Analysis of Biomedical Text for Chemical Names: A Comparison of Three Methods

W. John Wilbur[†], MD, PhD, George F. Hazard, Jr.[‡], PhD, Guy Divita[§], MS, James G. Mork[§], MS, Alan R. Aronson[§], PhD, Allen C. Browne[§], MS

[†]National Center for Biotechnology Information (NCBI),
[‡]Division of Specialized Information Services (SIS),
[§]Lister Hill National Center for Biomedical Communications (LHNCBC)
National Library of Medicine, Bethesda, MD 20894

At the National Library of Medicine (NLM), a variety of biomedical vocabularies are found in data pertinent to its mission. In addition to standard medical terminology, there are specialized vocabularies including that of chemical nomenclature. Normal language tools including the lexically based ones used by the Unified Medical Language System® (UMLS®) to manipulate and normalize text do not work well on chemical nomenclature. In order to improve NLM's capabilities in chemical text processing, two approaches to the problem of recognizing chemical nomenclature were explored. The first approach was a lexical one and consisted of analyzing text for the presence of a fixed set of chemical segments. The approach was extended with general chemical patterns and also with terms from NLM's indexing vocabulary, MeSH®, and the NLM SPECIALIST™ lexicon. The second approach applied Bayesian classification to n-grams of text via two different methods. The single lexical method and two statistical methods were tested against data from the 1999 UMLS Metathesaurus®. One of the statistical methods had an overall classification accuracy of 97%.

INTRODUCTION

Chemical nomenclature is used to represent a chemical concept in text. Systematic nomenclature is highly conjunctive, in that a single unbroken string can contain multiple concepts. Programs to un-embed important chemical concepts were originally developed for printed indexes[1]. This concept was extended later to online nomenclature retrieval, using a new algorithm[2]. Recently a method to detect chemical names in SGML patent text using segments and statistical methods has been described[3].

The UMLS Metathesaurus contains over 350,000 *chemical* English terms represented by a variety of types of nomenclature. There are systematic names such as *1,2-dimethoxyethane*, which describe the chemical structure, as well as generic drug names such as *zidovudine*, trade names such as *Maximum Strength Bayer Aspirin Caplets*, company codes such as *SKF-98625*, and formulations such as *Zovirax 250mg i-v infusion (pdr for recon)*.

The Natural Language Systems (NLS) program at NLM has produced a variety of tools which process text. These tools include Lexical Variant Generator (LVG)[4] which allows abstracting away from lexical variation and MetaMap[5] which maps free text to concepts in the Metathesaurus. The tools are oriented toward standard medical terminology and do not handle the manipulation of chemical names well. Consider the text … *the effect of the adenosine receptor agonist 5'-(N-ethylcarboxamido)adenosine (NECA)* …. MetaMap fragments the chemical into three phrases, *5'-*, *(N-ethylcarboxamido* and *)adenosine* because of the embedded parentheses. The fragmentation prevents adequate search for matching concepts in the Metathesaurus. Even if the fragmentation were prevented, matching would be difficult because the closest Metathesaurus string to the text is *5'-N-Ethylcarboxamidoadenosine* which has no parentheses. The presence of the acronym *NECA* in the text facilitates mapping since *NECA* does occur in the Metathesaurus, but proper handling of the full chemical name is still required.

NLS projects have begun collaborative efforts to improve the treatment of chemicals. The long-term goal is to correctly classify chemical terms occurring in text for use in phrase extraction, indexing (of both Metathesaurus concepts and bibliographic citations), synonym recognition and other text analysis applications. The immediate goal is to automatically recognize chemical terms in order to avoid subjecting them to inappropriate processing.

One approach to classifying chemical terms is based on the segmentation algorithms described above. The idea is to classify chemical terms by eliciting their chemical structure based on chemically meaningful segments. This approach requires *a priori* identification of segments that might be found in chemical nomenclature.

