CHAPTER 5

MEHODOLOGICAL FRAMEWORK FOR MULTIWORD EXPRESSIONS ACQUISAITION

This chapter presents a detail of implementation of multiword expressions acquisition, its system architecture, different paradigms and standard approaches. Furthermore, it also explains the feature extraction of Bengali Multiword Expressions for detection.

Corpus contains text data in Bengali language. The collection of written text data is done from corpora from historical background, banking sector, News Paper[1], Phrases and Idioms from various English and Bengali Books from different sources, shopping mall web sites etc. in which some MWEs are of popular use in our real life that contains a wide variety of texts corresponding to most common language use over a given time span. We consider word token to be an occurrence of a word in the corpus.

The steps followed while doing MWEs extraction and detection are

Step 1 - Corpora collection and preprocessing

Step 2 - Candidate Selection

Step 3 - Statistical Co-occurrence tests

Step 4 - Extracting Multiword Expressions Features

Step 5 - Detecting Multiword Expressions

5.1 Step 1 - Corpora Collection and Preprocessing

We collect corpora from different domain that contains a wide variety of texts corresponding to most common language use over a given time span. In preprocessing, we had to take some special attention in various phrases like tokenization in which some words were normally not tokenized.

[1]**https://www.anandabazar.com/**

Initially, essential preprocessing is done over the corpus and every word is assigned a set of possible POS tags. For extraction of Noun-Noun, Noun-Verb, Reduplication and Idiomatic compound noun, collocation of each sentence is further analyzed for identification of main verb. This step comprises of the following sub parts as follows

5.1.1 Text Preprocessing

Text preprocessing is a most important area of computational linguistics. Outputs depend on how efficiently we can represent in terms of word alignment, collocation of words etc.

Text preprocessing is done in two steps:

i. Sentence splitter and

ii. Tokenization

In each case we tried to eliminate stop words and applied stemming. The reason for eliminating stop words and stemming is explained below:

5.1.2 Stop words

Some information contained in a text corpus is useless and problematic for MWEs extraction which we called stop words. Stop words are words which are filtered out before and after preprocessing of NLP. The stop words usually refer to the most frequent words in a language. There is no single universal list of stop words used by Natural language tools. Some examples of stop words are - as, the, is, at, which, and so. Stop words can cause problems when searching for phrases that include them, such as 'who', 'take that'. To improve performance, it is required to remove stop words from the text. There is a set of blank words in every language that are common to all domains which are easily identified. For examples, articles, prepositions, conjunctions etc although, they can be verbs, adjectives and adverbs.

If words occurrences is frequent in the text corpus in a particular collection they are not good for discriminations. In fact, it is considered that word which appears more than 70% of a particular of the collection has no use for purpose of detection

and retrieval. These stop words are considered empty and are normally removed to avoid being considered as potential.

In our experiment, preprocessing step is to extract and eliminate stop words from the text corpus with the aim of dropping the context of the text to more specific expression which we called Multiword Expressions, containing the words that are useful and meaningful for the generation of Automatic MWEs detection.

5.1.3 Stemming

Stemming is the process of removing the affixes from inflected words to decrease all words with the same stem to a common form (without doing complete morphological analysis). Stemmer is one of word extracting IR (Information Retrieval) tools. It is useful in many areas of computational linguistics and information-retrieval works. It is assumed that complexity of the stemmer increases with increase of Morphological complexity of the language. Bengali is one of the good complex languages, which demands a good number of rules. Experimental results show that Information Retrieval performance is enhanced by 28%, 42% and 20% for Hindi, Marathi and Bengali (Dolamic, 2010).

We also have done sentence splitter as tokenization in preprocessing step. Sentence splitter divides the contents into token and non significant character are removed. In tokenization, tokens are divided by their POS tag and generally Noun-Noun (N-N), Noun-Verb (N-V), reduplication and Idiomatic Compound Noun can be declared as potential MWEs. The next step is to choose only common words instead of unique ones. Keywords are used for clustering and a keyword is useless when it is assigned to only a few documents, that is why we setup rules (features) for N-N, N-V, reduplication and Idiomatic compound Noun. If the features are matched with the words, they are considered as MWEs, otherwise they are it is missed by the system and will not come under MWEs categories. The remaining tokens and named entities are declared as keywords candidate.

