Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets

Burr Settles
Department of Computer Sciences
Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
Madison, WI, USA
bsettles@cs.wisc.edu

1 Introduction

As the wealth of biomedical knowledge in the form of literature increases, there is a rising need for effective natural language processing tools to assist in organizing, curating, and retrieving this information. To that end, named entity recognition (the task of identifying words and phrases in free text that belong to certain classes of interest) is an important first step for many of these larger information management goals.

In recent years, much attention has been focused on the problem of recognizing gene and protein mentions in biomedical abstracts. This paper presents a framework for simultaneously recognizing occurrences of `PROTEIN`, `DNA`, `RNA`, `CELL-LINE`, and `CELL-TYPE` entity classes using Conditional Random Fields with a variety of traditional and novel features. I show that this approach can achieve an overall F_1 measure around 70, which seems to be the current state of the art.

The system described here was developed as part of the BioNLP/NLPBA 2004 shared task. Experiments were conducted on a training and evaluation set provided by the task organizers.

2 Conditional Random Fields

Biomedical named entity recognition can be thought of as a sequence segmentation problem: each word is a token in a sequence to be assigned a label (e.g. `PROTEIN`, `DNA`, `RNA`, `CELL-LINE`, `CELL-TYPE`, or `OTHER`[1]). Conditional Random Fields (CRFs) are undirected statistical graphical models, a special case of which is a linear chain that corresponds to a conditionally trained finite-state machine. Such models are well suited to sequence analysis, and CRFs in particular have been shown to be useful in part-of-speech tagging (Lafferty et al., 2001), shallow parsing (Sha and Pereira, 2003), and named entity recognition for newswire data (McCallum and Li, 2003). They have also just recently been applied to the more limited task of finding gene and protein mentions (McDonald and Pereira, 2004), with promising early results.

Let $\mathbf{o} = \langle o_1, o_2, \ldots, o_n \rangle$ be an sequence of observed words of length n. Let S be a set of states in a finite state machine, each corresponding to a label $l \in L$ (e.g. `PROTEIN`, `DNA`, etc.). Let $\mathbf{s} = \langle s_1, s_2, \ldots, s_n \rangle$ be the sequence of states in S that correspond to the labels assigned to words in the input sequence $\mathbf{o}$. Linear-chain CRFs define the conditional probability of a state sequence given an input sequence to be:

$$P(\mathbf{s}|\mathbf{o}) = \frac{1}{Z_o} \exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(s_{i-1}, s_i, o, i)\right)$$

where Z_o is a normalization factor of all state sequences, $f_j(s_{i-1}, s_i, o, i)$ is one of m functions that describes a feature, and λ_j is a learned weight for each such feature function. This paper considers the case of CRFs that use a first-order Markov independence assumption with binary feature functions. For example, a feature may have a value of 0 in most cases, but given the text "the ATPase" it has the value 1 along the transition where s_{i-1} corresponds to a state with the label `OTHER`, s_i corresponds to a state with the label `PROTEIN`, and f_j is the feature function WORD=$ATPase \in o$ at position i in the sequence. Other feature functions that could have the value 1 along this transition are CAPITALIZED, MIXEDCASE, and SUFFIX=*ase*.

Intuitively, the learned feature weight λ_j for each feature f_j should be positive for features that are correlated with the target label, negative for features that are anti-correlated with the label, and near zero for relatively uninformative features. These weights are

[1] More accurately, the data is in IOB format. `B-DNA` labels the first word of a `DNA` mention, `I-DNA` labels all subsequent words (likewise for other entities), and `O` labels non-entities. For simplicity, this paper only refers to the entities, not all the IOB label variants.

