BIOINFORMATICS

Vol. 18 no. 8 2002
Pages 1124–1132

Tagging gene and protein names in biomedical text

Lorraine Tanabe and W. John Wilbur

National Center for Biotechnology Information, NLM, NIH, Bethesda, Maryland 20894, USA

Received on November 19, 2001; revised on February 6, 2002; accepted on February 14, 2002

ABSTRACT

Motivation: The MEDLINE database of biomedical abstracts contains scientific knowledge about thousands of interacting genes and proteins. Automated text processing can aid in the comprehension and synthesis of this valuable information. The fundamental task of identifying gene and protein names is a necessary first step towards making full use of the information encoded in biomedical text. This remains a challenging task due to the irregularities and ambiguities in gene and protein nomenclature. We propose to approach the detection of gene and protein names in scientific abstracts as part-of-speech tagging, the most basic form of linguistic corpus annotation.

Results: We present a method for tagging gene and protein names in biomedical text using a combination of statistical and knowledge-based strategies. This method incorporates automatically generated rules from a transformation-based part-of-speech tagger, and manually generated rules from morphological clues, low frequency trigrams, indicator terms, suffixes and part-of-speech information. Results of an experiment on a test corpus of 56K MEDLINE documents demonstrate that our method to extract gene and protein names can be applied to large sets of MEDLINE abstracts, without the need for special conditions or human experts to predetermine relevant subsets.

Availability: The programs are available on request from the authors.

Contact: tanabe@ncbi.nlm.nih.gov

INTRODUCTION

The automatic extraction of gene and/or protein names from the biological literature is a timely undertaking due to the current trend in molecular biology towards large-scale genetic analyses. The ability of scientists to produce data on the expression of thousands of genes at a time has led to an increased need for information regarding these genes and the proteins they encode. The prohibitive volume of information proliferating in the biomedical literature has prompted researchers to explore automated text processing techniques to make the task of managing all the relevant information more feasible. The recognition of gene and protein names in biomedical text is important for information filtering, information retrieval and automated knowledge acquisition for text mining. It remains a challenging task because many names are not proper nouns (*envelope, ran, frizzled*), several names are used to refer to the same entity (*caspase-3, CASP3*, *apoptosis-related cysteine protease, CPP32*), and names can be used in combination with other entities like cell lines and chemicals (*CHO-A(3), Ca*$^{2+}$*/calmodulin*). For compound names, there is the additional requirement of determining where the name begins and ends within a sentence. This can be particularly difficult when verbs and adjectives are embedded in names (*mullerian inhibiting substance, deleted in azoospermia-like*). Yet another difficulty arises when there are few morphological clues, making a legitimate gene or protein name hard to distinguish from the general language text surrounding it (*cysteine rich intestinal protein, d component of complement, never in mitosis*). Fortunately, there exists a fair amount of regularity in naming conventions, making automated methods possible.

The methods proposed for identifying gene and protein names in biomedical documents vary in their degree of reliance on dictionaries, statistical or knowledge-based approaches, manual versus automatic rule generation and ability to extract compound names. Most approaches require the use of a part-of-speech (POS) tagger, which is considered to be the most basic form of linguistic corpus annotation. The goal of a POS tagger is to automatically assign each word in a piece of text its part of speech, a task that can be easily and accurately performed on a computer. For the purpose of gene/protein name extraction, POS information can also be used in rule-based systems for rule conditions and/or error recovery, or as features in machine learning algorithms, for example, in Hidden Markov Models (HMMs) or decision trees. Lastly, POS information can aid in further linguistic processing including parsing and semantic interpretation.

One rule-based method, PROPER (PROtein Proper-noun phrase Extracting Rules), uses surface clues like capital letters, numbers and symbols to extract core terms for potential protein names which are later connected to other terms in the surrounding text (Fukuda *et al.*, 1998). The PROPER system does not rely on a dictionary, uses manually generated rules and can recognize compound words. It uses a POS tagger for rule conditions. A different approach uses a morphological POS tagger and a disambiguation program based on Hidden Markov Models (HMM) to find gene names in text by process of elimination. A sentence is split into tokens that are recognized by the morphological analyzer. If a term is not identified, it is assigned the *Guessed* tag and sent to the HMM disambiguator. This method uses a dictionary of general biological terms, a series of manually generated error recovery and contextual analysis rules and does not identify compound words (Proux *et al.*, 1998).

