Natural Language Engineering **1** (1): 000–000. Printed in the United Kingdom

The Importance of the Lexicon in Tagging Biological Text

LAWRENCE H. SMITH, THOMAS C. RINDFLESH,
and W. JOHN WILBUR
National Center for Biotechnology Information (1,3)
Lister Hill National Center for Biomedical Communications (2)
National Library of Medicine, Bethesda, Maryland
e-mail: {lsmith,wilbur}@ncbi.nlm.nih.gov, tcr@nlm.nih.gov

(*Received 26 January 2005*)

Abstract

Motivation: A part-of-speech tagger is a fundamental and indispensable tool in computational linguistics, typically employed at the critical early stages of processing. Although taggers are widely available that achieve high accuracy in very general domains, these do not perform nearly as well when applied to novel specialized domains, and this is especially true with biological text.
Results: We present a stochastic tagger that achieves over 97.44% accuracy on MEDLINE abstracts. A primary component of the tagger is its lexicon which enumerates the permitted parts-of-speech for the 10 000 words most frequently occurring in MEDLINE. We present evidence for the conclusion that the lexicon is as vital to tagger accuracy as a training corpus, and more important than previously thought.
Availability: Software, documentation, and a corpus of 5 700 manually tagged sentences is available at ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedPost.

1 Introduction

A growing number of researchers in the biomedical domain are engaged in text mining or investigating its methodology. Part-of-speech tagging is fundamental to this work, as so many NLP techniques rely on properly tagged text, *eg* parsing, bracketing, *etc*. But available taggers for English, trained on general text, perform poorly in this domain. In response to this, we introduced a tagger, called MedPost, trained specifically for biological text, namely abstracts in the MEDLINE database. A description of the tagger and tag set, with examples and instructions for download can be found in (Smith *et al.* 2004).

Our approach to designing MedPost was based on the fact that good baseline accuracy for part-of-speech tagging is achievable when words are assigned their most frequent tag (see for example (Brill 1992) where a 7.9% error rate on the Brown corpus is quoted). That is, the most important information for tagging is associated with the word itself and can be stored in a lexicon. But contextual information is also important, and this relationship can be modeled after a manually tagged training corpus. We will argue that tagging with high accuracy can be achieved most efficiently by manually constructing a lexicon and

independently a training corpus, rather than primarily a training corpus (Merialdo 1991; Elworthy 1994).

The MedPost tagger uses a simple hidden Markov model (HMM) to combine contextual information (tag bigrams) with lexical information (tag probabilities for known and unknown words) to improve on baseline tagging accuracy. The model itself is not new, being the core algorithm in the Xerox tagger (Cutting *et al.* 1992) and TnT (Brants 2000) to name two.

In the following sections we describe the MedPost tag set, corpus selection and annotation, the lexicon, and the algorithm for the stochastic tagger. We then discuss tagging effectiveness based on different ways of constructing the lexicon. As part of this discussion, we describe an analysis of the distributional characteristics of the referential vocabulary (that is, items referring to content rather than linguistic function) in three different genres of text. Finally, we compare some of the design decisions of the tagger culminating with an assessment of the relative benefit to accuracy of the manual effort to extend the lexicon versus the manual effort to extend the annotated training corpus.

2 Design of the Tagger

In this section we describe the architecture of the tagger. This includes (1) a new tag set, with some portability to other POS tag sets, (2) a generic stochastic tagger, (3) an extendible lexicon as well as (4) domain specific and *ad hoc* handling of unknown words. The effectiveness of the tagger, achieving an accuracy of 97.44% on our test set, is demonstrated by comparison with two publicly available taggers.

2.1 Tag Set and Conventions

The MedPost tag set consists of 60 tags. Most of these are from UCREL C5 and C7 (Garside *et al.* 1997) and the Penn treebank (Marcus *et al.* 1994) with minor changes to some names. The MedPost tag set adheres to linguistic principles (Quirk *et al.* 2000) as much as possible while preserving the ability to translate back to the original tag sets. The closed class set from C7 includes tags for punctuation, subordinators, determiners, coordinating conjunctions, the existential *there*, a genitive marker, prepositions, numbers, modal verbs, and the infinitive marker. Other closed class tags are for mathematical symbols, generic pronouns, determiner pronouns (*eg that*), possessive pronouns, and relative pronouns.

Open class tags cover common nouns, verbs, adjectives, and adverbs. Each verb is tagged with one of six inflections (base form, infinitive, past tense, past participle, present participle, and 3rd person singular). Verbs can also be tagged as present or past tense adjectival participles, or as gerunds. The verbs *be*, *do* and *have* have special tags.

The definitions for tags reflect and generalize features used by the Penn treebank (Marcus *et al.* 1994), the SPECIALIST Lexicon (NLM 2003), and a subset of the UCREL C7 tag set (Garside *et al.* 1997). Some tags from these sets are distributed into multiple tags in MedPost. For example the Penn treebank IN tag (Marcus *et al.* 1994) includes both prepositions and subordinating conjunctions, with the ambiguity resolved by reference to the bracketed corpus (and therefore, it is well defined only in that context). The MedPost tag set has separate tags for prepositions and subordinating conjunctions, explicitly resolv-

ing the ambiguity of IN. A side-effect is that the tagger is able to make better use of the different contexts in which prepositions and subordinating conjunctions tend to appear. Conversely, some tags in the MedPost tag set are distributed into multiple tags in other tag sets. For example, the preposition tag in MedPost (II) is usually mapped to the preposition tag (IN) in the Penn treebank tag set, except when the word is *to* in which case it is translated to the infinitive marker (TO). This illustrates that the translation from the MedPost tag set to the Penn treebank tag set may depend on both the tag and the token. There are only a few such exceptions, and these are enumerated in a translating module that automatically translates from MedPost tagged text to Penn treebank or SPECIALIST Lexicon tagged text (see (Smith *et al.* 2004)).

