BANNER: AN EXECUTABLE SURVEY OF ADVANCES IN BIOMEDICAL NAMED ENTITY RECOGNITION

ROBERT LEAMAN

Department of Computer Science and Engineering, Arizona State University

GRACIELA GONZALEZ[*]

Department of Biomedical Informatics, Arizona State University

There has been an increasing amount of research on biomedical named entity recognition, the most basic text extraction problem, resulting in significant progress by different research teams around the world. This has created a need for a freely-available, open source system implementing the advances described in the literature. In this paper we present BANNER, an open-source, executable survey of advances in biomedical named entity recognition, intended to serve as a benchmark for the field. BANNER is implemented in Java as a machine-learning system based on conditional random fields and includes a wide survey of the best techniques recently described in the literature. It is designed to maximize domain independence by not employing brittle semantic features or rule-based processing steps, and achieves significantly better performance than existing baseline systems. It is therefore useful to developers as an extensible NER implementation, to researchers as a standard for comparing innovative techniques, and to biologists requiring the ability to find novel entities in large amounts of text.

BANNER is available for download at http://banner.sourceforge.net.

1. Introduction

With molecular biology rapidly becoming an information-saturated field, building automated extraction tools to handle the large volumes of published literature is becoming more important. This need spawned a great deal of research into named entity recognition (NER), the most basic problem in automatic text extraction. Several challenge evaluations such as BioCreative have demonstrated significant progress [19, 20], with teams from around the world implementing creative solutions to the known challenges in the field such as the unseen word problem and the mention boundary problem. Although there are other open-source NER systems such as ABNER [11] and LingPipe [1] which are freely available and have been extensively used through the years as

[*] Partially supported by NSF CISE grant 0412000 (BPC supplement)

baseline systems, the advances since the creation of these systems have mostly remained narrated in published papers, and are generally not available as easily deployable code. Thus the field now sees a great need for a freely-available, open-source system implementing these advances for a more accurate reflection of what a baseline system should achieve, allowing researchers to focus on alternative approaches or extensions to the known techniques. In other words, the field needs an updated measuring stick.

We present here BANNER, an open-source biomedical named-entity recognition system implemented using conditional random fields, a machine learning technique. It represents an innovative combination of known advances beyond the existing open-source systems such as ABNER and LingPipe, in a consistent, scalable package that can easily be configured and extended with additional techniques. It is intended as an executable survey of the best techniques described in the literature, and is designed for use directly by biologists, by developers as a building block, or as a point of comparison when experimenting with alternative techniques.

2. Background

Named entity recognition (NER) is the problem of finding references to entities (*mentions*) such as genes, proteins, diseases, drugs, or organisms in natural language text, and labeling them with their location and type. Named entity recognition in the biomedical domain is generally considered to be more difficult than other domains, such as newswire, for several reasons. First, there are millions of entity names in use [19] and new ones are added constantly, implying that neither dictionaries nor training data will be sufficiently comprehensive. Second, the biomedical field is moving too quickly to build a consensus on the name to be used for a given entity [6] or even the exact concept defined by the entity itself [19], while very similar or even identical names and acronyms are used for different concepts [6], all of which results in significant ambiguities. Although there are naming conventions, authors frequently do not follow them and instead prefer to introduce their own abbreviation and use that throughout the paper [2, 19]. Finally, entity names in biomedical text are longer on average than names from other domains, it is generally much easier – for both humans and automated systems – to determine whether an entity name is present than it is to detect its boundaries [7, 19, 20].

Named entity recognition is typically modeled as a *label sequence problem*, which may be defined formally as follows: Given a sequence of input tokens $\boldsymbol{x} = (x_1 \dots x_n)$, and a set of labels $\boldsymbol{L}$, determine a sequence of labels $\boldsymbol{y} = (y_1, ..., y_n)$ such that $y_i \in \boldsymbol{L}$ for $1 \leq i \leq n$. In the case of named entity recognition the labels incorporate two concepts: the type of the entity (e.g. whether the name refers to a protein or a disease), and the position of the token within the entity. The simplest model for token position is the *IO* model, which indicates whether the token is **I**nside an entity or **O**utside of a mention. While simple, this model cannot differentiate between a single mention containing several words and distinct mentions comprising consecutive words [21]. The next-simplest model used is *IOB* [11], which indicates whether each token is at the **B**eginning of an entity, **I**nside an entity, or **O**utside. This model is capable of differentiating between consecutive entities and has good support in the literature. The most complex model commonly used is *IOBEW*, which indicates whether each token is at the **B**eginning of an entity, **I**nside an entity, at the **E**nd of an entity, a one-**W**ord entity, or **O**utside an entity. While the *IOBEW* model does not provide greater expressive power than the *IOB* model, some authors have found it to provide the machine learning algorithm with greater discriminative power, which may translate into higher accuracy [16]. Example sentences annotated using each label model can be found in table 1.

