Extracting the Names of Genes and Gene Products with a Hidden Markov Model

Nigel Collier, Chikashi Nobata and Jun-ichi Tsujii
Department of Information Science
Graduate School of Science
University of Tokyo, Hongo-7-3-1
Bunkyo-ku, Tokyo 113, Japan
E-mail:{nigel,nova,tsujii}@is.s.u-tokyo.ac.jp

Abstract

We report the results of a study into the use of a linear interpolating hidden Markov model (HMM) for the task of extracting technical terminology from MEDLINE abstracts and texts in the molecular-biology domain. This is the first stage in a system that will extract event information for automatically updating biology databases. We trained the HMM entirely with bigrams based on lexical and character features in a relatively small corpus of 100 MEDLINE abstracts that were marked-up by domain experts with term classes such as proteins and DNA. Using cross-validation methods we achieved an F-score of 0.73 and we examine the contribution made by each part of the interpolation model to overcoming data sparseness.

1 Introduction

In the last few years there has been a great investment in molecular-biology research. This has yielded many results that, together with a migration of archival material to the Internet, has resulted in an explosion in the number of research publications available in online databases. The results in these papers however are not available in a structured format and have to be extracted and synthesized manually. Updating databases such as SwissProt (Bairoch and Apweiler, 1997) this way is time consuming and means that the results are not accessible so conveniently to help researchers in their work.

Our research is aimed at automatically extracting facts from scientific abstracts and full papers in the molecular-biology domain and using these to update databases. As the first stage in achieving this goal we have explored the use of a generalisable, supervised training method based on hidden Markov models (HMMs) (Rabiner and Juang, 1986) for the identification and classification of technical expressions in these texts. This task can be considered to be similar to the *named entity* task in the MUC evaluation exercises (MUC, 1995).

In our current work we are using abstracts available from PubMed's MEDLINE (MEDLINE, 1999). The MEDLINE database is an online collection of abstracts for published journal articles in biology and medicine and contains more than nine million articles.

With the rapid growth in the number of published papers in the field of molecular-biology there has been growing interest in the application of information extraction, (Sekimizu et al., 1998)(Collier et al., 1999)(Thomas et al., 1999)(Craven and Kumlien, 1999), to help solve some of the problems that are associated with information overload.

In the remainder of this paper we will first of all outline the background to the task and then describe the basics of HMMs and the formal model we are using. The following sections give an outline of a new tagged corpus (Ohta et al., 1999) that our team has developed using abstracts taken from a sub-domain of MEDLINE and the results of our experiments on this corpus.

2 Background

Recent studies into the use of supervised learning-based models for the named entity task in the micro-biology domain have shown that models based on HMMs and decision trees such as (Nobata et al., 1999) are much more generalisable and adaptable to new classes of words than systems based on traditional hand-built patterns and domain specific heuristic rules such as (Fukuda et al., 1998), overcoming the problems associated with data sparseness with the help of sophisticated smoothing algorithms

Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000), Saarbrucken, Germany, July 31st--August 4th

(Chen and Goodman, 1996).

HMMs can be considered to be stochastic finite state machines and have enjoyed success in a number of fields including speech recognition and part-of-speech tagging (Kupiec, 1992). It has been natural therefore that these models have been adapted for use in other word-class prediction tasks such as the named-entity task in IE. Such models are often based on n-grams. Although the assumption that a word's part-of-speech or name class can be predicted by the previous n-1 words and their classes is counter-intuitive to our understanding of linguistic structures and long distance dependencies, this simple method does seem to be highly effective in practice. Nymble (Bikel et al., 1997), a system which uses HMMs is one of the most successful such systems and trains on a corpus of marked-up text, using only character features in addition to word bigrams.