Provision has also been made in the tag set for tagging multi-word lexical items. For example, the phrase *because of* should normally be tagged as a single preposition. And the phrase *in vitro* is tagged as an adjective since it frequently occurs as a prenominal modifier, as in *previous in vitro studies*. If these words were not tagged as a unit, then a simple bigram model would likely tag *because* as a subordinating conjunction and *in* as a preposition, making the syntax appear confused. In practice, only a small number of frequently occurring phrases are recognized, and less than 0.5% of the tokens in the test corpus belong to multi-word items.

2.2 Corpus Selection and Annotation

The corpus currently contains 5 700 sentences in 12 sets, with each set selected at random from 12 different groups of MEDLINE abstracts. Ten of these sets, MB_1 through MB_{10}, were selected from abstracts belonging to "themes" related to molecular biology using a query process described in (Wilbur 2002). These themes, in turn, were selected at random from the 10 largest branches of an *ad hoc* clustering of all themes related to molecular biology (based on a bag-of-words method). In addition to these 10, another set of sentences, MB, was selected from abstracts covering molecular biology in general, and another set, ML, was selected from random MEDLINE abstracts. A list of the sets with short descriptions, numbers of sentences and numbers of tokens is given in table 1. Throughout the development of the MedPost tagger, the MB and ML sets were not used for training, only for testing purposes; all training sets have been extracted from MB_1 through MB_{10}.

The text of the abstracts from each group were segmented into sentences by the tokenizer (described in §2.3) prior to random selection. Most of the selected sentences are the only ones from their corresponding abstract, but a few abstracts have two or more sentences selected. Some of the sentences that were improperly segmented (at an error rate of 1.2%) or that were uninformative (*eg* section headings) were rejected and replaced with better nearby representatives.

To manually tag the corpus, all sentences were first tagged using the evolving MedPost tagger (the earliest version of the tagger was trained on the Brown corpus), and the tags were then reviewed and edited by hand[1]. As work progressed, the reviewed portions of the corpus were used to train and improve the tagger, thereby reducing the number of

[1] The primary authorities for deciding membership in word classes are the works of (Quirk *et al.* 2000; Quirk *et al.* 2003).

Table 1. *Corpus composition*

set	# sent.	# tok.	description
MB_1	1000	28 608	Extracellular signaling
MB_2	1000	27 166	Sonic hedgehog
MB_3	1000	26 532	Metalloproteinases
MB_4	100	2 733	Human papillomavirus
MB_5	100	2 700	Hepatitis C
MB_6	100	2 705	Hepatitis B
MB_7	100	2 766	Proteoglycans
MB_8	100	2 892	Colony-stimulating factor
MB_9	100	2 615	Tissue plasminogen activator
MB_{10}	100	2 911	Chronic myeloid leukemia
MB	1000	27 786	General molecular biology
ML	1000	26 566	MEDLINE

corrections required. This approach may have introduced bias when a tagger-assigned tag is erroneous but left uncorrected. However, this bias was reduced through many iterations of review and cross-comparison.

2.3 Stochastic Tagging Algorithm

Tagging begins by obtaining a sequence of words from text. The tokenization algorithm is an elaboration of the conventions followed by the Penn treebank (Marcus *et al.* 1994). Words are delimited by white space and punctuation, but special conditions apply to hyphenated words (which are considered as a single word), abbreviations, numbers, and other creative uses of the character set. Sentences are delimited, or segmented, by looking for sentence-ending punctuation and ignoring probable abbreviations and other expressions containing non-sentence-ending periods.

The tagger itself reads tokenized input, and models it as a hidden Markov model. The model and algorithms are standard (Rabiner 1988), and will not be described exhaustively here. For completeness, the formula for the probability of a sequence of words $W_1^n = w_1, \ldots, w_n$ and tags $T_1^n = t_1, \ldots, t_n$, is

$$\begin{aligned}\Pr(T_1^n|W_1^n) &= \Pr(T_1^n, W_1^n)/\Pr(W_1^n) \\ &= \prod_{i=1}^{n} \Pr(w_i|t_i)\Pr(t_i|t_{i-1})/\Pr(w_i) \\ &= \prod_{i=1}^{n} \Pr(t_i|w_i)\Pr(t_i|t_{i-1})/\Pr(t_i).\end{aligned}$$

This derivation uses the definition of conditional probability, and appropriate independence assumptions. Given a sequence of words, the Viterbi algorithm is used to find the sequence of tags that maximizes this probability. The probabilities $\Pr(t_i|t_{i-1})$ and $\Pr(t_i)$ are estimated from unigram and bigram tag frequencies, in the supervised manner, from a pretagged training corpus with Witten-Bell discounting (Witten and Bell 1991).

