A Statistical Approach to Machine Aided Translation of Terminology Banks

Jyun-Sheng Chang
Department of Computer Science
National Tsing Hua University, Hsinchu, 30043, Taiwan
jschang@cs.nthu.edu.tw

Andrew Chang
Research and Development
Deuchemie Industries, Inc.
26, Kuangfu South Road, Hsinchu Industrial Park
Hsinchu, Taiwan

Tsuey-Fen Lin and Sur-Jin Ker
Department of Computer Science
SooChow University, Taipei, Taiwan

Abstract

This paper reports on a new statistical approach to machine aided translation of terminology bank. The text in the bank is hyphenated and then dissected into roots of 1 to 3 syllables. Both hyphenation and dissection are done with a set of initial probabilities of syllables and roots. The probabilities are repeatedly revised using an EM algorithm. After each iteration of hyphenation or dissection, the resulting syllables and roots are counted subsequently to yield more precise estimation of probability. The set of roots rapidly converges to a set of most likely roots. Preliminary experiments have shown promising results. From a terminology bank of more than 4,000 terms, the algorithm extracts 223 general and chemical roots, of which 91% are actually roots. The algorithm dissects a word into roots with around 86% hit rate. The set of roots and their hand-translation are then used in a compositional translation of the terminology bank. One can expect the translation of terminology bank using this approach to be more cost-effective, consistent, and with a better closure.

1. Introduction

Existing machine translators work well for limited domains (Slocum, 1985). When an MT system is transported to another domain, among other things, the domain specific terms have to be acquired and translated before the system can do any reasonable work again (Knowles, 1982). Current ways of handling this porting process are largely manual. Usually one either gleans domain specific terms from large amount of document at once and translates them one by one by hand, or translated each unknown term when it appears. These previous approaches all involve large amount of effort of more than one person. The long and tedious process may often result in inconsistent translation.

Furthermore, no dictionary is complete, but still we hope that the translation system produces some translation when encountering an unknown word. However, translation of terms on a one-for-one basis has no closure. When encountering an unknown term, however similar to a known one, the system will not be able to fall softly and produce some kind of reasonably acceptable translation like a human translator does. Similar consideration motives a text-to-speech research on producing pronunciation for an unknown words through morphological decomposition (Black et al. 1991).

This paper reports on a project experimenting on a new approach to this problem. The project involves statistical lexical acquisition from a large corpus of document to build a terminology bank, and automatic extraction of roots from the terminology bank. The idea is to perform human translation of these roots and to translate a term by composing the translation of its constituent roots. This idea is similar to the root-oriented dictionary proposed in (Tufis and Popescu, 1991). Certain amount of postediting is expected. However, over all, we expect this method to save significant amount of human effort, produce more consistent translation, and result in better closure such that the system can fall gracefully when encountering an unknown word.

The rest of the paper will focus on the acquisition of roots from a terminology bank. Section 2 states formally the problem. Section 3 describes our approach

to root acquisition. Section 4 describes the setup of our experiments and reports some preliminary results. Section 5 concludes the paper with some remarks and points out directions for future research.

2. The Problem of Root Acquisition

Suppose that we have a large amount of terms through a manual or automatic lexical acquisition process. In these terms, there is always certain degree of redundancy in the form of repeated occurrence of certain general or domain specific roots in different words (or words in noun-noun compounds). In order to take advantage of the redundancy and reduce the effort of translating these terms, there is the need for discovering the roots automatically. So given a set of terms, we are supposed to produce a list of roots that appear more than twice in the terminology bank. For example, given

acidimeter acidity amide antibiotic antiblocking
cyanoacrylate gloss glossmeter
hydroxybenzylmeth hydrometer mildew
mildewicide polyacryl polyacrylamide
polyacrylonitrile polyacrylsulfone acrylalkyd
pacrylate polyacrylate polyamide polyol
polytributyltinacrylate

we are suppose to produce

acryl, amide, amine, anti, block, cide, gloss,
meter, mildew, hydro, ol, poly

After hand translation, we get

acryl	丙烯
anti	防
block	貼合
cide	防　劑
gloss	光澤
meter	儀
mildew	霉
hydro	水
ol	醇
poly	聚

Now we are in a position to translate the original terminology bank by the composition of the translated roots:

antiblocking	防貼合
glossmeter	光澤儀
hydrometer	水份儀
mildewicide	防霉劑
polyacryl	聚丙烯
polyol	聚醇