set to maximize the conditional log likelihood of labeled sequences in a training set $D = \{\langle \mathbf{o}, \mathbf{l} \rangle_{(1)}, \ldots, \langle \mathbf{o}, \mathbf{l} \rangle_{(n)}\}$:

$$LL(D) = \sum_{i=1}^{n} \log \left(P(\mathbf{l}_{(i)} | \mathbf{o}_{(i)}) \right) - \sum_{j=1}^{m} \frac{\lambda_j^2}{2\sigma^2}.$$

When the training state sequences are fully labeled and unambiguous, the objective function is convex, thus the model is guaranteed to find the optimal weight settings in terms of $LL(D)$. Once these settings are found, the labeling for an new, unlabeled sequence can be done using a modified Viterbi algorithm. CRFs are presented in more complete detail by Lafferty et al. (2001).

These experiments use the MALLET implementation of CRFs (McCallum, 2002), which uses a quasi-Newton method called L-BFGS to find these feature weights efficiently.

3 Feature Set

One property that makes feature based statistical models like CRFs so attractive is that they reduce the problem to finding an appropriate feature set. This section outlines the two main types of features used in these experiments.

3.1 Orthographic Features

The simplest and most obvious feature set is the vocabulary from the training data. Generalizations over how these words appear (e.g. capitalization, affixes, etc.) are also important. The present model includes training vocabulary, 17 orthographic features based on regular expressions (e.g. ALPHANUMERIC, HASDASH, ROMANNUMERAL) as well as prefixes and suffixes in the character length range [3,5].

Words are also assigned a generalized "word class" similar to Collins (2002), which replaces capital letters with 'A', lowercase letters with 'a', digits with '0', and all other characters with '_'. There is a similar "brief word class" feature which collapses consecutive identical characters into one. Thus the words "IL5" and "SH3" would both be given the features WC=*AA0* and BWC=*A0*, while "F-actin" and "T-cells" would both be assigned WC=*A_aaaaa* and BWC=*A_a*.

To model local context simply, neighboring words in the window [-1,1] are also added as features. For instance, the middle token in the sequence "human UDG promoter" would have features WORD=*UDG*, NEIGHBOR=*human* and NEIGHBOR=*promoter*.

3.2 Semantic Features

In addition to orthography, the model could also benefit from generalized semantic word groups. If training sequences contain "PML/RAR alpha," "beta 2-M," and "kappa B-specific DNA binding protein" all labeled with `PROTEIN`, the model might learn that the words "alpha," "beta," and "kappa" are indicative of proteins, but cannot capture the fact that they are all semantically related because they are Greek letters. Similarly, words with the feature WC=*Aaa* are often part of protein names, such as "Rab," "Alu," and "Gag." But the model may have a difficult time setting the weights for this feature when confronted with words like "Phe," "Arg," and "Cys," which are amino acid abbreviations and not often labeled as part of a protein name.

This sort of semantic domain knowledge can be provided in the form of lexicons. I prepared a total of 17 such lexicons, which include 7 that were entered by hand (Greek letters, amino acids, chemical elements, known viruses, plus abbreviations of all these), and 4 corresponding to genes, chromosome locations, proteins, and cell lines, drawn from online public databases (Cancer GeneticsWeb,[2] BBID,[3] SwissProt,[4] and the Cell Line Database[5]). Feature functions for the lexicons are set to 1 if they match words in the input sequence exactly. For lexicon entries that are multi-word, all words are required to match in the input sequence.

Since no suitable database of terms for the `CELL-TYPE` class was found online, a lexicon was constructed by utilizing Google Sets,[6] an online tool which takes a few seed examples and leverages Google's web index to return other terms that appear in similar formatting and context as the seeds on web pages across the Internet. Several examples from the training data (e.g. "lymphocyte" and "neutrophil") were used as seeds and new cell types (e.g. "chondroblast," which doesn't even occur in the training data), were returned. The process was repeated until the lexicon grew to roughly 50 entries, though it could probably be more complete.