Table 1. Example sentences labeled using each of the common labeling models, taken from the BioCreative 2 GM training corpus [19].

Label model	Example
IO	Each|O immunoprecipitate|O contained|O a|O complex|O of|O N1|I-GENE (|I-GENE 3eltaic|I-GENE)|I-GENE and|O CBF1|I-GENE .|O
IOB	TNFalpha|B-GENE and|O IL|B-GENE -|I-GENE 6|I-GENE levels|O were|O determined|O in|O the|O culture|O supernatants|O .|O
IOBEW	CES4|W-GENE on|O a|O multicopy|O plasmid|O was|O unable|O to|O suppress|O tif1|B-GENE -|I-GENE A79V|E-GENE .|O

Conditional random fields (CRF) [14] are a machine learning technique that forms the basis for several other notable NER systems including ABNER [11]. The technique can be seen as a way to "capture" the hidden patterns of labels, and "learn" what would be the likely output considering these patterns. Like all supervised machine learning techniques, a CRF-based system must be trained on labeled data. In general, a CRF is modeled as an arbitrary undirected graph, but linear-chain CRFs, their linear form, are used for sequence labeling. In a CRF, each input x_i from the sequence of input tokens $\boldsymbol{x} = (x_1 \ldots x_n)$ is a vector of real-valued *features* or descriptive characteristics, for example, the part of speech. As each token is labeled, these features are used in conjunction with the pattern of labels previously assigned (the history) to determine the most likely label for the current token. To achieve tractability, the length of the history used, called the *order*, is limited: a 1st-order CRF uses the last label output, a 2nd-order CRF uses the last two labels, and so on. There are several good introductions to conditional random fields, such as [14] and [18].

As a discriminant model, conditional random fields use conditional probability for inference, meaning that they maximize $p(\boldsymbol{y}|\boldsymbol{x})$ directly, where $\boldsymbol{x}$ is the input sequence and $\boldsymbol{y}$ is the sequence of output labels. This gives them an advantage over generative models such as Hidden Markov Models (HMMs), which maximize the joint probability $p(\boldsymbol{x}, \boldsymbol{y})$, because generative models require the assumption that the input features are independent given the label. Relaxing this assumption allows discriminatively trained models such as CRFs to retain high performance even though the feature set contains highly redundant features such as overlapping n-grams or features irrelevant to the corpus to which it is currently being applied. This, in turn, enables the developer to employ a large set of rich features, by including any arbitrary feature the developer believes may be useful [14]. In addition, tolerating irrelevant features makes the feature set more robust with respect to applications to different corpora, since features irrelevant to one corpus may be quite relevant in another [6].

In contrast, another significant machine learning algorithm – support vector machines (SVMs) – also tolerate interdependent features, but the standard form of SVMs only support binary classification [21]. Allowing a total of only 2 labels implies that they may only recognize one entity type and only employ the *IO* model for label position, which cannot distinguish between adjacent entities.

3. Architecture

The BANNER architecture is a 3-stage pipeline, illustrated in Figure 1. Input is taken one sentence at a time and separated into *tokens*, contiguous units of meaningful text roughly analogous to words. The stream of tokens is converted to *features*, each of which is a name/value pair for use by the machine learning algorithm. The set of features encapsulates all of the information about the token the system believes is relevant to whether or not it belongs to a mention. The stream of features is then *labeled* so that each token is given exactly one label, which is then output.

The tokenization of biomedical text is not trivial and affects what can be considered a mention since generally only whole tokens are labeled in the output [20]. Unfortunately, tokenization details are often not provided in the biomedical named entity recognition literature. BANNER uses a simple tokenization which breaks tokens into either a contiguous block of letters and/or digits or a single punctuation mark. For example, the string “Bub2p-dependent” is split into 3 tokens: “Bub2p”, “-”, and “dependent”. While this simple tokenization generates a greater number of tokens than a more compact representation would, it has the advantage of being highly consistent.

