Enhanced Normalization of Parenthetic Plural Forms: (s), (es), (ies)

Chris J. Lu, Ph.D.[1], Guy Divita[1], Allen C. Browne[2]
[1] Management Systems Designers, Fairfax, Virginia
[2] National Library of Medicine, Bethesda, Maryland

ABSTRACT

Norm, one of the NLM Lexical tools, is a commonly used retrieval tool for natural language processing applications with the Unified Medical Language System® (UMLS). One of the core normalization techniques is to find the uninflected form of terms. Terms in UMLS Metathesaurus® have (s), (es), or (ies) attached to them as a shorthand way of representing either singular or plural nouns. An ideal normalization should strip these parenthetic plural forms of (s), (es), and (ies) from these terms to retrieve the uninflected nouns. Chemical terms, protein and gene names, mathematical terms, etc. also have (s) and should not be stripped through normalization. It was interesting to observe that this phenomenon of parenthetic plural form does not occur with acronyms and abbreviations. Stripping (s) appropriately in normalization is a challenge. An algorithm was developed and implemented to enhance Norm to handle this issue. The algorithm is accurate, fast, and expandable. This paper details the approach taken and an analysis of this work.

INTRODUCTION

Normalization

The lexical tool, Norm, distributed by NLM is widely used in natural language processing applications [1-9]. Since its introduction in 1994, the functionality of Norm has been incrementally improved. The 2005 release was enhanced to normalize spelling variants, strip diacritics, split ligatures, and return the synonyms of Unicode symbols [10]. Norm 2005 includes ten lvg flow components while there were only six lvg flow components in Norm 2002 [11].

One of the core normalization techniques is to retrieve the uninflected form of words. In the case of nouns, the uninflected form is the singular form. For example, "fingers" is a plural noun and is normalized to the singular noun "finger" by this uninflection operation. In 2004, SNOMED CT-® was inserted into UMLS Metathesaurus [12]. Clinical Terms in SNOMED have (s), (es), and (ies) attached to them as a shorthand way of representing either singular or plural nouns. These parenthetic singular/plural forms of (s), (es), and (ies) should be stripped in normalization.

Scope

We observed parenthetic plural forms within the 2004AC UMLS Metathesaurus and the 2005 SPECIALIST LEXICON. There are 2.8 million unique terms and 0.8 million inflected terms in UMLS Metathesaurus and the SPECIALIST LEXICON, respectively. Within this scope, about 2,800 terms with forms of (s), (es), and (ies) were found. We found that all terms with forms of (es) and (ies) represented singular/plural nouns. A first approximation at normalizing away from these parenthetic plural forms should thus include simply stripping out (es) and (ies).

Objectives

Chemical terms, protein and gene names, etc. have forms of (s) and do not represent singular/plural forms. The (s) within the terms "anatoxin-b(s)", "9(s)-erythromycylamine", and "XLalpha(s) protein" should be kept instead of being stripped because the (s) does not represent a shorthand for singular/plural forms. Our first objective should be to only strip (s) from terms where the (s) is found to indicate both a singular and plural form of a noun.

We also found terms in which (s) should neither be stripped nor kept. Instead, (s) should be replaced with a space for normalization purposes. For example, (s) in "O spontn disrptn/lig(s)knee" and "[X]O spontn disrptn/lig(s)knee" should be normalized to "O spontn disrptn/lig knee" and "[X]O spontn disrptn/lig knee". Our second objective should be to replace (s) with a space in such cases.

Our objectives can be summarized as to 1). strip (s) or 2). replace (s) with a space if the form of (s) represents a singular/plural noun. The key is to determine if the form of (s) in a term represents a singular/plural noun. Rationally, we could tokenize terms to find words with (s) and then apply an exhaustive algorithm to check if the tokenized words are nouns in the SPECIALIST LEXICON to decide if

they are singular/plural forms. This method is not a practical solution to this task due to slow performance. In addition, there are exceptions to this approach. Spurious matches will be found from The SPECIALIST Lexicon for terms where the (s) pattern has been stripped. The term "G(s), alpha Subunit" once rid of the (s) pattern would be left with the word fragment "G", which would spuriously match a Lexicon noun. A similar spurious match to the word fragment "su" for the term "su(s) protein, Drosophila".

METHODS

Patterns Observations

In order to understand the pattern of (s) used in non-singular/plural cases better, we manually examined through 2,800 terms with (s) and filtered out those terms that were not singular/plural nouns. As discussed previously, (s) in these terms should be kept instead of stripped. Table 1 shows some representative terms in this category.

