Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count

Ming-Wen Wu[†]
Keh-Yih Su[‡]

[†]Behavior Design Corporation
2F, 28, R&D Road II
Science-based Industrial Park
Hsinchu, Taiwan 300, R.O.C.
mingwen@bdc.com.tw

[‡]Department of Electrical Engineering
National Tsing-Hua University
Hsinchu, Taiwan 300, R.O.C.
kysu@bdc.com.tw

ABSTRACT

In machine translation systems, a computer-translated manual is usually concurrently processed by several posteditors; thus, to maintain the consistency of translated terminologies between different posteditors is very important. If all the terminologies used in the manual can be entered into the dictionary before machine translation, the consistency can be automatically maintained, which is a big advantage of machine translation over human translation. However, since new compounds are created from day to day, it is impossible to list them exhaustively in the dictionary being prepared long time ago. To guarantee subsequent parsing and translation to be correct, new compounds must be extracted from the text every time a new manual is to be translated and then entered into the dictionary. However, it is too costly and time-consuming to let the human inspect the entire text to search for the compounds. Therefore, to extract compounds automatically from the manual is an important problem. Traditional systems are to encode some sets of rules to extract compounds from the corpus. However, the problem with the rule-based approach is that not every compound obtained is desirable since it does not assign preferences to the candidates. It is not clear whether one candidate is more likely to be a compound than the other. The human effort required is still high because the lexicographer has to search for all the compound candidate list to find the preferred compounds. A new method is thus proposed in this paper to automatically extract compounds using the features of *mutual information* and *relative frequency count*. This method tests every n-gram (n is equal to 2 or 3 in this paper) formed in the manual to see whether it is a compound by checking those features. Those n-grams that pass the test are then listed in the order of significance to let the lexicographers to build into the dictionary. A significant cutdown in postediting time has been observed in our test.

1. Introduction

In technical manuals, technical compounds [Levi 78] are very common. Therefore, quality of their translations greatly affects the performance of machine translation. If a compound is not built into the dictionary before machine translation, in many cases, it would be translated incorrectly. One of the reasons is that many compounds are not *compositional*, which means that the translation of a compound is not the composite of the respective translations of the individual words [Chen 88]. For example, the translation of *green house* into Chinese is not the composite of the Chinese translations of *green* and *house*, so is *paper document*. Another advantage of building compounds into the dictionary before translation is to reduce number of parsing ambiguities. If a compound is not listed in the dictionary, it will be regarded as a group of single words. Since more than half of English words are of multi-categories, a group of words would cause more ambiguities, which will, in turn, reduce the accuracy rate of disambiguation and also increase translation time. In addition, as a manual is usually processed by several posteditors simultaneously, the translation consistency of terminologies is very important. If the compound is not present in the dictionary before machine translation, the posteditors have to spend a lot of time retrieving correct translation of the compound and checking the consistency between different posteditors. Therefore, if all the compounds can be built into the dictionary, quality of translation will be greatly improved, and lots of postediting time can be saved.

To solve the above problems, one might propose to build a huge dictionary which contains all compounds. However, compounds are rather productive, particularly in rapidly updating fields, such as information processing. New compounds are created from day to day. Hence, it is impossible to build a huge dictionary to store all compounds. Another approach is to let the human inspect the manual before machine translation to search for the compounds. Unfortunately, it is too costly and time-consuming because he has to spend a lot of time inspecting the whole manual. Once the compounds are selected, he has to check if the selected compounds are already in the dictionary. Moreover, he is not sure if it deserves the effort to enter the relevant information of the compound into the dictionary if it only appears a few times.

For these reasons, it is important that the compounds be found and entered into the dictionary before translation without much human effort. Hence, a tool to extract compounds automatically from the corpus using some quantitative criteria is seriously required. Several approaches have been proposed to extract compounds from corpus in the past [Bour 92, Calz 90]. Traditional rule-based systems are to encode some sets of rules to find the likely candidates. In LEXTER, a corpus of language texts on any subject is fed in, and the system proceeds in two stages (*analysis* and *parsing*) to produce a list of *likely terminological units* to be submitted to an expert to be validated [Bour 92]. The advantage is that since the analysis and parsing rules are simple and surface grammatical analysis instead of complete syntactic analysis is performed, it is easy to perform very frequent tests. However, since the process is done on the syntactic level without incorporating semantic information and domain knowledge, it might extract many noun phrases which are not desirable terminologies, and thus causes high false alarm. Also, it is not clear whether the terminology is a commonly used one. Since there is no performance evaluation reported, it is not clear how this approach works. Another approach is to adopt statistical measures as the selection criteria. In [Calz 90], the *association ratio* of a word pair and the *dispersion* of the second word in the word pair are used to decide if it is a fixed phrase (a compound). The drawback is that it does not take the number of occurrences of the word pair into account; therefore, it is not known if the word pair is commonly or rarely used. Also, no performance evaluation is given for this method.