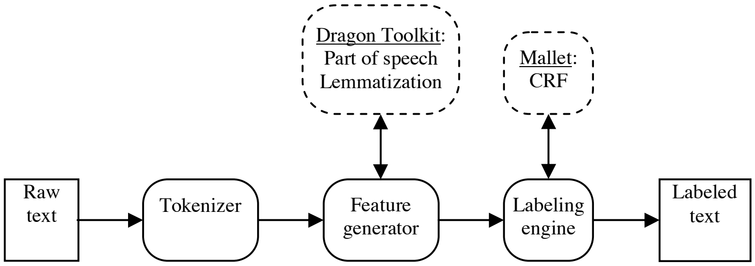

Figure 1. BANNER architecture. Raw sentences are tokenized, converted to features, and labeled. The Dragon toolkit [22] (POS) and Mallet [8] are used for part of the implementation.

BANNER uses the CRF implementation of the latest version of the Mallet toolkit (version 0.4) [8] for both feature generation and labeling using a second order CRF. The set of machine learning features used primarily consist of orthographic, morphological and shallow syntax features and is described in table 2. While many systems use some form of stemming, BANNER instead employs lemmatization [16], which is similar in purpose except that words are converted into their base form instead of simply removing the suffix. Also notable is the numeric normalization feature [15], which replaces the digits in each token with a representative digit (e.g. “0”). Numeric normalization is useful since entity names often occur in series, such as the gene names *Freac1*, *Freac2*, etc. The numeric-normalized value for all these names is *Freac0*, so that forms not seen in the training data have the same representation as forms which are seen. The entire set of features is used in conjunction with a token window of 2 to provide context, that is, the features for each token include the features for the previous two tokens and the following two tokens.

Table 2. The machine learning features used in BANNER (aside from the token itself), primarily based on orthographic, morphological and shallow syntax features.

Feature set description	Notes
The part of speech which the token plays in the sentence	Provided by the Dragon toolkit [22] implementation of the Hepple tagger [5]
The lemma for the word represented by the token, if any	Provided by the Dragon toolkit [22]
A set of regular expression features	Includes variations on capitalization and letter/digit combinations, similar to [9, 11, 15]
2, 3 and 4-character prefixes and suffixes	
2 and 3 character n-grams	Including start-of-token and end-of-token indicators
Word class	Convert upper-case letters to “A”, lower-case letters to “a”, digits to “0” and other characters to “x” [11]
Numeric normalization	Convert digits to “0” [15]
Roman numerals	
The names of the Greek letters	

There are features discussed in the literature which are not implemented in BANNER, particularly *semantic features* such as a match to a dictionary of names and *deep syntactic features*, such as information derived from a full parse of each sentence. Semantic features generally have a positive impact on overall performance [20] but often have a deleterious effect on recognizing entities not in the dictionary [11, 21]. Moreover, employing a dictionary reduces the flexibility of the system to be adapted to other entity types, since comparable performance will only be achieved after the creation of a comparable dictionary. While such application-specific performance increases are not the purpose of a system such as BANNER, this is an excellent example of an adaptation which researchers may easily perform to improve BANNER's performance for a specific domain.

Deep syntactic features are derived from a full parse of the sentence, which is a noisy and resource-intensive operation with no guarantee that the extra information derived will outweigh the additional errors generated [6]. The use of deep syntactic features in biomedical named entity recognition systems is not currently common, though they have been used successfully. One example is the system submitted by Vlachos to BioCreative 2 [16], where features derived from a full syntactic parse boosted the overall F-score by 0.51.

Unlike many similar-performing systems, BANNER does not employ rule-based post-processing steps. Rules created for one corpus tend to not generalize well to other corpora [6]. Not using such methods therefore enhances the flexibility of the system and simplifies the process of employing it on different corpora or for other entity types [9].

There are, however, two types of general post-processing which have good support in the literature and are sufficiently generic to be applicable to any biomedical text. The first of these is detecting when matching parenthesis, brackets or double quotation marks receive different labels [4]. Since these punctuation marks are always paired, detecting this situation is useful because it clearly demonstrates that the labeling engine has made a mistake. BANNER implements this form of processing by dropping any mention which contains mismatched parenthesis, brackets or double quotation marks.

