A Multivariate Gaussian Mixture Model for Automatic Compound Word Extraction

[+]Jing-Shin Chang and [+*]Keh-Yih Su

[+]Department of Electrical Engineering, National Tsing-Hua University, Hsinchu, Taiwan.

*Behavior Design Corporation, 2F, No. 5, Industrial East Road IV,

Science-Based Industrial Park, Hsinchu, Taiwan.

[+]shin@hermes.ee.nthu.edu.tw, [+*]kysu@bdc.com.tw

Abstract

An improved statistical model is proposed in this paper for extracting compound words from a text corpus. Traditional terminology extraction methods rely heavily on simple filtering-and-thresholding methods, which are unable to minimize the error counts objectively. Therefore, a method for minimizing the error counts is very desirable. In this paper, an improved statistical model is developed to integrate parts of speech information as well as other frequently used word association metrics to jointly optimize the extraction tasks. The features are modelled with a multivariate Gaussian mixture for handling the inter-feature correlations properly. With a training (resp. testing) corpus of 20715 (resp. 2301) sentences, the *weighted precision & recall* (WPR) can achieve about 84% for bigram compounds, and 86% for trigram compounds. The F-measure performances are about 82% for bigrams and 84% for trigrams.

1. Compound Word Extraction Problems

1.1 Motivation

Compound words are very common in technical manuals. Including such technical terms in the system dictionary beforehand normally improves the performance of an NLP system significantly. In a machine translation system, for instance, the translation quality will be greatly improved if such unknown compounds are identified and included before the translation process begins. On the other hand, if a compound is not in the dictionary, it might be translated incorrectly [Chen 88]. For example, the Chinese translation of 'green house' is not the composite of the Chinese translations of 'green' and 'house'. Furthermore, the number of parsing *ambiguities* will also increase due to the large number of possible parts of speech combinations for the individual words if such new compounds are unregistered. It will then reduce the *accuracy* rate in disambiguation, degrade the processing or translation *quality* and increase the *processing time*.

In addition, for some NLP tasks, such as machine translation, a computer-translated manual is usually concurrently processed by several posteditors in practical operations. Therefore, maintaining the consistency of the translated terminologies among different post-editors is very important. If all the terminologies can be entered into the dictionary beforehand, the consistency can be automatically maintained, the translation quality can be greatly improved, and lots of post-editing time and *consistency* maintenance cost can be saved.

Since compounds are rather productive and new compounds are created from day to day, it is impossible to exhaustively store all compounds in a dictionary. Furthermore, identifying the compounds by human inspection is too costly and time-consuming for a large input text. Therefore, spotting and updating such terminologies before translation without much human effort is important; an *automatic* and quantitative tool for extracting compounds from the text is thus seriously required.

1.2 Technical Problems in Previous Works

The extraction problem can be modeled as a two-class classification problem, in which potential compound candidates are classified into either the compound class or the non-compound class. Many English or Chinese extraction issues had been addressed in the literature [Church 90, Calzolari 90, Bourigault 92, Wu 93, Smadja 93, Su 94b, Tung 94, Chang 95, Wang 95, Smadja 96]. Our focus will be on statistical methods for English compound word extraction, since statistical approaches have many advantages for large-scale systems in automatic training, domain adaptation, systematic improvement, and low maintenance cost.

Most statistical approaches [Church 90, Smadja 93, Tung 94, Wang 95, Smadja 96] for terminology extraction rely on word association metrics, such as frequency [Wang 95, Smadja 96], mutual information [Church 90], dice metrics [Smadja 93] and entropy [Tung 94] to identify whether a group of words is a potential compound (or highly associated collocate). The mechanisms for applying such features are often based on simple filtering-and-thresholding statistical tests; a compound candidate will be filtered out (or classified as non-compound) if its association metric is below a threshold; when multiple features are available, the features are usually applied one-by-one independently with different heuristically determined thresholds. Such approaches can be implemented easily, and encouraging results were reported in various works. However, there are several technical problems with such filtering approaches.

First of all, most simple word association features, such as frequency and mutual information, can only indicate whether an n-gram (i.e., a group of n words) is highly associated; however, high association does not always implies that it is a compound, since there are other syntactic (and even semantic) constraints which will also produce highly associated n-grams. For instance, the word pair "is a" has sufficiently high frequency of occurrence and high mutual information. Nevertheless, it is not a compound word since such a construct is produced due to syntactic reasons. Many long collocates extractable by such filtering methods are also of this category [Smadja 96]. Therefore, many highly associated non-compound n-grams might be mis-recognized as compounds.

Although it is known that *syntactic information* is useful in resolving such problems, there are few works for integrating high level syntactic or semantic features, such as parts of speech, with known word association metrics in a *simple* and effective way. A part of speech related metric is therefore proposed in this paper to formulate the syntactic constraints among the constituents of potential compound candidates. Such integration between word association metrics and syntactic constraints in a uniform formulation is important, since syntactic constraints are closely related to the generation of the compounds, and it is desirable to apply simple statistical tests based on such features, instead of using complicated syntactic processing.

Second, since the association features are often applied independently for filtering even with multiple features available, it is impossible to jointly use all discrimination information to acquire the best system performance. For instance, by filtering out low frequency candidates and then filtering out candidates with low mutual information, we may filter out low frequency candidates which actually have high mutual information. If the filtering mechanism is based on *both* frequency and mutual information, the system performance is expected to be better. In fact, it is well known that the performance is usually improved if multiple features are jointly considered, instead of using a single feature or applying multiple features independently. Therefore, what is really important is an automatic approach which could combine all available features for acquiring the best performance in the extraction task.

However, several factors must be carefully considered in order to enjoy the discrimination information provided by multiple association features. For instance, many features proposed in the literatures are highly correlated. Therefore, the correlations among the association features must be included into the statistical model in order to acquire the best achievable performance. In this work, we will therefore use (a mixture of) multivariate Gaussian density functions to incorporate the effects of the inter-feature correlation. Furthermore, it is desirable to use only the most discriminative features and reject features that are either non-discriminative or redundant with respect to other more discriminative features when combining the features. In this paper we therefore propose an integrated method, which select the most appropriate features automatically, for combining a set of useful features. In particular, optimization based on frequency, mutual information, dice metric, contextual entropy and parts of speech information will be surveyed.

To sum up, current terminology extraction researches do not fully exploit techniques for (1) integrating high level syntactic information in a simple and effective way, (2) combining useful features jointly for discrimination. To attack such problems, the parts of speech information, which encodes syntactic constraints, is integrated with several known word association metrics in one unified scoring mechanism. The correlations among the features are taken into consideration in designing the classifier. A feature selection mechanism is used for incorporating as many discriminative and non-redundant features as possible so that the terminology extraction task is based on the joint observations of the most discriminative features. A minimum error classifier, based on likelihood ratio test, is used as the basis for minimizing the classification error in the extraction task.

In the following sections, we will therefore focus on the general issues to design a good minimum error classifier, which jointly considers a set of association features for achieving minimum classification error. The simulation result shows that the proposed approach gives promising results. The tool is also observed to be useful in cooperating with a machine translation system [Chen 91].