In this paper, a statistical approach to solve the compound finding problem is proposed. This method extracts compounds using *mutual information* and *relative frequency count* as the features for selection. The *likelihood ratio test* is then used to check whether an n-gram is a compound. As simulation results of the initial run show, the corpus-based approach works well except that the precision rate is too low. The reason is that there are many compound candidates, though passing the likelihood ratio test, are not suitable to be regarded as compounds, such as a preposition followed by an article (such as "in the") or an auxiliary preceded by a pronoun (such as "you can"). The performance can be improved by augmenting contents of the *exception table*, which stores scores of those entries. After involving the exception table, a significant cutdown of postediting time has been observed in our test, and quality of translation is greatly improved.

2. How to Form the Candidate List for Compounds

The first step to extract compounds is to find the candidate list for compounds. According to our operational experience on machine translation, most compounds are of length 2 or 3. Hence, only bigrams and trigrams in the corpus are of interest to us in compound extraction.

To prepare the raw compound list, a corpus is first fed into the morphological analyzer so that every word in the corpus is transformed into its stem form. The reason for storing the stem form (instead of surface form) of the word is to save memory space. Then, the manual is scanned to find the possible compound candidates. Each sentence is scanned from left to right with the window size 2 and 3. Each group of words within the window of size 2 is put into the bigram list, and each group of words within the window of size 3 is placed in the trigram list. Then, the mutual information and relative frequency count for each entry are computed.

3. Compound Extraction Procedure

3.1. Feature Selection

To find compounds from the file of bigrams and trigrams, we manage to choose some features which can discriminate compounds and non-compounds. Two quantitative features are adopted as selection features for classification, namely *mutual information* and *relative frequency count.* These two features will be discussed in more detail in the following subsections.

3.1.1 Mutual Information

Mutual information is a statistic measure of word associations. It compares the probability of a group of words to occur together (joint probability) to their probabilities of occurring independently. The mutual information in the bigram is computed by the formula [Chur 90]:

$$I(x;y) \equiv \log_2 \frac{P(x,y)}{P(x) \times P(y)},$$

where x and y are two words in the corpus, and $I(x;y)$ is the mutual information of these two words x and y (in this order). $P(x)$ is evaluated as the relative frequency of the number of occurrences of x with respect to the number of total instances of singletons. If there is a genuine association between x and y, i.e. x and y are likely to form a compound, then the joint probability $P(x,y)$ will be much larger than $P(x) \times P(y)$, and consequently $I(x,y) >> 0$. If there is no interesting relationship between x and y, i.e. x and y are not very likely to form a compound, then $P(x,y) \approx P(x) \times P(y)$, and thus $I(x,y) \approx 0$.

The mutual information of trigram is defined as follows [Su 91]:

$$I(x,y,z) \equiv \log_2 \frac{P_D(x,y,z)}{P_I(x,y,z)},$$

where $P_D(x,y,z)$ is defined as the probability for x, y and z to occur jointly ('D'ependently), and $P_I(x,y,z)$ is defined as the probability for x, y and z to occur by chance ('I'ndependently). That is:

$$\begin{aligned} P_D(x,y,z) &\equiv P(x,y,z) \\ P_I(x,y,z) &\equiv P(x) \times P(y) \times P(z) \\ &\quad + P(x) \times P(y,z) + P(x,y) \times P(z). \end{aligned}$$

3.1.2 Relative Frequency Count

The relative frequency count r_i for the *i-th* bigram (trigram) is defined as:

$$r_i = \frac{f_i}{K},$$

where f_i is the total number of occurrences of the *i-th* bigram (trigram), which is the number of occurrences of the entry in the manual, and K is the average number of occurrences of all the entries. In other words, f_i is normalized with respect to K to get the relative frequency.

As the more often a group of words appear together in the corpus, the more likely it will be a compound, the relative frequency count is used as a feature for selecting compounds. Since the cost of entering the relevant information of a compound into the dictionary is not low, it may not worth to enter a compound into the dictionary if it occurs only a few times. Moreover, for there is no inconsistency problem if a compound occurs only once, there is no need to build this kind of compounds into the dictionary.