Some researchers have taken a machine learning approach to gene/protein name extraction. One group uses an HMM trained on word features like digits, single capital letters, Greek letters and symbols (Collier *et al.*, 2000). Their method uses no dictionaries or handcrafted rules, and can detect compound words. They found that POS information did not significantly help their program's performance. The Naïve Bayes and decision tree methods have also been tried (Nobata *et al.*, 1999). In the statistical Bayesian approach, they used no POS information, and in the decision tree they used POS tags along with surface clues to build feature vectors. In both methods, they distinguished between classification (SOURCE, PROTEIN, DNA or RNA) and identification of the name. They found that identification was a much harder task than classification, due to the compositionality of terms according to a domain-specific model, and acknowledged that the absence of high level biological knowledge was a significant impediment to better performance of both methods. The Bayesian model used a small lookup list of gene-related headwords, and the decision tree used no dictionary. Both methods retrieved compound words.

EDGAR (Extraction of Drugs, Genes and Relations) is a more linguistically motivated system that uses an HMM tagger with an under specified syntactic parser and extensive knowledge resources to identify single and compound word gene names (Rindflesch *et al.*, 2000). The EMPathIE (Enzyme and Metabolic Pathways Information Extraction) and PASTA (Protein Active Site Template Acquisition) projects use dictionaries, context-free grammars and semantic interpretation to extract single and compound enzyme and protein names (Humphreys *et al.*, 2000). Their approach formalizes handcrafted and semi-automatically generated rules into phrasal grammars for enzyme and protein names.

In this paper, we show that a rule-based POS tagger can be trained to extract gene and protein names from MEDLINE abstracts. We extend the capabilities of the POS tagger by adding a GENE tag to the list of familiar part-of-speech tags available to the tagger, and training the tagger to recognize candidates for this new tag. We describe the rules that are automatically generated from our training set, followed by details of post-processing strategies to decrease false positives and false negatives. We present results and evaluation of an experiment conducted on a test set of 56 469 MEDLINE documents.

METHODS

We use a combination of statistical and knowledge-based strategies to extract gene and protein names from MEDLINE abstracts. First we apply automatically generated rules from the Brill POS tagger (Brill, 1994) to extract single word gene and protein names. These results are then filtered extensively using manually generated rules formed from morphological clues, low frequency trigrams, indicator words, suffixes, and part-of-speech information. A key step during this process is the extraction of multi-word gene and protein names that are prevalent in the literature but inaccessible to the Brill tagger. Finally, we apply Bayesian learning to rank the documents by similarity to documents with known gene names and show the effect of an assumption that documents below a certain threshold do not contain gene/protein names.

Training the transformation-based POS tagger

The Brill tagger is a transformation-based POS tagger that uses error-driven learning to induce rules for tagging parts of speech in text. It assigns each word its most likely tag if it is in the lexicon, and assumes that unknown words are nouns. Lexical rules are applied using prefixes, suffixes, infixes and word bigrams to challenge the first assignments. Contextual transformations improve the accuracy of the tagger.

We trained the Brill tagger to recognize protein and gene names in biomedical text. A set of 7000 sentences was hand-tagged using the new tag GENE. We updated the lexicon included in the Brill package (Brown Corpus plus Wall Street Journal corpus) with entries from the UMLS SPECIALIST lexicon (McCray *et al.*, 1994; Humphreys *et al.*, 1998), and generated a list of bigrams and a word list from all of MEDLINE to customize the training for our purposes. Training produced 78 new lexical rules and 81 new contextual rules involving the GENE tag. Examples of each type of rule are given in Table 1.

Post-processing of Brill output

The Brill tagger produces tagged text from untagged text. We use a variety of data sources to filter false positive and false negative gene and protein names from the Brill output. False negatives are given new tags to distinguish them from those found by the Brill tagger that carry the GENE tag. The order of filtering is: (1) eliminate false

Table 1. Examples of lexical and contextual rules learned by the Brill tagger NNP = proper noun, CD = cardinal number, CC = coordinating conjunction, JJ = adjective, VBG = verb, gerund/present participle

Lexical rule	**Description**
NNP gene fgoodleft GENE	*Change the tag of a word from NNP to GENE if the word gene can appear to the right*
-A hassuf 2 GENE	*Change the tag of a word from anything to GENE if it contains the suffix -A*
c- haspref 2 GENE	*Change the tag of a word from anything to GENE if it contains the prefix c-*
GENE cell fgoodright NNP	*Change the tag of a word from GENE to NNP if the word cell can appear to the left*
Contextual rule	**Description**
NNP GENE PREV1OR2WD genes	*Change the tag of a word from NNP to GENE if one of the two preceding words is genes*
NNP GENE NEXTBIGRAM (GENE	*Change the tag of a word from NNP to GENE if the two following words are tagged (and GENE*
CD GENE SURROUNDTAG CC)	*Change the tag of a word from CD to GENE if the preceding word is tagged CC and the following word is tagged)*
VBG JJ NEXTTAG GENE	*Change the tag of a word from VBG to JJ if the next word is tagged GENE*

positive names; (2) recover false negative names by lexical lookup or trigram matching and tag them NEWGENE; (3) recover false negative compound names and tag each component MULTIGENE; and (4) recover false negative names by applying non-specific contextual rules after retrieving as many false negatives as possible and tag them CONTEXTGENE. In this section we explain each filtering step in greater detail, and give examples of the data sources required for that step.