Although it is still early days for the use of HMMs for IE, we can see a number of trends in the research. Systems can be divided into those which use one state per class such as Nymble (at the top level of their backoff model) and those which automatically learn about the model's structure such as (Seymore et al., 1999). Additionally, there is a distinction to be made in the source of the knowledge for estimating transition probabilities between models which are built by hand such as (Freitag and McCallum, 1999) and those which learn from tagged corpora in the same domain such as the model presented in this paper, word lists and corpora in different domains - so-called *distantly*-labeled data (Seymore et al., 1999).

2.1 Challenges of name finding in molecular-biology texts

The names that we are trying to extract fall into a number of categories that are often wider than the definitions used for the traditional named-entity task used in MUC and may be considered to share many characteristics of term recognition.

The particular difficulties with identifying and classifying terms in the molecular-biology domain are an open vocabulary and irrgeular naming conventions as well as extensive cross-over in vocabulary between classes. The irregular naming arises in part because of the number of researchers from different fields who are working on the same knowledge discovery area as well as the large number of substances that need to be named. Despite the best efforts of major journals to standardise the terminology, there is also a significant problem with synonymy so that often an entity has more than one name that is widely used. The class cross-over of terms arises because many proteins are named after DNA or RNA with which they react.

TI - Activation of <PROTEIN> JAK kinases </PROTEIN> and <PROTEIN>STAT proteins </PROTEIN> by <PROTEIN> interleukin - 2 </PROTEIN> and <PROTEIN> interferon alpha </PROTEIN> , but not the <PROTEIN> T cell antigen receptor </PROTEIN> , in <SOURCE.ct> human T lymphocytes </SOURCE.ct> .
AB - The activation of <PROTEIN> Janus protein tyrosine kinases </PROTEIN> (<PROTEIN> JAKs </PROTEIN>) and <PROTEIN> signal transducer and activator of transcription </PROTEIN> (<PROTEIN> STAT </PROTEIN>) proteins by <PROTEIN> interleukin (IL) - 2 </PROTEIN> , the <PROTEIN> T cell antigen receptor </PROTEIN> (<PROTEIN> TCR </PROTEIN>) and <PROTEIN> interferon (IFN) alpha </PROTEIN> was explored in <SOURCE.ct> human peripheral blood - derived T cells </SOURCE.ct> and the <SOURCE.cl> leukemic T cell line Kit225 </SOURCE.cl> .

Figure 1: Example MEDLINE sentence marked up in XML for biochemical named-entities.

All of the names which we mark up must belong to only one of the name classes listed in Table 1. We determined that all of these name classes were of interest to domain experts and were essential to our domain model for event extraction. Example sentences from a marked up abstract are given in Figure 1.

We decided not to use separate states for pre- and post-class words as had been used in some other systems, e.g. (Freitag and McCallum, 1999). Contrary to our expectations, we observed that our training data provided very poor maximum-likelihood probabilities for these words as class predictors.

We found that protein predictor words had the only significant evidence and even this was quite weak, except in the case of post-class words which included a number of head nouns such as "molecules" or "heterodimers". In our

Class	#	Example	Description
PROTEIN	2125	*JAK kinase*	proteins, protein groups, families, complexes and substructures.
DNA	358	*IL-2 promoter*	DNAs, DNA groups, regions and genes
RNA	30	*TAR*	RNAs, RNA groups, regions and genes
SOURCE.cl	93	*leukemic T cell line Kit225*	cell line
SOURCE.ct	417	*human T lymphocytes*	cell type
SOURCE.mo	21	*Schizosaccharomyces pombe*	mono-organism
SOURCE.mu	64	*mice*	multiorganism
SOURCE.vi	90	*HIV-1*	viruses
SOURCE.sl	77	*membrane*	sublocation
SOURCE.ti	37	*central nervous system*	tissue
UNK	-	*tyrosine phosphorylation*	background words

Table 1: Named entity classes. # indicates the number of XML tagged terms in our corpus of 100 abstracts.

early experiments using HMMs that incorporated pre- and post-class states we found that performance was significantly worse than without such states and so we formulated the model as given in section 3.