The second type of generally-applicable post-processing is called abbreviation resolution [21]. Authors of biomedical papers often introduce an abbreviation for an entity by using a format similar to "antilymphocyte globulin (ALG)" or "ALG (antilymphocyte globulin)". This format can be detected with a high degree of accuracy by a simple algorithm [12], which then triggers additional processing to ensure that both mentions are recognized. The implementation of this form of post-processing is left as future work.

Extending BANNER for use in a specialized context or for testing new ideas is straightforward since the majority of the complexity in the implementation resides in the conversion of the data between different formats. For instance, most of the upgrades above the initial implementation (described in the next section) required only a few lines of code. Configuration settings are provided for the common cases, such as changing the order of the CRF model or adding a dictionary of terms.

4. Analysis

BANNER was evaluated with respect to the training corpus for the BioCreative 2 GM task, which contains 15,000 sentences from MEDLINE abstracts and mentions over 18,000 entities. The evaluation was performed by comparing the system output to the human-annotated corpus in terms of the precision (p), recall (r) and their harmonic mean, the F-measure (F). These are based on the number of true positives (TP), false positives (FP) and false negative (FN) returned by the system:

$$p = \frac{TP}{TP+FP} \qquad r = \frac{TP}{TP+FN} \qquad F = \left(\frac{p^{-1}+r^{-1}}{2}\right)^{-1} = \frac{2pr}{p+r}$$

The entities in the BioCreative 2 GM corpus are annotated at the individual character level, and approximately 56% of the mentions have at least one alternate mention annotated, and mentions are considered a true positive if they exactly match either the main annotation or any of the alternates. The evaluation of BANNER was performed using 5x2 cross-validation, which Dietterich shows to be more powerful than the more common 10-fold cross validation [3]. Differences in the performance reported are therefore more likely to be due to a real difference in the performance of the two systems rather than a chance favorable splitting of the data.

The initial implementation of BANNER included only a naïve tokenization which always split tokens at letter/digit boundaries and employed a 1st-order CRF. This implementation was improved by changing the tokenization to not split tokens at the letter/digit boundaries, changing the CRF order to 2, implementing parenthesis post-processing and adding lemmatization, part-of-speech and numeric normalization features. Note that both the initial and final implementations employed the *IOB* label model. In table 3 we present evaluation results for the initial and final implementations, as well as several system variants created by removing a single improvement from the final implementation.

Table 3. Results of evaluating the initial version of the system, the final version, and several system variants created by removing a single improvement from the final implementation.

BANNER System Variant	**Precision (%)**	**Recall (%)**	**F-Measure**
Initial implementation	82.39	76.21	79.18
Final implementation	**85.09**	**79.06**	**81.96**
With IO model instead of IOB	84.71	79.40	81.96
Without numeric normalization	84.56	79.09	81.74
With IOBEW model instead of IOB	85.46	78.15	81.64
Without parenthesis post-processing	84.05	79.27	81.59
Using 1st order CRF instead of 2nd order	84.49	78.72	81.50
With splitting tokens between letters and digits	84.54	78.35	81.33
Without lemmatization	84.44	78.00	81.09
Without part-of-speech tagging	84.02	77.83	80.81

The only system variant which had similar overall performance was the *IO* model, due to an increase in recall. This setting was not retained in the final implementation, however, due to the fact that the *IO* model cannot distinguish between adjacent entities. All other modifications result in decreased overall performance, demonstrating that each of the improvements employed in the final implementation contributes positively to the overall performance.

5. Comparison

We compare the performance of BANNER against the existing freely-available systems in use, we compare its performance against ABNER [11] and LingPipe [1], chosen because they are the most commonly used baseline systems in the literature [17, 19]. The evaluations are performed using 5x2 cross validation using the BioCreative 2 GM task training corpus, and reported in table 4. To demonstrate portability we also perform an evaluation using 5x2 cross validation on the disease mentions of the BioText disease-treatment corpus [10]. These results are reported in table 5. We believe that the relatively low performance of all three systems on the BioText corpus is due to the small size (3655 sentences) and the fact that no alternate mentions are provided.

Table 4. Results of comparing BANNER against existing freely-available software, using 5x2 cross-validation on the BioCreative 2 GM task training corpus.