Another approach to the chemical name classification problem is to use naive Bayesian classifiers[6,7,8]. Such an approach has the advantage of requiring no *a priori* knowledge of chemical name characteristics. If selected attributes occur independently in data, a naive Bayesian classifier will give ideal performance. Of course it is possible for attributes to have strong interdependencies, and in that case one may obtain better performance from a classifier of the rule-based type. But we have chosen the attributes to have more of a soft or statistical character.

METHODS

The Segmentation Approach (SEG method)

Initially, we decided to exploit the structure of chemical terms, analyzing them into their constituent chemical morphemes. We established a list of chemical morpheme segments and used the algorithm described in the *Registry File Basic Name Segment Dictionary*[2] to analyze chemical terms into constituents. The algorithm matches the longest left-most segment and proceeds across the term from left to right. If a term is analyzable into known chemical segment morphemes we can with a high level of confidence identify the term as a chemical term. For example *Triethylaminopropylisothiuronium* is analyzed into 8 constituent morphemes: TRI-ETHYL-AMINO-PROPYL-ISO-THI- URON-IUM.

The resulting segmentation algorithm does not handle generic and trade names well, since they are not fully constructed of significant systematic chemical segments. To remedy that situation the morpheme list was augmented with a list of biomedically significant segments an example of which is the segment *stigmast* representing the systematic parent of the *Sitosterol* class. The resulting list consists of 3,724 morpheme segments. We also created a supplemental list of chemical terms from 84,453 single-word MeSH chemical terms. In addition, pattern matching with regular expressions was used to identify recurring patterns such as numerical locants. These modifications allowed us to handle semi-systematic names such as *3’,5’-dichloromethotrexate* where *3',5'* is a locant pattern, *di* and *chloro* are systematic segments, and *methotrexate* is a generic drug name from MeSH. Pattern matching was also used to identify dosage and measurement patterns and other possibly nonchemical constituents of terms. A variety of other heuristics have been used as well. For example we have augmented the approach by identifying constituents not otherwise classified using information from the SPECIALIST lexicon in an effort to identify nonchemical components embedded within chemical terms. However, the lexicon also contains some chemicals which have been marked as such through a variety of methods. A final heuristic was added that consists of consulting a small list of terms such as *disease* and *syndrome* which completely disqualify a term being considered a chemical. The result of this segmentation provides both a lexical analysis of chemical terms and the means to classify them.

Once a term has been segmented, the segmentation algorithm assigns a score to each term representing the degree to which the term is a chemical. The scoring function has three components: provenance, cohesiveness, and coverage. Provenance computes the number of known chemical segments in a term. Segments from the chemical morpheme list and its supplemental lists give a term a higher provenance score. Certain patterns identified by regular expression matching also contribute to the provenance score. It is characteristic of chemical terms to contain internal punctuation. Provenance scores are therefore adjusted to take into account the amount of punctuation in a term. Cohesiveness and coverage are notions taken from the MetaMap algorithm[5]. Cohesiveness measures the maximum number of contiguous segments and coverage measures how many of the segments in a term are classifiable. The final score is (1/6 Coverage) + (1/6 Coherence) + (2/3 Provenance) yielding a value between 0 and 1. The provenance score, which is central to our analysis, has twice as much weight as coherence and coverage combined.

The Bayesian Classifier Approach (POS and TOTAL methods)

We have implemented and tested the Bayesian classifier in two different forms and we will describe the methodology in terms of what they have in common and then how they differ. All implementations depend on two parameters. One parameter is a small positive integer *n* which must be fixed before processing begins. It determines the *n*-gram size used in producing attributes. When *n* has been fixed, any string *STR* in the data set is processed as follows.

1) *STR* is lowercased.

2) *STR* is broken into terms at spaces and these individual terms are used to produce *n*-grams. Strings of length *n*+*k* produce *k*+1 overlapping *n*-grams, while any string of length *n* or shorter is taken as the only *n*-gram produced (for simplicity we shall refer to it as an *n*-gram even if shorter than *n*). All such *n*-grams are attributes of *STR*.

3) The first *n*-gram produced from each term derived from *STR* is marked at the right end by adding the character '!' and is included as an attribute.