3 Method

The purpose of our model is to find the most likely sequence of name classes (C) for a given sequence of words (W). The set of name classes includes the 'Unk' name class which we use for background words not belonging to any of the interesting name classes given in Table 1 and the given sequence of words which we use spans a single sentence. The task is therefore to maximize $Pr(C|W)$. We implement a HMM to estimate this using the Markov assumption that $Pr(C|W)$ can be found from bigrams of name classes.

In the following model we consider words to be ordered pairs consisting of a surface word, W, and a word feature, F, given as $< W, F >$. The word features themselves are discussed in Section 3.1.

As is common practice, we need to calculate the probabilities for a word sequence for the first word's name class and every other word differently since we have no initial name-class to make a transition from. Accordingly we use the following equation to calculate the initial name class probability,

$$\begin{aligned} Pr(C_t| < W_{first}, F_{first} >) = \\ \sigma_0 f(C_{first}| < W_{first}, F_{first} >) + \\ \sigma_1 f(C_{first}| < \_, F_{first} >) + \\ \sigma_2 f(C_{first}) \end{aligned} \quad (1)$$

and for all other words and their name classes as follows:

$$\begin{aligned} Pr(C_t| < W_t, F_t >, < W_{t-1}, F_{t-1} >, C_{t-1}) = \\ \lambda_0 f(C_t| < W_t, F_t >, < W_{t-1}, F_{t-1} >, C_{t-1}) + \\ \lambda_1 f(C_t| < \_, F_t >, < W_{t-1}, F_{t-1} >, C_{t-1}) + \\ \lambda_2 f(C_t| < W_t, F_t >, < \_, F_{t-1} >, C_{t-1}) + \\ \lambda_3 f(C_t| < \_, F_t >, < \_, F_{t-1} >, C_{t-1}) + \\ \lambda_4 f(C_t|C_{t-1}) + \\ \lambda_5 f(C_t) \end{aligned} \quad (2)$$

where $f(|)$ is calculated with maximum-likelihood estimates from counts on training data, so that for example,

$$f(C_t| < W_t, F_t >, < W_{t-1}, F_{t-1} >, C_{t-1}) \doteq \frac{T(< W_t, F_t >, C_t, < W_{t-1}, F_{t-1} >, C_{t-1})}{T(< W_t, F_t >, < W_{t-1}, F_{t-1} >, C_{t-1})} \quad (3)$$

Where $T()$ has been found from counting the events in the training corpus. In our current system we set the constants λ_i and σ_i by hand and let $\sum \sigma_i = 1.0$, $\sum \lambda_i = 1.0$, $\sigma_0 \geq \sigma_1 \geq \sigma_2$, $\lambda_0 \geq \lambda_1 \ldots \geq \lambda_5$. The current name-class C_t is conditioned on the current word and feature, the previous name-class, C_{t-1}, and previous word and feature.

Equations 1 and 2 implement a *linear-interpolating* HMM that incorporates a number

of sub-models (referred to from now by their λ coefficients) designed to reduce the effects of data sparseness. While we hope to have enough training data to provide estimates for all model parameters, in reality we expect to encounter highly fragmented probability distributions. In the worst case, when even a name class pair has not been observed before in training, the model defaults at λ_5 to an estimate of name class unigrams. We note here that the bigram language model has a non-zero probability associated with each bigram over the entire vocabulary.

Our model differs to a backoff formulation because we found that this model tended to suffer from the data sparseness problem on our small training set. Bikel *et al* for example considers each backoff model to be separate models, starting at the top level (corresponding approximately to our λ_0 model) and then falling back to a lower level model when there not enough evidence. In contrast, we have combined these within a single probability calculation for state (class) transitions. Moreover, we consider that where direct bigram counts of 6 or more occur in the training set, we can use these directly to estimate the state transition probability and we use just the λ_0 model in this case. For counts of less than 6 we smooth using Equation 2; this can be thought of as a simple form of 'bucketing'. The HMM models one state per name class as well as two special states for the start and end of a sentence.