Eliminate false positive names. During this step, the GENE tag is removed from a word if it matches a term from a list of 1505 precompiled general biological terms (acids, antagonist, assembly, antigen, etc.), 39 amino acid names, 233 restriction enzymes, 593 cell lines, 63 698 organism names from the NCBI Taxonomy Database (Wheeler *et al.*, 2000) or 4357 non-biological terms. Non-biological terms were obtained by comparing word frequencies in MEDLINE versus the Wall Street Journal (WSJ) using the following expression, where p is the probability of occurrence:

$$\log(p(\text{word occurs in MEDLINE})/ \\ p(\text{word occurs in WSJ})) < 1$$

Table 2. Examples of theme terms that appear directly before or after a gene name

Score range	**Before gene**	**After gene**
3.0–4.0	*Gene, truncated, express, recombinant, purified, chimeric*	*Transcript, mrnas, deficient, homolog, locus, deletion, transfected, encodes*
2.0–3.0	*Bind, activate, protein, inducible, rat, amplified, monomeric, factor*	*Regulates, translocates, heterodimers, binds, allele, plays, fusion*
1.0–2.0	*Oncogenic, fragment, subtypes, mutation, minimal, enzyme, avian, transporters*	*Levels, complex, domain, activator, stimulates, probe, isotype, antigen, region*

Additional false positives are found by regular expression matching to patterns indicating that the word is not a gene/protein name. These patterns include numbers followed by measurements (25 mg/ml) and common drug suffixes (-ole, -ane, -ate, -ide, -ine, -ite, -ol, -ose, cooh).

Recover false negative names by lexical lookup. We obtained a compilation of gene names for multiple organisms from LocusLink (Pruitt and Maglott, 2001) and The Gene Ontology Consortium (2000), including 34 555 single word names and 7611 compound word names that comprise our gold standard for gene names. During lexical lookup, we require that a single word name appear in a particular context due to ambiguity problems. Since compound gene names have less ambiguity, there is no similar contextual requirement.

The context words chosen to disambiguate single word gene names were automatically generated by a probabilistic algorithm. This requires a large set of known gene names (we use our gold standard set) and a large collection of texts in which these gene names occur. We computed a log odds score or Bayesian weight for all non-gene name words indicating their propensity to predict an adjacent gene name in the texts. We chose 1083 positive scoring words that could be useful for disambiguating single word gene names during lexical lookup. We refer to these positive scoring words as gene theme terms. If a text word matches a known gene name, its tag is changed to NEWGENE if it is preceded or followed by a gene theme term. A selection of gene theme terms is shown in Table 2.

Recover false negative names by trigram matching. We used a frequency distribution of trigrams of 500 000 terms occurring at least 3 times in MEDLINE to pick up potential gene/protein names overlooked by the Brill tagger. If a term contains one of 20 173 low frequency trigrams, it becomes a candidate for the NEWGENE tag. This is justified by the fact that many gene and protein names contain unusual trigrams that do not occur frequently in MEDLINE. However, other biological entities may also contain low frequency trigrams. For example, the low frequency trigram *jtc* occurs in the cell name *JTC-15*. To minimize these errors, we require that the trigram-containing word is either preceded or followed by a gene theme term.

Recover false negative compound names. This important third step involves assembling multi-word gene/protein names by applying manually generated rules to recognize where the name begins and ends. The Gene Ontology names comprise our gold standard for gene/protein names, because they represent multiple organisms' naming conventions. The compound names are particularly valuable because they give clues about the types of patterns that occur in gene/protein names. We observed that recombination of terms and/or morphological patterns can yield many different gene/protein names. Therefore, we start with a pool of terms and regular expressions that occur frequently in our gold standard and allow for recombination of these terms and patterns to identify new candidates.