After tokenization, we move for Part of Speech tagging for which we use POS tagger *viz.* TnT tagger, as explained in 4.1.1.3 as Trigrams's Tags or TnT is an efficient statistical Part of speech tagger. This tagger is based on HMM and uses some

optimization techniques for detecting and handling unknown words. It performs other current approaches, including the maximum entropy framework. TnT Tagger uses second order Markov models for part-of speech tagging. The states of the model represent tags, outputs represent the number of words. Transition state probabilities depend on the transition states, thus pairs of tags. Output probabilities merely depend on the most current transition states.

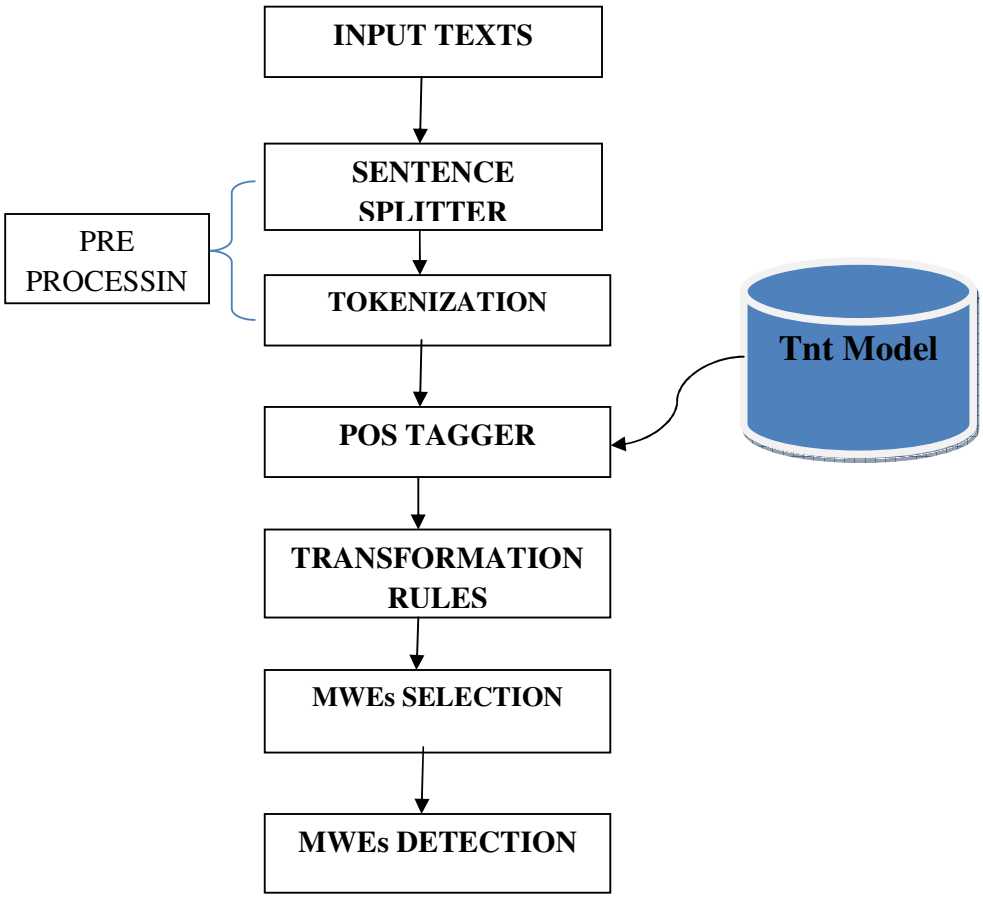

Figure 5.1: System architecture of MWEs Extraction and Detection

TnT Model follows *n*-gram model which is a statistical language model that is based on a probability distributions P(s) over all possible word sequences (or any other linguistic unit like words, sentences, paragraphs, documents, or spoken utterances). In TnT training mode, where the feature of Multiwords are clustered, first based on a certain criterion.

According to Siddiqui and Tiwary (2008), the aim of a statistical language model is to estimate the probability (likelihood) of a sentence. This is obtained by decomposing sentence probability into a product of conditional probabilities (here we considered words) using the chain rule as follows:

$$P(s) = P(w_1, w_2, w_3, \ldots, w_n)$$

$$= P(w_1)P(w_2/w_1)P(w_3/w_1w_2)P(w_4/w_1w_2w_3).....P(w_n/w_1w_2....w_{n-1}))$$

$$= \prod_{i=1}^{n} (P(w_i/h_i) \qquad (5.1)$$

where h_i is history of word w_i defined as

$$W_1 W_2 W_3 .....W_{i-1} \qquad (5.2)$$

So, in order to find sentence probability, we need to estimate the probability of a word, given the sentence of words preceding it. An *n*-gram model simplifies the task by approximating the probability of a word, given all the previous words by the conditional probability, given previous $n-1$ words only.