Once the state transition probabilities have been calculated according to Equations 1 and 2, the Viterbi algorithm (Viterbi, 1967) is used to search the state space of possible name class assignments. This is done in linear time, $O(MN^2)$ for M the number of words to be classified and N the number of states, to find the highest probability path, i.e. to maximise $Pr(W, C)$. In our experiments M is the length of a test sentence.

The final stage of our algorithm that is used after name-class tagging is complete is to use a clean-up module called *Unity*. This creates a frequency list of words and name-classes for a document and then re-tags the document using the most frequently used name class assigned by the HMM. We have generally found that this improves F-score performance by about 2.3%, both for re-tagging spuriously tagged words and for finding untagged words in unknown contexts that had been correctly tagged elsewhere in the text.

Word Feature	Example
DigitNumber	15
SingleCap	M
GreekLetter	alpha
CapsAndDigits	I2
TwoCaps	RalGDS
LettersAndDigits	p52
InitCap	Interleukin
LowCaps	kappaB
Lowercase	kinases
Hyphon	-
Backslash	/
OpenSquare	[
CloseSquare	]
Colon	:
SemiColon	;
Percent	%
OpenParen	(
CloseParen	)
Comma	,
FullStop	.
Determiner	the
Conjunction	and
Other	* + #

Table 2: Word features with examples

3.1 Word features

Table 2 shows the character features that we used which are based on those given for Nymble and extended to give high performance in both molecular-biology and newswire domains. The intuition is that such features provide evidence that helps to distinguish name classes of words. Moreover we hypothesize that such features will help the model to find similarities between known words that were found in the training set and unknown words (of zero frequency in the training set) and so overcome the unknown word problem. To give a simple example: if we know that *LMP - 1* is a member of PROTEIN and we encounter *AP - 1* for the first time in testing, we can make a fairly good guess about the category of the unknown word 'LMP' based on its sharing the same feature *TwoCaps* with the known word 'AP' and 'AP's known relationship with '- 1'.

Such unknown word evidence is captured in submodels λ_1 through λ_3 in Equation 2. We

consider that character information provides more meaningful distinctions between name classes than for example part-of-speech (POS), since POS will predominantly be noun for all name-class words. The features were chosen to be as domain independent as possible, with the exception of *Hyphon* and *GreekLetter* which have particular significance for the terminology in this domain.

4 Experiments

4.1 Training and testing set

The training set we used in our experiments consisted of 100 MEDLINE abstracts, marked up in XML by a domain expert for the name classes given in Table 1. The number of NEs that were marked up by class are also given in Table 1 and the total number of words in the corpus is 29940. The abstracts were chosen from a subdomain of molecular-biology that we formulated by searching under the terms *human*, *blood cell*, *transcription factor* in the PubMed database. This yielded approximately 3300 abstracts.

4.2 Results

The results are given as F-scores, a common measurement for accuracy in the MUC conferences that combines recall and precision. These are calculated using a standard MUC tool (Chinchor, 1995). F-score is defined as

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (4)$$

The first set of experiments we did shows the effectiveness of the model for all name classes and is summarized in Table 3. We see that data sparseness does have an effect, with proteins - the most numerous class in training - getting the best result and RNA - the smallest training class - getting the worst result. The table also shows the effectiveness of the character feature set, which in general adds 10.6% to the F-score. This is mainly due to a positive effect on words in the PROTEIN and DNA clases, but we also see that members of all SOURCE sub-classes suffer from featurization.

We have attempted to incorporate generalisation through character features and linear interpolation, which has generally been quite successful. Nevertheless we were curious to see just which parts of the model were contributing to the bigram scores. Table 4 shows the percentage of bigrams which could be matched against training bigrams. The result indicate that a high percentage of direct bigrams in the test corpus never appear in the training corpus and shows that our HMM model is highly dependent on smoothing through models λ_1 and λ_3. We can take another view of the training data by 'salami-slicing' the model so that only evidence from part of the model is used. Results are shown in Table 5 and support the conclusion that models λ_1, λ_2 and λ_3 are crucial at this size of training data, although we would expect their relative importance to fall as we have more direct observations of bigrams with larger training data sets.

