Automatic Multi-word Term Extraction and its Application to Web-page Summarization

by

Weiwei Huo

A Thesis
presented to
The University of Guelph

In partial fulfilment of requirements
for the degree of
Master of Science
in
Computer Science

Guelph, Ontario, Canada

ABSTRACT

Automatic Multi-word Term Extraction and its Application to Web-page Summarization

Weiwei Huo
University of Guelph, 2012

Advisor:
Prof. Fei Song

In this thesis we propose three new word association measures for multi-word term extraction. We combine these association measures with LocalMaxs algorithm in our extraction model and compare the results of different multi-word term extraction methods. Our approach is language and domain independent and requires no training data. It can be applied to such tasks as text summarization, information retrieval, and document classification.

We further explore the potential of using multi-word terms as an effective representation for general web-page summarization. We extract multi-word terms from human written summaries in a large collection of web-pages, and generate the summaries by aligning document words with these multi-word terms. Our system applies machine translation technology to learn the aligning process from a training set and focuses on selecting high quality multi-word terms from human written summaries to generate suitable results for web-page summarization.

Acknowledgments

I am grateful to Prof. Fei Song for his continued support, encouragement, and insights. And I would like to thank Prof. Mark Wineberg for his precious advice on my writing and thesis work. I would also like to thank my schoolmate Song Lin for his help and encouragement.

Special thanks to Prof. Yang Xiang and Prof. Fangju Wang to be on my thesis examination committee and for their helpful feedback.

Tables of Contents

Abstract

Acknowledgments

Tables of Contents

Chapter

List of Figures

List of Tables

Chapter 1

Introduction

A multi-word term (MWT), or simply a term for short in this thesis, is an expression consisting of more than one word with a grammatical structure and a specific meaning such as nouns phrases (e.g., swimming pool, Natural Language Processing), fixed collocations or idioms (e.g., blue moon, apple and orange), compound verbs (e.g., take into account), prepositional phrases (e.g., on the contrary), compound determiners (e.g., a piece of), and many others.

MWTs can intrinsically identify what a particular piece of writing is about, capturing the primary entities or key concepts for it. For example, given the sentence: "*Natural Language Processing* (NLP) is a *field of computer science* concerned with the understanding and generation of *human languages*.", the MWT "*Natural Language Processing*" corresponds to the major entity and other MWTs such as "*computer science*" and "*human languages*" are used as related concepts. Therefore, recognizing MWTs is both advantageous and important to text representation and understanding. Many tasks in Natural Language Processing require techniques to compute MWTs such as Information Retrieval (IR) (Witschel, 2005), Text Summarization (Dunning, 1993; Silva *et al*. 1999), Document Classification (Monta *et al*. 2005; Hovy *et al*. 2000), and Named Entity Recognition (Pal *et al*. 2010). With the rapid growth of digital documents on the Internet, the need for automatic multi-word term extraction is increasing.

In this thesis, we will examine the existing techniques for multi-word term extraction. We will also propose our own methods and compare them with some well-known existing techniques. In addition, we will explore the potential of using MWTs in the task of summarization of web pages. Due to the diversity of web-pages (some with a lot of text and some with few scattered phrases but plenty of graphics), MWTs can provide a suitable and useful solution to the summarization of such documents.

We will carry experiments to evaluate our method for multi-word term extraction on the data sets from Open Directory Project (OPD). The same data set and another data set from Document Understanding Conferences (DUC) will be used to evaluate web-page summarization based on machine evaluation from ROUGE measures.

1.1 Applications

Many Natural Language Processing areas call for multi-word term extraction techniques, such as Information Retrieval, Named Entity Recognition, Document Classification, and Text Summarization.

Information Retrieval needs MWTs in the index (Witschel, 2005) to improve the query search process. For example, if a user needs some documents about “hot dog”, multi-word term extraction can lead him/her directly to the documents that contain “hot dog” rather than the ones that contain “hot” and “dog” separately. In addition, multi-word term extraction techniques can refine queries from the users, making their search request more accurate. For example, Google offers such a service to help users refine their queries. When “text” is typed, a list of MWTs that start with “text” will appear automatically to help a user generate the query she/he wants.

Named Entity Recognition is the task to recognize such entities as person names, locations, and organizations automatically from a given corpus and then sort them into the corresponding categories. For instance, we may find “Michael Jackson” from a newspaper article and identify it as a person name. Multi-word term extraction techniques can be helpful since named entities are often described as MWTs. Thus, multi-word term extraction helps provide candidates for further processing.

Document Classification aims to group documents into pre-defined categories automatically. Most classification techniques tend to select discriminative words as features to represent documents and classify them based on the similarities between these documents. Multi-word term extraction techniques can help the task by providing high quality MWTs as features (Monta *et al*. 2005).

With the fast development of the Internet, there are thousands of huge number of documents available online that can be searched by the user. As a result, it is important to show the key contents for all the pages in a search result so that the user can decide which pages should be examined further. Text Summarization allows us to automatically generate short summaries for the related documents.

Traditional text summarization techniques are mostly focused on formal and well-structured text and try to extract important sentences as a short summary of the original text. This is because well-structured text is often coherent and contains topic sentences and key paragraphs to describe the main ideas. As a result, these techniques can just focus on the selection of the most discriminative sentences or paragraphs for generating summaries.

However, documents like emails and web pages are sometimes not formal and may not be coherently structured. For example, there can be multiple topics simultaneously described in short text along with distractive information such as advertisements. Some web pages may contain diverse contents (e.g., images, bullet points, short phrases) and irrelevant fragments (e.g., navigation bars, copyright notices). For text of this kind, paragraphs are not always available; sentences may contain redundant and conflicting information; and words may be inaccurate or even misspelled. As a result, MWTs may be the only suitable information units for describing the web content. In this thesis, we will apply our multi-word term extraction methods to web-page summarization and evaluate its performance with real web data.

In addition to the above areas, multi-word term extraction techniques can also contribute to many other NLP tasks such as Questing Answering (Hovy *et al*. 2000) and Machine Translation (Chiang, 2005). The wide range of applications makes it crucial and urgent to develop more effective and robust techniques for multi-word term extraction.

1.2 Motivation

Over the past twenty years, there has been a great deal of efforts directed toward efficient and robust systems for automatic multi-word term extraction. The mainstream methods assume that terms are composed of some grammatical patterns (Nenadić *et al*. 2002) or follow certain statistical distributions (Shimohata, 1997; Dunning, 1993). Linguistic rules can be constructed to recognize MWTs (Bourigault, 1992; Nenadić *et al*. 2002) or serve as filters to remove illegal MWTs. The performance of such methods is often not satisfactory since the rules of a language are too complicated to be fully captured under the current technologies. The statistical association measures try to distinguish MWTs by analyzing the word occurrences or co-occurrences. One simple measure is to use term frequencies (Salton and Buckley, 1988) to select terms. Other improved measures are TF×IDF and KEA (Witten *et al*. 1999) and C-Value/NC-Value (Frantzi *et al*. 2000). These methods tend to have poor performance since some MWTs may have low frequencies or do not follow the underlying statistical assumptions. In addition, these methods often need cutoff thresholds which are hard to optimize. Hybrid approaches try to combine linguistic rules and statistical measures (Seretan and Wehrli, 2006; Justeson and Katz, 1995), but the intrinsic problems of these two methods remain (Dias *et al*, 2000).

The LocalMaxs approach (Silva *et al*. 1999) is based on the assumption that MWTs have strong glues within them. It can apply various statistical association measures to weight n-grams and select MWTs without relying on language-specific information. However, we find that the original association measure, Symmetrical Conditional Probability (*SCP),* and its normalization in (Silva *et al*. 1999) and (Aires *et al*, 2008) can be improved. In our experiments, we tested the LocalMax approach with *SCP* association measures on our data set and found that the performance is not as good as expected. Further analysis shows that this is caused by the sparse data problem: the frequency of a short word sequence (usually composed by two words) is far greater than that of a longer one (composed by more than two words) in our corpus. For example, the word sequence “president of” may appear 100 times in a collection while a longer sequence “president of America” may occur only once. This huge frequency gap between n-grams makes *SCP* no longer suitable to differentiate between MWTs. To overcome this problem, we propose several new association measures based on smoothed n-gram probabilities and a different normalization method. Along with the LocalMaxs approach, we try to explore effective ways of extracting MWTs.

The fast growth of the Internet increases the need for web-page summarization. Traditional approaches based on sentence selection (Strzalkowski *et al*. 1998; Lin, 1999) are no longer suitable since web-pages are often poorly-structured and may not be coherent. Although extra information from web-pages has been utilized to enhance summarization such as hypertext links (Glover *et al*, 2002; Amitay and Paris, 2000; Delort *et al*, 2003) and click-through data (Sun *et al*, 2005), such information can also lead to inaccurate results as web-pages often contain lots of advertisement links and other irrelevant contents. We believe that automatic MWT extraction can provide a suitable solution to this problem as we explained earlier in this chapter. We will extract MWTs from human-edited summaries and use them to generate web-page summaries. We will demonstrate by experiments that such MWTs edited by humans are not only suitable but also readable for web page summarization.

1.3 Major Contributions

The main contribution of this thesis is a set of new association measures for automatic multi-word term extraction. Our approach is language independent, requires no training data, and can overcome the sparse data problem effectively. Moreover, it can successfully capture the longer MWTs from a corpus. It outperforms other related methods in our experiments and successfully locates more MWTs longer than two words. In addition, we introduce a simple smoothing method to calculate the probabilities. We use it to give more weights on longer MWTs so that they can be recognized. It can help overcome the sparse data problem and enhance the performance.