From the gold standard, we compiled a set of 415 terms that occur frequently in gene names. These terms include the digits 1–9, the letters a–z, the roman numerals, the Greek letters, functional descriptors (*adhesion, channel, coagulation, filament, junction, differentiation)*, organism identifiers (*feline, hamster, rabbit)*, activity descriptors *(regulated, releasing, promoting, stimulating)*, placement indicators *(early, central, downstream, epidermal, heart, liver)*, and generic descriptors *(light, major, non, red, small, smooth).* In addition to the 415 exact terms, we added regular expressions that allow for partial matches or special patterns such as words without vowels, words with numbers and letters, words in capital letters, and common prefixes and suffixes *(-gene, -like, -ase, homeo-)*.

Potential gene/protein names that match any of 312 stop words or 4357 non-biological terms (described previously in the false positive recovery section) are immediately eliminated as candidates. A potential gene/protein name begins with one of the matched terms or patterns and continues word by word until there is no match. At this point, the potential name is validated or rejected by applying a series of rules. If a rejecting rule is triggered, either the candidate is removed completely, or it is revised and re-evaluated. Criteria for complete removal include verbs followed by numbers, numbers followed by measurement terms, amino acid names, pH or pI numbers, molecular weights, IC_{50} numbers, and ATCC numbers. Criteria for revision include verbs at the beginning of the name, words ending in –ing or –ed at the end of the name and general biological terms in the name *(bacteria, base, dimers*).

Table 3. Contextual rules and constraints that will change the tag of x to CONTEXTGENE. ANYGENE = GENE, NEWGENE or MULTIGENE, CC = coordinating conjunction, x = current word, y = word near x

Contextual Rule	Constraints
ANYGENE , x y	x not a verb or adverb, y not a verb or symbol
x CC ANYGENE .	x must be a noun, adjective, cardinal number or preposition
ANYGENE CC x .	x contains a capital letter, dash or number and is not a verb or adverb
ANYGENE , CC x	x contains a capital letter, dash or number and is not a verb or adverb
x (ANYGENE y	x is not a verb or adverb, y is not a date, 'et', 'and', ',' or '='
ANYGENE (x y	x is not a verb or adverb and contains a capital letter, dash or number, y is not a date, 'et', 'and', ',' or '='
ANYGENE x)	x is not a verb or adverb, y is not a date, 'et', 'and', ',' or '='
x ANYGENE)	x is not a verb or adverb
x , ANYGENE	x is not a verb or adverb

The final validation step involves the last word in the name. This word has a more restrictive set of allowable terms/patterns than the other positions because many words that may occur within a name *(activating, feline, smooth)* are unacceptable at the end of a name. Thus we require that the last word in a name be a noun or contain special characters and/or numbers. As a final check, we employ the false positive filters of cell lines, organisms, restriction enzymes, and general biological terms to the gene/protein name. If the name passes, each separate word in the compound name is given the tag MULTIGENE.

Recover false negative names by context. After the NEWGENE and MULTIGENE tags have been assigned, a final pass is performed to pick up additional gene/protein name candidates using a small set of contextual rules. The CONTEXTGENE tag depends on the accuracy of the GENE, NEWGENE and MULTIGENE tags, which can cause cascading error problems. These problems arise because the rules are generic and simple, thus constraints are needed to lower the risk of carrying through an error from an incorrectly tagged GENE, NEWGENE or MULTIGENE to an error in the CONTEXTGENE tag.

An additional option to decrease errors is to check the potential CONTEXTGENE word for poor indicator suffixes. By calculating probability values based on the hypergeometric distribution of words that occur as head words (last position) of UMLS terms and those that do not, we concluded that the following suffixes are poor indicators of gene/protein names: *ar, ic, al, ive, ly, yl, ing, ry, ian, ent, ward, fold, ene, ory, ized, ible, ize, izes.* Other poor indicator suffixes are common drug suffixes and the very common suffixes *ed, tion, ity, ure, ence.*

During the CONTEXTGENE pass, we keep track of words that have been confirmed as gene/protein names, so that within a document, as long as the CONTEXTGENE name has been confirmed once, it is assumed to be confirmed throughout, without needing to trigger any rules. This automatic confirmation is done to pick up occurrences of the gene/protein name that do not occur in a typical gene/protein context. The assumption is that if the name has occurred in a gene/protein context at least once in the document, then it is safe to tag it as a gene/protein elsewhere in the document even if it does not occur in a gene/protein context. In other words, the gene/protein name has already 'proven itself.' We do this on a document boundary to eliminate errors that would occur if the boundary were extended to more than one document.