As an example suppose $n = 4$. Then if *STR* is the string *1-methyl MB*, it has attributes: *1-me*, *-met*, *meth*, *ethy*, *thyl*, *1-me!*, *mb* and *mb!*.

Once the attributes for all strings to be processed have been assigned, each attribute is assigned a weight based on the Bayesian formalism. Let n_c denote the number of strings that are classed as chemical names in the training set and let $n_{\bar{c}}$ denote the number of strings that are not classed as chemical names. Let *s* be an arbitrary attribute and suppose that in the training set n_{cs} denotes the number of chemical name strings that have the attribute and $n_{\bar{c}s}$ the number of nonchemical name strings that have the attribute. Then the weight assigned to the attribute *s* is given by

$$w_s = \log(p(1-q)) - \log(q(1-p))$$

where we define

$$p = (n_{cs} + \delta)/(n_c + 2\delta)$$
$$q = (n_{\bar{c}s} + \delta)/(n_{\bar{c}} + 2\delta) \quad .$$

Here δ implements uninformed priors[6] and is the second parameter that must be set in order to define the Bayesian classifier.

In addition to the parameters *n* and δ which must be chosen, we have implemented the Bayesian classifier in what we may call two flavors. One is just as described and all attributes are weighted whether they receive positive or negative weights. We will refer to this as the TOTAL method. It is important to note that in this approach the nonchemicals in the training set are just as important as the chemicals in discrimination between the two. Thus one may only expect to achieve top performance if the classifier is used to discriminate between chemicals names and strings which are something like the nonchemical strings in the training set. Because the world of nonchemical strings is much larger than the world of chemical strings and one may not be able to give prior characterization to the environment in which one may wish to detect chemical names, we also looked at a version of the classifier that only allows positive weights. In this version only the attributes that are more probable in the set of chemical names are weighted and all other attributes are given zero weight by default. In order to compensate somewhat for the lack of negative weights we treat each string as a document and the attributes as key terms and produce a vector length in the standard way employed in vector document retrieval[9]. The Bayesian score for a string is then divided by the vector length associated with that string in order to produce a final score for ranking purposes. This implementation we call the POS Bayesian classifier.

EVALUATION

Evaluation of the three methods was performed by constructing training and testing sets of both chemicals and nonchemicals from the strings in the 1999 UMLS Metathesaurus. First the set of (English) strings was divided into chemicals and nonchemicals according to semantic type. A string was considered to be chemical if it either had semantic type 'Clinical Drug' (a child of 'Manufactured Object') or was a descendent of 'Chemical' in the semantic hierarchy. Four semantic types below 'Chemical' were excluded because of their lack of chemical relevancy. Strings of type 'Chemical Viewed Functionally', for example, include *Lipstick* and *P&S Shampoo*; and 'Immunologic Factor' strings include *HLA-Cw9 antigen* and *Cryoproteins*. The list of semantic types defining the set of chemicals follows with the excluded semantic types lined through:

Chemical
  ~~Chemical Viewed Functionally~~
    Pharmacologic Substance
      Antibiotic
    ~~Biomedical or Dental Material~~
    Biologically Active Substance
      Neuroreactive Substance or Biogenic Amine
      Hormone
      Enzyme
      Vitamin
      ~~Immunologic Factor~~
      ~~Receptor~~
    Indicator, Reagent, or Diagnostic Aid
    Hazardous or Poisonous Substance
  Chemical Viewed Structurally
    Organic Chemical
      Nucleic Acid, Nucleoside, or Nucleotide
      Organophosphorus Compound
      Amino Acid, Peptide, or Protein
      Carbohydrate
      Lipid
        Steroid
        Eicosanoid
    Inorganic Chemical
  Element, Ion, or Isotope
…
Clinical Drug

The semantically determined chemical and nonchemical sets were each randomly divided into training and testing sets, 2/3 for training and 1/3 for testing. This produced a *Full Training Set* and a *Full Testset*. The two statistical methods were trained using the Full Training Set. Because the SEG method was developed using the entire 1998 Metathesaurus, however, we created a *99 Only Testset* removing strings that occurred in the 1998 Metathesaurus from the Full Testset. This ensured that the SEG method could be tested fairly with data it had not seen before.

