Entity identification in the molecular biology domain with a stochastic POS tagger: the BioCreative task

Shuhei Kinoshita[a,b] **Philip Ogren**[c,b]

K. Bretonnel Cohen[a] **Lawrence Hunter**[a,d]

1. Overview

Our approach to Task 1A was inspired by Tanabe and Wilbur's ABGene system, described in Tanabe and Wilbur (2002a and 2002b). Like Tanabe and Wilbur, we approached the problem as one of POS tagging, adding a GENE tag to the standard tag set. Where their system uses the Brill tagger, we used TnT, the *Trigrams 'n' Tags* HMM-based POS tagger described in Brants 2000. We also made use of Schwartz and Hearst's (2003) algorithm for abbreviation expansion. We implemented a set of post-processing rules to account for the specifics of the BioCreative task definition. We participated in both the "open" and the "closed" divisions; for the "open" division, we made use of data from NCBI.

2. The tagger

Past experience with the ABGene system in our lab suggested that the POS-tagging-based approach to entity identification is workable in the molecular biology domain. We noted some problems with the ABGene system that we felt were due to the Brill tagger that forms its heart, and hypothesized that for these types of problems, an HMM tagger might provide better results. Previous experiments with the TnT *Trigrams 'n' Tags* POS tagger, using the GENIA corpus for cross-validation, showed good results with no post-processing of the output. The TnT system is a stochastic POS tagger, described in detail in Brants (2000). It uses a second-order Markov model with tags as states and words as outputs. Transitions are defined over tags; outputs are predicted from the "most recent category." The probability of emitting a tag is calculated by:

$$\underset{t_1 \ldots t_T}{\operatorname{argmax}} \prod_{i=1}^{T} P(t_i|t_{i-1}, t_{i-2})\, P(w_i|t_i))\, P(t_{T+1}|t_T) \qquad (1)$$

Smoothing is by linear interpolation of uni-, bi-, and tri-grams, with λ estimated by deleted interpolation. Unknown words are handled by learning tag probabilities for word endings. As a POS tagger, the system has been tested on two languages, viz. English and German. It is publicly available at http://www.coli.uni-sb.de/~thorsten/tnt/. We were impressed by its availability on a variety of platforms, its intuitive interface, and the stability of its distribution, which installed easily and never crashed. We trained it on the full training corpus and tested it on the devtest data set. Performance of this system on the devtest data set, calculated by the BioCreative scoring software, was P = 67.8, R = 76.1, and F-measure = 71.7.

3. Post-processing of the tagger's output

We applied a number of post-processing procedures to the output of the tagger. Some of these were designed to deal with problems that arise on any definition of the problem, such as abbreviations and unknown words; others specifically address the BioCreative problem definition. These post-processing steps led to an increase from the level of performance of the tagger alone (P = 67.8, R = 76.1, and F-measure = 82.6) to P = 82.6, R = 82.5, and F-measure = 82.6 in the "closed" division, and P = 82.6, R = 83.5, and F-measure = 83.1 in the "open" division.

[a] Center for Computational Pharmacology, University of Colorado School of Medicine, Denver, Colorado
[b] Fujitsu Ltd., Bio-IT Laboratory, Chiba City, Japan
[c] Dept. of Computer Science, University of Colorado at Boulder, Boulder, CO
[d] Lawrence.Hunter@uchsc.edu, to whom correspondence should be addressed

3.1 Abbreviations

The tagger would sometimes recognize a full gene name but not its appositive parenthesized symbol/abbreviation, or vice versa. We implemented Schwartz and Hearst's (2003) algorithm to recognize abbreviations and their appositive definitions, such as *Insulin-like growth factor 1 (IGF-1)*. When one but not the other was tagged as GENE, we added the gene tag to the un-GENE-tagged member of the definition/abbreviation pair.

3.2 Rule-based post-processing

We used a number of rules to fix cases where the BioCreative task definition specified that a gene name should extend further to the right than the TnT tagger thought it should.

