF erased the F-measure and precision superiority of SIE in extracting unit-of-measure information, but their performance difference was not statistically significant. On the other tasks, SIE maintained its statistically significant superiority (with p < 0.01 in overlapping CIs).

**DISCUSSION**

SIE could recognize all analytes in the test data with a very high precision rate, mainly because SIE embodied a detection method for the headers and boundaries of document sections. Given the fact that many documents in the test set comprised a section with either an analyte or a measurand header, SIE did not have difficulty recognizing analyte names in those sections of the test data. Although all learning systems of this study were equipped with various NLP tools, none of them had an ability to perform a document structure analysis. Furthermore, SIE was the only system of this study that could utilize a large list of potential analytes. Because of these disparities, there was no real contest between SIE and machine learning systems on this task.

Limited training data and sparse feature set make rare or unseen entities in the documents too hard to find for any machine learning system. It also was the case for the three machine learning systems of this study. For example, while SIE was able to detect specimens in the following example, no machine learning system could do so: “Clinician-collected female endocervical and male urethral swab specimens, patient-collected vaginal swab specimens (in a clinical setting) and male and female urine specimens”.

Unlike specimen names, units of measures and detection limits did not collocate with analytes in the test data. Thus, extracting their information (relative to extracting specimen information) was far more difficult for SIE. In some documents, identifying them was quite challenging even for human experts.

Units of measures tend to have easily discernible patterns; thus, the machine learning methods had high recall rates on this task, but they dramatically failed on the same task in precision as they mistakenly predicted too many irrelevant units of measures, which were of interest for other analytes in other documents in the training data. HMM and CRF models of this study employed first- and second-order dependencies, respectively. Low precision scores of HMM models on labeling unit of measures could be associated with their suboptimal dependency.
structures. The failures of SVM models were surprising since SVMs are usually capable of detecting higher order dependencies with relatively low computational complexity.

Extracting detection limit usually requires the collocated unit-of-measure information. Therefore, the success in extracting unit of measure is expected to be the upper bound for extracting detection limit information. Since only less than three-fourths of decision summaries contained a detection limit and given the small training set, extracting the detection limit information was too difficult for all learning systems. Furthermore, they were not equipped with learning relevant numeric quantities from documents, especially if the text is saturated with numbers and measures most of which were not of interest. On this task, SVM failed completely and HMM’s precision was slightly better than SVM’s. Although CRF did well relative to the other learning systems and SIE performed better than CRF, none of the systems has shown competency on extracting the numeric quantity information.

Note that we developed SIE around the patterns that we observed in a separate set of decision summaries. The finite state automaton (FSA) in Figure 1 reflects those patterns. Furthermore, SIE had benefited from our context analysis and associated document sections with the degree of relevancy. None of these information pieces was available to the machine learning systems, yet they could compete with SIE despite the small training sample that we could afford to build. Therefore, the success of machine learning systems should not be judged solely on absolute performance metrics. These same systems could perform in another context at least as well as in this study without much retuning — we cannot claim the same for SIE.

Would there be a statistically significant improvement of any machine learning system performance, if we could afford to double the training size? Based on the experimental results reported in the literature, the answer is: Unlikely.

There were certain elements of SIE that were not specific to the study; e.g., document structure analysis, using sections as features, associating those features with decision thresholds, using external information or corpora to serve filters and trigger events, ranking candidates by their frequencies in the document, using collocations to relate separate entities such as numbers and units. All these characteristics of SIE were effective elements in this study and can easily be incorporated with any of these three learning systems of this study.

Conversely, SIE could benefit from machine learning methods as well. Note that SIE’s candidate ranking was rather ad hoc and might be suboptimal. SIE’s performance could be improved if its candidate ranking method were parameterized optimally through learning from data.