ID	Terms
1	9(s)-erythromycylamine
2	anatoxin-b(s)
3	Ap(s)pCHClpp(s)A
4	Bacillus phage rho11(s)
5	Cbz-AAPhepsi((s)-CH(OH)CH2)GlyVV-OMe
6	EAV G(s) glycoprotein
7	G(s), alpha Subunit
8	Histone H1(s)
9	J(s)(b) ANTIBODY
10	natoxin-a(s)
11	Salmonella II 6,7:(g),m,(s),t:1,5
12	(s)-(+)-citreofuran
13	su(s) protein, Drosophila
14	XLalpha(s) protein
15	[X]O spontn disrptn/lig(s)knee
16	O spontn disrptn/lig(s)knee

Table 1 Terms with non-singular/plural (s)

By observing above list, we found terms 1, 3, 6, 7, 8, 9, 12, 13 follow the same pattern. That is, the word in front of (s) has two or fewer characters. A word is defined as being separated by spaces or tabs. For example, "H1" is the word in front of (s) in term 8, "Histone H1(s)". The number of characters for the word in front of (s) is 2. This implies that nouns with two or fewer characters are not used in the shorthand way of singular/plural (s). The SPECIALIST LEXICON has 659 unique nouns that are two or fewer characters long. All of these nouns are either acronyms or abbreviations. It is interesting to know that acronyms and abbreviations are not used in the shorthand way of expressing singular/plural forms. We have adopted this result as pattern-1 for non-singular/plural (s):

Pattern-1:
(s) should be kept if the size of word in front of (s) is less than or equal to 2.

We observed that the characters in front of (s) in terms 1, 4, and 8 are Arabic numbers. For example, '1' is in front of (s) in term 4, "Bacillus phage rho11(s)". We hypothesized that nouns which end with an Arabic number do not follow regular rules of noun inflection. The SPECIALIST LEXICON contains 830 unique nouns that end in Arabic numbers. None of these nouns had a regular or Greco-Latin regular plural form [13]. In other words, nouns that end in Arabic number either do not have plural forms or a plural suffix s. We have adopted this result as pattern-2 for non-singular/plural (s):

Pattern-2:
(s) should be kept if the character in front of (s) is an Arabic number.

We also know that a regular noun usually does not end with punctuation. Terms 2, 5, 10, and 11 follow the same pattern of punctuation in front of (s) within a distance of 2. A distance unit is defined as a character. For example, '-' is in front of (s) at a distance of 2 in term 10, "natoxin-a(s)". The SPECIALIST LEXICON contains 404 unique nouns ending with punctuation. For the same reasons as discussed in pattern-2, there are no regular or Greco-Latin regular plural forms for nouns ending with punctuation. Thus, we adopted pattern-3 for non-singular/plural (s):

Pattern-3:
(s) should be kept if punctuation is in front of (s) within a distance 1 or 2.

The above three patterns cover most terms with non-singular/plural (s). Terms 3 and 14 are two exceptions that are not governed by those three patterns. From this experience, we learned that non-singular/plural (s) follows certain patterns in terms of the words that appear before it. Under this assumption, we observed two patterns, words ending in "pp" and "alpha" appear in front of (s) for non-singular/plural nouns. The SPECIALIST LEXICON contains 84 and 54 nouns ending with "pp" and "alpha", respectively. Only one term, "Lapp", has the regular plural form of "Lapps". Fortunately,

there are no “Lapp(s)” in the scope of our study. Thus, we adopted pattern-4 of non-singular/plural (s):

Pattern-4:
(s) should be kept if the word in front of (s) ends with “alpha” or “pp”.

Terms 15 and 16 represent the situation in which (s) should be replaced by a space instead of being stripped. An English word always follows the inserted space in these cases. Generally speaking, an English word starts with a letter. Accordingly, we discovered pattern-5, replacing (s) with a space when (s) is followed by a letter. There were exceptions to this pattern in our study - such as the terms “Ap(s)pCHClpp(s)A” and “G(s)alpha“. The letters ‘p’, ‘A’, ‘a’ follow (s) in these terms. However, (s) should be kept instead of being replaced by a space. Fortunately, these exceptions are governed by patterns 1-4. A detail discussion with examples on this issue is given in the Algorithm section. We have adopted pattern-5 for replacing (s) with a space:

Pattern-5:
Replace (s) with a space if (s) followed by a letter.

Rules Derivation
In order to achieve fast performance, a powerful search algorithm, the reversed trie, is used for this study [14]. The word trie is derived from the middle letters of the word "re**trie**val”. A reversed trie breaks a word into characters and stores each single character as a node in the tree structure in the reversed order. In this study, we use a regular expression syntax to represent the patterns we would like to find. Uppercase letters to represent wild cards as defined in table 2 while lowercase letters represent themselves.