$$P(w_i/h_i) \approx P(w_i/w_{i-n+1.....}w_{i-1}) \qquad (5.3)$$

Thus, an *n*-gram model calculates $P(w_i/h_i)$ by modeling language as Markov model of order *n*-1, i.e., by looking at previous *n*-1 words only. A model that limits the history to the previous one word only is termed as bi-gram (*n*=1) model. Similarly, a model that conditions the probability of word to the previous two words, is called a trigram (*n*=2) model. Words bigram and trigram language model denote *n*-gram model language models with *n* = 2 and *n* = 3, respectively. Using bi-gram and tri-gram approximation, the probability of a sentence can be calculated as:

$$P(s) \approx \prod_{i=1}^{n} P(w_i/w_{i-1}) \qquad (5.4)$$

and

$$P(s) \approx \prod_{i=1}^{n} P(w_i / w_{i-2}.w_{i-1}) \qquad (5.5)$$

Now, we can explain how to estimate probabilities of bigram and trigram. This is done by training the *n*-gram model on the training corpus. We estimate *n*-gram parameters using the Maximum Likehood Estimation (MLE) technique, i.e., using relative frequencies. We count a particular *n*-gram in the training corpus by dividing it by the sum of all *n*-grams that share the same prefix.

$$P(w_i / w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_{i-1}, w_i)}{\sum C(w_{i-1+1}, \ldots, w_{i-1}, w)} \qquad (5.6)$$

The sum of all *n*-grams that share first *n*-1 words is equal to the count of the common prefix $w_{i-n+1}, \ldots, w_{i-1}$. So, we rewrite the previous expression as follows:

$$P(w_i / w_{i-n+1}, \ldots, w_{i-1}) = \frac{C(w_{i-n+1}, \ldots, w_{i-1}, w_i)}{\sum C(w_{i-1+1}, \ldots, w_{i-1}, w)} \qquad (5.7)$$

The model parameter we have found using these estimates, maximizes the probability of the training set T given the model M, i.e., P(T/M). The frequency with which a word occurs in a text may not be same as in the training set. This model only provides the most likely solution.

Consider in a collection of 10,000 aligned English/Bengali sentence pairs the word mango appears 200 times (i.e., in 2% of documents). In the 200 or so Bengali documents aligned with the English documents containing mango we can count the frequencies of occurrence of each word that occurs. Suppose that 'আম' appears in 225 documents in the entire Bengali collection and 175 times in the 200 documents of interest. Pointwise Mutual Information (PMI) can be used to score 'আম' as a possible translation for mango. PMI is the log of the ratio between the joint probability and the product of prior probabilities. An example of scoring the word 'আম' given below:

$$\text{PMI}\ (\text{Mango},\ \text{আম}) = \left(\log \frac{P(Mango,\text{আম})}{P(Mango)*P(\text{আম})}\right)$$

$$= \log\left(\frac{\frac{175}{200}}{\frac{200}{10000} * \frac{225}{10000}}\right)$$

$$= \log\left(\frac{.875}{.0200 * .0225}\right)$$

$$= 10.924$$

Thus, আম receives a score of 10.924. Pointwise Mutual Information is an boundless metric where bigger values specify greater degree of association.

5.2 Step 2 - Candidate Selection

Proposed algorithm select candidate (text) in bigram and trigram in sequence from the tagged corpus. The tagset that we used for POS tagging consist of abbreviated forms such as Noun Common (NC), Noun Proper (NP), Verb Main (VM) and Verb Auxiliary (VA) etc. Thus we filter our bigram and trigram which have NC-NC, NP-VP, NP-VA and NC-NC-NC, NC-NC-VM etc. respectively.

Three approaches to MWEs extraction as proposed by Weiwei (2012) are:

1. Linguistic Approaches
2. Statistical Approaches
3. Hybrid Approaches

5.2.1 Linguistic Approaches

Construction of MWEs in each language is based on linguistic rules of some syntactic and morphological structures. In English, for example, MWEs like noun phrases are generally composed by nouns, prepositions and adjectives. If we can identify these syntactic and morphological structures, we will be able to recognize MWEs easily, since we have the knowledge about how the MWEs are composed. While most linguistic approaches tend to recognize MWEs according to their syntactic and morphological structures, there are other approaches which try to filter out MWEs by context analysis.

Bourigault (1992) in his work uses partial grammatical analysis to identify noun phrases and introduces *LEXTER:* an early multiword term extraction system for French. The system is composed of two steps: analysis and parsing. In the analysis step, text is annotated with grammatical information though analysis rules and each word is tagged with its grammatical category (part of speech). For a word sequence, the grammatical categories of words form a grammatical pattern such as noun-noun and adjective-noun.