The results suggest that machine learning systems can be optimized further by utilizing linguistic tools and methods more extensively. Even though all machine learning systems of this study came with NLP tools with varying degrees of sophistication, none was sophisticated enough to outperform SIE. SIE was clearly superior to others in analyte information extraction because, unlike others, SIE was equipped with a rudimentary, yet sufficient technique for document structure analysis and a large corpus of potential analytes for filtering the results.

On the other hand, it is also important not to read too much in the success of SIE in this study. Because, unlike the machine learning systems, SIE was tailored specifically for these tasks and it was sensitive to the characteristics of this particular corpus. For example, as FDA’s way of writing decision summaries changes over time, we would expect some degradation in SIE’s performance unless its algorithm is revised or retuned. Obviously, the same applies to the machine learning systems to a lesser degree as they would require new training data.

Although the performance of SIE may not be permanent and would be influenced by the changes of the corpus characteristics over time, the underlying methods of SIE and how to implement them successfully are the lasting features of this study. These methods, especially the metrics, can be used in similar tasks of information extraction from different biomedical corpora.

**CONCLUSION**

We developed a SIE method to extract four different types of diagnostic laboratory test information (Interested parties should contact the first author to obtain a copy of code to study). We evaluated SIE on this task along with three distinct machine learning methods, CRFs, SVMs, and HMMs. Evaluation results show that the machine learning methods competed well with varying degrees of success against SIE despite the small and sparse training data with little or no tuning. In particular, CRF, whose results we adjusted with our frequency metric in the weighted evaluation mode, performed better than SIE in extracting units of measures. On the other hand, this study also shows that a generic machine learning system may not outperform a well-tailored symbolic method in detecting more complex named entities and their relevancies.

This study scratches the surface of the BMIE problem for some diagnostic laboratory test entities (analytes, specimens, units of measures and detection limits) for the first time in the literature. Detecting such entities properly is an important first step for extracting laboratory
information from literature and narrative clinical reports. This study demonstrates the importance and the difficulties of identifying relevancy (e.g., the right units of measures and the right numeric quantities) among a multitude of candidates. It also highlights that it is not reasonable to successfully train a learning system with rare and complex features such as “clinician-collected female endocervical and male urethral swab specimens” without constructing a semi-sophisticated linguistic model.

The results strongly suggest that the developers of the next generation computational linguistic learning systems should not ignore the document structure analysis and use document sections as features to be associated with the outcome of interest. They should also be cognizant of the potential symbiosis between symbolic and machine learning methods. Finally, we also note that structure-learning methods have potentials to learn patterns in FSAs directly from data, but whether they can produce SIE like systems automatically is still an open question.