Another important contribution of the thesis is a generic web-page summarization system based on multi-word terms. The model applies machine translation techniques to generate an abstractive summary rather than an extractive one. The MWTs we select from the human-edited summaries of the training data are not only suitable but also readable for web page summarization. We carry out experiments by using both Open Directory Project (ODP) and Document Understanding Conferences (DUC). The result is encouraging and promising. It outperforms the similar summaries at word level while provides better readability.

1.4 Overview

The remaining thesis is organized into five chapters. Chapter 2 surveys the fields of automatic multi-word term extraction and web-page summarization. Methods and concepts relevant to this thesis are provided as background. We focus on the research with statistical approaches since they are language independent and more related to our work.

Chapter 3 presents our method for multi-word term extraction in detail. New statistical association measures are proposed using a smoothing method and a normalization approach. In addition, data preparation and post-processing filters are discussed in order to improve our results for multi-word term extraction.

Chapter 4 describes how our methods for multi-word term extraction are applied to web-page summarization. A general framework is introduced, followed by the detailed steps of content selection, machine translation, and summary generation for web-page summarization.

Chapter 5 discusses our experimental process and the evaluation measures. The data set is from the Open Directory Project and the MWTs are extracted from both human-edited summaries and the related web pages. The experimental results are compared and analyzed to gain further insights.

Chapter 6 concludes the thesis and describes some directions for future work.

Chapter 2

Multi-word Term Extraction and Text Summarization

2.1 Multi-word Terms

2.1.1 What is a Multi-word Term?

A multi-word term (MWT) is an expression consisting of more than one word with a grammatical structure and a specific meaning. A MWT can be a noun phrase such as “hot dog” and “president of America”, a fixed collocation or idiom such as “blue moon” and “orange and apple”, compound verbs such as “take into account”, compound prepositions such as “in order to”, compound conjunctions such as “on the contrary”, and compound determines such as “a piece of”, and so on. A multi-word term should satisfy the following four properties.

MWTs are cohesive

A multi-word term is a cohesive lexical unit which functions as a single concept or action. For example, “White House spokesman” or “to make a decision” expresses a concept or action in the text. Thus, the words in a MWT are closely connected and have some kind of glue between them. The presence of one or several words of a MWT often implies what word will appear next. For example, when “Kentucky Fried” appears, there is a high chance that the next word will be “Chicken”.

MWTs are language dependent

Non-native speakers often have a hard time to translate a MWT even though they understand the meaning of each single word in it such as “hot dog”. In a different language, the same concept may be represented by a different structure and/or different words, making word-to-word translation invalid. Terms like “hot dog” also indicate that the meaning for the sum is more than the combination of the meanings for the individual words and an accurate translation has to be done at a multi-word term level.

MWTs are domain dependent

A domain refers to a specific area of human endeavor or activity which requires specialized knowledge, such as computational linguistics and medicine. In each domain, numerous MWTs are created to stand for certain concepts, which seldom appear in general writing and are often unintelligible for the laymen. For instance, “hypertension relieving pill” refers to a drug in the medical domain and can be confusing for some people. Furthermore, familiar words can have different meanings in different domains. For example, in the domain of business, “red car” may stand for a taxi company; while in a children book, it may mean a car whose color is red.

MWTs are recurrent

Since MWTs refer to concepts or actions, they need to appear repeatedly in text in order to be recognized in a language and a domain. For example, “hot dog” is a term since it is used frequently in daily life.

2.1.2 Motivation for Extracting MWTs Automatically

In a sentence, a MWT is often the primary entity and by identifying it, the key information for the sentence can be captured. For instance, in an article about real estate, just by looking at words like “profit”, “property”, and “earn”, it will be hard to get the idea of what the article is about. However, with MWTs like “real estate” and “investment service”, we can know more about the topic and the context of the article.

Generally, MWTs provide more specific information than single words. Still taking an article about real estate, words like “real” and “estate” may provide a rough idea about the context, but there is a lot more to be desired. MWTs such as “real estate career” and “Los Angeles area” can offer more detailed information. An article about real estate career around the Los Angeles area can provide even more specific information than what we can get from single words.

MWTs are also valuable for many language processing tasks such as text summarization and document classification. However, it is nearly impossible to extract MWTs manually due to the high cost and the frequent introduction of new MWTs. In addition, different domains and regions may use different MWTs. Therefore, extracting MWTs automatically is not only necessary but also useful for natural language processing.

2.2 Automatic Multi-word Term Extraction

2.2.1 What MWTs to Extract?

Automatic multi-word term extraction is the task to recognize MWTs from text. However, it is neither practical nor necessary to identify all MWTs. Before discussing the technical challenges, we first clarify what MWTs we aim to extract.

A MWT can be of any length. For example, “real estate” contains two words; “real estate service”, three words; and “department of emergency medical services education”, six words. Early researchers (Bourigault, 1992) tried to find the maximal-length MWTs. However, longer is not always better. Although short MWTs may lack of detailed information, long MWTs can be too specific to be widely used. In practice, the commonly used MWTs are usually between two and six words.

Furthermore, a MWT can be interrupted or uninterrupted. A MWT is uninterrupted if each position of the term is occupied by only one possible word, such as “human rights”. On the contrary, a MWT is interrupted if we can have more than one possible word in a particular position, such as “them” in “take them into account”. Uninterrupted terms are easier to recognize and are more useful than interrupted ones, since they tend to represent an intact concept or piece of information. Interrupted multi-word terms often contain redundant and noisy words, making them less useful for many applications.

In this thesis, we aim to extract MWTs that are uninterrupted and contain two to six words. Such terms are informative and widely used and thus may help produce useful solutions for text summarization.

2.2.2 Challenges

Humans can recognize MWTs easily based on world knowledge and context information, but manual extraction is too expensive and impossible for a large corpus. To automatically extract MWTs, we need to decide how words are related to each other, where a MWT starts and ends, and whether a sequence of words stands for a specific meaning than those of the individual words. For example, “red car” is the name of a taxi company around the Guelph area, so it should be treated as a MWT.

As will be explained in section 2.3 on related work, current research on multi-word term extraction is mostly based on linguistic rules and/or statistical measures. For example, MWTs tend to appear in some grammatical patterns such as noun-noun, adjective-noun, and so on. We can learn these patterns from data and use them to find MWTs. Besides, the words in a MWT tend to occur together in a fixed order repeatedly, making it possible to find MWTs based on statistical measures like co-occurrences.

However, the real characteristics of MWTs are never as simple as an algorithmic interpretation. MWTs can occur at any position in a sentence with many grammatical combinations. Such characteristics of MWTs make these rules not as effective as expected. It is true that most MWTs follow some grammatical patterns such as noun-noun and adjective-noun, but there are always exceptions. MWTs can also be other combinations of words such as verb-noun, preposition-noun, noun-noun-verb, and many others. As we add more grammatical patterns to capture these MWTs, more meaningless word sequences may inevitably be recognized as MWTs as well.

In a like manner, statistical measures may also suffer from their own problems since some MWTs may appear hundred times in a document while others only appear once or twice. Take the MWT “hot dog” as an example. The two words together clearly stand for a kind of food. When the word “hot” appears in text, there is a good chance that the word “dog” will follow it. Likewise, if the word “dog” occurs, the probability of the word “hot” appearing before it is high as well. We say that the two words co-occur frequently. Nevertheless, it is also possible that there are other words appear after “hot” or before “dog”, like “hot weather” and “running dog”, except that the two words do not often appear together. The low frequency of “hot” and “dog” co-occurring in some domains of text may prevent us from recognizing “hot dog” as a MWT in such situations.

2.3 Related Work on Multi-words Term Extraction

The study on automatic multi-word term extraction can be traced back to early 1960’s. At that time, the computing power is quite limited in both time and space; so researchers focused on relatively simple and crude methods to identify the representative terms (MWTs or single words) from a corpus, which are mostly used to build an index for an information retrieval system. Since early 1990’s, along with the rapid development of the World Wide Web, a huge amount of electronic documents become available, making the automatic multi-word term extraction both important and urgent. In addition, the computing power has been multiplied hundreds of times, allowing researchers to apply more complex and expensive algorithms to find better solutions. A plenty of efforts has been dedicated to the task of automatic multi-word term extraction in the past decades, and we will review the related work in the next several subsections.

2.3.1 Linguistic Approaches

In each language, MWTs are constructed based on linguistic rules of some syntactic and morphological structures. In English, for example, MWTs such as 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 MWTs easily since we have the knowledge about how the MWTs are composed.

Although most linguistic approaches tend to recognize MWTs according to their syntactic and morphological structures, there are also other approaches that try to filter terms by context analysis. Work by (Bourigault, 1992) uses partial grammatical analysis to identify noun phrases and introduces *LEXTER:* an early multi-word 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 "hot" and "dog" are put together, a grammatical pattern of adjective-noun is formed. Some patterns are "negative" since they are never used for MWTs, while others are "positive" as they are often used for MWTs. The analysis step uses "negative" patterns as important clues to isolate the maximal-length noun phrases from text. The parsing step uses "positive" patterns to obtain the likely maximal-length noun phrases from the maximal-length ones. The author argues that partial grammatical analysis is advantageous over complete syntactic analysis. It focuses on the grammatical categories of words and the grammatical structures of word sequences rather than the actual position of words in the sentence, making the analysis more efficient and accurate.