3. Root Acquisition

A root can be anywhere between one and up to 11 characters (such as *phosphazene* in *phosphazene*, *polyaryloxyphosphazene*, and *polyphosphazene*). To carry out a statistical analysis on a letter by letter basis would mean searching for scarce roots (10^2-10^3) in a very large search space (10^{15}). However, a root can be either from one to 3 syllables long and there are but about some 2,000 syllables. So if we analyze the data as syllables, the search space is drastically reduced (10^{10}). So, we choose to hyphenate words in the terminology bank first and extract only roots that are made of 1 to 3 syllables.

If we had in advance the appearing frequency of the syllables and roots in the terminology bank, we could simply use them to compute the most likely hyphenation or dissection. After the whole term banks are hyphenated and dissected, we can then not only produce the list of the most likely roots in the terminology bank, but also produce the frequency count of each syllable or root. However, in most cases, we do not have the frequency count of syllables and roots in the first place, a dilemma.

Both hyphenation and root dissection are attacked using the EM algorithm (Dempster et al. 1977). In brief, the EM algorithm for the root dissection problem works like this: given some initial estimate of the root probability, any dissection of all the terms in the terminology bank into roots can be evaluated according to this set of initial root probability. We can compute the most likely dissection of terms into roots using the initial root probabilities. We then re-estimate the probability of any root according to this dissection. Repeated applications of the process lead to probability that assign ever greater probability to correct dissection of term into roots. This algorithm leads to a local but acceptable maximum.

3.1. Hyphenation

Previous methods for hyphenation are all based on rules about the nature of characters (consonant or vowel) and can only achieve about 90% hit rate (Knuth, 1985; Smith, 1989). The other 10% is done using an exception dictionary. These hyphenation algorithms are not feasible for our purpose because of the low rate and reliance on an exception dictionary. Therefore, we have developed a statistical approach to hyphenation. The idea is to collect frequency count of syllables in correctly hyphenated words. Then we use the frequency to estimate the likelihood of a syllable in

Algorithm 1. Hyphenation

Input:
  Word = $W_1W_2 \ldots W_n$ the word to hyphenate
  SylProb - probability of syllables
Output:
  Pos - positions of hyphens
Local:
  prob - probability of optimal hyphenation at a position
  prev - previous hyphenation position

1. prob[0] = 1.; prev[0] = 0;
2. For i = 1 to n do 3 & 4
3. $j^* = \max_{j} \text{prob}[i-j] \times \text{SylProb}(S_j)$
   where $S_j = W_{i-j+1} W_{i-j+2} .. W_i$.
4. prob[i] = prob[i-j*] * SylProb(Sj*);
   prev[i] = j*;
5. Compute Pos by tracing back the linked list starting from prev[len].

Algorithm 2. Root Dissection

Input:
  Word - the word to dissect
  RootProb - the estimated root probabilities
Output:
  Pos - the starting positions of roots
Local:
  prob - probability of optimal dissection at a position
  prev - previous dissecting position

1. Hyphenate Word into n syllables.
 Word = $S_1 S_2 \ldots S_n$
2. prob[0] = 1; prev[0] = 0;
3. For i := 1 to n do 4 & 5
4. $j^* = \max_{j=1,3} \text{prob}[i-j] \times \text{RootProb}(R_j)$
   where $R_j = S_{i-j+1} S_{i-j+2} \ldots S_i$
5. prob[i] = prob[i-j*] x RootProb(R_{j^*});
   prev[i] = j*
6. Compute Pos by tracing back prev links starting from prev[n].

a possible hyphenation and choose the hyphenation that consists of a most likely sequence of syllables. The optimization process is done through a dynamic programming algorithm described in Algorithm 1.