Table 6 shows the robustness of the model

Class	Base	Base-features
PROTEIN	0.759	0.670 (-11.7%)
DNA	0.472	0.376 (-20.3%)
RNA	0.025	0.000 (-100.0%)
SOURCE(all)	0.685	0.697 (+1.8%)
SOURCE.cl	0.478	0.503 (+5.2%)
SOURCE.ct	0.708	0.752 (+6.2%)
SOURCE.mo	0.200	0.311 (+55.5%)
SOURCE.mu	0.396	0.402 (+1.5%)
SOURCE.vi	0.676	0.713 (+5.5%)
SOURCE.sl	0.540	0.549 (+1.7%)
SOURCE.ti	0.206	0.216 (+4.9%)
All classes	0.728	0.651 (-10.6%)

Table 3: Named entity acquisition results using 5-fold cross validation on 100 XML tagged MEDLINE abstracts, 80 for training and 20 for testing. *Base-features* uses no character feature information.

	λ Model No.					
# Texts	0	1	2	3	4	5
80	0.06	0.22	0.10	0.67	0.93	1.0
40	0.06	0.19	0.10	0.63	0.94	1.0
20	0.04	0.15	0.09	0.59	0.89	1.0
10	0.03	0.12	0.08	0.52	0.83	1.0
5	0.02	0.09	0.06	0.41	0.68	1.0

Table 4: Mean number of successful calls to submodels during testing as a fraction of total number of state transitions in the Viterbi lattice. # Texts indicates the number of abstracts used in training.

Backoff models	λ_0,$\lambda_1 \ldots \lambda_5$	λ_0,$\lambda_2 \ldots \lambda_5$	λ_0,$\lambda_3 \ldots \lambda_5$	λ_0,λ_4, λ_5	λ_0,λ_5
F-score (all classes)	0.728	0.722	0.644	0.572	0.576

Table 5: F-scores using different mixtures of models tested on 100 abstracts, 80 training and 20 testing.

# Texts	80	40	20	10	5
F-score	0.728	0.705	0.647	0.594	0.534

Table 6: F-score for all classes against size of training corpus (in number of MEDLINE abstracts).

for data sparseness, so that even with only 10 training texts the model can still make sensible decisions about term identification and classification. As we would expect, the table also clearly shows that more training data is better, and we have not yet reached a peak in performance.

5 Conclusion

HMMs are proving their worth for various tasks in information extraction and the results here show that this good performance can be achieved across domains, i.e. in molecular-biology as well as using news paper reports. The task itself, while being similar to named entity in MUC, is we believe more challenging due to the large number of terms which are not proper nouns, such as those in the *source* sub-classes as well as the large lexical overlap between classes such as PROTEIN and DNA. A useful line of work in the future would be to find empirical methods for comparing difficulties of domains.

Unlike traditional dictionary-based methods, the method we have shown has the advantage of being portable and no hand-made patterns were used. Additionally, since the character features are quite powerful, yet very general, there is little need for intervention to create domain specific features, although other types of features could be added within the interpolation framework. Indeed the only thing that is required is a quite small corpus of text containing entities tagged by a domain expert.

Currently we have optimized the λ constants by hand but clearly a better way would be to do this automatically. An obvious strategy to use would be to use some iterative learning method such as Expectation Maximization (Dempster et al., 1977).

The model still has limitations, most obviously when it needs to identify term boundaries for phrases containing potentially ambiguous local structures such as coordination and parentheses. For such cases we will need to add post-processing rules.

There are of course many NE models that are not based on HMMs that have had success in the NE task at the MUC conferences. Our main requirement in implementing a model for the domain of molecular-biology has been ease of development, accuracy and portability to other sub-domains since molecular-biology itself is a wide field. HMMs seemed to be the most favourable option at this time. Alternatives that have also had considerable success are decision trees, e.g. (Nobata et al., 1999) and maximum-entropy. The maximum entropy model shown in (Borthwick et al., 1998) in particular seems a promising approach because of its ability to handle overlapping and large feature sets within a well founded mathematical framework. However this implementation of the method seems to incorporate a number of hand-coded domain specific lexical features and dictionary lists that reduce portability.