Later work by (Dagan *et al*. 1993) introduces Termight, a system helping recognize MWTs and their translations. In Termight, MWTs are identified by syntactic patterns. First a target document is parsed by a Part-Of-Speech (POS) tagging system, which associates each word with its corresponding lexical category such as noun, verb, adjective, and so on. Second, candidate MWTs are found according to a set of syntactic patterns defined by regular expressions. Third, all candidates are grouped by head words and then sorted by their frequencies. Fourth, all terms in a group are sorted alphabetically in the reverse order. The final MWTs are selected from the sorted groups by the associated concordance lines, which indicate how well a word is associated with other words. Thus the work can identify group-related MWTs.

Sophia Ananiadou (Ananiadou, 1994) proposes a system that extracts MWTs with the help of their morphological structures. Unlike grammatical analysis which annotates words with their grammatical categories (e.g., noun, verb, adjective), morphological analysis classifies the structure of words into four categories: Words, Affixies, Roots, and Combs. Each word can be morphologically represented by a combination of these four categories such as Affix+Word and Affix+Affix+Word. The work also distinguishes four levels: Non-native Compounding, Class I affixation, Class II affixation and Native Compounding. For example, the word “glorious” is represented as: “glory” ((category noun)(level 1)) and “ous” ((category suffix)(level 1)). When analyzing a sequence of words, the morphological structure of the words is marked by categories and levels. After that, a top-level filter is employed to determine if the words form a potential MWT by matching its morphological structure.

Christian Jacquemin (Jacquemin, 1999) offers a two-tier framework for multi-word term extraction which is composed of a paradigmatic level and a syntagmatic level. The paradigmatic level examines how terms are composed by lexical items such as words, while the syntagmatic level determines the syntactical structures of the terms. Figure 2.1 is an example taken from this work, where the term “speed measurement” can be represented as:

$$\begin{cases} Paradigm \quad : \left\{ \begin{array}{l} < N_1 lemma \; >= \; measuremen \quad t \\ < N_2 lemma \; >= \; speed \end{array} \right\} \\ Syntagm \quad : \{ N_0 \to N_2 N_1 \} \end{cases}$$

Figure 2. 1: Syntagmatic Relationships between Words

These two levels reflect the inner relationships between multi-word term variations in three linguistic dimensions (morphological, syntactic and semantic). Similar terms can also be found by considering the semantic information as available in WordNet (Miller *et al*, 1990), such as $N(speed) = \{fast_N, swift_N, rapid_N ...\}$. Thus, unknown MWTs can be recognized by matching against similar patterns with known terms in this approach.

2.3.2 Statistical Approaches

Statistical approaches aim to extract MWTs from text corpora by means of association measures (Church and Hanks, 1989; Shimohata, 1997; Witten *et al*. 1999). For example, the term “hot dog” often occurs repeatedly in text, indicating that there is some kind of “glue” or cohesiveness between the words. Statistical approaches apply statistical techniques to determine the degree of cohesiveness between the constituents of possible MWTs. Compared with linguistic approaches, statistical approaches are more popular since they are flexible and often domain/language independent. However, they usually require empirical thresholds to optimize the performance of selected MWTs.

Early in the 1960s, researchers started to employ statistical approaches, mainly focused on selecting significant single words for automatic indexing. (Edmundson and Wyllys, 1961) uses the frequency ratio to select significant terms, which is the ratio of the frequency of a word in a particular document *f* with its relative frequency *r* in general use. From 1970s to late 1980s, follow-up work studied various distributions to describe the co-occurrences of words. Researchers found that it is necessary to take into account the tendency of terms in cluster by modeling the co-occurrences of words with probability distributions. However, early research is mostly for theoretical interest since online documents are limited and the computing systems are not powerful enough at that time.

Since late 1980’s and early 1990’s, along with the advance of computing technologies, recognizing MWTs with statistical approaches becomes practical and popular. Co-occurrences of words are applied to large document corpora and more complex probability models are proposed. The work (Church and Hanks, 1989) introduces a 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:

$$I(x,y)=\log_2\frac{P(x,y)}{P(x)\cdot P(y)} \qquad (2.1)$$

Where *P(x)* and *P(y)* stand for the probability of *x* and *y* respectively, and *P(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 association ratio has the same definition as the mutual information, but it is different in two aspects. 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 window of *w* words; so the length of the window will affect *f(x, y)*. For example, in sentence “Each word is tagged with its grammatical category with the help of POS tagging”, *f(tagged, with)* = 2 instead of 1. Nevertheless, the association ratio can measure the relationship between two words, and is used in later work (Daille, 1995; Dagan *et al.* 1993). Note that the association ratio is unstable when the count is small, and as a result, it is mostly used to extract bigram terms.

Frank Smadja (Smadja, 1993) introduces a multi-word term extraction tool called Xtract, which is composed of three stages. Stage one performs statistical analysis on sentences to get the bigrams which have close lexical relationships. If two words frequently co-occur in a single sentence and there are fewer than five words between them, the word pair is taken as lexically related. Part-Of–Speech (POS) tagging is employed to help determine the possible combinations of the two words. For example, verb-verb is a poor combination and will be filtered out. Stage two generates maximal-length MWTs. First, sentences that contain a word pair are indentified and the frequency of each word around *w* (the first of the word pair) along with its relative distance from *w* is recorded. The words with relatively high probabilities in a fixed position around *w* are kept to form MWTs. Stage three is an enhancement step. Syntactic information of all possible MWTs is added to form a syntactic label which indicates its syntactic structure such as “verb-object” and “noun-object”. If a possible MWT has a stable label, it is then taken as a MWT. Although this work can generate MWTs longer than two words, many thresholds need to be optimized and if not done properly, some desirable MWTs can easily be filtered out.

J.F.Silva and his colleagues (Silva *et al.* 1999) introduce the LocalMaxs algorithm which assumes that MWTs have strong glues 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:

$$SCP = p(x \mid y) \cdot p(y \mid x) = \frac{p(x,y)^2}{p(x) \cdot p(y)} \qquad (2.2)$$

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 generalize this measure for n-grams, the authors introduce the Fair Dispersion Normalization which breaks an n-gram $w_1 w_2 ... w_n$ at different dispersion points and considers it as combinations of the two parts. For example, an n-gram $w_1 w_2 ... w_n$ can be broken into a bigram $w_1 w_2$ and a (n-2)-gram $w_3 w_4 ... w_n$ 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.

$$Avp = \frac{1}{n-1} \sum_{i=1}^{n-1} p(w_1 ... w_i) \cdot p(w_{i+1} ... w_n) \qquad (2.3)$$

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:

$$SCP\_f(w_1 w_2 ... w_n) = \frac{p(w_1 w_2 ... w_n)^2}{Avp} \qquad (2.4)$$

Based 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 MWTs.

Years later, the same group of authors (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 “glue” within a MWT. However, the SCP is calculated from the frequencies of the terms instead of their probabilities. Given a document 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$\approx$N-n+1. Thus

$$p(ngram) = \frac{freq(ngram)}{N - n + 1} \approx \frac{freq(ngram)}{N} \qquad (2.5)$$

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

$$Scp(x,y) = \frac{p(x,y)^2}{p(x) \cdot p(y)} = \frac{\left(\frac{freq(x,y)}{N}\right)^2}{\frac{freq(x)}{N} \cdot \frac{freq(y)}{N}} = \frac{freq(x,y)^2}{freq(x) \cdot freq(y)} \qquad (2.6)$$

In addition, the author utilizes a suffix array and the related structure to store the n-grams and their information associated such as frequencies, positions and lengths. The author applies the algorithm to the task of extracting Portuguese MWTs and shows that the smoothed Localmaxs algorithm can be carried out efficiently.

2.3.3 Hybrid Approaches

Linguistic approaches study the syntactic structures while statistical approaches focus on the recurrent characteristics of MWTs. Both have their advantages and limitations. Hybrid approaches try to combine linguistic and statistical techniques to extract MWTs. Linguistic approaches can be applied first to obtain multi-word term candidates, and then statistical approaches are used to select better candidates, or vice versa (Justeson and

Katz, 1995; Daille, 1995; Frantzi *et al*. 2000).

Gaël Dias and his colleagues (Dias *et al*. 2000) compare hybrid approaches with pure statistical approaches and points out that due to the separation of linguistic and statistical approaches, hybrid approaches usually do not bring substantial improvements. Moreover, hybrid approaches tend to filter out some MWTs like compound nouns. The experiments show that hybrid approaches only have a slight advantage over pure statistical approaches when extracting short MWTs such as bigrams. For long MWTs that contain more than two words, the two approaches have similar performance.

2.3.4 Graph-Based Approaches

Recent work starts to use web dictionary (Wikipedia) and graph-based techniques to identify MWTs (Mihalcea and Csomai, 2007; Grineva *et al*. 2009). Wikipedia is a free online encyclopedia that edited and maintained by volunteers all over the world. In the past 10 years, it has rapidly grown into the world's largest encyclopedia. Searchers find that Wikipedia sorts articles into hierarchical categories while provides cross-references between articles, which makes it a pretty resource to study term relationships.

For example, (Mihalcea and Csomai, 2007) proposes to extract MWTs by taking advantage of links and titles within Wikipedia articles. These titles and the texts associated with the links contain well-defined terms (single words or MWTs) and can be treated as a controlled vocabulary. Based on this controlled vocabulary, the author introduces Keyphraseness, which is a rank method to measure how likely a term candidate should be selected as a term. The Keyphraseness is defined as:

$$P_{keyphrase}(W) \approx \frac{count\left(D_W^{keyprhase}\right)}{count(D_W)} \qquad (2.8)$$

where $P_{keyphrase}(W)$ is the probability that word sequence *W* is selected as a key phrase in a new document; $count\left(D_W^{keyprhase}\right)$ is the number of the documents in which *W* is already selected as a key phrase; and $count(D_W)$ is the number of documents in which *W* appears. Given a document, all word sequences appear in the controlled vocabulary are taken as term candidates, and the Keyphraseness of each candidate is calculated. The top N candidates are then selected as terms. The author compares the Keyphraseness method with *TF×IDF* (Witten *et al*, 1999), and finds that the Keyphraseness method gives better performance.