The reason of using both the mutual information and relative frequency count as the features for selection is that using either of these two features alone can not provide enough information for compound finding. The problem with using relative frequency count alone is that it is likely to choose the bigram (trigram) with high relative frequency count but low mutual information among the words comprising the compound. For example, let the relative frequency of word x be $P(x)$, and the relative frequency of word y be $P(y)$. If $P(x)$ and $P(y)$ are very large, which may cause a large $P(x,y)$ even they are not related. However, $\frac{P(x,y)}{P(x) \times P(y)}$would be small for this case.

On the other hand, the problem with using mutual information alone is that it is highly unreliable if $P(x)$ and $P(y)$ are too small. The chosen compound has high mutual information not because the words within it are highly correlated but due to a large estimation error. Furthermore, it may not worth the cost of entering the compound into the dictionary if it occurs very few times. Actually, the relative frequency count and mutual information supplement each other. A group of words of both high relative frequency count and mutual information is most likely to be composed of words which are highly correlated, and very commonly used. Hence, it is a preferred compound candidate.

3.2. Establishing Statistics of Training Corpus

The corpus which has been processed before and checked by the human can be used as the knowledge source, because all the real compounds in the corpus have already been built into the dictionary. The corpus is divided into two parts, one as the training corpus, and the other the testing set. Every word in the corpus is first converted into its stem form through morphological analysis. In this paper, the number of words in the training corpus is 74,404.

The bigrams and trigrams in the training corpus are divided into two clusters. The compound cluster comprises the bigrams and trigrams already in the dictionary, and non-compound cluster is composed of the bigrams and trigrams which are not in the dictionary. After the distribution statistics of two clusters are first estimated, we calculate the mean and standard deviation of mutual information and relative frequency count. The entries with outlier values (outside the range of 3 standard deviations of the mean) are discarded for the *robustness* of estimating statistic parameters. And, the entries of frequency count 1 are deleted for it is of little importance because there is no inconsistency problem with the term which occurs only once. Then, the statistics are estimated once again. The means and variances of mutual information and relative frequency count in both clusters are then estimated using the following formulae [Papo 90]:

$$\begin{aligned}\hat{\mu}_m &= \frac{1}{n}\sum_{i=1}^{n} m_i,\ \hat{\sigma_m^2} = \frac{1}{n-1}\sum_{i=1}^{n}(m_i-\hat{\mu}_m)^2\\ \hat{\mu}_r &= \frac{1}{n}\sum_{i=1}^{n} r_i,\ \hat{\sigma_r^2} = \frac{1}{n-1}\sum_{i=1}^{n}(r_i-\hat{\mu}_r)^2\end{aligned}$$

where m_i is the mutual information of the *i-th* bigram (trigram), r_i is the relative frequency count of the *i-th* bigram (trigram), $\hat{\mu}_m$ is the estimated mean of mutual information, $\hat{\mu}_r$ is the estimated mean of relative frequency count, $\hat{\sigma_m^2}$ is the estimated variance of mutual information, $\hat{\sigma_r^2}$ is the estimated variance of relative frequency count, and *n* is the number of bigrams (trigrams). Since we regard the bigram and trigram models as different models, the distribution statistics are estimated separately.

The covariance μ_{mr} and correlation coefficient r_{mr} of the two clusters can be estimated as follows [Papo 90]:

$$\begin{aligned}\hat{\mu}_{mr} &= \frac{1}{n-1}\sum_{i=1}^{n}(m_i-\hat{\mu}_m)(r_i-\hat{\mu}_r)\\ \hat{r}_{mr} &= \frac{\hat{\mu}_{mr}}{\sigma_m\sigma_r}\end{aligned}$$

The distribution statistics of the training corpus is shown in Table 1 and 2. (MI: mutual information, RFC: relative frequency count, cc: correlation coefficient, sd: standard deviation)

	number	mean of MI	sd of MI	mean of RFC	sd of RFC	covariance	cc
bigrams	659	6.799	3.011	1.674	1.726	-0.139	-0.027
trigrams	169	6.955	2.635	2.950	2.747	-0.930	-0.128

Table 1: Distribution statistics of compounds

	number	mean of MI	sd of MI	mean of RFC	sd of RFC	covariance	cc
bigrams	8151	4.116	3.271	1.436	1.473	-0.670	-0.139
trigrams	9392	4.859	2.763	1.627	0.706	-0.279	-0.143

Table 2: Distribution statistics of non-compounds

From Table 1 and 2, we can see that the means of mutual information and relative frequency count of compound cluster are larger than those in non-compound cluster. And, mutual information and relative frequency count are almost uncorrelated in both clusters since the correlation coefficients are close to 0.