2. Optimal Classifier Design

2.1 Optimization Criteria in Compound Extraction

In a compound retrieval task, it is desirable to recover from the corpus as many real candidates as possible; in addition, the extracted compound word list should contains as little ‘false alarm’ (i.e., incorrect candidates) as possible. The ability to extract real candidates in the corpus is defined in terms of the recall rate, which is the percentage of real compounds that are extracted to the compound list by the classifier; on the other hand, the ability to exclude false alarm from the extracted compound list is defined in terms of the precision rate, which is the percentage of real compounds in the extracted compound list. Let $n_{\alpha\beta}$ be the number of class-α input tokens which are classified as class-β (α, β = 1 for compound, and 2 for non-compound, respectively), and, let n_1 represent the number of real compounds in the corpus. The precision p and recall r are defined as follows:

$$p = \frac{n_{11}}{n_{11} + n_{21}}$$

$$r = \frac{n_{11}}{n_{11} + n_{12}} = \frac{n_{11}}{n_1}$$

The precision and recall rates are, in many cases, two contradictory performance indices especially for simple filtering approaches. When one of the performance index is raised, another index might degrade. To make fair comparison in performance, a joint performance indice or criterion function O(p,r) of the precision (p) and recall (r) rates is usually used to evaluate the system performance, instead of evaluating precision or recall alone. In the following sections, the *weighted precision & recall* (WPR) and the *F-measure* (FM) will be adopted as the optimization criteria. The weighted precision and recall (WPR), which reflects the average of these two indices, is proposed here as the weighting sum of the precision and recall rates:

$$WPR\left(w_p : w_r\right) = w_p * p + w_r * r \qquad (w_p + w_r = 1)$$

where w_p, w_r are weighting factors for precision and recall, respectively. The F-measure (FM) [Appelt 93, Hirschman 95, Hobbs 96], defined as follows, is another joint performance metric which allows lexicographers to weight precision and recall differently:

$$FM(\beta) = \frac{\left(\beta^2 + 1\right) p r}{\beta^2 p + r}$$

where β encodes user preference on precision or recall. When β is close to 0 (i.e., FM is close to p), the lexicographer prefers the system with higher precision; when β is large, the lexicographer prefers the system with higher recall. We will use Wp=Wr=0.5 and β=1, throughout this work, which means that no particular preference over precision or recall is imposed. If β=1, FM reduces to $\frac{2pr}{p+r}$, which appreciates the balance between precision and recall in the sense that equal precision and recall is most preferred if $p + r$ is identical. With the optimization criteria defined, our goal is to design an optimal classifier which could maximize the WPR and FM.

2.2 Task Definition for Optimal Classifier Design

Conventional extraction methods tend to use a list of word association related constraints for filtering out candidates of low likelihood based on certain word association metrics and empirical thresholds for the metrics. Unfortunately, there are no simple rules, other than trial-and-error, for such methods to acquire the optimal thresholds for acquiring the required precision or recall performance. In general, when the precision is raised by using high thresholds the recall degrades, and *vice versa*. The lexicographers could only use such tools by guessing. It is very difficult to automatically fit the lexicographers' preference on the precision-vs-recall performance. Such difficulty can be resolved if we can design an optimal classifier for automatically maximizing the performance criterion, such as WPR or FM, which encode user preference in the pre-specified weights.

The extraction problem can be regarded as a two-class classification problem in which each n-gram candidate is assigned either the compound label or the non-compound label based on the feature vector **x** associated with the candidate. To design a compound extractor is therefore equivalent to designing a discrimination function g(**x**;Λ) (which is capable of scoring how likely a candidate comes from the compound class), and using a set of decision rules to decide which n-gram candidate is a compound. (The symbol Λ refers to the parameters of the discrimination function, such as distributional means or variances of the probability density functions used in a statistical model.)

Different discrimination functions and decision rules will classify the input candidates differently, and thus have different performance in terms of a performance criterion. Designing an optimal classifier for a particular criterion function is therefore equivalent to finding a partition of the feature space into the decision regions for the compound class and non-compound class; feature vectors belonging to the compound decision regions are classified as compound, otherwise, they are classified as non-compound. Our main task is therefore to design an optimal classifier (or equivalently the corresponding discrimination function $g^*_{O(p,r)}(\mathbf{x};\Lambda)$) which could maximize an objective criterion function O(p,r) of the precision (p) and recall (r) rates.

2.3 Optimal Classifier for Precision and Recall Optimization

Given the underlying distributions, $f(\mathbf{x}|\mathbf{C})$ and $f(\mathbf{x}|\overline{\mathbf{C}})$, of the feature vectors **x** in the compound class (**C**) and non-compound class ($\overline{\mathbf{C}}$), it is possible to estimate the error probabilities associated with any decision region (or equivalently, any threshold, decision rules or statistical tests which could be used to define such a region) for a class. Therefore, it is possible to design the optimal classifier for some simple criterion functions if the feature distribution is very simple. In fact, procedures for designing optimal classifiers, such as the minimum error classifier, had been well studied in the speech, communication and pattern recognition communities [Devijver 82, Juang 92]. For example, the decision rule that minimizes the expected probability of classification error turns out to be a likelihood ratio test in the 2-class classification case [Devijver 82].

However, since WPR and FM are non-linear functions of classification errors (i.e., a non-linear function of n_{12} and n_{21}), it is hard to find a simple analytical discrimination function $g^*_{O(p,r)}(\mathbf{x};\Lambda)$ for testing whether an n-gram is a compound, such that the joint performance O(p,r) is maximum. Therefore, a two stage optimization scheme is proposed here in order to optimize a user specified criterion function of precision and recall, while retaining a small error rate. In the first stage, a minimum error classifier, $g^*_e(\mathbf{x};\Lambda)$, (which satisfies the minimum error criterion) is used as the base classifier to minimize the error rate (e) of classification. In the second stage, a learning method is applied, starting from the minimum error status, to optimize a user-specified criterion function of the recall and precision rates by adjusting the parameters of the classifier according to mis-classified instances.

Figure 1 shows the block diagram for training such a classifier. In the training flow, the n-grams in the training text corpus are extracted and manually inspected; those real compounds within the text corpus are used to construct a compound dictionary. The feature vectors associated with the n-grams are divided into the compound and non-compound classes according to the compound dictionary. The parameters for the compound class (Λ_c) and non-compound class ($\Lambda_{\bar{c}}$) are estimated from the distributions of the two classes. The training n-grams are then classified by the minimum error classifier. The result is compared with the compound dictionary afterward. Those misclassified n-grams are then used to adjust the parameters iteratively so that the criterion function is maximized.

The first optimization stage serves to determine the appropriate thresholds (or, more precisely, the decision boundaries) in the feature space so that as little misclassification is attained as possible. In this way, the precision and recall are expected to be improved indirectly. The second stage, on the other hand, adjusts the parameters of the classifier to achieve a local optimum of the joint precision-recall performance, starting from the minimum error status, instead of optimizing the precision and recall from arbitrary decision boundary. In other words, we are not trying to find some simple analytical discrimination function which are capable of identifying the optimal decision boundaries for precision-recall optimization. Instead, we first establish reasonably optimized decision boundaries by using the simple discrimination function for the minimum error classifier, and then modify the decision boundaries by changing the parameters of the distribution functions of the minimum error classifier to maximize the joint precision-recall performance.