2.4 Web-Page Summarization

With the fast development of the Internet and the World Wide Web, there are vast amounts of web contents available to the users for easy access, but at the same time, it creates the information explosion, making it difficult for the user to find and digest the relevant information they need.

The “inefficiency” problem for information access exists in two aspects. The first is about the search process. Finding the right web contents which contain the very information needed by the users in the huge database of World Wide Web is a complicated and time consuming task. Fortunately, search engines such as Google have provided solutions to this problem by just providing the web pages that are relevant to a user’s queries. Although the current searching techniques still have rooms to improve, users are more or less satisfactory with their efficiency and accuracy.

The second is about the browsing process, where a user interacts with a search engine. Current systems put a heavy burden on the user in providing the right query and extracting the answers from the search results. The user often needs to take a long time to examine the returned pages and if the results are not relevant, the user has to formulate another query and repeat the process. In the early days, a search engine simply provides a short description for a web page based on the first few lines in the given web page. However, such short descriptions are often off the topic and do not reflect what is in a page or a specific portion of a page.

In order to alleviate the user from the heavy burden of search and browsing in the age of information overload, we need a better way to provide a brief and meaningful summary for a web page so that the user can quickly decide whether the page is relevant or not. Web-page summarization is the task to identify the key ideas of a web-page and generate a summary for it. By providing carefully extracted summaries or gists, the user can get an overview of the related web-pages before deciding whether to read them in details. This can not only help the user access the web-pages more efficiently but also allow the user to access the search results via small-display devices such as cell phones. Furthermore, it can also help the user browse a hierarchical organization for web-pages such as Yahoo’s topic hierarchy in order to find the information he/she wants.

2.4.1 Challenges for Web-Page Summarization

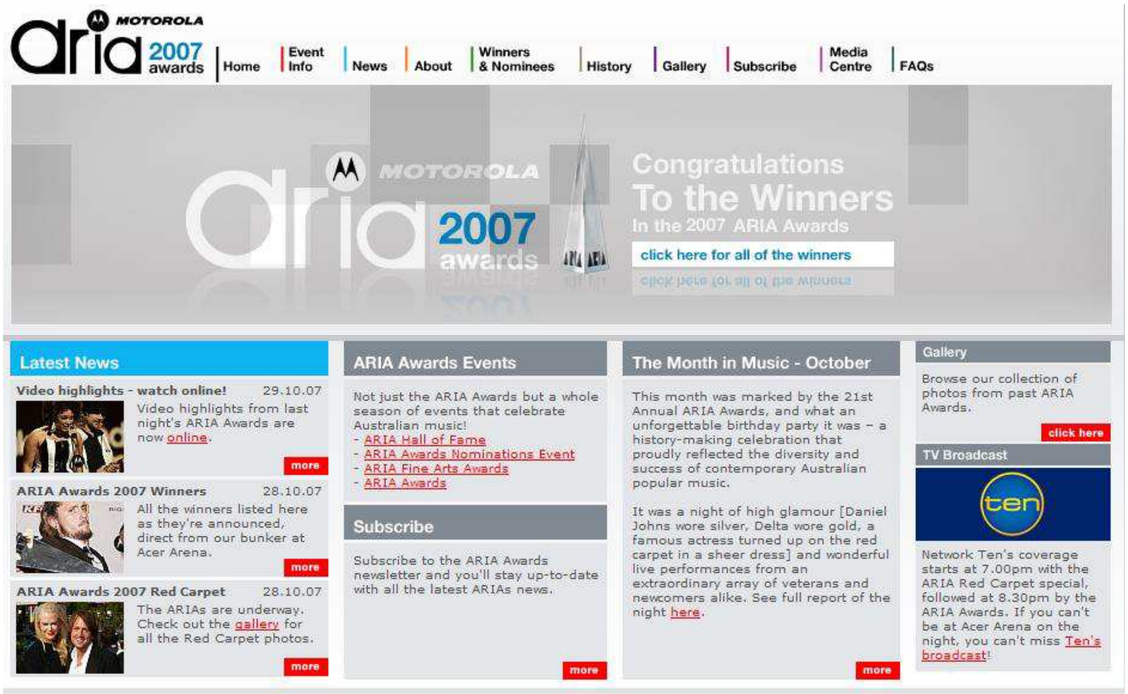
Regular documents are usually well structured with cohesive and coherent contents. Each document typically has one or more central topics, and there are paragraphs grouped by the related topics. Each paragraph also contains a topic sentence while the other sentences provide supporting details.

Web-pages, on the other hand, are published by all kinds of individuals, companies and organizations in the world, and as a result, the quality of the structure and the content can be quite varied. In addition, some web-pages may contain multilingual contents, making it difficult for automatic text summarization.

Moreover, current techniques allow us to use multimedia: videos, audios, pictures, and other kinds of dynamic contents that can all appear in a web-page along with textual information. There are also a lot of auxiliary fragments in a web-page such as navigation bars to make it more artistic or easier to use. These multimedia and auxiliary contents can appear anywhere in a web-page, often partitioning it into different frames or areas and making the structure appear to be random and scattered.

Furthermore, the contents of web-pages are generally diverse. There are advertisements and hyperlinks with anchor text that are usually irrelevant. There are also short paragraphs that describe multiple topics collectively. In short, there is a mass of distractive information in web-pages and their central ideas are not always obvious and prevailing in the text.

Figure 2.2 shows a typical web page. We can see that the page contains pictures, hyperlinks, navigation bars, and textual information. The textual information includes titles, links, and texts, and the texts are not cohesive and coherent, since they are divided into several sections and each section has its own idea which is irrelevant to others. Such a structure makes it challenging to automatic web-page summarization, but the need becomes increasingly important and urgent.



Latest News

Video highlights - watch online! 29.10.07
Video highlights from last night's ARIA Awards are now online.

ARIA Awards 2007 Winners 28.10.07
All the winners listed here as they're announced, direct from our bunker at Acer Arena.

ARIA Awards 2007 Red Carpet 28.10.07
The ARIAs are underway. Check out the gallery for all the Red Carpet photos.

ARIA Awards Events

Not just the ARIA Awards but a whole season of events that celebrate Australian music!
- ARIA Hall of Fame
- ARIA Awards Nominations Event
- ARIA Fine Arts Awards
- ARIA Awards

Subscribe

Subscribe to the ARIA Awards newsletter and you'll stay up-to-date with all the latest ARIAs news.

The Month in Music - October

This month was marked by the 21st Annual ARIA Awards, and what an unforgettable birthday party it was – a history-making celebration that proudly reflected the diversity and success of contemporary Australian popular music.

It was a night of high glamour [Daniel Johns wore silver, Delta wore gold, a famous actress turned up on the red carpet in a sheer dress] and wonderful live performances from an extraordinary array of veterans and newcomers alike. See full report of the night here.

Gallery

Browse our collection of photos from past ARIA Awards.

TV Broadcast

Network Ten's coverage starts at 7.00pm with the ARIA Red Carpet special, followed at 8.30pm by the ARIA Awards. If you can't be at Acer Arena on the night, you can't miss Ten's broadcast!

Figure 2. 2: A Typical Web-page

2.4.2 Automatic Text Summarization

Text summarization aims to create short summaries for textual data such as documents, news articles and reports. The earliest work on text summarization is dated in 1950's

(Baxendale, 1958; Luhn, 1958). Over the past sixty years, text summarization has been actively studied and many techniques have been proposed to generate summaries of different forms.

Text summarization can be distinguished into single document summarization and multi-document summarization. Single document summarization focuses on generating a summary for just one document and the main challenge is to distinguish the more important parts of the document from the less important ones. Multi-document summarization is more complicated in that it aims to create a short summary for multiple documents. These documents can have information that overlaps, supplements or contradicts with each other. So it is important to recognize disparate and similar topics, and make a balanced and coherent summary for the related documents.

Methodologies are quite diverse for text summarization and can be divided into three categories: extraction, abstraction, and compression. Extractive summarization (Abracos and Lopes, 1997; Lin, 1999) focuses on selecting the original paragraphs, sentences, or words from text and uses them to form a summary. Abstractive summarization (Zechner, 1996) employs language generation techniques to create a more coherent summary that may use words that do not appear in the original text. Compress summarization (Radev *et al*. 2002) aims to filter out unimportant and irrelevant words from the extracted paragraphs or sentences.

Furthermore, text summarization can also be divided into generic, query-based, or domain specific. Generic summarization aims to cover as much information of the text as possible and simplify the content to form summaries. Query-based approaches generate summaries based on the parts of text that are related to the queries. Domain-specific summarization focuses on generating summaries about a specific domain.

There are many techniques available for text summarization. For single document summarization, machine learning techniques are often employed to extract key contents from a target document. For example, Naïve-Bayes method (Kupiec et al, 1995; Aone *et al*. 1999) and decision trees (Lin, 1999) rank the sentences according to the combinations of many features such as sentence positions, sentence lengths, cue phrases, TF×IDF weights, and so on. The top N sentences are then extracted to form the summaries. Hidden Markov Model (Conroy and O'leary, 2001) treats the sentence selection as a sequence problem. It acquires the transition probabilities of the sentences from a training set, and then uses them to identify sentences that can be in a summary. Other rank algorithms (Svore *et al*, 2007; Burges *et al*, 2005) consider additional features such as n-gram positions and their frequencies and use third party databases, e.g., Wikipedia and WordNet to select sentences or words to form summaries.