Apply Bayesian learning to rank documents

While we use our extension of the Brill tagger and several rule-based processing steps to identify potential gene names based on morphology, context, and grammatical considerations, such processing does not take into account the general context of the whole document being processed. In order to take advantage of the general context of all the words in a document we use Naïve Bayes learning (Langley, 1996; Mitchell, 1997; Wilbur, 2000) to predict the overall likelihood that a document contains a gene name. First our gold standard set of gene names is used to find documents containing these names in all of MEDLINE. A Naïve Bayesian classifier is then trained to distinguish these documents from the remainder of MEDLINE. Examination of the results show that those documents that receive high scores from the classifier almost always contain gene names even if they are not names contained in the gold standard set of gene names used in selecting the training set. We are then able to make use of these scores to rate the chance that a document contains a gene name and to eliminate from consideration documents that almost certainly do not contain gene names (e.g. a score less than –20).

EXPERIMENT AND RESULTS

In this section, we describe an experiment to explore the utility of our approach to the task of finding gene/protein names in biomedical abstracts. The test set consists of the complete set of 56 469 abstracts introduced into MEDLINE between 15 June–24 September, 2001. No attempt was made to narrow the set using query terms. We ran the abstracts through the Brill tagger, false positive/negative filters and ranking procedure. We checked 100 random sentences out of every 50K sentences in the test set ordered by descending Bayes classifier score. Precision and recall results are shown for each individual score range in Table 4, and cumulative results are shown in Table 5. For a compound gene name to be considered a true positive, it must be both complete and accurate. For example, *histocompatibility complex genes* was tagged as a MULTIGENE, but not counted as a true positive because *major* was not included. General terms like *leukocyte RNA, growth factors,* and *N-linked glycosylation sites* were considered to be false positives. The heuristics to detect invalid combinations of gene components in compound word names are imperfect, allowing false positive gene names like *region containing a short sequence.* Other notable false positives include entities related to genes, for example, *plasmid pMC1403N* and *RNA phage Qbeta*.

Next we provide illustrative examples of the output obtained by applying our methods to the large test set. The examples are excerpts from randomly chosen abstracts in three score ranges: high (200s), medium (100s) and low (−20s). We give the interim outputs to show how the tags shift during each step.

High-scoring example, Score = 254.24

The results after applying the Brill tagger and false positive filter are:

> We/PRP conclude/VBP that/IN a/DT series/NN of/IN sites/NNS (/(NF-kappaB/GENE ,/, IRF/GENE ,/, GRE/NNP ,/, and/CC the/DT E/NN box/NN)/SYM are/VBP not/RB required/VBN for/IN efficient/JJ viral/JJ spread/NN in/IN the/DT sheep/NN model/NN ,/, although/IN mutation/NN of/IN some/DT of/IN these/DT motifs/NNS might/MD induce/VB a/DT minor/JJ phenotype/NN during/IN transient/JJ transfection/NN assays/NNS in/IN vitro/FW ./.

We see that the tagger has correctly identified *NF-kappaB* and *IRF*, but has missed *GRE* and the *E box*. After applying the false negative filter, we obtain:

> We/PRP conclude/VBP that/IN a/DT series/NN of/IN sites/NNS (/(NF-kappaB/GENE ,/, IRF/GENE ,/, GRE/CONTEXTGENE ,/, and/CC the/DT E/MULTIGENE box/MULTIGENE)/SYM are/VBP not/RB required/VBN for/IN efficient/JJ viral/JJ spread/NN in/IN the/DT sheep/NN model/NN ,/, although/IN mutation/NN of/IN some/DT of/IN these/DT motifs/NNS might/MD induce/VB a/DT minor/JJ phenotype/NN during/IN transient/JJ transfection/NN assays/NNS in/IN vitro/FW ./.

GRE has triggered a CONTEXTGENE rule, and *E box* has been extracted as a compound gene name.

Table 4. Precision and recall for each score range TP+FN = number of gene names; B = Brill tagger; B FP = Brill tagger and false positive filter; B FP FN (L/T) = Brill tagger, false positive filter, false negative filter using gene list and trigrams; B FP FN (L/T, M) = Brill tagger, false positive filter, false negative filter using gene list, trigrams and multiple word rules; B FP FN (L/T, M, C) = Brill tagger, false positive filter, false negative filter using gene list, trigrams, multiple word rules and final pass contextual word rules.