The term $\Pr(t_i|w_i)$ is the probability that a tag t_i will occur given the word w_i. The

lexicon stores specific words together with explicit probabilities $\Pr(t_i|w_i)$ for the possible tags. However, most words in the MedPost lexicon (98.7%) specify equal probabilities divided among a set of permitted tags. Despite the absence of prior probabilities for tags of many words, the tagger is able to achieve very high accuracy using its bigram language model and unknown word model.

The tokenizer is also able to recognize multi-word lexical items (*because of, in vitro, etc*) that appear in the lexicon, and the tagger treats these as single tokens. Thus the tokenizer makes a commitment to a multi-word item and the tagger is forced to tag all of the words of the phrase as a unit. The drawback of this is that the words of a phrase might appear together accidentally, without being used as a phrase. For example, out of 44 occurrences of the phrase *due to* in the corpus, all but one of them function as prepositions, as in *hypoxia due to apnea*. One of them however, was more properly tagged with *due* as an adjective and *to* as an infinitive marker as in *embryos due to be pattern-deleted*. Because of this inflexibility, only the high frequency and problematic phrases (those whose constituents would tend to be tagged in a way that would obscure the syntactic structure) were included in the lexicon.

2.4 Lexicon

The lexicon file, containing a list of words with their permitted parts of speech, is divided into three sections: a *word ending* section, a *closed class* section, and an *open class* section. Each section is a list of entries, which may be a word, phrase, or word ending, followed by a list of the tags permitted for it, and probabilities for each tag. For example,

```
^feeding_VVG_VVGJ_VVGN
```

is the entry for the whole word *feeding*, and indicates the possible tags are present participle verb (VVG), participial adjective (VVGJ), or gerund (VVGN), and these are assumed to be equally probable (in this case, $1/3$ probability each). The lexicon itself may be modified using a text editor, taking care to use the correct format for each entry, as illustrated. Different inflections of words are entered separately, and the tagging algorithm makes no connection between words having the same lexeme.

The closed class section consists of words (currently 360) that have at least one closed class tag (determiners, conjunctions, *etc*) together with a list (currently 72) of multi-word lexical items (*because of, in vitro, etc*). The words themselves were obtained from the high frequency words in MEDLINE supplemented with lists found in (Quirk *et al.* 2000), and the assignment of closed class tags was based on principles described in (Quirk *et al.* 2000). Multi-word items were selected from various on-line and print sources, including (Quirk *et al.* 2000), and from experience with the training corpus.

The open class section of the lexicon contains nouns, verbs, adjectives, and adverbs obtained from MEDLINE. All open class words were sorted in decreasing frequency of occurrence. The final lexicon contains entries for the first 10 000 words on this list. We also evaluated the effect of an "extended" lexicon containing 100 000 of the most frequent words in MEDLINE. However, since only the test sets (MB and ML) were evaluated using the extended lexicon, it was only required to enter the subset of the 100 000 words actually occurring in MB and ML (the words not occurring in the test set could not have any

effect), and this amounted to an additional 1 852 words. Therefore the open class section of the extended lexicon contained a total of 11 852 words. In §3.4, we compare the relative effectiveness of three different methods for developing the open class lexicon: derivation from the training set of tagged sentences, generation by machine learning of word classes, and construction by hand.

The word ending section of the lexicon contains high frequency word endings (currently 757 of them), from 0 to 4 letters in length used to handle unknown words. This is explained in the next section.

2.5 Handling of Unknown Words

The machinery for tagging words that are not found in the closed class or open class sections of the lexicon relies mainly on the word endings and *ad hoc* constraints.

Word endings. Both the orthography of a word (*eg* whether it is written as all lower case, all upper case, mixed letters and numbers, *etc*), as well as the suffixes are predictive of word class. To capitalize on this, the word ending section of the lexicon contains entries combining this information. Each entry specifies one of eight identified orthographic classes together with a word ending of 0 to 4 letters in length, and the probability of each possible tag is stored with it. In constructing this section, the training corpus was used, and only those combinations of class and word ending with 20 or more instances were kept. When the tagger encounters a word that is not in the lexicon, the probabilities for each tag, $\Pr(t_i|w_i)$, are obtained from the lexicon entry in the word ending section that has the same orthographic class and the most specific (*ie* longest) word ending. For example, in searching for the unknown word *preadipocytes*, the entry for orthographic class "all lower case" and the word ending *ytes* is used, but not the entries for the word endings *tes*, *es*, or *s*.

Ad hoc constraints. After consulting the lexicon, a short list of constraints is applied to $\Pr(t_i|w_i)$, also based on the orthography of the word. These constraints were coded by hand to address common tagging errors, and are applied only to words that are not found in the lexicon. For example, expressions referring to numbers like -1, 10,000, and 402-405 and spelled numbers like *forty two* are recognized and always assigned the number tag with probability 1. Other practical constraints were also implemented. For example, novel hyphenated words are almost never verbs, so the probabilities for all of the verb tags are set to 0 for such words (frequent hyphenated verbs, like *up-regulated*, appear in the lexicon as verbs). Also, there is a recognizer for generic acronyms, like *ERK*, and *MMPs*, which are very often correctly tagged as noun and plural noun, respectively.