For multi-document summarization, solutions such as (Radev and McKeown, 1998; Radev et al, 2004) often take two steps. The first is called “fusion or merging”. As there are multiple topics in a set of documents, we need to identify similar and dissimilar topics before merging or fusing them. The second is called “representing”, which generates appropriate textual units (sentences or words) to represent these topics in a summary.

Query-based summarization is an active topic in multi-document summarization. Jaime G Carbonell and Jade Goldstein (Carbonell and Goldstein, 1998) introduce the maximal marginal relevance measure to balance the coverage and reduce the redundancy in a summary. Inderjeet Mani and Eric Bloedorn (Mani and Bloedorn, 1997) propose a graph-based method. A query is taken as an entry node to find the other related nodes and once the graph is established, words are selected based on their weights in the corresponding modes of the graph.

Although these diversified approaches can be applied to web-page summarization, not all are suitable due to its unique challenges. First, the techniques for long and single documents are not appropriate for web-pages. Such techniques are usually paragraph based, which use a paragraph to address a single topic in a summary. Obviously web-pages are not structured in this way where paragraphs are always available and can be extracted. In fact, web-pages are more like multi-document which usually contain several topics with short descriptions.

Second, the techniques that rely on natural language cues and discourse models are not appropriate either. They identify topic sentences or terms based on features such as linguistic annotation, order of words, list of predefined phrases and lexical choices in the text. Such techniques can be too domain specific to be applied to web-pages. In addition, they usually need adequate contexts to perform such linguistic analysis. For web-pages, however, the textual information can be short with just few sentences, making such analysis impossible.

This leaves us with techniques based on the repetitions of terms and phrases. Summaries made of terms and phrases will not be coherent, but they preserve as much information as possible from original text. Researchers have extended such techniques for summarizing informal text such as e-mails (Carenini, 1997). They are also useful and suitable for summarizing web-pages due to their unique challenges and characteristics.

In this thesis, we explore abstractive approaches to generate generic summaries for web-pages. In particular, we try to find effective ways of extracting MWTs and use them for web-page summarization.

2.4.3 Related Work for Web-Page Summarization

Although the techniques described in the previous subsection can be applied to web-page summarization either through extraction (Strzalkowski *et al*. 1998; Lin, 1999) or abstraction (Berger and Mittal, 2000; Boguraev *et al*. 1998), web-page summarization has its own characteristics and challenges.

Some approaches take advantage of the extra information in web-pages such as hyperlinks and their descriptions for web-page summarization. We call them context-based approaches. (Glover *et al*. 2002; Amitay and Paris, 2000; Delort *et al*. 2003) use the hypertext structures of web-pages and exploit anchor text found in the links to a given web-page, which usually contains short descriptions about the web page. (Sun et al, 2005) extracts additional information from the click-through data of a web search engine to improve web-page summarization.

In the work (Amitay and Paris, 2000), the authors survey the techniques in automatic text summarization and introduce the InCommonSence system. The system relies on the paragraph conventions found in web pages to generate summaries for them. Web-pages are often linked to other web-pages and resources such as annotations, notes, and descriptions in the Internet. The InCommonSence system detects such links within web-pages and uses them to summarize the web-pages. Figure 2.3 is taken from the work (Amitay and Paris, 2000) and illustrates how the system works. The anchors appearing in the graph stand for the links within the paragraphs.

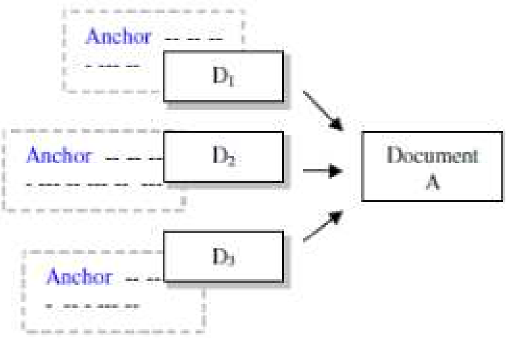
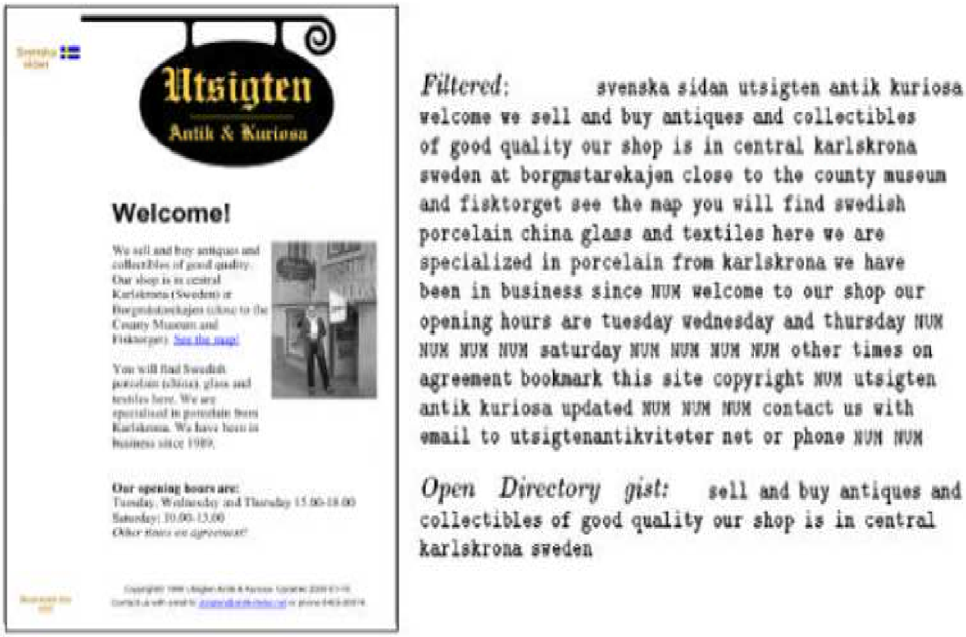


Figure 2. 3: Work Flow in InCommoneSense (Amitay and Paris, 2000)

As can be seen from Figure 2.3, there can be more than one description for a web-page. The system develops a ranking method to choose the best online description for the web-page. More than 60 features such as lengths, punctuations, verbs, positions of verbs, and many others are employed to distinguish descriptions into 5 levels (5 is the best and 1 is the worst) based on empirical experiments. The description with the highest rank is selected to represent the web-page.

The solution from (Sun *et al*, 2005) is similar to (Amitay and Paris 2000), but instead of using descriptions from hypertext links, it employs click-through data to help create summaries. The click-through data are mainly obtained from the search queries submitted by the users. The experiments show that these queries contain high quality terms to form summaries for web-pages. The work proposes two different methods to calculate the weights of the terms in the click-through data in order to select ones that better describe a web-page: adapted significant word method and adapted latent semantic analysis method. The first method selects words based on their frequencies in web-pages and queries respectively. The second method is an improvement of the first, and it picks up words based on their frequencies in sentences and queries, respectively, instead of web-pages. The work shows that method two has advantages over method one since it can capture low frequency terms in web-pages. (Sun *et al*, 2005) also presents a method for web-pages which are not covered by click-through data. The method relies on the hierarchical structure of OPD to generate summaries for web-pages.

Adam L. Berger and Vibhu O. Mittal (Berger and Mittal, 2000) propose OCELOT system for web-page summarization. The author points out that for some web-pages it is impossible to generate coherent summaries and words become feasible information units to form summaries for web-pages. The system takes advantage of the data from Open Directory Project (OPD) and uses them to train the parameters of machine learning algorithms. The ODP data is a web directory where each web-page has a human edited summary. OCELOT system employs machine translation techniques to learn how the words in web-pages are related to the words in human edited summaries. By training on the web-page and summary pairs of OPD, the system can generate new words from the text of web-pages, creating abstractive summary rather than extractive summary. Given a web page, the system first performs a series of filtering steps such as removing all the links, pictures and punctuations, converting all letters to lowercases, and others. Then, it takes the words that have the top N frequencies in the remaining text of the web page, and generates an abstractive summary using the matched words in the summaries of the training documents. Figure 2.4 is taken from (Berger and Mittal, 2000) and shows a summary generated by OCELOT system along with its original web-page.



Utsigten

Antik & Kuriosa

Welcome!

We sell and buy antiques and collectibles of good quality. Our shop is in central Karlskrona (Sweden) at Borgmästarekajen (close to the County Museum and Fisktorget). See the map!

You will find Swedish porcelain (china), glass and textiles here. We are specialised in porcelain from Karlskrona. We have been in business since 1989.

Our opening hours are:
Tuesday, Wednesday and Thursday 15.00-18.00
Saturday: 10.00-13.00
Other times on agreement!

Filtered: svenska sidan utsigten antik kuriosa welcome we sell and buy antiques and collectibles of good quality our shop is in central karlskrona sweden at borgmstarekajen close to the county museum and fisktorget see the map you will find swedish porcelain china glass and textiles here we are specialized in porcelain from karlskrona we have been in business since NUM welcome to our shop our opening hours are tuesday wednesday and thursday NUM NUM NUM NUM saturday NUM NUM NUM NUM other times on agreement bookmark this site copyright NUM utsigten antik kuriosa updated NUM NUM NUM contact us with email to utsigtenantikviteter net or phone NUM NUM

Open Directory gist: sell and buy antiques and collectibles of good quality our shop is in central karlskrona sweden

Figure 2. 4: Output From OCELOT System

Chapter 3

Multi-word Term Extraction Methods

In this chapter, we describe our methods for multi-word term extraction along with three prototype extraction systems. In particular, we propose a new association measure and a smoothing method that works along with the LocalMaxs algorithm for multi-word term extraction. In addition, we include a simple filtering step that helps improve the performance of our methods for multi-word term extraction.