With all this information at the model's disposal, it can still be difficult to properly disambiguate between these entities. For exam-

[2] http://www.cancerindex.org/geneweb/
[3] http://bbid.grc.nia.nih.gov/bbidgene.html
[4] http://us.expasy.org/sprot/
[5] http://www.biotech.ist.unige.it/interlab/cldb.html
[6] http://labs.google.com/sets

ple, the acronym "EPC" appears in these static lexicons both as a protein ("eosinophil cationic protein" [sic]) and as a cell line ("epithelioma papulosum cyprini"). Furthermore, a single word like "transcript" is sometimes all that disambiguates between `RNA` and `DNA` mentions (e.g. "BMLF1 *transcript*"). The CRF can learn weights for these individual words, but it may help to build general, dynamic keyword lexicons that are associated with each label to assist in disambiguating between similar classes (and perhaps boost performance on low-frequency labels, such as `RNA` and `CELL-LINE`, for which training data are sparse).

These keyword lexicons are generated automatically as follows. All of the labeled terms are extracted from the training set and separated into five lists (one for each entity class). Stop words, Greek letters, and digits are filtered, and remaining words are tallied for raw frequency counts under each entity class label. These frequencies are then subjected to a χ^2 test, where the null hypothesis is that a word's frequency is the same for a given entity as it is for any other entity of interest (i.e. `PROTEIN` vs. `DNA + RNA + CELL-LINE + CELL-TYPE`, such that there is only one degree of freedom). All words for which the null hypothesis is rejected with a p-value < 0.005 are added to the keyword lexicon for its majority class. Some example keywords are listed in table 1.

Keyword	χ^2 value	Lexicon
protein	1121.5	`PROTEIN`
gene	984.3	`DNA`
line	618.1	`CELL-LINE`
promoter	613.4	`DNA`
factor	563.2	`PROTEIN`
site	399.8	`DNA`
receptor	338.7	`PROTEIN`
complex	312.8	`PROTEIN`
mRNA	292.2	`RNA`
sequence	196.5	`DNA`
peripheral	57.8	`CELL-TYPE`
lineage	56.1	`CELL-TYPE`
jurkat	45.2	`CELL-LINE`
culture	41.3	`CELL-LINE`
transcript	40.9	`RNA`
clone	38.1	`CELL-LINE`
mononuclear	30.2	`CELL-TYPE`
messenger	12.3	`RNA`

Table 1: A sample of high-ranking semantic keywords and the lexicons to which they belong.

Orthographic Features Only					
Entity	R	P	F_1	L-F_1	R-F_1
`PROTEIN`	76.3	68.4	72.1	77.4	79.2
`DNA`	62.4	68.2	65.2	68.5	73.8
`RNA`	61.9	62.9	62.4	64.9	75.2
`CELL-LINE`	53.8	54.0	53.9	58.5	65.1
`CELL-TYPE`	63.6	78.5	70.3	72.6	80.4
Overall	70.3	69.3	69.8	74.2	77.9

Complete Feature Set					
Entity	R	P	F_1	L-F_1	R-F_1
`PROTEIN`	76.1	68.2	72.0	77.3	79.2
`DNA`	62.1	67.9	64.9	67.7	74.1
`RNA`	65.3	64.2	64.7	66.4	73.9
`CELL-LINE`	57.4	54.1	55.7	59.2	64.2
`CELL-TYPE`	61.7	78.4	69.1	71.3	79.7
Overall	70.0	69.0	69.5	73.7	77.7

Table 2: Detailed performance of the two features sets. Relaxed F_1-scores using left- and right-boundary matching are also reported.

4 Results and Discussion

Two experiments were completed in the time allotted: one CRF model using only the orthographic features described in section 3.1, and a second system using all the semantic lexicons from 3.2 as well. Detailed results are presented in table 2. The orthographic model achieves an overall F_1 measure of 69.8 on the evaluation set (88.9 on the training set), converging after 230 training iterations and approximately 18 hours of computation. The complete model, however, only reached an overall F_1 of 69.5 on the evaluation set (86.7 on the training set), converging after 152 iterations in approximately 9 hours.