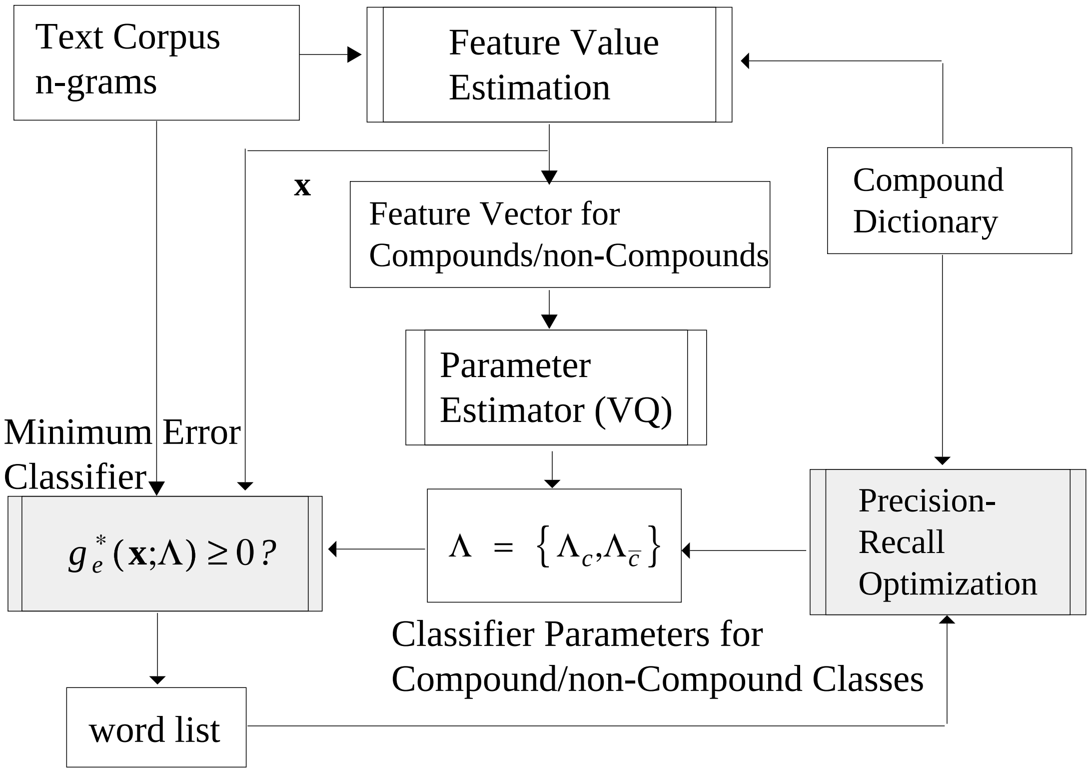

Figure 1 Supervised Training of Classifier Parameters for Precision-Recall Maximization.

The minimum error classifier is adopted at the first optimization stage since reducing classification error, in general, will improve precision and recall. In addition, it is relatively easy to implement a minimum error classifier [Devijver 82, Juang 92], and it is believed that a better local optimum could be found near the minimum error status. To see the relationship between the error rate and the precision/recall rates, first note that $p = \left(1 + n_{21}/n_{11}\right)^{-1}$ and $r = \left(1 + n_{12}/n_{11}\right)^{-1}$. The precision and recall can thus be improved by reducing n_{21} and n_{12}, respectively. Since, the error rate is propotional to $n_{12} + n_{21}$, the minimum error rate (i.e., minimum $n_{12} + n_{21}$) status is a good initial point for further optimizing the precision, recall or WPR performance. As far as F-measure is concerned, it is easy to prove that maximizing the F-measure is equivalent to minimizing $\left(n_{12} + n_{21}\right)/n_{11}$ ([Chang 97b].) Therefore, it is also appropriate for using minimum error as the criterion of the first optimization stage. In fact, if we plot the WPR (or FM) graph as a function of n_{21} and n_{12}, moving toward minimum error tends to have higher WPR (or FM) in general.

There are several issues related to the design of the minimum error classifier. As mentioned previously, it is desirable to use features that encode syntactic information, such as parts of speech, in the feature set to reject highly associated non-compound candidates; it is also desirable to use multiple features jointly to enjoy all the information contained in the features. However, the feature correlation among the features must be carefully handled in order to model the distributions of the features properly. Redundant or non-discriminative features should be removed when combining the features. Furthermore, the parameters must be estimated in a way as to minimize the classification errors. These issues will be addressed in the following sections. Due to the length limitation, we will focus ourselves on the designing issues of the minimum error classifier in this paper. Interested readers on the precision-recall optimization techniques at the second learning stage are referred to [Chang 97b].

3. The Minimum Error Classifier for Compound Extraction

A likelihood ratio test method, which was proved to be the most powerful test [Papoulis 90], can be used as the baseline classifier to achieve minimum classification error if the distributions of the feature vectors of the two classes are known. In fact, it implicitly implies the use of the optimal thresholds (or decision boundaries) which minimizes the misclassification costs in Bayesian decision points of view [Devijver 82] if the cost for each misclassification is unity. In other words, it minimizes the probability of errors for two classes of known distributions.

To identify whether an n-gram is a compound or a non-compound, each n-gram is associated with a feature vector $\mathbf{x}$, it is then judged to see whether it is more likely to be generated from the compound class $\mathbf{C}$ or the non-compound class $\overline{\mathbf{C}}$ based on the following (log-)likelihood ratio:

$$\lambda = \frac{f(\mathbf{x}|\mathbf{C})P(\mathbf{C})}{f(\mathbf{x}|\overline{\mathbf{C}})P(\overline{\mathbf{C}})}$$

$$g(\mathbf{x}) \triangleq \log \lambda = \log f(\mathbf{x}|\mathbf{C}) - \log f(\mathbf{x}|\overline{\mathbf{C}}) + \log P(\mathbf{C}) - \log P(\overline{\mathbf{C}}),$$

where $P(\mathbf{C})$ and $P(\overline{\mathbf{C}})$ are the prior probabilities of the two classes and $f(\mathbf{x}|\mathbf{C})$ (resp. $f(\mathbf{x}|\overline{\mathbf{C}})$) is the probability density function of the feature vector $\mathbf{x}$ in the compound (resp. non-compound) class. If the likelihood ratio $\lambda \geq 1$ (i.e., the discrimination function $g(\mathbf{x})$, or the log-likelihood ratio, $\log \lambda \geq 0$) for an n-gram, then it is classified as a compound; otherwise, it is classified as a non-compound. The model parameters for the two classes are referred to as Λ_c and $\Lambda_{\bar{c}}$. They correspond to the means, variances, prior probabilities, etc. (depending how the density functions are formulated), in the above formula.

4. Features for Compound Extraction

The performance upper bound of the classifier depends on the distribution of the input feature vector $\mathbf{x}$. Many statistical features are used in various applications. In particular, the *normalized frequency* (NF) [Wu 93, Su 94b] of an n-gram, the *mutual information* (MI) [Church 90, Su 91, Chang 95] among the words within an n-gram, the *dice* metric (D) [Smadja 96, Chang 97b] among the words of the n-gram, the *contextual entropy* (H) [Tung 94, Chang 95, Chang 97a] of the neighboring words of the n-grams, are used in the classification task. (The definitions of such association features and their extension are given in the Appendix.) In addition, we will introduce a *part of speech discrimination* metric (Dpos) in this paper; it is proposed in this paper to encode syntactic information so that syntactic information could be integrated with other simple word association metrics in a simple and effective way, without resorting to complicated syntactic processing. When all such features are used, we will have the following discrimination function:

$$g(\mathbf{x}) = \log \frac{f(NF, MI, D, H, D_{pos}|\mathbf{C})P(\mathbf{C})}{f(NF, MI, D, H, D_{pos},|\overline{\mathbf{C}})P(\overline{\mathbf{C}})}$$

Since such features might contain redundant information, only a subset of the features will be automatically selected with a feature selection mechanism for classification.