3.2.1 "Keywords"

If a word tagged as GENE is followed by a word such as *gene*, *mutant*, etc., and the following word is not tagged as GENE, then the tag on the following word is changed to GENE.

3.2.2 Numbers and Greek letters

If a word tagged as GENE is followed by a number, a Roman numeral, or a Greek letter, and that following number or letter is not tagged as GENE, then its tag is changed to GENE.

3.2.3 Parentheses

If a word is tagged as GENE and it is followed by a 5-character-or-shorter stretch of parenthesized material, and that parenthesized material is not tagged as GENE, then its tag is changed to GENE.

3.3 Statistically-based post-processing

We used a small set of rules based on distributions of words in name-initial and name-final positions to modify the boundaries of multi-word gene names on the right and left edges.

3.4 Dictionary-based post-processing in the "open" division

In the "open" division, we made use of data from NCBI. We applied this data just in cases where:

- A word was not found in the statistical model, and
- …it was tagged as a noun, and
- …it was four characters or greater in length.

We first looked for such words in LocusLink symbol fields. If we found it in a LocusLink symbol field, then we tagged it as GENE. If we did not find it in a LocusLink symbol field, then we queried the NCBI website through Entrez, specifying `db=nucleotide` and restricting our search to the gene name field. If any items were returned by Entrez, then we tagged the word as GENE.

4. Results on training and devtest data

4.1 Overall

We did five rounds of cross-validation, training on four subsets of the data and testing on a fifth. We evaluated our results using the scoring software provided with the BioCreative data. The resulting average precision and recall were .68 and .77 without post-processing (i.e. just based on the output of the tagger). The resulting average precision and recall were .82 and .81 with post-processing. The averaged results of the cross-validation runs are shown in Figure 1.

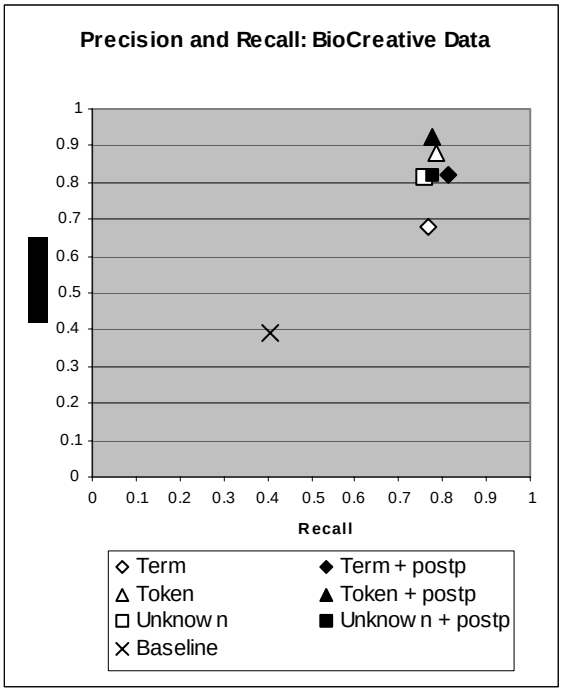

Figure 1 Precision and recall for the BioCreative training and devtest data.

4.2 Term-level precision and recall

Term-level scores (i.e., for performance on full gene names, analogous to the *strict* metric of Olsson et al. 2002) were obtained using the BioCreative scoring software. We evaluated performance both with and without post-processing. Without performing post-processing, average precision and recall were .68 and .77. When we then applied rule-based post-processing as described in 4.3 *Postprocessing the BioCreative output* above, average precision and recall were .82 and .81. Post-processing improved both the precision and the recall, having a much larger effect on precision than on recall.

4.3 Baseline, and normalizing for the difficulty of the task

As a baseline for understanding the difficulty of the task, we determined the performance that would be achieved by simply assigning each word the most frequent tag seen with that word in the training set. This baseline strategy achieved an average precision of .39 and an average recall of .41—considerably worse than even our without-post-processing results.