Score Range	*no. of words tested*	*TP + FN*	*B*	*B FP*	*B FP FN (L/T)*	*B FP FN (L/T, M)*	*B FP FN (L/T, M, C)*
−60 to −40	3403	0	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000	0.000 0.000
−40 to −20	12 490	51	0.084 0.156	0.200 0.146	0.319 0.288	0.347 0.463	0.386 0.593
−20 to 0	2760	59	0.409 0.143	0.700 0.125	0.632 0.211	0.633 0.322	0.657 0.371
0 to 20	1884	53	0.679 0.373	0.714 0.294	0.760 0.358	0.769 0.556	0.744 0.571
20 to 40	1362	52	0.640 0.333	0.810 0.321	0.778 0.396	0.778 0.528	0.789 0.566
40 to 60	1324	54	0.714 0.417	0.783 0.333	0.778 0.389	0.744 0.571	0.756 0.607
60 to 80	468	25	0.857 0.240	0.857 0.240	0.909 0.400	0.762 0.593	0.739 0.654
80 to 100	413	21	0.833 0.714	0.833 0.714	0.789 0.714	0.630 0.810	0.630 0.810
100 to 120	416	24	0.667 0.348	0.727 0.333	0.769 0.417	0.850 0.708	0.850 0.708
120 to 140	339	27	1.000 0.680	1.000 0.630	1.000 0.630	1.000 0.815	1.000 0.815
140 to 160	304	31	0.933 0.467	0.933 0.452	0.941 0.516	0.833 0.806	0.833 0.806
160+	101	9	1.000 0.417	1.000 0.444	1.000 0.444	0.857 0.667	0.857 0.667

Medium-scoring example, Score = 102.98

The results after applying the Brill tagger and false positive filter are:

> Furthermore/RB ,/, the/DT yeast/NN ELAV/NNP homolog/NN ,/, Pub1p/GENE ,/, regulates/VBZ the/DT stability/NN mediated/JJ by/IN the/DT TNFalpha/NN ARE/VBP ./.

We see that the tagger has correctly identified *Pub1p*, but has missed *TNFAlpha*. After applying the false negative filter, we obtain:

> Furthermore/RB ,/, the/DT yeast/MULTIGENE ELAV/MULTIGENE homolog/MULTIGENE ,/, Pub1p/GENE ,/, regulates/VBZ the/DT stability/NN mediated/JJ by/IN the/DT TNFalpha/NN ARE/VBP ./.

A compound name is found, *yeast ELAV homolog*, however, *TNFalpha* remains unchanged. This is due to the fact that, even though *TNFalpha* is a member of our gold standard, we required that a gene theme term occur directly before or after a word tagged NEWGENE. The compound name *TNFalpha ARE* is not found, because ‘are’ is on the stop list, immediately eliminating it as a MULTIGENE candidate.

Low-scoring example, Score = −20.538

The results after applying the Brill tagger are:

> NE/NN threshold/NN sensitivity/NN (/(pD(T20/GENE)/SYM =/SYM -log/CD of/IN 20%/CD response/NN dose)/NN was/VBD analyzed/VBN ./. pD(T20)/GENE was/VBD significantly/RB decreased/VBN in/IN A1/NN ,/, pA3/GENE ,/, and/CC dA3/CD of/IN 1-hit/CD 24-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ,/, and/CC was/VBD further/RB decreased/VBN in/IN all/DT vessels/NNS of/IN 2-hit/JJ 72-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ./.

We see that the tagger output for all three GENE tags is incorrect. After applying the false positive filter, we obtain:

Table 5. Cumulative precision and recall using the score as a lower threshold. B = Brill tagger; B FP = Brill tagger and false positive filter; B FP FN (L/T) = Brill tagger, false positive filter, false negative filter using gene list and trigrams; B FP FN (L/T, M) = Brill tagger, false positive filter, false negative filter using gene list, trigrams and multiple word rules; B FP FN (L/T, M, C) = Brill tagger, false positive filter, false negative filter using gene list, trigrams, multiple word rules and final pass contextual word rules

Score	B	*B FP*	*B FP FN (L/T)*	*B FP FN (L/T, M)*	*B FP FN (L/T, M, C)*
−60	0.482	0.684	0.693	0.671	0.673
	0.348	0.321	0.394	0.571	0.609
−40	0.523	0.703	0.708	0.683	0.684
	0.348	0.321	0.394	0.571	0.609
−20	0.729	0.823	0.810	0.771	0.769
	0.373	0.345	0.410	0.587	0.611
0	0.774	0.832	0.831	0.788	0.784
	0.424	0.386	0.448	0.639	0.660
20	0.795	0.853	0.844	0.791	0.792
	0.435	0.406	0.467	0.657	0.680
40	0.833	0.863	0.861	0.794	0.793
	0.462	0.429	0.487	0.692	0.711
60	0.878	0.889	0.889	0.811	0.806
	0.478	0.467	0.526	0.741	0.754
80	0.881	0.892	0.886	0.821	0.821
	0.532	0.518	0.554	0.777	0.777
100	0.898	0.915	0.922	0.886	0.886
	0.489	0.473	0.516	0.769	0.769
120	0.973	0.972	0.974	0.898	0.898
	0.537	0.522	0.552	0.791	0.791
140	0.950	0.947	0.952	0.838	0.838
	0.452	0.450	0.500	0.775	0.775
160	1.000	1.000	1.000	0.857	0.857
	0.417	0.444	0.444	0.667	0.667