3.1 Statistical Association Measures

Researchers assume that words that form a multi-word term (MWT) should have relatively strong “glue” with each other since a MWT has the characteristic of being used repeatedly. Based on this assumption, statistical association measures are employed to calculate the “glue” values within n-grams for the purpose of extracting MWTs.

A simple and crude 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 “glue” within them since they often occur together. Generally speaking, 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 MWTs that have low frequencies in a corpus of reasonable size, as illustrated in Figure 3.1. The X axis is for the ranks of words based on their frequencies. The Y axis is for the frequencies of the corresponding words. As we can see, there are a small number of words at the top that have high frequencies, but there are a large number of words at the tail that have low frequencies. In particular, after rank 1000, most words only have a frequency of 1. This means that frequencies can only identify MWTs with high frequencies but fail on those with low frequencies. Furthermore, there are some irrelevant and meaningless n-grams that can have relatively high frequencies such as “in a” and “is that” which are not considered as MWTs.

In order to identify more MWTs with a reasonable precision, more sophisticated association measures have been developed such as mutual information (Daille. 1995; 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 “glue” within possible MWTs without using the frequency values directly. However, as discussed in Chapter 2, these association measures still fail to capture long MWTs with low frequencies.

In this chapter, we propose a new association measure that helps measure the “glue” within MWTs 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 MWTs effectively.

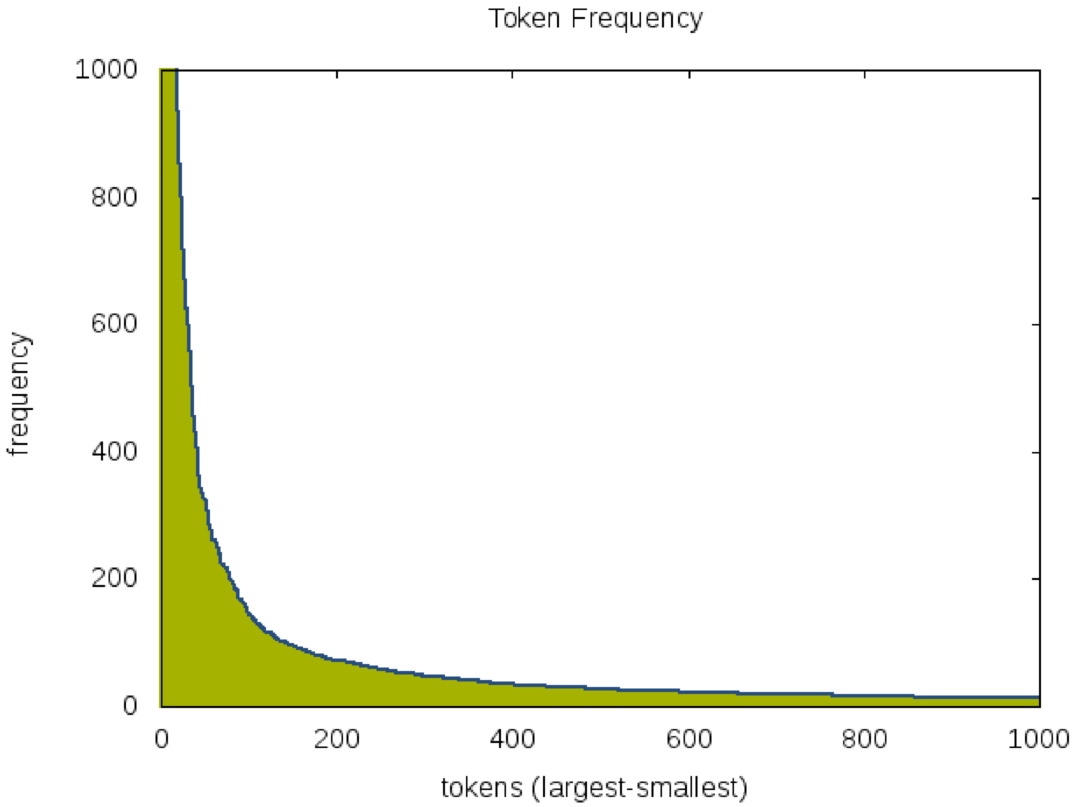


Figure 3. 1: Ranks of Tokens According to Frequencies

3.2 LocalMaxs Algorithm

The LocalMaxs algorithm is originally proposed by (Silva *et al*, 1999). It is language and domain independent and takes a text collection as input and produce MWTs as output. It selects MWTs from the n-grams based on two assumptions. First, the more cohesive a term is, the higher the association measure it should have. Second, MWTs are localized n-grams which have strong associations within. Before we describe the LocalMaxs algorithm, we need to introduce two concepts first, *antecedent* and *successor.*

- An *antecedent* of an *n*-gram $w_1 w_2 ... w_n$ is a sub-*n*-gram with size *n-1, either* $w_1 ... w_{n-1}$ or

$w_2 ... w_n$. We denote the set of all antecedents for an n-gram *W* as: $ant(W)$

• A *successor* of an *n*-gram $w_1 w_2 ... w_n$ 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)*

The LocalMaxs algorithm assumes that for an n-gram, if its association measure value is higher than or equal to the association measure values of all its antecedents and also higher than the association measure values of all its successors, then the n-gram is considered as a MWT:

$$\begin{array}{lll} g(W) \geq g(ant(W)) \wedge g(W) > g(succ(W)) & W\text{'s size} > 3 & (3.1) \\ g(W) > g(succ(W)) & W\text{'s size} = 2 & \end{array}$$

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

The LocalMaxs algorithm is flexible in that it allows various association measures to be used as long as they obey the first assumption (the more cohesive a term is, the higher the association measure it should have). Many experiments are performed with different association measures in (Silva *et al*. 1999; Dias *et al*. 1999; Dias *et al*. 2000; Aires *et al*. 2008).

Furthermore, the LocalMaxs algorithm can extract long MWTs such as compound nouns by comparison. For example, the MWT “House speaker Nancy Pelosi” is made up by two shorter ones “House speaker” and “Nancy Pelosi”. The LocalMaxs algorithm can extract such MWTs by comparing their association measure values. The association measure value for “House speaker” is high since it appears multiple times in a corpus. The association measure values for “House speaker Nancy” and “_ House speaker” are lower than that for “House speaker” in the corpus because there can be many words other than “Nancy” that appear before or after “House speaker”. Thus, “House speaker” should be chosen as a MWT while “House speaker Nancy” should not. However, when “Pelosi” is added after “House speaker Nancy”, the association measure value for “House speaker Nancy Pelosi” is higher than those for its antecedents “House speaker Nancy” and “speaker Nancy Pelosi” as well as its successors “_ House speaker Nancy Pelosi” and “House speaker Nancy Pelosi _”. Thus, “House speaker Nancy Pelosi” should also be chosen as a MWT. As illustrated in Figure 2.3, the peaks in the graph correspond to MWTs.

Instead of selecting the n-grams whose association measure values are locally maximal, the work (Aires *et al*. 2008) proposes the *Smoothed LocalMaxs* algorithm which requires the association measure value of a MWT 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 a MWT, we can describe the *Smoothed LocalMaxs* algorithm as follow:

$$\begin{aligned} &g(W) > \frac{\max(g(ant(W))) + \max(g(succ(W)))}{2} && W\text{'s size} > 3 \qquad (3.2)\\ &g(W) > \max(g(succ(W))) && W\text{'s size} = 2 \end{aligned}$$

The *Smoothed LocalMaxs* algorithm provides a global standard to decide if a *n*-gram is a

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

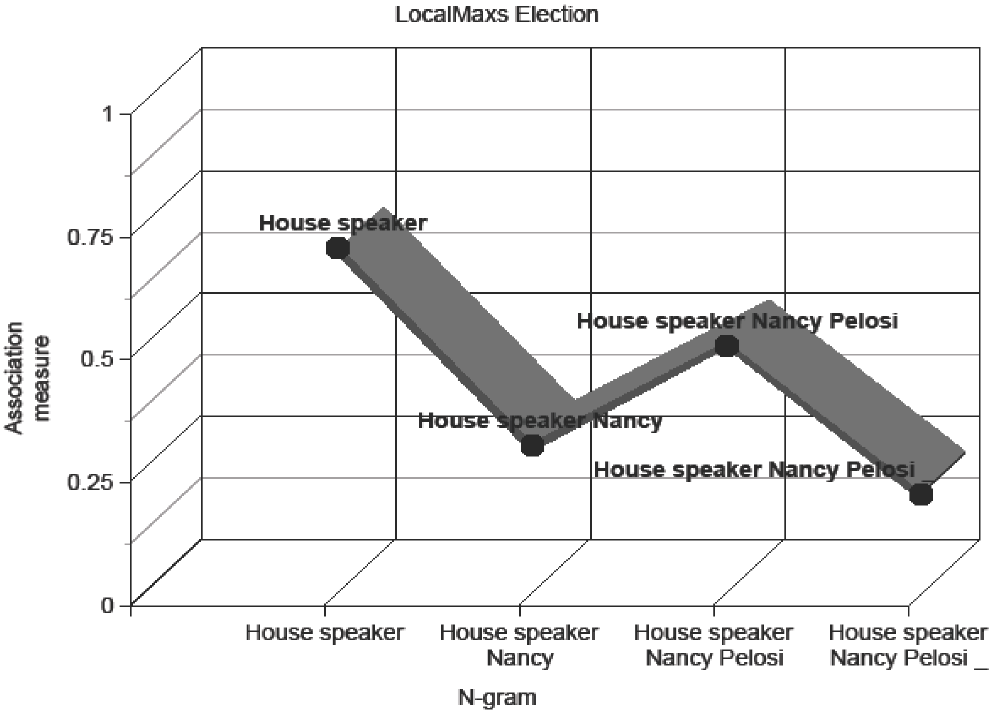


Figure 3. 2: Association Measures Fluctuating under an Ideal Situation

Although the LocalMaxs algorithm is powerful and useful, there are still rooms for improvement by developing better association measures. In a text collection, there are usually huge frequency gaps between n-grams. We find that as we get to trigram and four-gram levels, the frequencies observed are much smaller than those for the bigrams. While the LocalMaxs algorithm compares the association measure values of an n-gram with its antecedents/successors directly, the huge frequency gaps will cause the association measure value of a bigram to be far larger than those of its successor trigrams.