RESULTS

All three methods produce a score for each candidate string and a threshold must be chosen above which a score is used to classify a string as a chemical. For purposes of testing and comparison of methods as presented here, thresholds were chosen to minimize the overall error rate for both chemicals and nonchemicals. The Bayesian methods also require the setting of the n-gram size *n* and the prior confidence level δ. The optimal value of *n* was found to be 4 for the POS method and 7 for the TOTAL method. For both of these methods a δ of 0.01 proved optimal or near optimal and was used.

For completeness all methods were tested against both the Full and 99 Only Testsets. The full testset contained 118,034 chemicals and 210,898 nonchemicals. Each of the three methods correctly identified at least 84% of the chemicals and 87% of the nonchemicals (see Table 1) with the TOTAL method performing significantly better than the other methods. It correctly identified 96% of the chemicals and 97% of the nonchemicals.

	SEG Method	**POS Method**	**TOTAL Method**
Chem Found	99,649	103,180	113,571
% Found	84.4%	87.4%	96.2%
Nonchem Found	182,388	197,734	204,488
% Found	86.5%	93.8%	97.0%
Found Wtd. Avg.	85.7%	91.5%	96.7%
Missed Wtd. Avg.	14.3%	8.5%	3.3%

Table 1: Results of Full Testset

Similarly, the 99 Only Testset contained 35,113 chemicals and 44,321 nonchemicals. Each of the three methods correctly identified at least 84% of both the chemicals and nonchemicals (see Table 2) with all three methods scoring somewhat less on nonchemicals than in the Full Testset. Overall performance for the SEG and POS methods declined slightly, but TOTAL's overall performance remained the same due to an increase in its performance for chemicals. The TOTAL method was still the overall best scoring method.

	SEG Method	**POS Method**	**TOTAL Method**
Chem Found	29,494	31,951	34,102
% Found	84.0%	91.0%	97.1%
Nonchem Found	37,137	40,146	42,700
% Found	83.8%	90.6%	96.3%
Found Wtd. Avg.	83.9%	90.8%	96.7%
Missed Wtd. Avg.	16.1%	9.2%	3.3%

Table 2: Results of 99 Only Testset

DISCUSSION

For both testsets all three classification methods provide a high level of accuracy. The TOTAL method clearly achieves the best results for classifying terms and is likely to be useful for both indexing and retrieval of such terms as well as detecting chemicals in free text. However the segmentation approach offers a lexical analysis of chemical terms which can support tasks in which chemical nomenclature is important. These tasks include recognizing synonyms of a given chemical and normalization of chemical terms.

The most important observation regarding the Bayesian methods was that a δ of only 0.01 gave a significant boost to performance when compared with a more conventional choice. The usual interpretation of δ is a number of prior observations so that it would be set to a small positive integer, frequently 1[6]. The use of 0.01 has essentially no effect in the calculations unless either n_{cs} or $n_{\overline{cs}}$ is zero. If, for example $n_{cs} = 0$, it has the effect of adding the value $\log(\delta)$ to the weight one would have obtained when $n_{cs} = 1$. This abrupt change in the weight is a form of soft rule that says if an attribute is encountered that was never seen in a chemical, then it is probably not a chemical and the score should undergo a quantum decrease. Likewise when an attribute is observed that was never seen in a nonchemical in training, the score should undergo a quantum increase by $-\log(\delta)$. With almost 1 million strings in the Metathesaurus, when *n* is 4 there are just over 440 thousand attributes and when *n* is 7 there are over 1.3 million attributes. Thus it is not easy to hand code all the rules that might be useful in distinguishing chemicals from nonchemicals. The soft rules are automatically in effect in the Bayesian classifiers and allow for some rule like behavior which proves beneficial.