REFERENCES

1. Aronson AR, Mork JG, Gay CW, Humphrey SM, Rogers WJ. The NLM indexing initiative's medical text indexer. Stud Health Technol Inform 2004;107:268‑72.
2. Humphrey SM, Névéol A, Gobel J, Ruch P, Darmoni SJ, Browne A. Comparing a rule based vs. statistical system for automatic categorization of MEDLINE documents according to biomedical specialty. J Am Soc Inf Sci Technol 2009;60:2530‑9.
3. Kastrin A, Peterlin B, Hristovski D. Chi-square-based scoring function for categorization of MEDLINE citations. Methods Inf Med 2010;49:371‑8.
4. Erhardt RA, Schneider R, Blaschke C. Status of text-mining techniques applied to biomedical text. Drug Discov Today 2006;11:315‑25.
5. Fukuda K, Tamura A, Tsunoda T, Takagi T. Toward information extraction: Identifying protein names from biological papers. Pac Symp Biobios Comput 1998:707‑18.
6. Zhou G, Zhang J, Su J, Shen D, Tan C. Recognizing names in biomedical texts: A machine learning approach. Bioinformatics 2004;20:1178‑90.
7. Wilbur WJ, Hazard GF, Diviga G, Mork JG, Aronson AR, Browne AC. Analysis of biomedical text for chemical names: A comparison of three methods. Proc AMIA Symp 1999:176‑80.
8. PLOS Collections Table of Contents: Text Mining. Available from: http://www.ploscollections.org/article/browseIssue.action?issue = info. [Last accessed on 2013 Jun 8].
9. U.S. Government Printing Office. Federal Food, Drug, and Cosmetic Act (FD&C Act). FD&C Act Section number: Sec. 510(k). Title: Sec. 360 - Registration of producers of drugs or devices. Available from: http://www.gpo.gov/fdsys/pkg/USC40-2010-title21/html/USC40-2010-title21-chap9-subchapV-parA-sec360.htm. [Accessed on 2012 Oct 1].
10. Food and Drug Administration. Premarket Notification (510k). Available from: http://www.fda.gov/MedicinalDevices/DeviceRegulationsandGuidance/HowtoMarketYourDevice/PremarketSubmissions/PremarketNotification510k/default.htm. [Last accessed on 2012 Oct 1].
11. Food and Drug Administration. 510(k) Premarket Notification. Available from: http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfPMN/PMN.cfm. [Last accessed on 2012 Oct 1].
12. Regenstrief Institute. I. Logical Observation Identifiers Names and Codes. Available from: http://www.loinc.org. [Last accessed on 2010 Jan 10].
13. U.S. National Institutes of Health. ClinicalTrials.gov. Available from: http://www.clinicaltrials.gov. [Last accessed on 2013 Jan 8].
14. U.S. National Library of Medicine. PubMed.gov. Available from: http://www.ncbi.nlm.nih.gov/pubmed. [Last accessed on 2013 Jun 8].
15. U.S. National Library of Medicine. Databases, Resources & APIs. Available from: http://wwww2.nlm.nih.gov/nlm_eresources/eresources/search_database.cfm. [Last accessed on 2013 Jun 8].
16. Ciravegna F, Lavelli A, Mana N, Matiashek J, Gilardoni L, Mazza S, et al. FACILE: classifying texts integrating pattern matching and information extraction. In: Proceedings of 16th International Joint Conference on Artificial Intelligence. Stockholm: Morgan Kaufmann Publishers; 1999. p. 890‑5.
17. Grishman R. The role of syntax in information extraction. Proc Tipster Text Program 24‑month Conference 1996:139‑42.
18. Hobbis JR, Appelt DE, Bear J, Tyson M, Magerman D. The TACITUS System: The MUC‑3 Experience. Menlo Park, CA; SRI, 1991.
19. Narayananwamy M, Ravi Kumar KE, Vijay Shanker K. A biological named entity recognizer. Pac Symp Biobios Comput 2003:427‑38.
20. Wilks Y. Information extraction as a core language technology. Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology Lecture Notes in Computer Science. Vol. 1229. Springer‑Verlag London, UK; 1997. p. 1‑9.
21. Collier N, Nobata C, Tsujii J. Extracting the names of genes and gene products with a hidden Markov model. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000). Saarbrucken, Germany: Association for Computational Linguistics Stroudsburg, PA, USA; 2000. p. 201‑7.
22. Lee KJ, Hwang YS, Kim S, Rim HC. Biomedical named entity recognition using two-phase model based on SVMs. J Biomed Inform 2004;37:436‑47.
23. Lin YF, Tsai TH, Chou WC, Wu KP, Sung TY, Hsu WL. A maximum entropy approach to biomedical named entity recognition. In: The 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics. Seattle, WA; 2004. p. 56‑61.
24. McCallum AK. Efficiently inducing features of conditional random fields. In: Proceedings of Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA; Morgan Kaufmann Publishers Inc; 2003. p. 403‑10.
25. Szarvas G, Forkas R, Koscor A. A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms. In: Proceedings of the 9th international conference on Discovery Science. Springer‑Verlag Berlin, Heidelberg; 2006. p. 267‑78.
26. Sithari R, Niu C, Li W. A hybrid approach for named entity and sub-type tagging. In: Proceedings of the 6th Conference on Applied Natural Language Processing, Stroudsburg PA; USA: Association for Computational Linguistics 2000. p. 247‑54.
27. Chapman BE, Lee S, Kang HP, Chapman WW. Document-level classification of CT pulmonary angiography reports based on an extension of the ConText algorithm. J Biomed Inform 2011;44:728‑37.
28. Liu K, Hogan WR, Crowley RS. Natural language processing methods and systems for biomedical ontology learning. J Biomed Inform 2011;44:163‑79.
29. Mansouri A, Affendey LS, Mamat A. Named entity recognition using a new fuzzy support vector machine. Int J Comput Sci Netw Secur 2008;8:320‑5.
30. Shladover JH. Escaping brittleness: The possibility of general purpose learning Algorithms applied to parallel rule-based systems. Computation & intelligence. Menlo Park, CA, USA: American Association for Artificial Intelligence; 1995. p. 275‑304.
31. Freitag D. Using grammatical inference to improve precision in information extraction. In: Proceedings of the Workshop on Grammatical Inference, Automata Induction, and Language Acquisition (ICML’97). Nashville, TN, San Mateo, CA: Morgan Kaufmann Publishers; 1997.
32. Crowley RS, Castine M, Mitchell K, Chavan G, McSherry T, Feldman M. caTIES: A grid based system for coding and retrieval of surgical pathology reports and tissue specimens in support of translational research. J Am Med Inform Assoc 2004;11:268‑72.
33. Mansouri A, Affendey LS, Mamat A. Named entity recognition using a new fuzzy support vector machine. Int J Comput Sci Netw Secur 2008;8:320‑5.
34. Quinlan JR. C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.
35. Riloff E. Automatically generating extraction patterns from untagged text. In: Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI) 1991;725‑32.
36. Quinlan JR. C4.5: Programs for Machine Learning, San Mateo, CA: Morgan Kaufmann, 1993.
37. Riloff E. Automatically generating extraction patterns from untagged text. In: Proceedings of the 13th National Conference on Artificial Intelligence (AAAI). AAAI Press; 1996. p. 1044‑9.
38. Sonderland S. Fisher D, Aseltine J, Lehert W. CRYSTAL: Inducing a conceptual dictionary. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI), Vol. 2. San Francisco, CA, USA; Morgan Kaufmann Publishers Inc; 1995. p. 1314‑9.
39. Sonderland S. Learning information extraction rules for semi-structured and free text: Mach Learn 1999;34:233‑72.
APPENDIX 1: CONFIGURATION PARAMETER SETTINGs FOR MACHINE LEARNING SYSTEMS