2.6 Effectiveness

The resulting tagger achieves 97.44% accuracy on the 1 000 sentences of the ML set. We quote this number as our final figure of performance with a 95% confidence interval of $\pm$ .19. We compare this figure with the performance results obtained from other taggers trained on their original corpora. The Brill tagger (Brill 1992), trained on the Brown corpus using the Penn treebank tag set (Marcus *et al.* 1994), achieves an accuracy of 86.8% on the ML set. And the Xerox tagger (Cutting *et al.* 1992), also trained on the Brown corpus but

modified to use the tag set of the SPECIALIST Lexicon (Rindflesh *et al.* 2000), achieves an accuracy of 93.1%. By translating the MedPost tagger output to those respective tag sets, it achieves an accuracy of 97.7% and 98.9% respectively. This comparison is not intended to be a fair comparison of tagger algorithms, yet the differences can be explained by the training of the MedPost tagger on a corpus extracted from MEDLINE, the reduced ambiguity of its tag categories, and the specificity of its lexicon for MEDLINE.

3 Analysis of Tagger Effectiveness

We found that an effective tagger requires (1) that the tag set is carefully defined, (2) that the lexicon contains domain specific words, and (3) that the lexicon is of high quality and manually reviewed. We also found that an efficient development effort would place about as many token types in the lexicon as tokens in the training set. In this section we analyze and present evidence for each of these findings in the context of our experience with tagging biological text.

3.1 Comparison of Tag Sets

As described above, the MedPost tag set was derived from part-of-speech class definitions from the Penn treebank, UCREL, and the SPECIALIST Lexicon. But we departed from those tag sets in order to reduce contextual ambiguity, and to make the task of tagging easier. At the same time, the tags were chosen to facilitate automatic conversion to other tag sets.

To evaluate this, we applied the MedPost algorithm on the ML set using alternative tag sets. With the MedPost tag set, the tagger achieved 97.44% $\pm$ .19 accuracy. When the output of the tagger and the test set tags were both translated to the Penn treebank tag set before comparing them, the accuracy was 97.68% $\pm$ .18 (this increase is not statistically significant). Finally, when the MedPost tagger was converted to a Penn treebank based tagger (by translating the training set, lexicon, and ad hoc constraints to Penn treebank tags) the tagging accuracy was 96.29% $\pm$ .22. This decrease (significant with $p < .05$) supports the hypothesis that the MedPost tag set more accurately models the language. The same trend was found when the evaluation was done on the MB set (97.31%, 97.52%, and 96.09% respectively).

3.2 Comparison of Vocabulary

Evidence of language specialization can be seen in a comparison of words occurring in MEDLINE with the words occurring in the Brown corpus (Marcus *et al.* 1994) and the AP corpus (1988/1989 version), as shown by the Venn diagram of figure 1. To construct this figure, we first took the token types from each corpus separately, sorted them by decreasing order of frequency, and selected the highest ranked token types on the list that accounted for 92.3% of the tokens in the corpus. This percentage was chosen so that the list of types for the MEDLINE corpus comprised the top 10 000 types. With these lists of token types, the token types in each intersection were counted; the whole numbers in the figure give

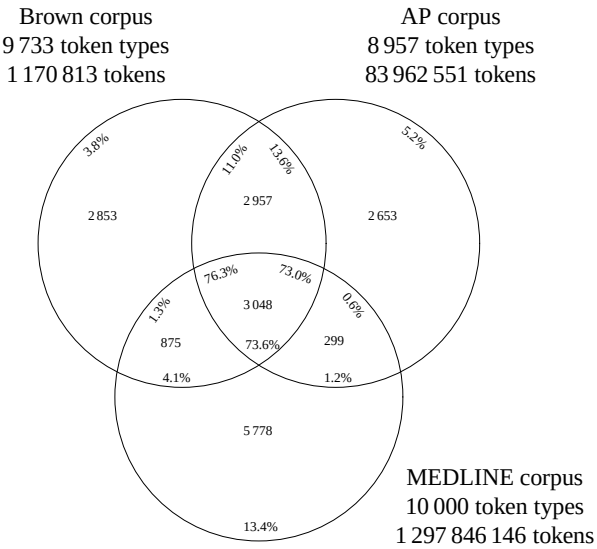

Fig. 1. A comparison of the terminology of Brown, AP, and MEDLINE corpora. The number in each intersection is the number of token types in the intersection, and the percentages are the percentage of tokens represented in the corpus on the corresponding area.

the number of token types in each intersection, while the percentages reflect the total number of tokens represented by these types in the corresponding corpus (percentages sum to 92.3%). A set of 3 048 types is common to all three corpora, accounting for around 75% of the tokens of each. Outside of this common set of words, the next most significant overlap is 2 957 token types uniquely shared by the Brown and AP corpora, accounting for 11.0% and 13.6% of their tokens, respectively. Yet there are only 875 types uniquely shared by MEDLINE and Brown and 299 types uniquely shared by MEDLINE and AP, and a significant 5 778 types unique to MEDLINE alone. The types unique to MEDLINE account for 13.4% of that corpus, while the types unique to Brown account for 3.8% and those of AP account for 5.2%. This difference in terminology may explain why taggers trained on other corpora perform poorly when applied to MEDLINE, and the emphasis of the lexicon in our approach addresses this phenomenon.