Let *M* and *R* be the random variables which denote the mutual information and relative frequency count, respectively. Assume *M* and *R* are of Gaussian distribution. Let μ_m be the mean of mutual information of compound cluster, and μ'_m of non-compound cluster, μ_r be the mean of relative frequency count of compound cluster, and μ'_r of non-compound cluster, σ_m be the standard deviation of mutual information of compound cluster, and σ'_m of non-compound cluster, σ_r be the standard deviation of relative frequency count of compound cluster, and σ'_r of non-compound cluster, r be the correlation coefficient of mutual information and relative frequency count in compound cluster, and r' in non-compound cluster. The bivariate probability density function of the compound and non-compound clusters can be expressed as [Papo 90]:

$$\begin{aligned}
&f(M,R \mid Compound)\\
&= \frac{1}{2\pi\sigma_m\sigma_r\sqrt{1-r^2}} exp\left\{-\frac{1}{2(1-r^2)}\left(\frac{(M-\mu_m)^2}{\sigma_m^2} - 2r\frac{(M-\mu_m)(R-\mu_r)}{\sigma_m\sigma_r} + \frac{(R-\mu_r)^2}{\sigma_r^2}\right)\right\}\\
&f(M,R \mid Non-Compound)\\
&= \frac{1}{2\pi\sigma'_m\sigma'_r\sqrt{1-r'^2}} exp\left\{-\frac{1}{2(1-r'^2)}\left(\frac{(M-\mu'_m)^2}{\sigma'^2_m} - 2r'\frac{(M-\mu'_m)(R-\mu'_r)}{\sigma'_m\sigma'_r} + \frac{(R-\mu'_r)^2}{\sigma'^2_r}\right)\right\}
\end{aligned}$$

3.3. Two Cluster Classification

Given the joint distribution of mutual information and relative frequency count, the compound extraction problem can be formulated as a *two cluster classification problem*; that is, to assign a group of words x into either one of two clusters: compound cluster or non-compound cluster. If we form the ratio [Papo 90]

$$\lambda = \frac{f\left(X \mid it\ is\ \ a\ compound\right)}{f\left(X \mid it\ is\ not\ a\ compound\right)},$$

then a test based on the statistic λ is called the *likelihood ratio test.* If $\lambda > 1$, it is more likely that x belongs to the compound cluster. Otherwise, it is assigned to the non-compound cluster. Alternatively,

$$\begin{aligned} If \quad \lambda &= \frac{f\left(X \mid it\ is\ a\ compound\right)}{f\left(X \mid it\ is\ not\ a\ compound\right)}, \\ \text{then} \quad \ln\lambda &= -\frac{1}{2\left(1-r^2\right)}\left(\frac{\left(M-\mu_m\right)^2}{\sigma_m^2} - 2r\frac{\left(M-\mu_m\right)\left(R-\mu_r\right)}{\sigma_m\sigma_r} + \frac{\left(R-\mu_r\right)^2}{\sigma_r^2}\right) \\ &+ \frac{1}{2\left(1-r'^2\right)}\left(\frac{\left(M-\mu_m'\right)^2}{\sigma_m'^2} - 2r'\frac{\left(M-\mu_m'\right)\left(R-\mu_r'\right)}{\sigma_m'\sigma_r'} + \frac{\left(R-\mu_r'\right)^2}{\sigma_r'^2}\right) \\ &+ \ln\left(2\pi\sigma_m'\sigma_r'\sqrt{1-r'^2}\right) \\ &- \ln\left(2\pi\sigma_m\sigma_r\sqrt{1-r^2}\right). \end{aligned}$$

Therefore, if $\ln\lambda > 0$, there are more chances that x is a compound than it is not. To take the a priori probabilities of $P\left(x\ is\ a\ compound\right)$ and $P\left(x\ is\ not\ a\ compound\right)$ into consideration, the above equation can be changed to if $\ln\lambda > \beta$, then x is a compound, where β is a function of $P\left(x\ is\ a\ compound\right)$ and $P\left(x\ is\ not\ a\ compound\right)$. In our test, β is set to 0.

3.4. Extraction Procedure

After testing the above formula, we have found that there are some bigrams (trigrams) which have a large λ (greater than 1), but are not suitable to be regarded as compounds. For example, a preposition followed by an article (like "in the") has a very large λ, but it is not reasonable to regard it as a compound. Therefore, we use the *exception table* to store those entries. If a bigram (trigram) is found to be in the exception table, it will no longer be considered as a compound candidate.

To put it briefly, each bigram and trigram in the testing corpus is put into the following algorithm to see if it is a compound.

```
For each bigram (trigram) $x$
    if ($\lambda_x < 1$) then
        $x$ is not a compound; exit
    else
        if $x$ is in the dictionary then
            ignore
        else
            if $x$ is in the exception table then
                $x$ is not a compound; exit
            else
                place $x$ into the compound list file
End
```

The entries in the compound list file are listed in the order of significance (in the descending order of λ) to be examined by lexicographers.