Failure analysis

The results of a preliminary failure analysis are shown in Table 3. The table shows the number of incorrectly identified chemicals and nonchemicals for each method together with the number of failures unique to the method. The last row gives the number of failures common to all methods. 266 chemicals were not identified as such by any method. Ninety-eight of these had the semantic type ‘Pharmacologic Substance’. *Agents for alcohol related cognitive impairment* is an example of such a missed chemical. Sixty-four cases had the semantic type ‘Organic Chemical’. Such organic chemicals as *Jim’s juice* and *Devil’s Red* were missed. Examples like these are difficult to detect because although they represent chemicals they do not have the characteristic pattern of chemical terms. Some of the sixty-four organic chemicals, *IS 145* for instance, involved English words used as acronyms.

Method	Chemicals Not Identified		Nonchemicals Identified as Chemicals	
	Total	Unique	Total	Unique
SEG	5,619	4,656	7,184	5,107
POS	3,162	1,893	4,175	1,892
TOTAL	1,011	274	1,621	601
All Methods	266		474	

Table 3. Errors (99 Only Testset)

Many of the failures of the SEG and POS methods were terms composed of two-, three- and four-character segments that are acronyms and abbreviations. *CyH-CHID* is an example.

The TOTAL method (using both positive and negative evidence) failed to recognize some chemical terms involving dosages or units of measurement such as *CYCLOSERINE 250 MG capsule*. It also failed for some terms such as *Somnifacient* that the SEG method retrieved because the terms appeared on one of the supplemental lists.

All methods had problems identifying terms denoting allergies (e.g., *Fludrocortisone allergy*) as non chemicals. Over half of the shared failures were of this variety. Terms involving complex punctuation and subterms which denote chemicals caused problems for both the SEG and POS methods. The terms *AMYLASE.S1:CCNC:PT:SER:QN* and *Accid pois - petroleum naphtha* are examples. The TOTAL method had less trouble with this sort of example. Terms that involved overdoses and terms that involved intentions were also missed by the SEG method. One example is *Piracetam overdose of undetermined intent*.

Future work

A straightforward way of taking advantage of the results described here is to add the TOTAL classification method to MetaMap’s tokenization algorithm. This is actually being done as part of an effort to recognize higher-order tokens of various types including author-defined acronyms and chemicals.

A more ambitious extension of this work is to combine the strengths of the Bayesian classification and segmentation approaches. We believe that a combined approach would enhance recognition of chemical terms while retaining the segmentation analysis which has potential applications to our text analysis efforts. We are investigating techniques for appropriate lexical normalization of chemical terms based on segmentation. We are also investigating techniques to discover the bounds of chemical terms so that they can be recognized in free text. Recognition of parent substituent and modifier segments of chemical terms within the segmentation analysis is feasible and should enable us to recognize synonymy between chemical terms.

References

1. Heym DR, Siegel H, Steensland MC, and Vo HV. Computer Recognition and Segmentation of Chemically Significant Words for KWIC Indexing. *J. Chem. Inf. Comput. Sci*., 16:173-176, 1976.

2. Chemical Abstracts Service. *Registry File Basic Name Segment Dictionary.* June, 1993.

3. Kemp N and Lynch M. Extraction of Information from the Text of Chemical Patents. 1. Identification of Specific Chemical Names. *J. Chem. Info. Comput. Sci*., 38:544-551, 1998.

4. McCray AT, Srinivasan S, and Browne AC. Lexical methods for managing variation in biomedical terminologies. In Ozbolt JG (ed.) *Proceedings of the 18th Annual SCAMC*, 235-239, 1994.

5. Aronson AR, Rindflesch TC, and Browne AC. Exploiting a large thesaurus for information retrieval. *Proceedings of RIAO 94*, 197-216, 1994.

6. Langley P. *Elements of Machine Learning*. Morgan-Kaufmann Publishers, Inc., San Francisco, CA, 1996.

7. Langley P and Sage S. Induction of selective Bayesian classifiers. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 399-406, 1994.

8. Langley P, Iba W, and Thompson K. An analysis of Bayesian classifiers. *Proceedings of the Tenth National Conference on Artificial Intelligence*, 223-228, 1992.

9. Salton, G. *Automatic Text Processing*. Addison-Wesley, Reading, MA, 1989.