3.3 Comparison of Tagger Components

The relative impact of different components of the tagger was measured by tagging after effectively removing each component in turn, with results shown in the upper half of table 2. The components that were removed were the *ad hoc* constraints and word ending section of the lexicon (separately and together), the open class section of the lexicon, the closed class section of the lexicon, and the bigram frequencies. To effectively remove the bigram frequencies and word ending frequencies, all probabilities were assigned the same value. The *ad hoc* constraints were removed simply by commenting out the code implementing them. The open and closed class sections of the lexicon were removed by deleting all of their entries from the lexicon. Obviously the most important component of the tagger is

Table 2. *Accuracy of tagging with varying components. The top half of the table lists accuracies obtained by removing: ad hoc constraints, word endings in the lexicon, ad hoc constraints and word endings in the lexicon, open class words in the lexicon, closed class words in the lexicon, and bigram data. The bottom half lists accuracies obtained using lexicons obtained by different methods: the trained lexicons (trained on 10 and 3 700 sentences from MB_1 through MB_{10}), the automated lexicon, and the manual lexicon. 95% Confidence intervals are approximately given by* $\pm$ *.25.*

lexicon	ML	Δ	MB	Δ
w/o ad hoc constraints	97·36%	-0·08	97·12%	-0·32
w/o word endings	95·26%	-2·18	95·12%	-2·32
w/o endings or ad hoc	94·88%	-2·56	94·40%	-3·04
w/o open class	94·86%	-2·58	95·58%	-1·86
w/o closed class	59·66%	-37·78	60·86%	-36·58
w/o bigrams	92·12%	-5·32	92·51%	-4·93
trained (10)	94·53%	-2·91	95·06%	-2·38
trained (3 700)	95·76%	-1·68	96·25%	-1·19
automated	96·37%	-1·07	96·34%	-1·10
manual	97·44%	-	97·31%	-

the closed class section of the lexicon. This represents a large portion of words appearing in English text, which are "function" words. Without the closed class section, the tagger is unable to tag any of these words with their correct tag, because only open class tags are permitted for unknown words. Next in importance, for this evaluation, is the bigram frequency data. The *ad hoc* constraints play a minor role in determining accuracy.

When one considers that the word ending section and bigram data are both derived from the training corpus, this argues that the training set controls a greater share of accuracy than the open class section. But this does not tell us how the accuracy changes if these components are *varied*, only how the accuracy changes if the components are removed altogether. The incremental effect will be examined more closely in §3.5 after considering the effect of employing different methods of constructing the open class lexicon.

3.4 Comparison of Lexicons

We experimented with three different methods for constructing the open class section of the lexicon. The trained lexicon was derived by a method similar to that used in (Brill 1992) and (Cutting *et al.* 1992), using a training set to build a lexicon of words and the frequency of their tags. The automated lexicon was constructed using a machine learning algorithm and multiple lexical sources to predict lexicon entries. Finally, the manual lexicon was created by hand. Here we describe each of these in detail and the resulting tagging accuracy. The lexicons are presented in the order of their performance, with the trained lexicon performing worst and the manual lexicon best.

Trained lexicon. The training set consists of 3 700 tagged sentences that were hand corrected. Its main purpose is for training the Markov transition probabilities between tags, and to construct the word ending section of the lexicon. The training set may also be used

as a source of tagged words for creating a lexicon, which we call the *trained lexicon*. All open class words appearing in the training set were entered into the trained lexicon, together with the frequency observed in the training set for each tag. As described in §2.3, to compute $\Pr(t_i|w_i)$ for a word w_i using this lexicon, the tag frequencies are smoothed with default tag frequencies using Witten-Bell discounting. As shown in the lower half of table 2, a training set of 3 700 sentences of MB_1 to MB_{10} was used with an accuracy of 95.76%, and another with 10 sentences for comparison, with an accuracy of 94.53%. Other methods of extracting a lexicon from the training set were explored, but these did not result in clear improvements in accuracy and are not reported here.

Automated lexicon. To construct the automated lexicon, a machine learning algorithm was used to combine several sources of information to produce lexicon entries, *ie* a list of permitted tags for each word. The features used in machine learning were obtained from the letters of the word, the words of MEDLINE, WordNet, and the SPECIALIST Lexicon. For example, the word *feeding* is alphabetic (consisting of letters alone) and ends in the letters *ng*, and these facts are coded as features FORM=alpha, END1=g and END2=ng. It also occurs in both WordNet and the SPECIALIST Lexicon as a noun, as does the word *feed*, which are coded as features WNLEX=noun, SPLEX=noun, WNLEX-ing=noun (which means that the word *feed* occurs in WordNet as a noun and is obtained from *feeding* by removing the ending *ing*) and SPLEX-ing=noun. The words *feed, feedings, feeder*, and *feeds* all occur in MEDLINE, and these facts are coded as the features MEDLINE-ing, MEDLINE-ing+s (which means that the word *feeds* occurs in MEDLINE and is obtained from *feeding* by removing the ending *ing* and adding the ending *s*) and MEDLINE-ing+er. In addition to these, we also reasoned that an unknown word should occur frequently in unambiguous contexts where the tagger is likely to obtain the correct tag. We therefore tagged all of MEDLINE with MedPost using a version of the lexicon whose open class section contained a few hundred of the most frequently occurring words in MEDLINE. For each word, the majority tag and tags occurring more than 10% were used to form features. To continue the example, the word *feeding* received the tag of a gerund (VVGN) most often but also frequently received the tag of an adjectival participle (VVGJ) and these facts are encoded as features TAGMJ=VVGN, TAGHI=VVGN and TAGHI=VVGJ.

For each open class tag, the CMLS algorithm (Zhang and Oles 2001) was applied to a training set of known words to learn to recognize words that can occur with that tag. The lexicon entry in the automated lexicon for each word was constructed by listing all of the tags for which the corresponding recognizer accepted the word.