4. Simulation Results

The following experiment is conducted to investigate the performance of the compound extracting method for the training corpus and the testing set. Each bigram (trigram) is put into the above algorithm for testing. If it passes the test (i.e. $\lambda > 1$ and not in the exception table), it will be recognized as a compound. Otherwise, it will be regarded as a non-compound.

There are totally 6014 bigrams and 8620 trigrams in the testing set. The performance of compound extraction for bigrams and trigrams is shown in Table 3 and 4. The simulation results are quite satisfactory

	training corpus	testing set
recall rate	68.736	60.218
precision rate	66.985	55.380

Table 3: Performance for bigrams (%)

	training corpus	testing set
recall rate	68.853	63.830
precision rate	62.687	39.474

Table 4: Performance for trigrams (%)

Table 5 shows the first five bigrams and trigrams with the largest λ and not in the exception table for the testing set. Among them, four out of five bigrams (except *select text*) and three out of five trigrams (except *dialog box display* and *mouse pointer assume*) are plausible compounds. Also, we can see if the part of speech can be adopted as another feature in modeling, the recall rate and precision rate can be greatly improved simultaneously.

bigram	trigram
paragraph style	dialog box display
insertion point	paragraph style set
dialog box	mouse pointer assume
select text	unit of measurement
style sheet	main document text

Table 5: The first five bigrams and trigrams with the largest λ and not in the exception table for the testing set

5. Conclusion

In machine translation systems, information of the words of source language should be available before any translation process can begin. The new simple words can be found by spelling check, and they are not as productive as compounds, so that the relevant information of simple words can be entered into the dictionary before translation. However, the handling of compounds is more difficult. Since compounds are very productive, new compounds are created from day to day in every domain. Obviously, it is impossible to build a huge dictionary to contain all compounds. To guarantee correct parsing and translation, new compounds must be extracted from the text to be translated and entered into the dictionary. However, it is too costly and time-consuming for the human to inspect the entire text to find the compounds. Therefore, a method to extract compounds from corpus automatically is required.

The method proposed in this paper uses mutual information and relative frequency count as two features for selection to discriminate compounds and non-compounds. The compound extracting problem is formulated as a two cluster classification problem in which a bigram (trigram) is assigned to one of those two clusters. If a bigram (trigram) is assigned to the compound cluster, it will be put into the list of potential compound candidates. Otherwise, it is discarded. The entry already in the dictionary or in the exception table will be discarded, too. With this method, the time for posteditors to retrieve the correct translation of missed compounds can be greatly reduced, and the consistency between different posteditors is easier to maintain.

The recall rate and precision rate can be further improved if the part of speech of each word can be used as a feature in modeling. Therefore, in future research, the part of speech will be adopted as another feature for selection besides mutual information and relative frequency count.

References

[Bour 92] Bourigault, D., 1992. "Surface Grammar Analysis for the Extraction of Terminological Noun Phrases," *Proceedings of COLING-92*, vol. 4, pp. 977–981, 14th International Conference on Computational Linguistics, Nantes, France, Aug. 23–28, 1992.

[Calz 90] Calzolari, N. and R. Bindi, 1990. "Acquisition of Lexical Information from a Large Textual Italian Corpus," *Proceedings of COLING-90*, vol. 3, pp. 54–59, 13th International Conference on Computational Linguistics, Helsinki, Finland, Aug. 20–25, 1990.

[Chen 88] Chen, S.-C. and K.-Y. Su, 1988. "The Processing of English Compound and Complex Words in an English-Chinese Machine Translation System," *Proceedings of ROCLING I*, Nantou, Taiwan, pp. 87–98, Oct. 21–23, 1988.

[Chur 90] Church, K.-W. and P. Hanks, 1990. "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, pp. 22–29, vol. 16, Mar. 1990.

[Levi 78] Levi, J.-N., "The Syntax and Semantics of Complex Nominals," *Academic Press, Inc.*, New York, NY, USA, 1978.

[Papo 90] Papoulis, A., "Probability & Statistics," *Prentice Hall, Inc.*, Englewood Cliffs, NJ, USA, 1990.

[Su 91] Su, K.-Y., Y.-L. Hsu and C. Saillard, 1991. "Constructing a Phrase Structure Grammar by Incorporating Linguistic Knowledge and Statistical Log-Likelihood Ratio," *Proceedings of ROCLING IV*, Kenting, Taiwan, pp. 257–275, Aug. 18–20, 1991.