3.2. Root Dissection

One can set the initial estimate of the probability of single-, bi-, and tri-syllable roots as follows:

polychloroprene flexible foam
polychloroprene rubber
polycondensate
polycondensation
polycondensation resin
polydiallylphthalate
polydiene
polydimethyl butadiene
polydimethylsiloxane
polydiolefin
polydioxyarylene diphenyl silane
polydioxycycloalkylene diphenyl
polydiphenylsulphonemaleimide
polydispersity
polyelectrolyte
polyene
polyepichlorohydrin
polyepichlorohydrin rubber
polyester
polyester acrylate resin
polyester amide
polyester dithiol

Figure 1. An excerpt from a chemical terminology bank

a 241	a. 13	ab 25	ac 354
ac. 1	act. 1	ad 47	aer 6
af 7	ag 59	age. 15	air 4
air. 2	al 141	al. 26	am 49
an 106	an. 8	ance. 7	and. 2

Figure 2. Independent syllable probabilities

abi 3	abil 3	able.9
abra 6	absorb 2	absorp 9
accel 8	accep 3	ace 40
acene. 4	aci 7	acid. 177
acous 2	acri 3	acro 3

Figure 3. Syllable bigrams

$$\text{Prob}(R) = \begin{cases} \text{SylProb}(S), & \text{for is a single-syllable root } R = S \\ = \text{Bigram}(S_1S_2), & \text{for a bi-syllable root } R = S_1S_2 \\ = \text{Min}(\text{Bigram}(S_1S_2), \text{Bigram}(S_2,S_3)) & \text{for a tri-syllable root } R = S_1S_2S_3, \end{cases}$$

The root dissection is done using Algorithm 2 which is similar to the hyphenation algorithm.

a 2	a. 2	able. 4
abrasion. 3	abrasive. 3	absorption. 7
accele 4	acene. 2	acetal. 6
acetate. 2	acetic. 4	aceto 3
acety 5	acetyl. 2	acid. 177
acidi 22	acoustic. 2	acridine. 2
acryl 3	acryl. 2	acrylat 4

Figure 4. Roots extracted after the first iteration

plastic 5 plasticisation plasticised plasticiser plasticity
po 6 antipode epichlopohydrin pored porosity poteniometric
polari 4 polarisation polarity polarization
poly 302 polyacetal polyacrolein polyacrylamide polyacrylate
polymer 28 copolymerization polymerisation copolymerisate
polymer. 14 prepolymer terpolymer photopolymer biopolymer
port. 4 export import support
position. 5 composition decomposition
pre 18 prechrome precipitate precipitated precipitation
prene. 5 chloroprene polychloroprene
pri 3 prileshajev primary primerless

Figure 5. Roots extracted after the last iteration

4. Experimental Results

The experiment has been carried out on a personal computer running a C++ compiler under DOS 5.0. The terminology bank consists of more than 4,000 lines of chemical terms compiled by a leading chemical company in Germany for internal use. Each line consists of from 1 to 5 words and a word can be any where from 1 to 15 syllables long or 2 to 31 characters long.

The initial syllable probabilities used in the hyphenation algorithm are the appearance counts of some 1,800 distinct syllables in a partially hyphenated data, which is the result of running Latex (Knuth 1986) on the terminology bank itself.

The root dissection algorithm uses the syllable probability and bigram of syllables to start the EM algorithm. Small segments of the bigram and root probabilities produced in the first iteration are shown in Figure 2 and Figure 3 respectively.