4.1 POS Discrimination (Dpos)

Part of speech (POS) is an important syntactic feature for extracting compounds. For instance, many compound words are associated with the part of speech patterns: {noun, noun}, {adjective, noun} (for bigrams) or {noun, noun, noun}, {adjective, noun, noun} (for trigrams). Previous frameworks [Wu 93, Su 94b] show that simple word association metrics are useful for extracting highly associated compound words. However, many non-compound n-grams, like 'is a', which have high association and high frequency of occurrence are also recognized as compounds. Such n-grams could be rejected if syntactic information is available.

One way to use the POS information is to measure how similarly the candidate and the compound class are tagged with different POS patterns. For instance, if the compound words are tagged as {noun, noun} in 80% of the cases and as {adjective, noun} in 20% of cases, then a candidate which was tagged as {noun, noun} and {adjective, noun} in most of the cases is very likely to be a compound. In this paper, we thus suggest the following *POS Discrimination* metric for measuring the similarity or distance between a compound word candidate x_i and the compound word class, in terms of their tagged POS patterns. The *discrimination* metric [Blahut 87] is defined as follows in terms of the distribution P_{ij} of the POS patterns of the candidate and the distribution P_j of the POS patterns of the compound class:

$$D_{pos}\left(x_i; \{P_{ij}\}, \{P_j\}\right) = \sum_j P_{ij} \log \frac{P_{ij}}{P_j}$$

$$P_{ij} \equiv P\left(j|w_i\right), \qquad P_j \equiv P(j)$$

where P_{ij} is the probability for the *i*th compound word candidate (or n-gram) to be tagged with the part of speech pattern j (such as a {noun, noun} tag pair) and P_j is the probability for any compound word to be tagged as j.

Intuitively, the log-likelihood ratio of P_{ij} over P_j indicates how close or similar (in terms of probability of occurrence) the particular POS pattern j is, in comparison with the probability for the whole class. If the two probabilities are nearly identical, that is, $P_{ij} \approx P_j$, the log-likelihood ratio will be close to zero. Otherwise, the 'distance' will be large. The probability P_{ij} preceding the log-likelihood ratio is a weighting factor indicating how often such a 'distance' is observed; the discrimination metric is thus the expected distance between the two probability distributions of POS tagging patterns. When a compound word candidate has exactly the same distribution as the distribution for the compound class ($P_{ij} = P_j$ for all j), the 'distance' will be exactly zero. Therefore, we can gather the POS distributions of the n-grams, and use the distributions of such a distance measure in the two classes to see whether the candidate comes from the compound class.

Since this metric assumes continuous values, the distribution of this metric can be expressed in a parametric form and the parameters of the probability density functions can be estimated from a training corpus. We can thus easily incorporate such POS information for identifying compound terminologies with a few such parameters in a very simple and effective way.

5. Experiment Environments

To investigate the various models, a corpus of 23,016 sentences (188,267 words) is prepared. The corpus is collected from a technical manual for cars. It is first processed by a morphological analyzer to normalize every word into its stem form, instead of its surface form, to reduce the number of possible variants. Since parts of speech are used as a compound extraction feature, the text is tagged by a discrimination oriented probabilistic lexical tagger [Lin 92, Lin 95] in advance. The corpus is then divided into two parts; 90% of the sentences (i.e., 20,715 sentences, 169,237 words) are used as the training corpus, and the remaining 10% (2,301 sentences, 19,030 words) are used as the testing set.

According to our experience in machine translation, most interested compounds are of length 2 or 3. Longer compounds only constitute a small fraction of interested compounds; and such long compounds can be extended by slightly modifying the definition for some association metrics. Hence, only bigrams and trigrams compounds are investigated in the current work. The corpus is therefore scanned from left to right with the window sizes 2 and 3. The lists of bigrams and trigrams thus acquired then form the lists of compound candidates of interest.

All bigrams and trigrams are submitted to three independent lexicographers of a local MT-based service translation center. The lexicographers inspect all n-grams and decide which n-grams should be considered as compounds and entered into the compound dictionary for the MT system. When there is inconsistency among their choices, the lexicographers will negotiate for a compromise. The final candidates are then used as the standard for evaluating the performance of the proposed compound extraction method. Since all the bigrams and trigrams are scanned for qualification before any experiment is conducted, the performance will reasonably reflect the performance against human judgement, the criterion for including an n-gram or not will thus not be biased by the algorithm designer's intention to have high performance.

The parameters for the compound model Λ_c and non-compound model $\Lambda_{\bar{c}}$ are evaluated from the above-mentioned training corpus, which is tagged with parts of speech and normalized into stem forms. The n-grams in the training corpus are further divided into two classes. The compound class comprises the n-grams in the compound dictionary, which was constructed by the lexicographers as described above; and the non-compound class consists of the remaining n-grams which are not in the compound dictionary. However, n-grams that occur only once or twice are excluded from consideration because such n-grams rarely introduce inconsistency and the estimated feature values are highly unreliable.

For each class, the means and standard deviations of the mutual information, normalized frequency, dice metric, contextual entropy and POS discrimination are estimated. The outlier entries (outside the range of 3 standard deviations from the mean) are discarded before estimating the model parameters so that the estimated parameters are more robust.

6. Baseline Models

To achieve minimum error classification, several factors must be carefully considered, including the features to be used, the model for formulating the underlying probability density functions of the two-classes, and the estimation to the parameters of the density functions. In the simplest form, only one feature is used for classification, and the probability density function is assumed to be a normal distribution. We then have the following baseline models:

$$\lambda = \frac{f\left(X_i|\mathbf{C}\right)P(\mathbf{C})}{f\left(X_i|\overline{\mathbf{C}}\right)P(\overline{\mathbf{C}})}$$

where X_i refers to any of the features among normalized frequency (NF), mutual information (MI), dice metric (D), contextual entropy (H) and POS discrimination (Dpos). Such baseline models are used to evaluate the performance for the individual feature; they will also be compared with other more complicated models to justify our proposals.

The following table gives the performance using only one feature. The shaded areas highlight the error rate performance, which is the optimization criterion at the current stage. The features are arranged in increasing order of error rates for bigrams.

		Training Set					Testing Set				
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
2-gram Baseline	Recall	11.09	0.0	4.87	6.01	12.33	8.07	0.0	1.35	2.69	36.77
	Precision	100.0	*	30.92	30.69	37.07	100.0	*	23.08	33.33	57.75
	Error Rate	11.03	12.41	13.15	13.34	13.47	21.20	23.06	23.78	23.68	20.79
	WPR(1:1)	55.54	*	17.90	18.35	24.70	54.03	*	12.22	18.01	47.26
	F-measure	19.97	*	8.41	10.05	18.50	14.93	*	2.55	4.98	44.93
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
3-gram Baseline	Recall	0.0	0.0	13.99	10.20	7.58	0.0	0.0	12.07	3.45	39.66
	Precision	*	*	42.11	22.58	25.49	*	*	58.33	66.67	41.07
	Error Rate	4.95	4.95	5.21	6.18	5.67	11.51	11.51	11.11	11.31	13.49
	WPR(1:1)	*	*	28.05	16.39	16.54	*	*	35.20	35.06	40.37
	F-measure	*	*	21.00	14.05	11.69	*	*	20.00	6.56	40.35

Table 1 Error Rate Performance Using only One Feature
(*: undefined, i.e., all candidates are classified as non-compound.).