Undoubtedly we could incorporate richer features into our model and based on the evidence of others we would like to add head nouns as one type of feature in the future.

Acknowledgements

We would like to express our gratitude to Yuka Tateishi and Tomoko Ohta of the Tsujii laboratory for their efforts to produce the tagged corpus used in these experiments and to Sang-Zoo Lee also of the Tsujii laboratory for his comments regarding HMMs. We would also like to thank the anonymous referees for their helpful comments.

References

A. Bairoch and R. Apweiler. 1997. The SWISS-PROT protein sequence data bank and its new supplement TrEMBL. *Nucleic Acids Research*, 25:31–36.

D. Bikel, S. Miller, R. Schwartz, and R. Wesichedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the Fifth Confererence on Applied Natural Language Processing*, pages 194–201.

A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. 1998. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proceedings of the Workshop on Very Large Corpora (WVLC'98)*.

S. Chen and J. Goodman. 1996. An empirical study of smoothing technniques for language modeling. *34st Annual Meeting of the Association of Computational Linguistics, California, USA*, 24–27 June.

N. Chinchor. 1995. MUC-5 evaluation metrics. In *In Proceedings of the Fifth Message Understanding Conference (MUC-5), Baltimore, Maryland, USA.*, pages 69–78.

N. Collier, H.S. Park, N. Ogata, Y. Tateishi, C. Nobata, T. Ohta, T. Sekimizu, H. Imai, and J. Tsujii. 1999. The GENIA project: corpus-based knowledge acquisition and information extraction from genome research papers. In *Proceedings of the Annual Meeting of the European chapter of the Association for Computational Linguistics (EACL'99)*, June.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the 7th International Conference on Intelligent Systemps for Molecular Biology (ISMB-99)*, Heidelburg, Germany, August 6–10.

A.P. Dempster, N.M. Laird, and D.B. Rubins. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society (B)*, 39:1–38.

D. Freitag and A. McCallum. 1999. Information extraction with HMMs and shrinkage. In *Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction*, Orlando, Florida, July 19th.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. 1998. Toward information extraction: identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing'98 (PSB'98)*, January.

J. Kupiec. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech and Language*, 6:225–242.

MEDLINE. 1999. The PubMed database can be found at:. http://www.ncbi.nlm.nih.gov/PubMed/.

DARPA. 1995. *Proceedings of the Sixth Message Understanding Conference(MUC-6)*, Columbia, MD, USA, November. Morgan Kaufmann.

C. Nobata, N. Collier, and J. Tsujii. 1999. Automatic term identification and classification in biology texts. In *Proceedings of the Natural Language Pacific Rim Symposium (NLPRS'2000)*, November.

Y. Ohta, Y. Tateishi, N. Collier, C. Nobata, K. Ibushi, and J. Tsujii. 1999. A semantically annotated corpus from MEDLINE abstracts. In *Proceedings of the Tenth Workshop on Genome Informatics*. Universal Academy Press, Inc., 14–15 December.

L. Rabiner and B. Juang. 1986. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–16, January.

T. Sekimizu, H. Park, and J. Tsujii. 1998. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. In *Genome Informatics*. Universal Academy Press, Inc.

K. Seymore, A. McCallum, and R. Rosenfeld. 1999. Learning hidden Markove structure for information extraction. In *Proceedings of the AAAI'99 Workshop on Machine Learning for Information Extraction*, Orlando, Florida, July 19th.

J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. 1999. Automatic extraction of protein interactions from scientific abstracts. In *Proceedings of the Pacific Symposium on Biocomputing'99 (PSB'99)*, Hawaii, USA, January 4–9.

A. J. Viterbi. 1967. Error bounds for convolutions codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269.