For example, when words "blue" and "ribbon" (Bengali words like কাচা and মাল) are put together, a grammatical pattern of adjective-noun is formed. Some patterns are "negative" since they are never used for MWEs, while others are "positive" as they are frequently used for MWEs. The analysis step uses "negative" patterns as an important conclusion to isolate the maximal-length noun phrases from text. The parsing step uses "positive" patterns to attain the likely maximal-length noun phrases from the maximal-length ones. It is seen that partial grammatical analysis is advantageous over complete syntactic analysis. The linguistic approach focuses on the grammatical categories of words and the grammatical structures of word sequences other than the actual position of words in the sentence, making the analysis more efficient and accurate (Weiwei, 2012).

Jacquemin and Christian (1999) proposed a two-tier framework for multiword expression extraction which is composed of a paradigmatic level and a syntagmatic level. The paradigmatic level examines how expressions are composed by lexical items such as words, while the syntagmatic level determines the syntactical structures of the expression. As shown in Fig. 5.2, an example taken from his experiment, where the expression "speed measurement" can be represented as:

$$\left\{ \begin{array}{l} Paradigm : \left\{ \begin{array}{l} < N_1 lemma >= measurement \\ < N_2 lemma >= speed \end{array} \right. \\ Syntagm : \{ N_0 \rightarrow N_2 N_1 \} \end{array} \right\}$$

Fig 5.2. Syntagmatic relationships between Words

These two levels reflect the inner relationships between Multiword Expression variations in morphological, syntactic and semantic relations. Likewise other

expressions can be found by considering the semantic information as available in WordNet (Miller *et al.*, 1990), *viz.* $N(speedup) = \{fast_N, swift_N, rapid_N ....\}$ Thus, unknown MWEs can be predicted by matching against similar patterns with known expressions in this linguistic approach.

5.2.2 Statistical Approaches

Statistical approaches focus to extract MWEs from text corpora by means of association measures (Church and Hanks, 1989). For example, the term "খাওয়া দাওয়া" often occurs repeatedly in text, indicating that there is some kind of "bond" or cohesiveness between the words. Statistical approaches apply statistical techniques to determine the degree of cohesiveness between the constituents of possible MWEs. Compared with linguistic approaches, statistical approaches are more popular since they are flexible and often domain/language independent. In statistical approaches, co-occurrences of words are applied to large document corpora and more complex probability models are proposed. Church and Hanks (1989) in their work introduces measurement for words cohesiveness called the association ratio. It is based on mutual information. The mutual information of two given words *x* and *y,* I(x, y) is defined as:

$$I(x, y) = \log_2 \frac{P(x, y)}{P(x).P(y)} \qquad (5.8)$$

where P(x) and P(y) stand for the probability of x and y respectively (x, y) is the joint probability of x and y.

Mutual information is a measurement that identifies the co-occurrence probability of two words. If the words x and y are closely associated, the joint probability P(x, y) will be greater than the product P(x)· P(y); then I(x, y) >>0. If the two words are completely independent, the joint probability P(x, y) should be equal to P(x)·P(y); then I(x, y) ≈0.

The definition of mutual information is same as association, but it is different in two logic. First, mutual information is symmetric since P(x, y)=P(y. x), but the association ratio is not symmetric, since f(x, y), the co-occurrence of word x followed by word y, is different from f(y, x). Second, f(x, y) is often counted in a series of *w*

words, so the length of the series will affect f(x, y). However, the relationship between two words can be measured by the association ratio (Daille, 1995). It is noted that the association ratio is unstable when the word count is small, and as a result, it is mostly used to extract bigram expressions.

Silva *et al*. (1999) introduced the LocalMaxs algorithm which assumes that MWTs have strong relation within them. The authors define a new association measure for terms called Symmetrical Conditional Probability (SCP) for measuring the "correlation" between two words as follows:

$$SCP = p(x/y).p(y/x) = \frac{p(x,y)^2}{p(x).p(y)} \qquad (5.9)$$

where p(x, y) is the probability of the bigram (x, y) appearing in the corpus, p(x) is the probability of the unigram *x* appearing in the corpus, and p(y) is the probability of the unigram *y* appearing in the corpus.