BANNER (The CRF Algorithm)
We used BANNER’s default training parameters in our experiments as shown below. We consulted these parameter settings with BANNER developers and these settings were confirmed to be valid.

```plaintext
lemmatiserDataDirectory = nlpdata/lemmatiser
posTagger = dragon.nlp.tool.HeppleTagger
posTaggerDataDirectory = nlpdata/tagger
tokenizer = banner.tokenization.SimpleTokenizer
tagFormat = IOB
useNumericNormalization = true
useParenthesisPostProcessing = true
order = 2
useFeatureInduction = False
textDirection = Forward
```

GATE (The SVM Algorithm)
We used the following primary parameter settings. We consulted these parameter settings with GATE developers and these settings were confirmed to be valid.

```plaintext
thresholdProbabilityClassification = 0.5
thresholdProbabilityBoundary = 0.4
thresholdProbabilityEntity = 0.2
Kernel_type (t) = polynomial
degree in polynomial Kernel (d) = 2
cost associated with allowing training errors (c) = 0.7
value of uneven margins (τ) = 0.4
```

LingPipe (The HMM Algorithm):
The `CharLmRescoringChunker` has three primary configuration settings, the number of n-best chunkings to be rescoring, the maximum length of n-grams to be used, and whether to use smoothing. We used the following parameter settings for evaluation.

```plaintext
number of n-best chunkings = 64
maximum length of n-grams = 12
smoothing = true
```