The deleterious effect of the semantic lexicons is surprising and puzzling.[7] However, even though semantic lexicons slightly decrease overall performance, it is worthwhile to note that adding lexicons actually improves both recall and precision for the `RNA` and `CELL-LINE` entities. These happen to be the two lowest frequency class labels in the data, together comprising less than 10% of the mentions in either the training or evaluation set. Error analysis shows that several of the orthographic model's false negatives for these entities are of the form "*messenger* accumulation" (`RNA`) or "nonadherent *culture*" (`CELL-LINE`). It may be that keyword lexicons contributed to the model identifying these low frequency terms more accurately.

[7] Note, however, that these figures are on a single training/evaluation split without cross-validation, so differences are likely not statistically significant.

Also of note is that, in both experiments, the CRF framework achieves somewhat comparable performance across all entities. In a previous attempt to use a Hidden Markov Model to simultaneously recognize multiple biomedical entities (Collier et al., 2000), HMM performance for a particular entity seemed more or less proportional to its frequency in the data. The advantage of the CRF here may be due to the fact that HMMs are generative models trained to learn the joint probability $P(\mathbf{o}, \mathbf{l})$ — where data for $\mathbf{l}$ may be sparse — and use Bayes rule to predict the best label. CRFs are discriminative models trained to maximize $P(\mathbf{l}|\mathbf{o})$ directly.

5 Conclusions and Future Work

In short, I have presented in detail a framework for recognizing multiple entity classes in biomedical abstracts with Conditional Random Fields. I have shown that a CRF-based model with only simple orthographic features can achieve performance near the current state of the art, while using semantic lexicons (as presented here) do not positively affect performance.[8]

While the system presented here shows promise, there is still much to be explored. Richer syntactic information such as shallow parsing may be useful. The method introduced in section 3.2 to generate semantic keywords can also be adapted to generate features for entity-specific morphology (e.g. affixes) and context, both linearly (e.g. neighboring words) and hierarchically (e.g. from a parse).

Most interesting, though, might be to investigate why the lexicons do not generally help. One explanation is simply an issue of tokenization. While one abstract refers to "IL12," others may write "IL-12" or "IL 12." Similarly, the generalization of entities to groups (e.g. "x antibody" vs. "x antibodies") can cause problems for these rigid lexicons that require exact matching. Enumerating all such variants for every entry in a lexicon is absurd. Perhaps relaxing the matching criteria and standardizing tokenization for both the input and lexicons will improve their utility.

[8] More recent work (not submitted for evaluation) indicates that lexicons are indeed useful, but mainly when training data are limited. I have also found that using orthographic features with part-of-speech tags and only the `RNA` and `CELL-LINE` (rare class) lexicons can boost overall F_1 to 70.3 on the evaluation data, with particular improvements for the `RNA` and `CELL-LINE` entities.

Acknowledgements

I would like to thank my advisor Mark Craven for his advice and guidance, as well as Andrew McCallum and Aron Culotta for answering my questions about the MALLET system. This work is supported by NLM training grant 5T15LM007359-02 and NIH grant R01 LM07050-01.

References

Nigel Collier, Chikashi Nobata, and Jun ichi Tsujii. 2000. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the International Conference on Computational Linguistics*, pages 201–207. Saarbrucken, Germany.

Michael Collins. 2002. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the Association for Computational Linguistics Conference*, pages 489–496. Philadelphia, USA.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*. Williamstown, MA, USA.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Conference on Natural Language Learning*, pages 188–191. Edmonton, Canada.

Andrew McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Ryan McDonald and Fernando Pereira. 2004. Identifying gene and protein mentions in text using conditional random fields. In *Proceedings of BioCreative: Critical Assessment for Information Extraction in Biology*. Grenada, Spain.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*. Edmonton, Canada.