To evaluate the tagger using the automated lexicon, the machine learning algorithm was trained on a training set of 1 548 open class words. This was then used to obtain lexicon entries for all 11 852 words in the lexicon (this list includes the original 1 548 words of the training set). The accuracy that resulted from training with this automated lexicon on the ML set was 96.37%. Interestingly, a comparable accuracy was obtained when the automated lexicon was trained without using features from WordNet or the SPECIALIST Lexicon. For this study, we did not explore the effects of varying the size of the training set or employing alternative learning algorithms in constructing an automated lexicon.

Manual lexicon. The lexicon entries from the automated lexicon for all 11 852 words were manually edited to obtain a version of the lexicon we refer to as the *manual lexicon*. Wherever possible, the permitted tags for each word excluded those tags that in our judg-

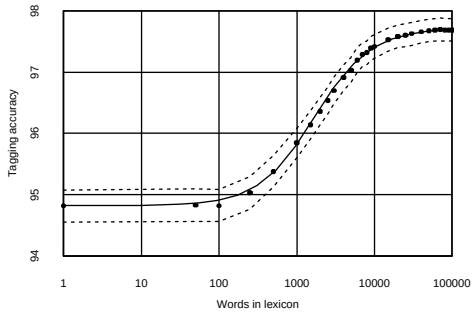

Fig. 2. Accuracy of the tagger on the ML set as a function of the size of the lexicon. The dotted lines are 95% confidence intervals.

ment were inappropriate to scientific text, though such tags would necessarily appear in a lexicon appropriate for some genres of text. The manual entries did not contain probability information for 9 913 open class words out of 10 000, or 99.13%. Those few words that were given explicit prior probabilities were examples where a tag was added with low probability to allow for rare usages (as determined from experience with the training corpus). For example, the word *inverse* occurs most often as an adjective, but occasionally it can be used as a noun, so its lexicon entry was entered as

```
^inverse_JJ:1000_NN:1
```

Using the manually corrected lexicon resulted in an accuracy of 97.44% on the ML set.

Summary of comparison of lexicons. The accuracy of tagging is shown in the lower half of table 2 for MB and ML for all 3 lexicons, with the trained lexicon shown for training sets of 10 and 3 700 sentences. The accuracy of the tagger using the trained lexicon is less than both the manual and the automated lexicons. All numbers in the table are significantly different ($p < 0.5$), except for the MB set where the accuracy resulting from using the trained (3 700) lexicon accuracy is not significantly different from that using the automated lexicon. The small difference between training with 10 sentences and 3 700 sentences is a result of the tagger's ability to tag unknown words. The tagging of unknown words depends on the word ending section of the lexicon which was constructed using 3 700 sentences, and this factor was not varied in comparing lexicons.

3.5 Comparison of Lexicon and Training Set

This project began with the goal of developing a tagger for MEDLINE text. And from the beginning, we were focused on obtaining an accurately tagged training corpus (Merialdo 1991). But our initial experience with manual tagging suggested that the lexicon is very valuable. In this section we present experimental evidence that quantifies the relative importance of the lexicon. We compare the incremental effects of two variables, the size of the lexicon and size of the training corpus, where the latter quantity varies the amount of data used to construct both the word ending section of the lexicon and the bigram probabilities. By controlling these variables separately, we will be able to observe their relative impacts without biasing the conclusion. In all experiments, the *ad hoc* constraints and closed class section of the lexicon are the same.

Figure 2 shows the accuracy on the ML set, with 95% confidence intervals, as the number of open class words in the lexicon varies up to 100 000 (confidence intervals are provided here and elsewhere to indicate how the computed accuracy depends on the size of the test set). Note that the data has a sigmoid-curve shape, and a logistic curve has been fitted to the data. The lowest accuracy (94.82%) is determined by the ability of the tagger to tag open class words when they are all unknown. The highest accuracy (97.69%) approaches the limit that is achievable by this tagger when *all* the words occur in the lexicon. The difference between the higher figure 97.69% and the reported figure 97.44% is similarly due to the difference between tagging with a lexicon of 100 000 words and with a lexicon of 10 000 words.

A similar sigmoid-curve shape can be seen (graph not shown) as the number of words in the training set varies, thereby varying the data used to derive the word ending section of the lexicon and bigram probabilities. The shapes of these curves suggest the possibility of modeling tagger accuracy as a double sigmoid-curve in these two variables, the size of the lexicon and the size of the training set, which would be valid at least over the range of the available data. Such a closed formula makes it possible to predict the incremental improvement in accuracy as the lexicon and training set are varied in size. In other words, for any given size of the lexicon and training set, it is possible to predict whether to put effort into one or the other in order to get the maximal improvement in accuracy. Of course, these predictions cannot be expected to extrapolate beyond the bounds of the data, or to hold for other tagging projects.

To carry out this plan, the 1 000 sentences of the ML set were tagged by the tagger using differing sizes of training set and lexicon. A total of 336 tagging experiments were performed using one of 12 different training sets and one of 28 different lexicons. The 12 training sets were extracted from MB_1 through MB_{10} and consist of 10 to 3 700 sentences, or 297 to 101 216 words respectively. The 28 lexicons effectively vary from 0 to the 100 000 most frequently occurring words in MEDLINE.