To simplify this measure for *n*-grams, the authors introduce the fair dispersion normalization which breaks an *n*-gram $w_1w_2.....w_n$ at different dispersion points and considers it as combinations of the two parts. For example, an *n*-gram $w_1w_2.....w_n$ can be broken into a bigram w_1w_2 and a (*n*-2) *n*-gram $w_3w_4.....w_n$ if we choose the dispersion point between w_2 and w_3. To measure the "cohesiveness" between the words in an *n*-gram, we calculate the average of the products for the two parts at different dispersion points of the *n*-gram.

$$Avp = \frac{1}{n-1}\sum_{i=1}^{n-1} p(w_i.....w_j).p(w_{i+1}....w_n) \qquad (5.10)$$

where *n* is the length of the *n*-gram and $p(w_1...w_n)$ is the probability for the word sequence $w_1...w_n$. Fair Dispersion Normalization then uses the average product to normalize association measure for a given *n*-gram, which is defined as:

$$SCP\_f(w_1w_2....w_n) = \frac{p(w_1w_2...w_n)^2}{Avp} \qquad (5.11)$$

Depending on Fair Dispersion Normalization, the LocalMaxs algorithm tries to find an *n*-gram that has a stronger *SCP_f value* than any (*n*-1)-gram within it and any (*n*+1)-grams containing it, and treat such *n*-grams as MWEs.

Aires *et al*. (2008) proposes an improvement on the original Localmaxs algorithm (Silva *et al.,* 1999) by introducing a smoothed LocalMaxs algorithm, which extends the search from local maxima to global maxima. The Smoothed LocalMaxs algorithm still uses Symmetrical Conditional Probability (SCP) as the association measure to rank the "bond" within MWEs. However, the SCP is calculated from the frequencies of the terms instead of their probabilities. Given a text file that contains N words, the number of unigrams will be N, the number of bigrams will be N-1, and the number of *n*-grams will be N-n+1. When N>>n, N>>N-n+1.

Thus,

$$P(ngram) = \frac{freq(ngram)}{N-n+1} \approx \frac{freq(ngram)}{N} \tag{5.12}$$

and the SCP of a word sequence *x* and a word sequence *y* can be computed as follows:

$$Scp(x,y) = \frac{p(x,y)^2}{p(x).p(y)} = \frac{\frac{freq(x,y)^2}{N}}{\frac{freq(x)}{N}.\frac{freq(y)}{N}} = \frac{freq(x,y)^2}{freq(x).freq(y)} \tag{5.13}$$

In addition, Aires *et al*, (2008) utilizes a suffix array and the related structure to store the n-grams and their information associated with such as frequencies, positions and lengths. They also find out efficient accuracy in their improvement. We apply n-gram model for our experiment.

5.2.3 Hybrid Approaches

Statistical approaches focus on the repeated characteristics of MWEs while linguistic approaches study the syntactic structures. Both have their advantages and limitations. Hybrid approaches attempt to join linguistic and statistical techniques to extract MWEs. Linguistic approaches can be applied first to find multiword expression candidates, and then statistical approaches are used to select improved candidates, or vice versa (Justeson and Katz, 1995).

Hybrid approaches tend to filter out some MWEs like compound nouns. The hybrid approaches help in extracting MWEs such as bigrams. For lengthy MWEs that contain more than two words, the two approaches have similar performance.

5.3 Step 3 - Statistical Co-Occurrence Tests

This step involves various statistical measurements through which we can test the connectedness of the collocation. It further exploits whether the pattern are habitual or accidental. For both the measures, frequency is counted for those context containing bigrams that are either in open or hyphenated form. For example, ধীরে ধীরে (Dheere Dheere, means slowly) is considered to be same in Bengali. For comparisons, the occurrence of words, We apply PMI and to observe frequencies we used Chi-Square Test.

5.4 Step 4 - Extracting Multiword Expressions Features

5.4.1 Extraction Methods

The proposed method for Multiword Expression extraction along with three other models of extraction systems are described below. In particular, we propose a new association measure and a smoothing method that works along with the LocalMaxs algorithm for multiword expression extraction. In addition, we include a simple filtering step that helps to improve the performance of our methods for multiword expression extraction.

5.4.1.1 Statistical Association Measures

Researchers realised that words that form a MWE should have relatively strong "bond" with each other since an MWE has the characteristic of being used repeatedly. Based on this assumption, statistical association measures are employed to calculate the "bond" values within *n*-grams for the purpose of extracting MWEs.

A simple and basic association measure is frequency. Given an *n*-gram, if it has a high frequency in a corpus, we can say that the words of the *n*-gram have strong "bond" within them since they often occur together. However, frequency is not a good association measure since it usually leads to poor results. According to Zipf's law (Li, 1992), there are always a large number of MWEs that have low frequencies in a corpus of reasonable size.