The error rates are in the ranges of 11.03%-13.47% and 20.79%-23.78% for bigrams in the training set and the testing set respectively; for 3-grams the error rates are in the ranges of 4.95%-6.18% (training set) and 11.11%-13.49% (testing set); such performance corresponds to accuracy rates of 87-89% (76-79%) and 94-95% (87-89%) in classifying the bigram and trigram training (testing) set. Using the minimum error classifier thus achieves moderately low error rates both for the training set and testing set, without resorting to arbitrary thresholding.

Initially, however, the precision and recall are not sufficiently high except for the bigram POS discrimination case since the classifier tends to recognize most n-grams (or even all n-grams) as non-compounds. The 0% recalls and undefined precisions (designated as ‘*’) in the table are the results of classifying all entries as non-compound as suggested by the assumed normal distributions. Such initial precisions and recalls are not a critical problem at the current stage where minimization of error counts is the major goal. It will be shown in later sections that, by incorporating more features, the error rates will be further reduced and the precision and recall will be indirectly improved toward high precision and moderate recall.

There are several problems to achieve the minimum error criterion by using the above baseline models.

First of all, various features are not used jointly to supplement each other so as to reduce the error rate. Second, the distributions are not necessarily normal for some features. (For instance, the normalized frequency is more likely to have an exponential distribution. Fortunately, the comparison between the baseline models and other more complicated models in the current work will not be affected significantly, since we actually get almost the same error rates for such features by using the exponential distribution assumption.) To resolve such problems, we will propose some methods in the following sections to improve the error rate performance further so that the first optimization stage is better conducted.

7. Feature Integration and Optimal Feature Selection

7.1 Integration of the Features

While each of the above features provides moderately good initial error rate performance in the above baseline models, it is known that jointly considering all the features would, in general, achieve better performance. It is also known that step-by-step filtering approaches, which were commonly used in traditional extraction tasks, tend to raise the precision rate at the cost of lowing the recall, since a filtering module may filter out potential candidates without using all available information; it is then not likely to acquire the global optimal precision and recall achievable by using such features. Using all features jointly in one step for optimizing the extraction task is thus emphasized here, instead of using the multiple features step-by-step in multiple filtering modules.

However, increasing the number of features may increase the modeling complexity of the classifier [Devijver 82] without increasing much performance, since some of the features might be highly correlated, and thus much redundant information will be contained in the whole set of features. Therefore, an automatic mechanism for choosing the right features is proposed here, so that only a subset of the most discriminative features are used for efficient computation without losing discrimination power.

Since our goal is to minimize the error rate performance, our strategy for finding the best feature set is to combine the current feature set (which is initially empty) with each feature not in the current feature set for conducting the likelihood ratio test. The feature which enable the classifier to minimize the error rate performance, when jointly considered with the current set of optimal features, is then added to the optimal set of features. This process starts from the baseline models and stopped when the inclusion of new feature do not improve the training set performance further. This strategy can be characterized as a kind of sequential forward selection (SFS) in the literature [Devijver 82].

7.2 Optimization Using Independent Normal Model

The performance of the classifier will also depend on how good the density function of the features fits the real training data, in addition to the feature set being used. In the simplest model, the joint probability of the features is approximated as the product of the probabilities for the individual features (by assuming that they are mutually independent), and each feature is assumed to be normally distributed. The corresponding log-likelihood ratio then becomes:

$$\log\lambda = \sum_{i=1}^{D}\left[\log f\left(x_i|\mathbf{C}\right) - \log f\left(x_i|\overline{\mathbf{C}}\right)\right] + \left[\log P(\mathbf{C}) - \log P\left(\overline{\mathbf{C}}\right)\right]$$

where the summation is taken over all features being used, and D is the dimension of the feature vector. In other words, all features are assumed to be independent in such a simplified model. With such assumptions, uncorrelated (complementary) features are likely to be included earlier than highly correlated features since features with smaller correlation coefficients tend to be closer to the independent assumption and are likely to have better performance. The mechanism can thus select the most useful and complementary features automatically and leave redundant features unused. Table 2 shows the performances for using different numbers of features, which are selected, in sequence, by the automatic feature selection method described in the previous section, using independent normal assumption.

By applying the feature selection mechanism over all the features, the Dpos (discrimination), H (entropy), MI (mutual information), NF (normalized frequency) and D (dice) features are selected in sequence for bigrams; on the other hand, the best feature sequence for trigrams, under the current model, is Dpos, MI, H, D, NF. The SFS strategy results in the following error rate performance, where the features are arranged in the same order as the sequence in the feature selection process. For instance, the second column of the bigram performance table shows that the error rate is 8.07% when the entropy feature, H, is added to the feature set with other preceding features (in this case, the discrimination feature, Dpos).

		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	11.09	40.41	54.61	35.34	31.30	8.07	35.43	60.54	33.63	50.67
	Precision	100.0	88.04	77.39	71.04	49.67	100.0	89.77	92.47	82.42	66.47
	Error Rate	11.03	8.07	7.61	9.81	12.46	21.20	15.82	10.24	16.96	17.27
	WPR(1:1)	55.54	64.23	66.00	53.19	40.49	54.04	62.60	76.51	58.03	58.57
	F-measure	19.97	55.39	64.03	47.20	38.40	14.93	50.81	73.17	47.77	57.50
Feature Sequence		Dpos	MI	H	D	NF	Dpos	MI	H	D	NF
3-gram	Recall	0.0	14.29	33.53	29.45	26.24	0.0	17.24	44.83	56.90	48.28
	Precision	*	100.0	70.99	46.98	33.83	*	100.0	86.67	49.25	47.46
	Error Rate	4.95	4.24	3.97	5.14	6.19	11.51	9.52	7.14	11.71	12.10
	WPR(1:1)	*	57.15	52.26	38.22	30.04	*	58.62	65.75	53.08	47.87
	F-measure	*	25.01	45.55	36.20	29.56	*	29.41	59.09	52.80	47.86

Table 2 Error rate performances of the independent normal model.

The shaded areas highlight the error rate performance, which is the optimization criterion at the current stage. The parts of speech discrimination is selected first in the two feature sequences, since the parts of speech information provide the best error rate performance among all using the normal assumption. For the bigram case, the error rate is reduced by 26.8% (from 11.03% to 8.07) when the contextual entropy information, H, is included. The inclusion of the the mutual information further reduces the error rate performance to 7.61%, corresponding to a reduction of 5.7% of the remaining errors. For trigrams, the error rates are improved slightly from 4.95% to 4.24% to 3.97 when the second and the third features (i.e., MI and H) are included, corresponding to the error reduction rates of 14% and 6%, respectively.

In addition to the improvement in error rate performance, the extra features do improve the precision and recall performance (WPR or FM, or both) as well. Although the error rate is only slightly improved (and the system retains essentially the same low error rates), the precision and recall performance is shifted away from the initial low precesion and recall status significantly. Such observations partially justify our two-stage arguments to optimize the precision and recall performance starting from a minimum error status.

However, it fails to further improve the error rate performance as the feature dimension increases further, since the mutually independent assumption for the joint density function becomes harder and harder to be true as the feature dimension increases. For instance, the dice metric (D) and mutual information (MI) has a high correlation coefficient of about 0.6 in bigrams and 0.4 in trigrams. Another example would be the NF (normalized frequency) and H (contextual entropy), which have correlation coefficients of about 0.4-0.5 in the bigram and trigram data. The problem is resolved by considering the feature correlation and using better density functions to approximate the joint distribution as follows.