To facilitate human translation, in the last iteration, we produce the exemplary words along side with the root found. A small segment is shown in Figure 5.

Following the terminology of research in information retrieval, we can evaluate the performance of this root extraction method:

$$\text{precision} = \frac{\text{number of correct roots}}{\text{number of roots found}}$$

$$\text{recall} = \frac{\text{number of correct roots}}{\text{number of actual roots}}$$

These two numbers can be calculated for all appearances of roots or for the set of distinct roots respectively. We have extracted 223 distinct and more frequently occurring root and 203 of them are valid roots. To analyze precision and recall for all occurrences, we have randomly sampled 100 terms, in which a domain expert identified 237 roots and our algorithm split into 195 valid roots in 226 proposed roots. Thus, counting all occurrences of root, the precision and recall rates are as follows:

precision	recall
86.3%=(195/226)	82.3%=(195/237)

If distinct roots are counted, the precision and recall rates are as follows:

precision	recall
91.0%=(203/223)	Not available

5. Concluding Remarks

Our approach is very similar to the research on identifying Chinese words in the absence of delimiters (such as spaces in English) by Sproat and Shih (1990). They have used a greedy method and the words identified are limited to 2-syllable words. In comparison, we use a global optimization algorithm through dynamic programming and identify roots up to 3 syllables long.

The results have shown that statistical approaches are very robust and through an EM algorithm, we can extract roots effectively to cut down cost in translation, achieve better consistency and closure.

The limitations of the current approach include the following: (1) Some roots do not end at syllable boundary and that results in acquisition of incomplete roots or no acquisition at all. (2) Currently, we are not performing any kind of prefix or suffix analysis. Therefore, some words having the same character sequence are incorrectly split. That results in over generation of roots.

We are now working on the following: (1) Changing the root splitting algorithm in the target language process from syllable-based to letter-based. (2) Translation of roots. (3) Formulation of the process of generating term translation in Chinese from translated roots.

Acknowledgment

This research was supported by the National Science Council, Taiwan, under Contracts NSC 81-0408-E007-13 and -529.

References

Black, A.W., J. van de Plassche and B. Williams. *Analysis of Unknown Words through Morphological Decomposition*, In Proceedings of the 5th Conference of the European Chapter of the ACL, pages 101-107, 1991.

Chang, J.S., C.D. Chen, and S.D. Chang. *Chinese Word Segmentation through Constraint Satisfaction and Statistical Optimization*, In Proceedings of ROC Computational Linguistics Conference, pages 147-166, Kenting, Taiwan, 1991, (in Chinese).

Chang, J.S., S.D. Chen, and J.D. Chen. *Conversion of Phonemic Input to Text through Constraint Satisfaction*, In Proceedings of Internal Conference on Computer Processing of Chinese and Oriental Languages, pages 30-36, Nankuan, Taiwan, 1991.

Dempster, A.P., N.M. Laird and D.B. Rubin. Maximum Likelihood from incomplete Data via the EM algorithm, J. of the Royal Statistical Society 39, pages 1-38, 1977.

Knowles, F.E. *The pivotal Role the Various Dictionary in an MT system*, in Practical Experience of Machine Translation, V. Lawson, Ed. North-Holland, Amsterdam, pages 149-162, 1982.

Knuth, D. *The TeXbook*, Prentice Hall, Reading, Massachusetts, 1985.

Slocum, J. *A Survey of Machine Translation*, Computational Linguistics 11, pages 1-15, 1985.

Smith, A. *Text Processing*, MIT Press, 1989.

Sproat, R. and Ch. Shih. *A Statistical Method for Finding Word Boundaries in Chinese Text*, Computer Processing of Chinese and Oriental Languages, pages 336-351, 1990.

Tufis, D. and O. Popescu. *A Unified Management and Processing of Word-forms, Idioms and Analytical Compounds*, In Proceedings of the 5th Conference of the European Chapter of the ACL, pages 95-100, 1991.