To identify more MWEs with a reasonable precision, more sophisticated association measures have been developed such as mutual information (Dagan *et al.*, 1993), the dice measure (Dunning,1993) and symmetric conditional probability (Silva

et al., 1999). These association measures are intended to capture the "bond" within possible MWEs without using the frequency values directly.

We propose a new association measure that helps measure the "bond" within MWEs with relatively low frequencies in a corpus and reduce the noise from irrelevant n-grams with high frequencies. It will be used with the LocalMaxs algorithm in order to extract MWEs effectively.

5.4.1.2 LocalMaxs Algorithm

The LocalMaxs algorithm is originally proposed by (Silva *et al.*, 1999). The LocalMaxs algorithm is domain as well as language independent and takes collection of texts as input and gives MWEs as output. It selects MWEs the *n*-gram based on two assumptions. First, the more cohesive an expression is, higher the association measure it should have, Second, MWEs are localized *n*-grams which have strong association within it. The assumptions follow antecedent and successor. Before we describe the algorithm, we need to introduce these two concepts first, antecedent and then successor.

An antecedent of an *n*-gram $w_1 w_2 \ldots . w_n$ is a sub-*n*-gram with size *n*-1, either $w_1 \ldots w_{n-1}$ or w_2, …, w_n. We denote the set of all antecedents for an n-gram *W* as: ant*(W)*.successor of an n-gram $w_1 w_2 \ldots . w_n$ is a super-*n*-gram containing an additional word before (to the left) or after (to the right) of the *n*-gram. An *n*-gram can have more than one successor, since any word can appear before or after it. We denote the set of all successors for an *n*-gram *W* as: succ(W)

$g(w) \geq g(succ(w)) \wedge g(w) > g(succ(w))$ W's size>3

$g(w) > g(succ(w)$ W's size=2

where g(.) is a function that assigns an association measure value to the *n*-gram W.

The LocalMaxs algorithm is flexible as it allows different association measures to be used as long as they obey the first assumption (the more cohesive an expression, the more association measure it should have). Many experiments are performed with different association measures in (Silva *et al*., 1999; Dias *et al*., 2000). However, the

LocalMaxs algorithm can extract long MWEs such as compound nouns by comparison.

A part from selecting the n-grams whose association measure values are locally maximal, the work of (Aires *et al*., 2008) proposes the Smoothed LocalMaxs algorithm which requires the association measure value of an MWE to be higher than the average value of *max(g(ant(W)))* (the highest association measure value of all its antecedents) and max(g(succ(W))) (the highest association measure value of all its successors). If an *n*-gram *W* is an MWE, we can describe the Smoothed LocalMaxs algorithm as follow:

$$g(w) \succ \frac{max(g(ant(w))) + max(g(succ(w)))}{2} \quad \text{W's size>3}$$

$$g(w) \succ \max(g(succ(w)))^2 \quad \text{W's size=2}$$

The Smoothed LocalMaxs algorithm provides a global standard to decide if an *n*-gram is an MWE. According to the original LocalMaxs algorithm, if an n-gram is selected as MWE, neither its antecedents nor successors will be selected as MWEs. The Smoothed LocalMaxs algorithm can select an MWE even if it is not a local maximum. As a result, an *n*-gram and it successors or antecedents can be selected as MWEs at the same time.

5.4.1.3 Smoothed Probabilities of *n*-Gram

Basically association measures are based on the frequencies of n-grams. One difficulty is that there is usually a large number of missing *n*-grams in a corpus of reasonable size, controlled by the Zipf law (Power and David, 1998), which is called the sparse data problem. If the order of the *n*-grams increases, the more missing *n*-grams we have in a corpus. For example, the bigram "school of" will positively occur much more often than the four-gram "school of physical sciences". In theory, if there are 20,000 words in a corpus, then we can have 400 million ($20{,}000^2$) possible bigrams and 8 trillion ($20{,}000^2$) possible trigrams. In practice, however, the number of *n*-grams actually covered by a corpus is much smaller. The huge gaps between the possible words of real bigrams and trigrams make the occurrences of the average bigram much bigger than that of the average trigram. Since the LocalMaxs algorithm

relies on the comparison of the association measures of an *n*-gram with those of its antecedents or successors, such huge gaps make the algorithm not effective for extracting long MWEs.