System	Precision (%)	Recall (%)	F-Measure
BANNER	85.09	79.06	81.96
ABNER [11]	83.21	73.94	78.30
LingPipe [1]	60.34	70.32	64.95

Table 5. Results of comparing BANNER against existing freely-available software, using 5x2 cross-validation on the disease mentions from the BioText disease/treatment corpus [10].

System	Precision (%)	Recall (%)	F-Measure
BANNER	68.89	45.55	54.84
ABNER [11]	66.08	44.86	53.44
LingPipe [1]	55.41	47.50	51.15

Like BANNER, ABNER is also based on conditional random fields; however it uses a 1^{st}-order model and employs a feature set which lacks part-of-speech, lemmatization and numeric normalization features. In addition, it does not employ any form of post-processing, though it does use the same *IOB* label model. ABNER employs a more sophisticated tokenization than BANNER, however this tokenization is incorrect for 5.3% of the mentions in the BioCreative 2 GM task training corpus.

LingPipe is a well-developed commercial platform for various information extraction tasks that has been released free-of-charge for academic use. It is based on a 1^{st}-order Hidden Markov Model with variable-length n-grams as the sole feature set and uses the *IOB* label model for output. It has two primary configuration settings, the maximum length of n-grams to use and whether to use smoothing. For the evaluation we tested all combinations of max n-gram={4…9} and smoothing={*true*, *false*} and found that the difference between the maximum and the minimum performance was only 2.02 F-measure. The results reported here are for the maximum performance, found at max n-gram=7 and smoothing=*true*. Notably, LingPipe requires significantly less training time than either BANNER or ABNER.

The large number of systems (21) which participated in the BioCreative 2 GM task in October of 2006 provides a good basis for comparing BANNER to the state of the art in biomedical named entity recognition. Unfortunately, the official evaluations for these systems used a test corpus that has not yet been made publicly available. The conservative 5x2 cross-validation used for evaluating BANNER still allows a useful direct comparison, however, since BANNER achieves higher performance than the median system in the official BioCreative results, even with a significant handicap against it: the BioCreative systems were able to train on the entire training set (15,000 sentences) while BANNER was only trained on half of the training set (7,500 sentences) because the other half was needed for testing. These results are reported in table 6.

Table 6. Comparison of BANNER to select BioCreative 2 systems [19]. A difference in the F-measure of 1.23 or more is significant and a difference of 0.35 or less is not ($p < 0.05$).

System or author	Rank at BioCreative 2	Precision (%)	Recall (%)	F-Measure
Ando [19]	1	88.48	85.97	87.21
Vlachos [16, 19]	9	86.28	79.66	82.84
BANNER	–	85.09	79.06	81.96
Baumgartner et. al. [19]	11 (median)	85.54	76.83	80.95
NERBio [15, 19]	13	92.67	68.91	79.05

Unlike BANNER, most of the systems submitted to BioCreative 2 were competitive systems employing features or post-processing rules specific to genes [19], a notable exception being the system submitted by Vlachos [16]. The results reported for those systems may therefore not generalize to other entity types or corpora. Moreover the authors are unaware of any of the BioCreative 2 GM systems being publicly available, as of July 2007, except for NERBio [15], which is available for limited manual testing over the Internet*, but not for download.

6. Conclusion & Future Work

We have shown that BANNER, an executable survey of advances in named entity recognition, achieves significantly better performance than existing open-source systems. This is accomplished using features and techniques which are well-supported in the more recent literature. In addition to confirming the value of these techniques and indicating that the field of biomedical named entity recognition is making progress, this work demonstrates that there are sufficient known techniques in the field to achieve good results using known techniques.

We anticipate that this system will be valuable to the biomedical NER community both by providing a benchmark level of performance for comparison and also by providing a platform upon which more advanced techniques can be built. We also anticipate that this work will be immediately useful for information extraction experiments, possibly by including minimal extensions such as a dictionary of names of types of entities to be found.

Future work for BANNER includes several general techniques which have good support in the literature but have not yet been incorporated. For example, authors have noted that part-of-speech systems trained on biomedical text gives superior performance to taggers such as the Hepple tagger which are not specifically intended for biomedical text [6]. We performed one experiment using the Dragon toolkit implementation of the MedPost POS tagger [13], which resulted in slightly improved precision (+0.18%), but significantly lower recall (-1.44%), degrading overall performance by 0.69 F-measure. We plan to test other taggers trained on biomedical text and anticipate achieving a small improvement to the overall performance.