> NE/NN threshold/NN sensitivity/NN (/(pD(T20/NN)/SYM =/SYM -log/CD of/IN 20%/CD response/NN dose)/NN was/VBD analyzed/VBN ./. pD(T20)/NN was/VBD significantly/RB decreased/VBN in/IN A1/NN ,/, pA3/GENE ,/, and/CC dA3/CD of/IN 1-hit/CD 24-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ,/, and/CC was/VBD further/RB decreased/VBN in/IN all/DT vessels/NNS of/IN 2-hit/JJ 72-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ./.

The filter has eliminated 2/3 false positives. Next we apply the false negative filter:

> NE/NN threshold/NN sensitivity/NN (/(pD(T20/NN)/SYM =/SYM -log/CD of/IN 20%/CD response/NN dose)/NN was/VBD analyzed/VBN ./. pD(T20)/NN was/VBD significantly/RB decreased/VBN in/IN A1/CONTEXTGENE ,/, pA3/GENE ,/, and/CC dA3/CONTEXTGENE of/IN 1-hit/CD 24-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ,/, and/CC was/VBD further/RB decreased/VBN in/IN all/DT vessels/NNS of/IN 2-hit/JJ 72-h/JJ septic/JJ rats/NNS (/(P/NN </SYM 0.05/CD)/SYM ./.

Here we see an example of cascading errors as pA3 gives rise to two additional errors, A1 and dA3. A simple way to decrease these errors is to restrict the CONTEXTGENE tag to abstracts with higher Bayesian scores, which we have found useful. Unfortunately, cascading errors can also occur in abstracts with high scores, due to Brill tagging errors that affect the triggering of rules. For example, our method extracts the incorrect gene name *inhibiting NF-kappaB* from the text fragment *down-regulate LPS-induced COX-2 expression by inhibiting NF-kappaB*. The tag provided by the Brill tagger for *inhibiting* is JJ (adjective). If the correct verbal tag had been found, the post-processing program would have removed the verb from the beginning of the name. To address this problem, we tested a Support Vector Machine (Platt, 1998, 1999; Burges, 1999) to disambiguate adjectives and verbs containing the suffix–*ing*. Using 268 examples of correctly tagged words and 424 examples of incorrectly tagged words, we were able to predict 77% of the correct and 93% of the incorrect training examples using a leave-one-out cross validation procedure.

RELATED WORK

Direct comparison of gene/protein name extraction methods is difficult because some of the methods distinguish between genes, proteins and enzymes and some do not, and there is wide variation in both the type and size of the test sets used by each group to evaluate their methods. In general, small, specialized test sets tend to perform better than large, general ones. For example, the PROPER system of Fukuda *et al.* achieves a precision value of 0.95 and recall of 0.99 using a set of 30 abstracts about the SH3 protein domain (Fukuda *et al.*, 1998). Proux *et al.* obtain a precision of 0.91 and recall of 0.94 using 1200 sentences from FlyBase where each gene name uses the correct gene symbol, and all sentences contain at least 2 gene symbols (The FlyBase Consortium, 1998). The precision ratio on a larger, more general set of 25K MEDLINE abstracts was 0.70, and the test set was constrained by the query 'Drosophila' (Proux *et al.*, 1998). Collier *et al.* achieve F-scores of 0.76, 0.47 and 0.03 for proteins, DNA and RNA respectively on a set of 100 abstracts controlled by the query 'human, blood cell, transcription factor' (Collier *et al.*, 2000). Nobata *et al.* achieve F-score ranges of 0.05–0.15 and 0.63–0.72 for DNA and PROTEIN, respectively, using the decision tree method, and 0.07–0.19 and 0.43–0.48 using the Naïve Bayes method on a set of 100 MEDLINE abstracts (Nobata *et al.*, 1999). Humphreys *et al.* achieve a precision of 0.87 and recall of 0.97 for proteins, 0.61 and 0.84 for sites and 0.44 and 1 for regions on a set of 52 abstracts using the PASTA system (Humphreys *et al.*, 2000). Rindflesch *et al.* (2000) use gene name extraction as part of a larger semantic interpretation, and no precision/recall values are available for this task alone.