While we cannot do much about the frequency of an n-gram, we can enhance the way we calculate its probability through smoothing techniques, thus making the association measures as good as possible. Many association measures of an n-gram $w_1 w_2 ... w_3$ are based on the joint probability $p(w_1 w_2 ... w_3)$, which means the probability of words $w_1, w_2 ...$ and w_3 appearing nearest to each other in text. The joint probability is symmetric in that *P*(*AB*) = *P*(*BA*), which implies that the word order does not matter. However, a good association measure should reflect the 'bond' value within an *n*-gram based on the order of its words; so we apply the chain rule to extend the joint probability into the product of a series of conditional probabilities as follows:

$$p(w_1 w_2 ... w_n) = p(w_1)\, p(w_2 | w_1)\, p(w_3 | w_1 w_2) ... p(w_n | w_1\, w_2 ... w_{n-1}) \qquad (5.14)$$

Where $p(w_n | w_1 w_2 ... w_{n-1})$ is the conditional probability of word W_n occurring after the sequence $w_1 w_2 ... w_{n-1}$ in text.

The conditional probability is asymmetric since it takes the order of words for its account. Other benefit is that it is less dependent on large frequencies. Given a bigram $w_1 w_2$ the conditional probability $p(w_1 | w_2)$ is defined as follows:

$$p(w_1 | w_2) = \frac{freq(w_1 w_2)}{freq(w_1)} \qquad (5.15)$$

where $freq(w_1 w_2)$ is the frequency of bigram $w_1 w_2$ and is the frequency of word $freq(w_1)$. If both $w_1 w_2$ and w_1 have low frequencies, the conditional probability $p(w_1 | w_2)$ can still be relatively high. On the other hand, if $w_1 w_2$ is a high frequency bigram but an unrelated expression, the frequency of word w_1 should also be high, making the conditional probability $p(w_1 | w_2)$ which is relatively small. Thus, the conditional probability can manage itself with the frequencies: high frequency *n*-grams do not always get high conditional probabilities

while low frequency *n*-grams do not always get small conditional probabilities. Such a property is very much important for extraction of MWEs.

According to W.Huo (2012), conditional probabilities are less dependent on frequencies, the sparse data problem still exists in conditional probabilities. To get around the problem, we apply the shrinkage method to smooth order conditional probabilities. The shrinkage method is based on the hypothesis that the chance of an *n*-gram occurring in a text can somehow be approximated by a shorter (*n*-1)-gram. For instance, whenever a trigram $w_1 w_2 w_3$ occurs, its shorter bigram $w_2 w_3$ will also occur in text. When a shorter (*n*-1)-gram has a high chance to occur, the n-gram itself also occur more frequent. Since we required a reasonable value to estimate the probability of a long *n*-gram, we combine the probabilities of the *n*-gram and its (*n*-1)-gram linearly with appropriate weights as follows:

$$p(w_n \mid w_1 w_2 ... w_{n-1}) = (1-\lambda) p_n(w_n \mid w_1 w_2 ... w_{n-1}) + \lambda p_{n-1}(w_n \mid w_2 w_3 .... w_{n-1}) \quad (5.15)$$

where $p(w_n \mid w_1 w_2 ... w_{n-1})$ is the conditional probability for *n*-gram $w_1 w_2 ... w_n$, $p_{n-1}(w_n \mid w_2 w_3 ... w_{n-1})$ is the conditional probability for the (*n*-1)-gram $w_2 w_3 ... w_n$, and λ is a parameter used to adjust the weights for these two parts.

According to formula (5.13), when we compute the conditional probability for the *n*-gram $w_1 w_2 ... w_n$, we need the probability of its shorter (*n*-1)-gram $w_2 w_3 ... w_n$. This can be extended to a recursive process, and different λ 's can be used for combining conditional probabilities of *n*-grams with different size of lengths.

5.4.1.4 Normalized Sequence Probabilities

Depending on the smoothed probabilities of *n*-grams, we propose a new association measure for a sequence of words that calculates the "bond" within an n-gram with the normalized sequence probability for the *n*-gram. We view an *n*-gram as a sequence of words: if the "bond" within the sequence is strong, the words composing the sequence tend to occur together in the given order. As a result, the joint probability in words occurring together, should be high as well. In particular, we define the normalized sequence probability of an n-gram in our experiment for Multiwords containing reduplication properties as follows:

$$seq\_p(w_1 w_2 \ldots . w_n) = \sqrt[n]{p(w_1 w_2 \ldots w_n)} \qquad (5.16)$$

where we apply the chain rule and use the smoothed conditional probabilities to compute the joint probability for the *n*-gram. In addition, by taking the n^{th} root to normalize the joint probability, the "affinity" values of n-grams of different sizes can be compared accurately.