A second technique which has strong support in the literature but is not yet implemented in BANNER is feature induction [7, 9, 15]. Feature induction is the creation of new compound features by forming a conjunction between adjacent singleton features. For example, knowing that the current token contains capital letters, lower-case letters and digits (a singleton pattern probably indicating an acronym) *and* knowing that next token is “gene” is a

* http://140.109.19.166/BioNER

stronger indication that the current token is part of a gene mention than either fact alone. Feature induction employs feature selection during training to automatically discover the most useful conjunctions, since the set of all conjunctions of useful length is prohibitively large. While this significantly increases the amount of time and resources required for training, McDonald & Pereira [9] report an increase in the overall performance of their system by 2% F-measure and we anticipate BANNER would experience a similar improvement.

Acknowledgements

The authors wish to thank Jörg Hackenburg for helpful discussions and for suggesting the BioText corpus. The authors would also like to thank the anonymous reviewers for many useful suggestions.

References

1. Baldwin, B.; and B. Carpenter. LingPipe. *http://www.alias-i.com/lingpipe/*
2. Chen, L.; H. Liu; and C. Friedman. (2005) Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics* 21, pp 248-255.
3. Dietterich, T. (1998) Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10, pp. 1895-1923.
4. Dingare, S.; et al. (2005) A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations. *Comparative and Functional Genomics* 6, pp. 77-85.
5. Hepple, M. (2000) Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, Hong Kong.
6. Leser, U.; and J. Hakenberg. (2005) What makes a gene name? Named entity recognition in the biomedical literature. *Briefings in Bioinformatics* 6, pp. 357-369.
7. McCallum, A. (2003) Efficiently Inducing Features of Conditional Random Fields. *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, San Francisco, California, pp. 403-441.
8. McCallum, Andrew. (2002) MALLET: A Machine Learning for Language Toolkit. *http://mallet.cs.umass.edu*
9. McDonald, R.; and F. Pereira. (2005) Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics* 6 (Suppl. 1):S6.

10. Rosario, B.; M. A. Hearst. (2004) Classifying Semantic Relations in Bioscience Text. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*
11. Settles, B. (2004) Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. *Proceedings of the COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP).*
12. Schwartz, A.S.; and Hearst, M.A. (2003) A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. *PSB 2003* pp 451-462.
13. Smith, L.; T. Rindflesch; and W.J. Wilbur. (2004) MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics* 20, pp. 2320-2321.
14. Sutton, C.; and A. McCallum. (2007) An Introduction to Conditional Random Fields for Relational Learning. *Introduction to Statistical Relational Learning*, MIT Press.
15. Tsai, R.; et al. (2006) NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. *BMC Bioinformatics* 7 (Suppl. 5):S11.
16. Vlachos, A. (2007) Tackling the BioCreative 2 gene mention task with Conditional random fields and syntactic parsing. *Proceedings of the Second BioCreative Challenge Workshop* pp. 85-87.
17. Vlachos, A.; C. Gasperin; I. Lewin; and T. Briscoe. (2006) Bootstrapping the recognition and anaphoric linking of named entities in Drosophila articles. *PSB 2006* pp. 100-111.
18. Wallach, H.M. (2004) Conditional Random Fields: An Introduction. University of Pennsylvania CIS Technical Report MS-CIS-04-21.
19. Wilbur, J.; L. Smith; and T. Tanabe. (2007) BioCreative 2. Gene Mention Task. *Proceedings of the Second BioCreative Challenge Workshop* pp. 7-16.
20. Yeh, A.; A. Morgan; M. Colosimo; and L. Hirschman. (2005) BioCreAtIvE Task 1A: gene mention finding evaluation. *BMC Bioinformatics* 6 (Suppl. 1):S2.
21. Zhou, G.; et al. (2005) Recognition of protein/gene names from text using an ensemble of classifiers. *BMC Bioinformatics* 6 (Suppl 1):S7.
22. Zhou, X.; X. Zhang; and X. Hu. (2007) Dragon Toolkit: Incorporating Auto-learned Semantic Knowledge into Large-Scale Text Retrieval and Mining. *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI).*