7.3 Model Refinement with Mixture of Gaussian Density Function

There are two sources of errors for including new features in the previous model, which assumes that the features are *mutually independent* and *normally distributed*. First, the independent assumption might not be true for some feature pairs. In fact, the correlation matrices for the features indicate that some of the features are highly correlated. Therefore, it is desirable to use a multivariate normal (i.e., Gaussian) distribution [Roussas 73, Rabiner 93], which encode feature correlations with a covariance matrix, to consider the effects of the correlations among the features. Second, the distributions of some features are not similar to a normal distribution. Therefore, using a mixture of the multivariate normal distribution would be a better way to fit the density functions. By increasing the number of mixtures, it is possible, in theory, to fit the shapes of the real distributions better, and thus have better estimation on the likelihoods of the joint feature vectors.

7.3.1 Using Multivariate Gaussian with Fixed Number of Mixtures

To fit the training data into a mixture of multivariate Gaussian distribution, we must estimate the means and co-variances of each mixture or cluster. The clusters are acquired using a standard vector quantization (VQ) technique [Duda 73]. For a K-mixture distribution, the feature vectors are clustered into K clusters; the mean vectors, covariance matrices and prior probabilities of the clusters are then estimated from the clustering results.

Since the number of mixtures for the underlying distributions of the joint features of various dimensions are not known, we fixed the number of mixtures (K) throughout the whole feature selection process to find the best performance. The cases for fixing K=1, 2, 3 are tried in order to find the best number of mixtures to use. The best results for 2-grams and 3-grams are given in the Tables 3-4. The comparison between the independent normal model and the K-mixture multivariate normal model (using fixed K throughout the feature selection process) is summarized in Table 5.

		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.50	71.61	50.67	51.71	69.06	71.30	69.96	67.26	47.09
	Precision	100.0	97.87	88.93	62.93	45.53	100.0	95.78	93.41	80.65	52.24
	Error Rate	3.74	3.73	4.63	9.82	13.67	7.14	7.34	8.07	11.27	22.13
	WPR(1:1)	84.92	84.69	80.27	56.80	48.62	84.53	83.54	81.68	73.95	49.66
	F-measure	82.24	82.63	79.34	56.14	48.42	81.70	81.75	80.00	73.34	49.53

Table 3 The Best Bigram Performance of the Minimum Error Rate Classifier Using a 2-Mixture Multivariate Normal Density Function (K=2).

Feature Sequence		Dpos	H	MI	D	NF	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.90	54.23	75.86	74.14	74.14	36.21	37.93
	Precision	100.0	95.12	90.91	80.91	39.08	100.0	97.73	95.56	95.45	41.51
	Error Rate	1.82	1.75	1.96	2.99	6.45	2.78	3.17	3.37	7.54	13.29
	WPR(1:1)	81.63	81.67	78.98	66.40	46.65	87.93	85.93	84.85	65.83	39.72
	F-measure	77.50	79.45	77.18	63.24	45.43	86.27	84.32	83.50	52.50	39.64

Table 4 The Best Trigram Performance of the Minimum Error Rate Classifier Using a 3-Mixture Multivariate Normal Density Function (K=3).

N		Training Set					Testing Set				
	Model && Features	P	R	E	WPR	FM	P	R	E	WPR	FM
2	IN: Dpos+H	88.04	40.41	8.07	64.23	55.39	89.77	35.43	15.82	62.60	50.81
	IN: Dpos+H+MI	77.39	54.61	7.61	66.00	64.03	92.47	60.54	10.24	76.51	73.17
	Mx: Dpos+H (K=2)	97.87	71.50	3.73	84.69	82.63	95.78	71.30	7.34	83.54	81.75
3	IN: Dpos+MI	100.0	14.29	4.24	57.15	25.01	100.0	17.24	9.52	58.62	29.41
	IN: Dpos+MI+H	70.99	33.53	3.97	52.26	45.55	86.67	44.83	7.14	65.75	59.09
	Mx: Dpos+H (K=3)	95.12	68.22	1.75	81.67	79.45	97.73	74.14	3.17	85.93	84.32

Table 5 Comparison between Independent Normal (IN) Model and K-mixture Multivariate Normal (Mx) Model. (2: 2-gram, 3: 3-gram, P: Precision, R: Recall, E: Error Rate, WPR: Weighted Precision/Recall with equal weights, FM: F-measure.)

For bigram compound word detection, the best (training set) error rate performance is found in Table 3 when Dpos (parts of speech discrimination) and H (contextual entropy) are used jointly using a 2-mixture multivariate (bivariate) normal density function. The best feature sequence is identical to the normal independent model. In this case, the error rate, 3.73%, is only about 49% of the best normal independent model (using Dpos, H and MI), whose error rate is 7.61%. The WPR is also significantly improved from 66.00 to 84.69, and the FM from 64.03 to 82.63. The precision and recall for this case are 97.87% and 71.50%, respectively.

Trigram compound detection also acquires the best results by using Dpos and H, but with a 3-mixture multivariate normal density function (Table 4). The error rate is 1.75% in this case, which is only 44% of its counterpart using the independent normal model, i.e., 3.97% (using Dpos, MI and I). The results demonstrates that using a mixture of multivariate normal density function to include the correlation and fit the density function of the training data does reduce the error rate and improve the precision, recall, WPR and FM significantly.

Again, the WPR and FM are, in general, improved when the error rate is reduced. However, the tables indicate that the error rates do not decrease monotonically as the number of features are increased for a given K; the error rate decrease only for the first two or three features in the feature sequence. Besides, the error rates do not decrease monotonically either when the number of mixtures increased when comparing the performance for a specific number of features. There are several possibilities which make the fitting of the training data to a K-mixture D-variate density function imperfect in the above process; the performance thus is not monotonically increased with K or D [Chang 97b].

In particular, the number of mixtures for the underlying density function of the joint features may not be characterized by a small K when the number of features increases to some extent. In fact, it is known in statistical pattern recognition community [Devijver 82] that when the number of features increases, the best number of mixtures for modeling the joint distribution of the features, in general, will increase quickly. For instance, two features, each having two normal mixtures, when considered jointly, may have as many as four mixtures if they are independently distributed. The number of mixtures tends to grow exponentially with the number of features in the worst cases. As a result, the real K may far exceed our searching range (K=1-3) when new features are included.

7.3.2 Improvement by Searching for the Best Number of Mixtures

The above identified problems in using a fixed number of mixtures *throughout* the whole feature selection process indicates several ways to improve the error rate performance. The simplest way would be to set an upper bound, Kmax, and tries all K <= Kmax *during* the feature selection process for each feature dimension. We thus tries several Kmax and find the best K (K*) for such searching ranges. The following table shows the results when Kmax=3.

The numbers in the parentheses indicate the best number of mixtures (K*) used. For instance, the Dpos(.)-H(2) feature sequence means that a local optimal is found when Dpos and H are jointly considered using 2-mixtures.

		Training Set					Testing Set				
Feature Sequence		Dpos(2)	H(2)	MI(3)	NF(3)	D(1)	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.50	72.12	67.05	32.12	69.06	71.30	70.40	65.92	44.39
	Precision	100.0	97.87	90.74	83.70	56.78	100.0	95.78	94.01	93.63	68.28
	Error Rate	3.74	3.73	4.37	5.71	11.45	7.14	7.34	7.86	8.89	17.58
	WPR(1:1)	84.92	84.69	81.43	75.37	44.45	84.53	83.54	82.21	79.77	56.34
	F-measure	82.24	82.63	80.37	74.46	41.03	81.70	81.75	80.51	77.37	53.80
Feature Sequence		Dpos(3)	H(3)	MI(3)	D(3)	NF(1)	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.90	24.49	75.86	74.14	74.14	36.21	44.83
	Precision	100.0	95.12	90.91	80.91	33.60	100.0	97.73	95.56	95.45	48.15
	Error Rate	1.82	1.75	1.96	2.99	6.13	2.78	3.17	3.37	7.54	11.90
	WPR(1:1)	81.63	81.67	78.98	66.40	29.04	87.93	85.93	84.85	65.83	46.49
	F-measure	77.51	79.45	77.19	63.24	28.34	86.27	84.32	83.50	52.50	46.43

Table 6 The Performance of the Minimum Error Rate Classifier
Using Multivariate Normal Density Function up to 3 Mixtures (Kmax=3).