4.4 Per-token precision and recall

We then determined the results on a per-word basis. This is equivalent to Olsson et al.'s *protein name parts* metric. As would be expected, performance on single words is better than the term-level results, with an average precision of .88 and average recall of .79 without post-processing, and an average precision of .92 and average recall of .78 with post-processing. Post-processing yielded some improvement in precision, although not of the magnitude observed for full gene names. It actually degraded recall somewhat.

4.5 Performance on unknown words

For unknown words, average precision was .81 and average recall was .76 without post-processing. Average precision was .82 and average recall was .78 with post-processing. Post-processing yielded no improvement in performance for unknown words.

4.6 Effect of term length on performance

Figure 4 shows the effect of term length on precision and recall. Again, there is no drastic drop in performance until names reach a length of 5 or more words.

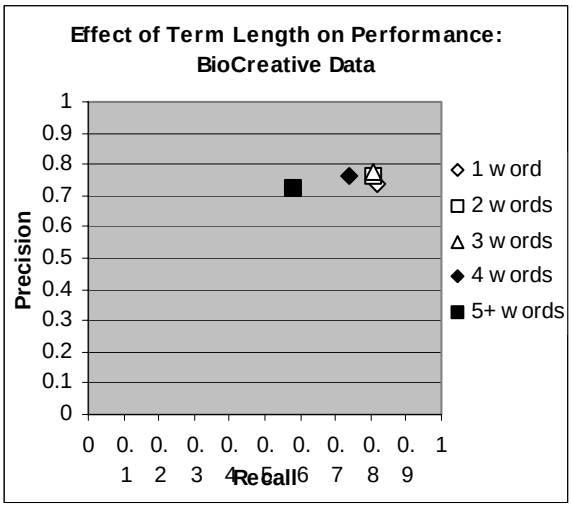

Figure 2 Effect of term length on performance for the training and devtest data.

4.7 Overall effects of rule-based post-processing

The main effect of post-processing is an increase in precision. For full gene names, average precision increased from .68 to .82, and average recall increased from .77 to .81. On the level of individual words, post-processing had a much smaller, and not always positive, effect.

5 Final scores on the test data

Table 1 shows the results on the official test data. It conforms closely to the results for our cross-validation runs on the training and devtest data.

	TP	FP	FN	P	R	F
C	4767	1161	1182	.804	.801	.803
O	4840	1187	1109	.803	.814	.808
C	4858	1208	1091	.801	.817	.809

O	4867	1213	1082	.800	.818	.809

4 Conclusion

The POS-tagging-based approach that we took from the ABGene system worked reasonably well, considering the small amount of training data available, and our results with the GENIA corpus suggest that it is robust with respect to different corpora and different problem definitions. Post-processing rules, both pattern-based and statistical, worked well to increase both precision and recall, with the F-measure rising from 71.7 (without post-processing) to 82.6 (with post-processing) on the devtest data set. Domain-specific dictionaries were less helpful, giving an increase of only .5 in F-measure (to 83.1) compared to the post-processing-without-dictionaries approach.

References

Brants, Thorsten. 2000. TnT - A Statistical Part-of-Speech Tagger. *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000).*

Olsson, Fredrik; Gunnar Eriksson; Kristofer Franzén; Lars Asker; and Per Lidén (2002). Notions of correctness when evaluating protein name taggers. *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp. 765-771.

Schwartz, Ariel S., and Marti A. Hearst. 2003. A Simple Algorithm For Identifying Abbreviation Definitions in Biomedical Text. *Proceedings of the Pacific Symposium on Biocomputing* 8:451-462.

Tanabe, Lorraine, and W. John Wilbur. 2002(a). Tagging gene and protein names in biomedical text. *Bioinformatics* 18(8):1124-1132.

Tanabe, Lorraine, and W. John Wilbur. 2002(b). Tagging gene and protein names in full text articles. *Proceedings of the workshop on biomedical natural language processing in the biomedical domain*, pp. 9-13. Association for Computational Linguistics.