Wild card	Definition
^	Starting mark of a term
$	Ending mark of a term before (s)
C	Any character
D	Any digit, [0-9]
L	Any letter, [a-Z]
P	Punctuation, [-(,]
S	Space, [ ]

Table 2 Wild cards definition

Rules with wild card representation can be derived to construct the trie tree structure based on the above definition. For example, “^$” is used as a rule representation for the pattern in which (s) appears at the start of “(s)-(+)-citreofuran”. Similarly, J (which is a character) begins the term of “J(s)(b) ANTIBODY” and thus “^C$” is used as the rule representation. We also use lowercase in rule representations. For examples, “pp” and “alpha” are both recognized words in front of (s) in pattern-4. The rules can be represented as “pp$” and “alpha$”. Table-3 shows the derived rules in our study.

Pat.	Example	Rule
1	(s)-(+)-citreofuran	^$
1	J(s)(b) ANTIBODY	^C$
1	EAV G(s) glycoprotein	SC$
1	su(s) protein, Drosophila	^CC$
1	Histone H1(s)	SCC$
2	9(s)-erythromycylamine	D$
3	Salmonella II 6,7:(g),m,(s),t:1,5	P$
3	natoxin-a(s)	PC$
4	Ap(s)pCHClpp(s)A	pp$
4	XLalpha(s) protein	alpha$

Table 3 Rule representation

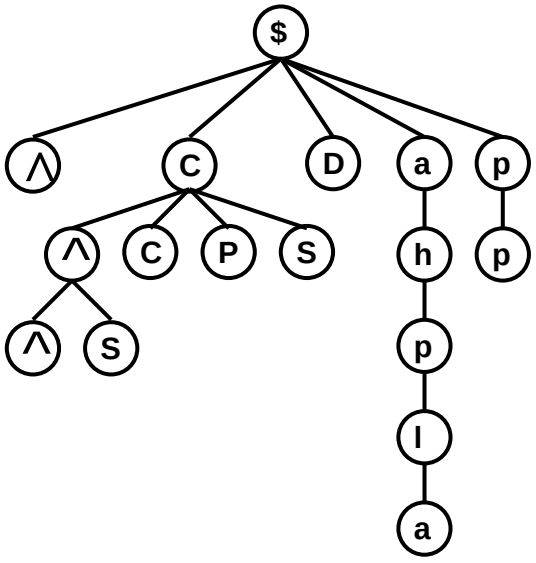

Figure 1 Tree structure of reversed trie

Reversed Trie
A reversed trie tree structure was built based on the derived rules in table 3. This tree is shown in Figure 1. A traversal algorithm on the reversed trie was then used to find if a term matches patterns described in the derived rules. Let’s use the term of “anatoxin-b(s)” as an example. First, the position of (s) is identified. This position is where the root ($) of the reverse trie is. Second, the character before (s) is ‘b’. ‘b’ is a character and matches node C while traversing the reversed trie. Third, the character

before 'b' is '-'. '-' is punctuation and matches node P. P node is at the end of a branch in the tree which means the tested term matches the pattern of derived rules, PC$. Consequently, no stripping on (s) should be performed for this term.

Algorithm

Figure 2 shows the flow chart of the core algorithm. This algorithm strips (s), (es) and (ies) as well as replaces (s) with a space correctly. (s) in "O spontn disrptn/lig(s)knee" is replaced by a space in this algorithm. The reasons are: 1) the parenthetic plural form (s) was found; 2) no rule matched while traversing the reversed trie; and 3) the character following (s) is a letter, 'k'. Forms of (s) in "Ap(s)pCHClpp(s)A" and "G(s)alpha" are kept because there are rule matches while traversing the reversed trie.

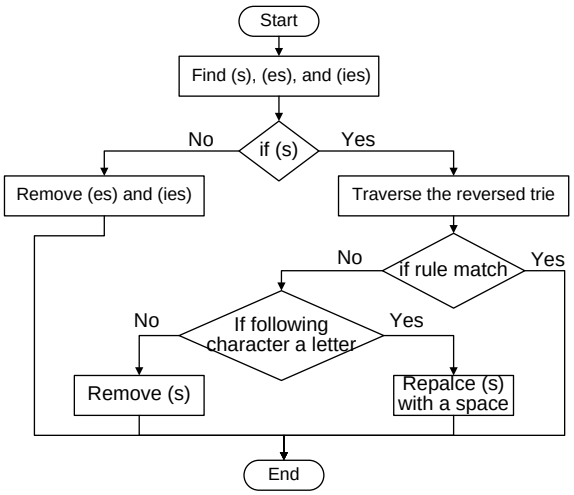

Figure 2 Flow chart of core algorithm

RESULTS & DISCUSSIONS

We implemented above algorithm and tested it on terms (3.6 million) in both UMLS Metathesaurus and the SPECIALIST LEXICON. The results show all terms (2,800) with (s), (es), and (ies) are kept, stripped, or replaced by a space correctly. We also found all words with (s) stripped are known nouns in the SPECIALIST LEXICON with only 7 exceptions, as shown in table 4. Five of these 7 exceptions are shorthand representations of nouns. The other two are drug and catalog codes.