CONCLUSION

Our gene and protein tagging method has some limitations, for example, it can miss single word gene names that occur without contextual gene theme terms. It can also incorrectly tag some related entities like plasmids and phages as genes. Our heuristics to detect invalid combinations of gene components in compound word names are imperfect. One of the most problematic results of the staged discovery of names during multiple passes is cascading errors. We can decrease these errors in abstracts unrelated to genes by applying a Bayesian score cutoff after which the contextual rules are discontinued. However, the problem also arises in abstracts that contain valid gene/protein names. Here we can stop the error cascade at its source, the POS tagger, by using a larger, more general training set for the Brill tagger, or by adding another step to identify incorrect tags used in post-processing rules. We have successfully experimented with the second alternative using a Support Vector Machine.

The extraction of gene and protein names from biological texts remains a challenging computational problem due to informal naming conventions and morphological similarity to other biological entities including cell line names. However, there is enough structure and regularity in the naming conventions, making automated methods possible. The results presented in this paper show that a rule-based POS tagger can be trained to automatically identify gene and protein names in biomedical text. We have also demonstrated the utility of using a combination of filters to pick up compound word names, using a variety of knowledge sources and manually generated rules. Our method can be used on large and unconstrained sets of MEDLINE documents.

ACKNOWLEDGEMENTS

We thank Allen C. Browne for providing the Penn Treebank tags for SPECIALIST lexicon entries, and Donald C. Comeau for the list of low frequency trigrams in MEDLINE.

REFERENCES

Brill,E. (1994) Some advances in transformation-based part of speech tagging. *Proceedings of the National Conference on Artificial Intelligence*. AAAI Press, pp. 722–727.

Burges,C.J.C. (1999) A tutorial on support vector machines for pattern recognition, Bell Laboratories, Lucent Technologies.

Collier,N., Nobata,C. and Tsujii,J. (2000) Extracting the names of genes and gene products with a hidden markov model. *Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000)*. pp. 201–207.

Fukuda,K., Tsunoda,T., Tamura,A. and Takagi,T. (1998) Toward information extraction: identifying protein names from biological papers. *Proceedings of the Pacific Symposium on Biocomputing (PSB98)*. pp. 705–716.

Humphreys,B.L., Lindberg,D.A., Schoolman,H.M. and Barnett,G.O. (1998) The Unified Medical Language System: an informatics research collaboration. *J. Am. Med. Inform Assoc.*, **5**, 1–11.

Humphreys,K., Demetriou,G. and Gaizauskas,R. (2000) Two applications of information extraction to biological science journal articles: enzyme interactions and protein structures. *Proceedings of the Pacific Symposium on Biocomputing (PSB2000)*. pp. 502–513.

Langley,P. (1996) *Elements of Machine Learning*. Morgan Kaufmann, San Francisco.

McCray,A.T., Srinivasan,S. and Browne,A.C. (1994) Lexical methods for managing variation in biomedical terminologies. *SCAMC '94*. pp. 235–239.

Mitchell,T.M. (1997) *Machine Learning*. WCB/McGraw-Hill, Boston.

Nobata,C., Collier,N. and Tsujii,J. (1999) Automatic term identification and classification in biology texts. *Proceedings of the Natural Language Pacific Rim Symposium*. pp. 369–374.

Platt,J. (1998) How to implement SVMs. *IEEE Intell. Syst.*, **13**, 26–28.

Platt,J.C. (1999) Fast training of support vector machines using sequential minimal optimization. In Scholkopf,B., Burges,C.J.C. and Smola,A.J. (eds), *Advances in Kernal Methods*. MIT Press, Cambridge, MA, pp. 185–208.

Proux,D., Rechenmann,F., Julliard,L., Pillet,V. and Jacq,B. (1998) Detecting gene symbols and names in biological texts: a first step toward pertinent information extraction. *Proceedings of the Ninth Workshop on Genome Informatics*. pp. 72–80.

Pruitt,K.D. and Maglott,D.R. (2001) RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Res.*, **29**, 137–140.

Rindflesch,T.C., Tanabe,L., Weinstein,J.W. and Hunter,L. (2000) EDGAR: extraction of drugs, genes and relations from the biomedicalliterature. *Proceedings of the Pacific Symposium on Biocomputing (PSB2000)*. pp. 514–525.

The FlyBase Consortium (1998) FlyBase—A *Drosophila* database. *Nucleic Acids Res.*, **26**, 85–88.

The Gene Ontology Consortium (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Wilbur,W.J. (2000) Boosting naive bayesian learning on a large subset of MEDLINE. *American Medical Informatics 2000 Annual Symposium*. Los Angeles, CA, pp. 918–922.

Wheeler,D.L., Chappey,C., Lash,A.E., Leipe,D.D., Madden,T.L., Schuler,G.D., Tatusova,T.A. and Rapp,B.A. (2000) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **28**, 10–14.