This problem can lead to poor performance on recall and fail to capture MWTs longer than two words. For example, “department of arts” is a MWT we may extract from a data set, but since the bigram “department of” is a popular word sequence and has a much higher frequency than that of “department of arts”, “department of” is automatically recognized as a MWT instead of “department of arts”. As illustrated in Figure 3.3, the dark line shows the ideal distribution of associate measure values for the n-grams within “House speaker Nancy Pelosi”, while the gray line show the real distribution of these associate measure values.

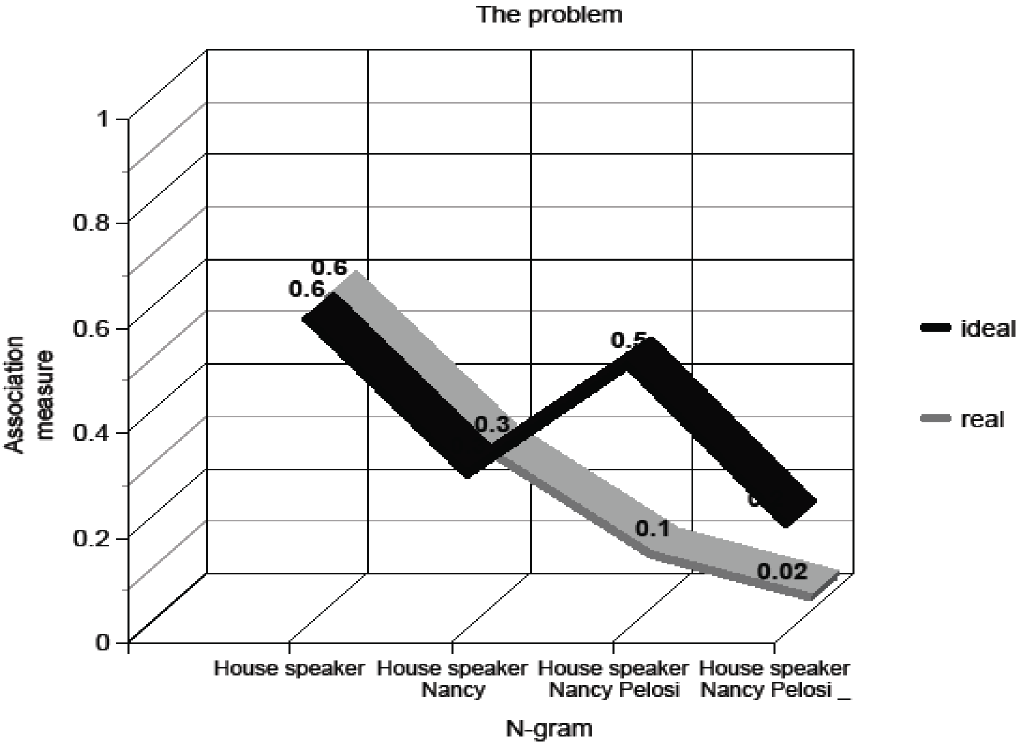


Figure 3. 3: Association Measures Fluctuating under the Actual Situation

In Chapter 5 on experiments, we will explore different ways to improve the statistical weights with the *LocalMaxs* algorithm in order to overcome the problem.

3.3 Smoothed Probabilities of N-Grams

Most association measures are based on the frequencies of n-grams. One problem is that there are usually a large number of missing n-grams in a corpus of reasonable size, controlled by the Zipf law. This is called the sparse data problem. The higher the order of the n-grams, the more missing n-grams we have in a corpus. For example, the bigram "department of" will certainly occur much more often than the four-gram "department of computing science". 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 ($20{,}000^3$) possible trigrams. In practice, however, the number of n-grams actually covered by a corpus is much smaller. The huge gaps between the possible and real bigrams and trigrams make the frequency 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/successors, such hug gaps render the algorithm not effective for extracting long MWTs.

Although we cannot do much about the frequency of an n-gram, we can improve the way we calculate its probability through smoothing techniques, thus making the association measures more comparable. Many association measures of an n-gram $w_1 w_2 ... w_n$ are based on the joint probability $p(w_1 w_2 ... w_n)$, which means the probability of words w_1, w_2 … and w_n appearing adjacent to each other in text. The joint probability is symmetric in that $P(AB) = P(BA)$, implying that the word order does not matter. However, a good association measure should reflect the "glue" 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:

$$p(w_1w_2...w_n) = p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1w_2)\ldots p(w_n \mid w_1w_2...w_{n-1}) \qquad (3.3)$$

where $p(w_n \mid w_1w_2...w_{n-1})$ is the conditional probability of word w_n occurring after the sequence $w_1w_2...w_{n-1}$ in text.

The conditional probability is not symmetric since it takes the order of words into account. Another advantage is that it is less dependent on large frequencies. Given a bigram w_1w_2, the conditional probability $p(w_2 \mid w_1)$ is defined as follows:

$$p(w_2 \mid w_1) = \frac{freq(w_1w_2)}{freq(w_1)} \qquad (3.4)$$

where $freq(w_1w_2)$ is the frequency of bigram w_1w_2 and $freq(w_1)$ is the frequency of word w_1. If both w_1w_2 and w_1 have low frequencies, the conditional probability $p(w_2 \mid w_1)$ can still be relatively high. On the other hand, if w_1w_2 is a high frequency bigram but an irrelevant term, the frequency of word w_1 should also be high, making the conditional probability $p(w_1 \mid w_2)$ relatively small. Thus, the conditional probability can adjust 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 highly desirable to extract useful MWTs.

Although conditional probabilities are less dependent on frequencies, the sparse data problem still exists. To get around the problem, we apply the shrinkage method to smooth high-order conditional probabilities. The shrinkage method is based on the assumption 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 should also occur more often. Since we need 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:

$$p(w_n \mid w_1 w_2 .. w_{n-1}) = (1-\lambda) p_n (w_n \mid w_1 w_2 .. w_{n-1}) + \lambda p_{n-1}(w_n \mid w_2 w_3 ... w_{n-1}) \qquad (3.5)$$

where $p_n (w_n \mid w_1 w_2 .. w_{n-1})$ 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 3.5, 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 $\lambda' s$ can be used for combining conditional probabilities of n-grams with different lengths. For example, we need 4 parameters to estimate the conditional probability for a 5-gram and we can name them $\lambda_2, \lambda_3, \lambda_4$ and λ_5. The subscript of λ indicates what n-gram the parameter is applied for: λ_2 is for bigram; λ_3 is for trigram; and so on. There is no λ_1 since the probability of a unigram is computed directly from its frequency. The optimal values for these parameters are not known in advance, but can be set intuitively or empirically estimated from a training data set. We assume that longer n-grams have more weights than its antecedents. Thus, λ_2 should be the smallest while λ_5 should be the largest. Moreover, most commonly used

MWTs are bigrams and trigrams, making λ_2 and λ_3 more important than the other parameters since they have greater influence on the performance. More details for estimating the values of these parameters will be discussed in Chapter 5.

3.4 Normalized Sequence Probabilities

Based on the smoothed probabilities of n-grams, we propose a new association measure for a sequence of words that calculates the "glue" 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 "glue" within the sequence is strong, the words composing the sequence tend to occur together in the given order. As a result, the joint probability, which is the probability of the words occurring together, should be high as well. More specifically, we define the normalized sequence probability of an n-gram as follows:

$$seq\_p(w_1 w_2 ... w_n) = \sqrt[n]{p(w_1 w_2 ... w_n)} \qquad (3.6)$$

Where we apply the chain rule and use the smoothed conditional probabilities to compute the joint probability for the n-gram. Furthermore, by taking the *nth* root to normalize the joint probability, the "glue" values of n-grams of different sizes can be compared directly and meaningfully.

3.5 Post-Processing Filters

As discussed earlier, stop-words are the main source of interference, and can seriously affect the precision of multi-word term extraction. For example, word combinations such as “of a” and “is the” are often selected as MWTs due to their high frequencies. We can remove such undesirable n-grams by applying a post-processing filter. In addition, we can remove certain auxiliary words from MWTs, such as “the” from “the hot dog”, making the terms more concise and meaningful.

We use a list of 675 common stop words for English downloaded from the Internet for stop word removal. If a stop word is found at the beginning or at the end of a MWT, it will be removed. If the term still contains more than two words after the stop-word removal, we will keep it as a MWT; otherwise, it will be dropped from further consideration.