5.5 Data Preparation

Some information contained in a text corpus is useless or even create problem for MWEs extraction. As a result, we need to filter out such information through pre-processing. Our data preparation is based on annotation which we check in two ways

i. **Manual annotation:** In manual annotation, we go through the list of data, making a binary decision on whether the proposed word combination is true MWEs. This process depends on availability of native speakers to perform the annotation. Actually large sample required to be annotated in order to achieve more reliable evaluation measure. But this process is quite time consuming depending on the type of expression and required expert linguistics. Again, it is not possible to perform manual annotation several times.

ii. **Automatic Annotation:** In automatic annotation we consider that corpus containing large target MWEs that exist in our corpus. This may be a regular dictionary or a simple list of MWEs which called is gold standard (GS) or reference dictionary (Ramisch, 2015). To complete this we consider that gold standard is complete or at least that it contain broad coverage of the target MWEs. Hence, we considered that assumed MWEs are true positives while we considered those not contained in the gold standard we considered them as false MWEs.

We made annotations so that different kinds of tokens can be distinguished and selected. We distinguish the following kinds of tokens: word made of letters; numbers made of digits apostrophized words; hyphenated words; idioms, abbreviations, words

connected by ampersands, whitespaces and newlines, and end-of-sentence marks (including "." "," "?" "!"). After the tokenization, the following steps are performed

a. Eliminate all illegal and meaningless characters.
b. Eliminate all non-textual information.
c. Separate all hyphenated words if there are more than one "-"mark in the tokens.
d. Separate all words connected by "&" if there are more than one "&"mark in the tokens.
e. Eliminate suffixes to extract stems.
f. Keep the original format of the text, including the spaces, line breaks and end-of-sentence marks.
g. Remove tokens that contain both letters and digits with lengths longer than 5 characters.

After eliminating all the "unwanted" information, we normalize all MWEs based on their properties and start to extract n-grams along with their frequencies. Although the most commonly used MWEs are between two and three words, we restrict ourselves to *n*-grams of up to trigram (n=3) based on computing power of our implemented system tool.

5.6 Multiword Expression Extraction System

Here we describe our system model for extracting MWEs based on the new association measures and the LocalMaxs algorithm. We implement four versions of our system that work for feature extractions so that we can determine the effects of smoothed probabilities and normalized sequence probabilities through experiments in Chapter 6.

5.6.1 Method Based On Sequence Frequencies

Here word occurrences are checked in the system. It is based on the Symmetrical Conditional Probability (SCP) method from (Silva *et al*., 1999). However, instead of using probabilities to calculate the SCP value, we use frequencies directly as recommended by the work (Aires *et al*. 2008), since when the number of words N in a corpus is large, the formula based on frequencies is equivalent to that

based on probabilities. By using frequencies, we also simplify the calculations for extracting MWEs.

5.6.2 Method Based On Smoothed Probabilities

This method replaces the joint probabilities in the calculations of SCPs with our smoothed conditional probabilities as discussed in section 5.2.2. We want to know if the smoothed method is helpful to stabilize the association measure and leads to a better performance.

5.6.3 Method Based On Normalized Sequence Frequencies

This method explores the potential of our normalized sequence probabilities when they are combined with the calculations of probabilities based on the frequencies as described in above. We want to know if the normalization method itself is effective in extracting MWEs when used with the LocalMaxs algorithm.

5.6.4 Method Based On Normalized Smoothed Probabilities

This method combines the normalized sequence probabilities with the smoothed conditional probabilities of n-grams. We consider that such a combination can not only address the sparse data problem for high-order *n*-grams, but also bring better performance due to the direct comparisons between the probabilities of different *n*-grams after the normalization.

All four methods share the extraction process. Likewise, all the results from these methods are further processed for filtering so that stop words and words with not matching with the given features are eliminated. In addition, all four methods use the LocalMaxs algorithm in selecting final MWEs. These four methods help us to measure how our proposed approach performs for the Multiword Expression extraction. We found that method based on Normalized Smoothed Probabilities (NSP) has the best performance which will be demonstrated by the experiments in Chapter 6.

5.7 STEP5 - DETECTING MULTIWORD EXPRESSIONS DETECTION

The step 5 of system architecture is described in detail in the chapter 6, in which evaluation method, system performance, experimental result, result analysis and finally detection methods are explained thoroughly.

5.8 Chapter Summary

In this chapter we proposed an evaluation framework for evaluating automatic Multiword Expression extraction methods under common terms, so as to be able to compare their performance. We discussed how we train our data using TnT model and various approaches for MWEs extraction. We also discussed how we prepare data based on annotation following under set of rules.