These 336 accuracy values were then smoothed by fitting to a parameterized logistic curve using the nonlinear least squares method. The root-mean-square error of the fit is 0.243 (units of percent), with 95% of the fitted values within 0.481 (units of percent) of their actual value. As an example of the goodness-of-fit, the accuracy of tagging the ML set with bigrams and word endings trained on the 3 700 sentences of MB_1 through MB_{10} using the 10 000 word manual lexicon is 97.44% while the value obtained from the empirical formula is 97.40%, a difference of .04.

The smoothed data is represented by the solid lines in the contour plot of figure 3. Each point in this graph corresponds to a size for the lexicon, given on the x-axis, and a size for the training set, given on the y-axis. Both axes are represented logarithmically by powers of 10. Therefore, to read the graph to estimate the accuracy of tagging with a given size for the lexicon and training set, locate this point on the graph and interpolate from the accuracies of the nearest (solid line) contours. Though not shown, when the actual accuracy data is plotted on this graph (using linear interpolation to obtain the contours, for example) the contours conform to the fitted curves, with small deviations and jagged edges.

The empirical formula for accuracy can be used to predict the improvement in accuracy from adding a word to the lexicon, call this i_L, and the improvement to accuracy from adding a word to the training set, call this i_T (these values both depend on the current size

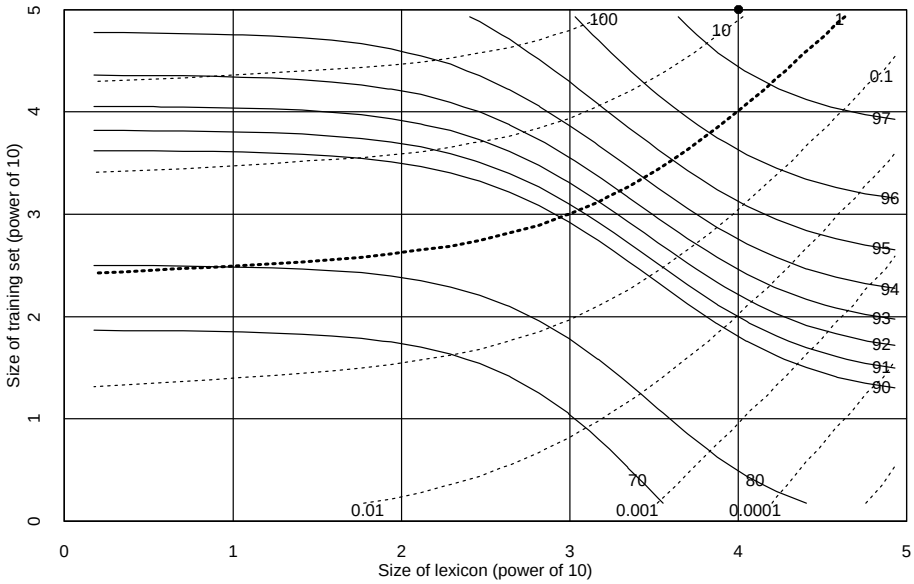

Fig. 3. Contour plot of smoothed accuracy (solid lines) as a function of the size of the lexicon and training set. The dashed lines are the ratio of the change in accuracy from adding an (open class) word to the lexicon to the change in accuracy from adding a word to the training set. The training set is used to train both the language model (tag bigrams) and the word ending section of the lexicon. The large dot at $(10^4, 10^5)$ is the current status of the MedPost tagger.

of the lexicon and training set). If $i_L > i_T$, then more improvement is to be gained by adding a word to the lexicon, than by adding a word to the training set. The ratio i_L/i_T is plotted in the contour plot in figure 3 as the dashed lines. The bold dashed line, the "equilibrium curve," is where the ratio is 1, that is, $i_L = i_T$. Above this line is a region where $i_L > i_T$, and below it a region where $i_L < i_T$.

At any stage of development of the lexicon and training set (occurring in figure 3), the effort required to transition from one region to the other is surprisingly small. For example, with 100 words in the lexicon and no training set, the training set should be increased to about 400 words. With 1 000 words in the training set, the lexicon should be extended to about 1 000 words. The current training set of MB_1 through MB_{10} contains a little over 100 000 words, and the formula predicts that all effort should be put into the lexicon until it contains 40 000 words. Our tagger is clearly in a region of diminishing returns (figure 2).

4 Discussion

We do not believe that tagging accuracy depends critically on the underlying algorithm, but rather on the number of words in the lexicon and training set (always, a complete closed class lexicon is necessary). For example, by translating our manual lexicon (of open and closed class words) into a form usable by the Brill tagger (Brill 1992) or the Xerox tagger (Cutting *et al.* 1992), and dividing our corpus into a training and testing set, we would

expect either of these taggers could achieve an accuracy comparable to 97.44%. But a tagging system that derives its lexicon from the training corpus, and does not use a manually generated lexicon, should require a much larger training set to achieve comparable accuracy. This would apply, for example, to the Maximum Entropy Model tagger (Ratnaparkhi 1996).

Using MedPost, we have shown that the requirement for manually tagged training sentences is substantially reduced if an adequate lexicon is used. It remains to argue that the most efficient way to obtain a lexicon is by direct manual creation of the lexicon entries. To make this argument, we will derive an estimate of the manual effort required to produce the MedPost lexicon and training corpus and compare it with an estimate of the manual effort required to produce a training corpus alone, assuming that a comparable lexicon is extracted from it.