The natural characteristics of the trie algorithm provide this algorithm high performance. In addition, new rules can be derived and added into the reversed trie tree if new patterns are observed. Rules with exceptions can also be implemented in the system if needed. Currently, no exceptions were observed in our testing data. In conclusion, this system is accurate, expandable, and extreme fast.

Exceptional Word	Notes
fing(s)	finger(s)
methamphetamine(s)	stimulant drug
muscl(s)	muscle(s)
musc(s)	muscle(s)
olbwt(s)	catalog code
rship(s)	Relationship(s)
tend(s)	tendon(s)

Table 4 Words with (s) and are not in LEXICON

Future Work

This algorithm has yet to be integrated with NLM Lexical tool, Norm. We plan to run this algorithm through additional datasets and derive additional rules to make this system more sophisticated prior to it's integration. Our plan is to integrate this method into Norm 2006.

ACKNOWLEDGEMENTS

The authors wish to thank Ms. Laura Roth and Ms. Destinee Nace for their support on tracking down the terminology. The authors also wish to thank Ms. Stephanie Lipow for discovering the problem.

REFERENCES

1. Xiaoyan Wang, Hui Nar Quek, Michael Cantor, Pauline Kra, Aylit Schultz, Yves A. Lussier. Automated Terminology Networks for the Integration of Heterogeneous Databases. Proc MedInfo, 2004. p. 555-9.
2. Aziz A. Boxwala, Qing T. Zeng, Anthony Chamberas, Luke Sato, Meghan Dierks. Coverage of patient safety terms in the UMLS Metathesaurus. Proc AMIA Symp, 2003. p. 110-4.
3. Qinghua Zou, Wesley W. Chu, Craig Morioka, Gregory H. Leazer, and Hooshang Kangarloo. IndexFinder: A Method of Extracting Key Concepts from Clinical Texts for Indexing. Proc AMIA Symp, 2003. p. 763-7.
4. Joshua C. Denny, Jeffrey D. Smithers, Anderson Spickard, III, Randolph A. Miller. A New Tool to Identify Key Biomedical Concepts in Text Documents, with Special Application to Curriculum Content. Proc AMIA Symp, 2002. p. 1007.

5. Christopher G. Chute. The Horizontal and Vertical Nature of Patient Phenotype Retrieval: New Directions for Clinical Text Processing. Proc AMIA Symp, 2003. p. 165-9.
6. Olivier Bodenreider, Joyce A. Mitchell, Alexa T. McCray. Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics, Proc AMIA Symp, 2003. p. 61-5.
7. Suresh Srinivasan, Thomas C. Rindflesch, William T. Hole, Alan R. Aronson, and James G. Mork. Finding UMLS Metathesaurus Concepts in MEDLINE, Proc AMIA Symp, 2002. p. 727-31.
8. Tammy Powell, Suresh Srinivasan, Stuart J. Nelson, William T. Hole, Laura Roth, Vladimir Olenichev2. Tracking Meaning Over Time in the UMLS® Metathesaurus®. Proc AMIA Symp, 2002. p. 622-6.
9. Olivier Bodenreider. Using UMLS Semantics for Classification Purposes. Proc AMIA Symp, 2000. p. 86-90.
10. National Library of Medicine: Lexical Tools: http://umlslex.nlm.nih.gov/lvg/2005/docs/userDoc/norm.html
11. National Library of Medicine: Lexical Tools: http://umlslex.nlm.nih.gov/lvg/2002/docs/userDoc/norm.html
12. National Library of Medicine: Unified Medical Language System: http://www.nlm.nih.gov/research/umls/Snomed/snomed_represented.html
13. Allen C Browne, Alexa T. McCray, Suresh Srinivasan. The SPECIALIST LEXICON. National Library of Medicine Technical Reports, 2000. p. 18-21.
14. Alfred V. Aho, Jeffrey D. Ullman, John E. Hopcroft. Data Structure and Algorithms. Addison Wesley, 1983. p. 163-9.