Table 6 demonstrates that, by searching for the best K in [1, Kmax] for each feature dimension, the error rate performance is always better than (or identical to) its counterpart in Tables 3-4 of the same number of features. This justify our arguments that K must be searched for a local optimum instead of using a fixed number of mixtures all the time.

Table 6, however, still do not show monotonic decreasing of the error rates when the number of features are increased. In fact, the error rates no more decrease after the third feature is included, just like Tables 3-4. The problem is that Kmax=3 is still too small to search for a better performance even with only 3 features. In fact, we could further enlarge the searching range Kmax, and it is demonstrated in [Chang 97b] that the training set error rates for any given number of features do decrease monotonically as the searching range [1, Kmax] is increased. We can thus expected that the error rate will decrease monotonically as the number of features are increased if we allow a much larger searching ranges. In the current task, however, it is observed that the best number of features even for 3 features are more than ten (i.e., K*>10). This would requires a very lengthy time to converge. Furthermore, the features at the tail of the feature are highly correlated with features at the front of the list, which means that they may provide little additional information once those features selected earlier are used for classification. Improving the estimation of the density functions for including such features thus would not likely to produce significant improvement. Therefore, compromise must be taken between modeling complexity and computation costs.

With Kmax=3 and two features (in which the training set error rates are minimal), we actually have testing set WPR performance of 84% (bigram) and 86% (trigram); the F-measures are about 82% and 84% for bigram and trigram, respectively.

Given the above error rate performance, it is still possible to further improve the error rate performance and thus indirectly improve the precision and recall rate performance. However, such approaches do not guarantee to get the best joint precision-recall performance, since the minimum error rate criterion, eventually, is not equivalent to maximum precision-recall. Therefore, optimizing the precision and recall performances by adjusting the parameters of the classifiers afterward is desirable. Such optimization issues and the resultant improvement, however, is beyond the scope of the current paper. Interested readers are referred to [Chang 97b].

8. Concluding Remarks

Most simple mechanisms for terminology extraction rely on trial-and-error to setup empirical thresholds for each available features, and use such features to filter out inappropriate candidates step-by-step using one feature per step. Such simple filtering-and-thresholding approaches cannot automatically optimize a user specified criterion function of precision and recall. To resolve such optimization problems, a two-stage optimization scheme is proposed. In the first stage, the system tries to reach minimum classification error to optimize the precision and recall performance indirectly, by using a two-class classifier with a likelihood test method. In the second stage, an adaptive learning method is then applied to directly optimize a criterion function of precision and recall; such a criterion function can be pre-specified by a lexicographer based on the preference over the precision and recall performance. Optimization through error rate minimization in the first stage, in particular, is addressed in detail in this paper.

The method proposed in this paper integrates mutual information, normalized frequency, dice, contextual entropy and part of speech information as the features for discriminating compounds and non-compounds. The POS discrimination metric, in particular, is proposed in the current work for encoding the syntactic constrains over possible compound candidate. Syntactic constraints can thus be easily integrated quantitatively for jointly optimizing the system performance with other word association metrics.

To reach minimum error rate in the first optimization stage, all association features are jointly considered so that all available information could be enjoyed by the system; an automatic feature selection mechanism is applied so that only the most discriminative features are used to jointly qualify compound candidates. Various models are used to fit the training data to various density functions so as to minimize the system error rate. The correlations among the features are taken into account by including the correlation matrices into the density functions, and the density functions are formulated using a mixture of multivariate

Gaussian density functions so as to well characterize the distribution of the training data.

With a training (resp. testing) corpus of 20715 (resp. 2301) sentences sampled from technical manuals about cars, the *weighted precision & recall* (WPR) using the proposed approach can achieve about 84% for bigram compounds, and 86% for trigram compounds. The F-measure performances are about 82% for bigrams and 84% for trigrams.

Appendix: Association Features for Two-Class Classification

1. Normalized Frequency (NF)

The normalized frequency for the i^{th} n-gram is defined as:

$$r_i = \frac{f_i}{\bar{f}}, \qquad \bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i$$

where f_i is the total number of occurrences of the i^{th} n-gram in the corpus, and $\bar{f}$ is the average frequency of all the entries. In other words, r_i is the normalized frequency with respect to the average frequency $\bar{f}$.

2. Mutual Information (MI)

Mutual information is a measure of word association. It is the ratio between the joint probability for a group of words to appear in the same n-gram window and the probabily for such words to occur in the same window independently. The bigram mutual information $I(x;y)$ is known as [Church 90]:

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

where x and y are two words in the corpus. The mutual information of a trigram is defined as [Su 91]:

$$I(x,y,z) = log \frac{P_D(x,y,z)}{P_I(x,y,z)} = log \frac{P(x,y,z)}{P_I(x,y,z)}$$

$$P_I = P(x)P(y)P(z) + P(x)P(y,z) + P(x,y)P(z)$$

where $P_D(x,y,z) \equiv P(x,y,z)$ is the joint probability for x, y, z to appear jointly as a group of words in a trigram window, and $P_I(x,y,z)$ is the probability for x, y, z to appear, independently, as a group by chance. Note that the three product terms in $P_I(x,y,z)$ correspond to three different ways in which the constituents of the trigram appear in the same trigram window by chance; $P_I(x,y,z)$ is the total probability of the various possible combinations. In general, $I(\cdot) >> 0$ implies that the words in the n-gram are strongly associated. Otherwise, their appearance as one group of words may be simply by chance.

3. Dice Metric (D)

The dice metric is commonly used in information retrieval tasks [Salton 83] for identifying closely related binary relations. The dice metric for a pair of words x, y is defined as follows [Smadja 96]

$$D_2(x,y) = \frac{P(x=1,y=1)}{\frac{1}{2}[P(x=1) + P(y=1)]},$$

where x=1 and y=1 correspond to the events that x appears in the first place and y appears in the second place of a bigram respectively. Intuitively, the dice metric is the likelihood ratio between the joint probability for two words (or events) to occur simultaneous and the average probability for each individual word (or event to) occur in bigram pairs. Therefore, a high dice value tends to mean that x and y are highly associated.