To account for the fact that some annotation tools are easier to use than others, we assume, as in our work, that a person is presented with precomputed annotations and lexicon entries and is asked merely to independently verify their correctness; and that when corrections *are* required, the computer interaction is minimal. With this understanding, we found that it was feasible for one person to create lexicon entries for 1 000 new words per day, while about 200 sentences per day could be annotated. Therefore, our estimate of the time to produce the MedPost lexicon of 10 000 words and a training corpus of 3 700 sentences is 28.5 days.

Many taggers derive a lexicon by tabulating token types and their parts of speech observed in a training corpus. Suppose, for instance, that a usable lexicon entry can be derived for words observed at least twice. Assuming that words occur independently with their observed frequencies in MEDLINE, a Monte Carlo simulation shows that approximately 18 000 sentences would be needed in order for the resulting lexicon to contain all of the 10 000 most frequent words. (We point out that the resulting lexicon would still underrepresent many of the 10 000 words, and their tag probabilities would need to be estimated by using a smoothing approximation.) Again using our work estimate of 200 sentences per day, this training corpus of 18 000 sentences would require 90 days, which is more than three times the effort estimated to produce the MedPost lexicon and training corpus.

The reason many more sentences are required to derive the lexicon from a training corpus is simply a matter of efficiency. As more (random) sentences are considered, each sentence contributes less and less *new* information to the lexicon. For example, in the first batch of 200 sentences there would be approximately 2.7 new words encountered per sentence, with most of them being high frequency words. But in succeeding batches, this number would quickly drop to 0.8 new words per sentence, corresponding to lower frequency words in MEDLINE. Moreover, many of these words would be very low frequency and not in the 10 000 word lexicon.

Finally, we speculate that in some sense our results are not a consequence of our implementation or even of our definition of the tagging problem. Rather, we think that the results originate from the characteristic behavior of English word classes, and may generalize to any tagging problem involving a human language similar to English, *ie* word order dependent and not highly inflected, where a relatively small tag set is appropriate. Natural languages obey the Zipf distribution (Zipf 1949), and this explains the diminishing utility of effort spent tagging sentences to create a lexicon. One may hypothesize a simple para-

metric "law" of tagging that describes the equilibrium between lexicon size and training set size (the bold dashed line in figure 3), whose parameters would depend only slightly on the tagging problem and the implementation.

5 Conclusions

In our effort to develop a tagger trained for biological text, we have achieved a high level of accuracy, an efficient tag set and a corpus that can be used for training, testing, or any other purpose. All of this material is available online[2], where additions, revisions and notes may also appear in the future.

We have also found that lexicon development is an important requisite for accurate tagging in this, and probably any, specialized domain. Our results suggest a strategy for building a highly accurate tagger:

- manually tag a modest amount of text for training and testing, about 100 000 words as in our project, (important for learning the language model or how parts of speech relate to and follow one another in a sentence),
- construct a lexicon, either manually or with careful manual review, with permitted parts of speech for all closed class words and punctuation, and a number of the most frequently occurring open class words comparable to the number of tokens in the training set, and
- tag text with a stochastic model based on smoothed relative frequency data.

Acknowledgements

We would like to thank Lorrie Tanabe for providing a critique of an early draft of this paper.

References

T. Brants. 2000. TnT - A Statistical Part-of-Speech Tagger. *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ACL, 2000.

E. Brill. 1992. A simple rule-based part of speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, ACL, 1992.

D. Cutting, J. Kupiec, J. Pedersen and P. Sibun. 1992. A practical part-of-speech tagger. *Proceedings of the Third Conference on Applied Natural Language Processing*, ACL, 1992.

D. Elworthy. 1994. Does Baum-Welch re-estimation help taggers? *Proceedings of the 4th ACL conference on Applied Natural Language Processing*, 53–58.

R. Garside, G. Leech and A. McEnery. 1997. *Corpus Annotation*. Longman, London and New York.

M.P. Marcus, B. Santorini and M.A. Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2), 313-330.

B. Merialdo. 1991. Tagging text with a probabilistic model. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1991.

National Library of Medicine. 2003. *UMLS Knowledge Sources, 14th Edition*.

R. Quirk, S. Greenbaum, G. Leech and J. Svartvik. 2000. *A Comprehensive Grammar of the English Language*. Longman, London and New York.

R. Quirk *et al.* (eds) 2003. *Dictionary of Contemporary English*. Longman, London and New York.

[2] see ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedPost,

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. *Proc. of the First Conference on Empirical Methods in Natural Language Processing*, 1996.

L.W. Rabiner. 1988. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition *Proc. of the IEEE* 77, 2, 257-286.

T.C. Rindflesch, J.V. Rajan and L. Hunter. 2000. Extracting molecular binding relationships from biomedical text. *Proceedings of the 6th Applied Natural Language Processing Conference*, 188-195.

L.H. Smith, T.C. Rindflesch and W.J. Wilbur. 2004 MedPost: A Part of Speech Tagger for BioMedical Text. *Bioinformatics* 20(14), 2320-2321.

W.J. Wilbur. 2002. A Thematic Analysis of the AIDS Literature. *Pacific Symposium on Biocomputing* 7, 386-397.

I.H. Witten and T.C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), 1085-1094.

T. Zhang and F.J. Oles. 2001. Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval* 4, 5-31.

G.K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Reading, MA.