We can also define the dice metric for triple relations following the same spirit in defining the 3-gram mutual information. However, note that in defining the bigram dice metric, the joint probability $P(x=1,y=1)$ is normalized with respect to the *average* of the marginal probabilities, $P(x=1)$ and $P(y=1)$, of the constituents instead of the *product* of the marginal probabilities (i.e., the probability of independent occurrence). Therefore, we have three different ways to normalize the joint probability with respect to the averages of the marginal constituent probabilities as follows:

$$\frac{P(x=1,\ yz=1)}{\frac{1}{2}[P(x=1)+P(yz=1)]},\ \frac{P(xy=1,\ z=1)}{\frac{1}{2}[P(xy=1)+P(z=1)]},\ or\ \frac{P(x=1,\ y=1,\ z=1)}{\frac{1}{3}[P(x=1)+P(y=1)+P(z=1)]}$$

where $P(x=1,y=1,z=1)$ is the probability that x, y and z appear simultaneously in the first, second, and third places of a trigram, $P(xy=1)$ (i.e., $P(x=1,y=1)$) is the probability that x and y appear simultaneously in the first and second places of a trigram, and $P(yz=1)$ (i.e., $P(y=1,z=1)$) stands for the probability that y and z appear in the second and third places of a trigram simultaneously. (Note that the first two normalized metrics are simply the bigram dice metrics for [x, yz] and [xy, z], respectively.) If any of the above three normalized association metrics is small, then the trigram is likely to belong to different words. Therefore, we shall use the *minimum* of the three normalized likelihood ratios to indicate the association of the trigram. The trigram dice metric is then defined as follows.

$$\begin{aligned} D_3(x,y,z) &= min\left[\frac{2P(x=1,y=1,z=1)}{P(x=1)+P(yz=1)}, \frac{2P(x=1,y=1,z=1)}{P(xy=1)+P(z=1)}, \frac{3P(x=1,y=1,z=1)}{P(x=1)+P(y=1)+P(z=1)}\right] \\ &\triangleq \frac{P(x=1,y=1,z=1)}{P_I'} \\ P_I' &= max\left[\frac{1}{2}[P(x=1)+P(yz=1)], \frac{1}{2}[P(xy=1)+P(z=1)], \frac{1}{3}[P(x=1)+P(y=1)+P(z=1)]\right] \end{aligned}$$

The three terms in the bracket of the *min* operator indicate three different ways in which the three words do not belong to the same lexical entry. The *min* operator means to choose the weakest evidence of association for comparison with a threshold. If the weakest evidence of association is greater than a threshold, then the trigram dice measure gives a strong indication that the three words belong to the same lexical entry. Given the above definition, only those trigrams which appear simultaneously with significantly higher probability than the maximum probability of the various other combinations of the constituents are considered compound candidates.

4. Contextual Entropy (H)

The left and right contextual entropies [Tung 94] are defined respectively as follows:

$$\begin{aligned} H_L(x) &= -\sum_{w_i} P_L(w_i;x) \log P_L(w_i;x) \\ H_R(x) &= -\sum_{w_i} P_R(x;w_i) \log P_R(x;w_i) \end{aligned}$$

where w_i is the left or right neighboring word of an n-gram x, and the probability that w_i appear as the left or right neighbor of an n-gram x is represented as $P_L(w_i;x)$ and $P_R(x;w_i)$, respectively. If the contextual entropy is large, which means that the neighbors of x are randomly distributed, then x tends to be a lexical unit by itself; otherwise, x and w_i are likely to appear simultaneously, which implies that x is unlikely to be a lexical unit by itself. In the current work, we use the average of H_L and H_R as a single feature instead of using the two entropy metrics.

References

Appelt, Douglas E., Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson, "FASTUS: A Finite-State Processor for Information Extraction from Real-World Text," *Proc. IJCAI-93*, Chambery, France, Aug. 1993.

Blahut, Richard E., *Principles and Practice of Information Theory,* Addison-Wesley Publishing Company, MA, USA, 1987.

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

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

Chang, Jing-Shin, Yi-Chung Lin and Keh-Yih Su, "Automatic Construction of a Chinese Electronic Dictionary," *Proceedings of the Third Workshop on Very Large Corpora*, pp. 107-120, MIT, June, 1995.

Chang, Jing-Shin and Keh-Yih Su, 1997a. "An Unsupervised Iterative Method for Chinese New Lexicon Extraction", to appear in *International Journal of Computational Linguistics & Chinese Language Processing*, 1997.

Chang, Jing-Shin, 1997b. *Automatic Lexicon Acquisition and Precision-Recall Maximization for Untagged Text Corpora*, PhD dissertation, National Tsing-Hua University, Hsinchu, Taiwan, July 1997.

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

Chen, S.-C., J.-S. Chang, J.-N. Wang and K.-Y. Su, "ArchTran: A Corpus-Based Statistics-Oriented English-Chinese Machine Translation System," *Proceedings of Machine Translation Summit III*, pp. 33--40, Washington, D.C., USA, July 1--4, 1991.

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

Devijver, Pierre A. and Josef Kittler, 1982. *Pattern Recognition: A Statistical Approach,* Prentice-Hall Inc., N.J., USA, 1982.

Duda, Richard O. and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, NY, USA, 1973.

Hirschman, Lynette and Marc Vilain, *Extracting Information from the MUC,* Tutorial of the ACL 95, MIT, Cambridge, MA, June 16, 1995.

Hobbs, Jerry R. "FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text," *Proc. of ROCLING IX*, pp. 199-231, Natl. Cheng-Kung Univ., Tainan, Taiwan, Aug. 1996.

Juang, B.-H. and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Signal Processing,* vol. 40, no. 12, pp. 3043-3054, Dec. 1992.

Lin, Y.-C., T.-H. Chiang and K.-Y. Su, "Discrimination Oriented Probabilistic Tagging," In *Proceedings of ROCLING V,* Taipei, Taiwan, pp. 85--96, Sep. 18--20, 1992.

Lin, Y.-C., T.-H. Chiang and K.-Y. Su, "The effects of learning, parameter tying and model refinement for improving probabilistic tagging," *Computer Speech and Language*, vol. 9, no. 1, pp. 37-61, Academic Press. Jan. 1995.

Papoulis, A., *Probability & Statistics*, Prentice Hall, Inc., Englewood Cliffs, NJ, USA, 1990.

Rabiner, L., and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1993.

Roussas, G. G., *A First Course in Mathematical Statistics*, Addison-Wesley Publishing Company, 1973.

Salton, Gerard and Michael J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1983.

Smadja, Frank, "Retrieving Collocations from Text: Xtract," Computational Linguistics, vol. 19, no. 1, pp. 143-177, 1993.

Smadja, Frank, Kathleen R. McKeown and Vasileios Hatzivassiloglou, "Translating Collocations for Bilingual Lexicons: A Statistical Approach," Computational Linguistics, vol. 22, no. 1, pp. 1-38, 1996.

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

Su, K.-Y.. and C.-H. Lee, 1994a, "Speech recognition using weighted HMM and subspace projection approaches," *IEEE Trans. Speech and Audio Processing*, vol. 2, no.1, pp. 69-74, Jan. 1994.

Su, K.-Y., M.-W. Wu and J.-S. Chang, 1994b. "A Corpus-based Approach to Automatic Compound Extraction", *Proceedings of ACL 94*, 32nd Annual Meeting of the ACL, pp. 242-247, New Mexico State University, 27-30 June 1994.

Tung, Cheng-Huang and Hsi-Jian Lee, "Identification of Unknown Words from a Corpus," *Computer Processing of Chinese & Oriental Languages*, Vol. 8, pp. 131-145, (*Proceedings of ICCPOL-94*, pp. 412-417, Taejon, Korea,) Dec. 1994.

Wang, Mei-Chu, Chu-Ren Huang and Keh-Jiann Chen, "The Identification and Classification of Unknown Words in Chinese: An N-Grams-Based Approach," In Ishikawa, Akira and Yoshihiko Nitta, Eds. Festschrift for Professor Akira Ikeya, pp. 113-123. Tokyo: The Logico-linguistics Society of Japan, 1995.

Wu, Ming-Wen and Keh-Yih Su, "Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count", In *Proceedings of ROCLING VI*, Nantou, Taiwan, ROC Computational Linguistics Conference VI, pp. 207-216, Sep. 2-4, 1993.