3.6 Data Preparation

Some information contained in a text corpus is useless or even harmful for multi-word term extraction. As a result, we need to filter out such information through pre-processing. Our data preparation is based on regular expressions so that different kinds of tokens can be distinguished and selected. We distinguish the following kinds of tokens: words made of letters, numbers made of digits, URLs, e-mails, apostrophized words, hyphenated words, abbreviations, words connected by ampersands, whitespaces and newlines, and end-of-sentence marks (including “.” “,” “?” “!”). After the tokenization, the following steps are also performed:

- ✓ Remove all illegal and meaningless characters.
- ✓ Remove all non-textual information.
- ✓ Separate all hyphenated words if there are more than one “-“ mark in the tokens.

- ✓ Separate all words connected by “&” if there are more than one “&“ mark in the tokens.
- ✓ Remove suffixes to extract stems.
- ✓ Keep the original format of the text, including the spaces, line breaks and end-of-sentence marks.
- ✓ Convert all numbers with the symbol “NUM”
- ✓ Convert all e-mail addresses with the symbol “E-MAIL”
- ✓ Remove tokens that contain both letters and digits with lengths longer than 5 characters.

After removing all the “unwanted” information, we normalize all letters to lower cases and start to extract n-grams along with their frequencies. Although the most commonly used MWTs are between two and six words, we restrict ourselves to n-grams of up to 5 words due to the limited computing power of our machines. Nevertheless, we think that terms of up to 5 words should cover a majority of MWTs in a text corpus. We store all n-grams in the inverted files so that the association measures can be computed efficiently.

3.7 Multi-word Term Extraction System

Here we describe our prototype systems for extracting MWTs based on the new association measures and the LocalMaxs algorithm. We implement four versions of our system so that we can determine the effects of smoothed probabilities and normalized sequence probabilities through experiments in Chapter 5.

- ♦ Method based on Sequence Frequencies (SF)

The first version is the baseline for our system. It is a re-implementation of 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 document is large, the formula based on frequencies is equivalent to that based on probabilities. By using frequencies, we also simplify the calculations for extracting MWTs.

- ♦ Method based on Smoothed Probabilities (SP)

The second version replaces the joint probabilities in the calculations of SCPs with our smoothed conditional probabilities as discussed in section 3.3. We want to know if the smoothed method is helpful to stabilize the association measure and leads to a better performance.

- ♦ Method based on Normalized Sequence Frequencies (NSF)

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

- ♦ Method based on Normalized Smoothed Probabilities (NSP)

The fourth version combines the normalized sequence probabilities with the smoothed conditional probabilities of n-grams. We expect 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 versions share the same preprocessing steps. Similarly, all the results from these versions are further processed by our post-processing filters. In addition, all four versions will use the LocalMaxs algorithm in selecting final MWTs. These four versions should help us measure how our proposed approach performs for the multi-word term extraction. We expect that the version based on NSP will have the best performance which will be demonstrated by the experiments in Chapter 5.

Chapter 4

Application to Web-page Summarization

Multi-word term extraction techniques are useful in many research areas in Natural Language Processing. They serve as underlying tools for the applications that use multi-word terms (MWTs) for text representation. We have introduced many applications for multi-word term extraction techniques in Chapter 1. In this Chapter, we apply our multi-word term extraction methods to the task of generic web-page summarization.

For the task of generic web-page summarization, a summarizer needs to deal with web-pages of various forms and structures. In order to make the system general and effective for most web-pages (including pages with lots of graphics but few words), we often need to rely on small lexical units such as words, phrases, or sentences to form summaries. Some researchers simply utilize words or multi-word terms from the original web-pages to compose the summaries. However, such methods will not work well for web-pages that contain multiple topics. Some extraction methods try to create a gist for each topic regardless of duplications and inconsistencies, which can make the summaries less concise and meaningful. We believe that a more suitable solution should be able to abstract the content of a web-page while keeping it short and concise.

The work (Berger and Mittal, 2000) introduces the OCELOT system, which abstracts summaries based on words. This work inspires us to use MWTs rather than words so that we can make the summaries more readable and meaningful. Thus, our contribution here is to extend the OCELOT system and use MWTs for web page summarization.

4.1 System Overview

Our web-page summarization system is mainly composed by three steps: content selection, translation, and ordering. When a human editor generates a summary for a web page, he/she will not include every detail but try to identify the more important information from the less important, and focus on the parts of the content containing the key information. Thus, content selection should be the first step. After that, the editor will try to compose some words or word sequences (usually sentences) that are related to the key information in meaning but more succinct and concise (usually shorter than the selected context), which can be viewed as a “translation” process. Translation is usually between two different languages, but here we use the term to describe the mapping between different words or word sequences in a web-page and its summary. Finally, the editor will try to choose the optimal combination of words or word sequences to form a summary that is both meaningful and readable, which we define as ordering.

Our summarization system is built on a machine learning algorithm which is trained on a huge web-page collection along with the related human-edited summaries. It can generate new words or MWTs that may not appear in the original web-page and order them into a readable and meaningful summary. Thus, it is an abstractive method for web-page summarization. We take Open Directory Project (OPD), a human-edited web directory as our training data set for the machine learning algorithm. More details about the data from ODP will be introduced in Chapter 5 on experiments. We believe that the special structures of generic web-pages make MWTs more suitable for web-page summarization than words and sentences. Of course, we can still take single words into consideration when MWTs are not available to create summaries for some web pages.

4.2 Content Selection

The content selection task aims to locate words and MWTs that contain the important information about a web-page. A web-page can contain plenty and various contents, and it is impossible for a summary to cover the key information in it. Thus, we need to select the important information from the web-page and rely on it to form a summary. For this purpose, we need to start with a filter for web-pages. We only use the textual content since the other information such as pictures, videos and audios cannot be effectively summarized, nor do they need to be summarized. Furthermore, even the textual content may contain lots of invalid information that needs to be filtered out. For example, meaningless and illegal expressions are commonly found in web-pages, as well as misspellings. We apply a relatively strong filter to get rid of these unwanted words or word sequences as much as possible.

We first tokenize the textual content into tokens and employ the following list of rules in our filter.

- ✓ Remove all illegal and meaningless tokens. (e.g. words that contain over 20 characters)
- ✓ Perform stemming.
- ✓ Remove all stop words.

- ✓ Separate all hyphenated tokens if there are more than two “-“ marks in them.
- ✓ Separate all tokens connected by the “&” marks if there are more than two “&“ marks in them.
- ✓ Keep the original format of the textual information – including spaces, line breaks, and sentence terminators.
- ✓ Replace all number tokens with the symbol “NUM”
- ✓ Replace all e-mail address tokens with the symbol “E-MAIL”
- ✓ Remove tokens that contain both letters and digits and whose lengths are longer than 5 characters.

After removing all the unwanted content from the web-page, we break the remaining text into words and MWTs. We prepare a multi-word term dictionary which is learned from our training set (discussed in detail in the next section). We perform a greedy search to match the longest MWT in the text. Since the maximum MWTs we can recognize are 5-grams, we conduct a search by scanning the text in a window size of 5. We first look for the 5-gram from the window in our dictionary. If we find it in the dictionary, we take it as a MWT; otherwise, we remove the last word in the window and search the 4-gram in our dictionary to see if it is a MWT, and so on. If no n-grams (n>1) are recognized as MWTs according to the dictionary, then the first word is treated as a single word and the window is moved forward, starting from the next word. However, if there is an n-gram (1<n<6) recognized as a MWT, we will not look for shorter n-grams within the window and move to the next word beyond the current window and continue to scan the text in a window of size 5 from there.

Based on the words and MWTs, or simply called terms together, we propose two standards to select the content that we need to represent the web-pages. The first one is based on the frequency. We assume that if a term appears more times in the original page, then it should also have a higher chance to occur in a summary:

$$E[freq(w \mid s)] = E[freq(w \mid d)] \qquad (4.1)$$

where $E[.]$ is the expectation operator; $freq(w \mid s)$ is the frequency of term *w* in summary *s*; and $freq(w \mid d)$ is the frequency of term *w* in web-page *d*. This agrees with the simple assumption that the more a lexical unit appears in a text, the more important it is (Nenkova and Vanerwende, 2005).

Although frequency is a useful standard to locate the important content for summarization, it is not adequate. When the frequencies of terms are close to each other or even the same from a text, they can lead to poor performance and results. Our second standard to select important terms is based on the position. According to other researchers (Cruz and Urrea, 2005; Katragadda et al, 2005), the position of a term in text can provide useful information on how important it is. Human writers tend to put the key points at the beginning and/or at the end of an article. Thus, the terms which are located near these two ends should have higher chances to carry the useful information. We adjust the frequency of a term by taking the positions of their original tokens into account. We calculate the adjusted count of term *W* as follows:

$$adjusted\_count(W) = \alpha + (1-\alpha)\beta^{\min - dis\tan ce} \qquad (4.2)$$

where α and β are both real numbers between [0, 1], and they work as parameters to adjust the weights between the frequency and the position. The min-distance is the minimal word distance between the term which produces *W* and the beginning/end of the text. For example, if the term that produces *W* is the 5th token in the text with 15 tokens, then the min-distance is set to 4 since the word distance from the beginning is 4 and the one from the end is 10. When the term that produces *W* is the first or the last token, the min-distance is 0, and the adjusted count is 1. When the min-distance is approaching infinite, the adjusted count is α. So the maximal value of an adjusted count is 1, and the minimal value is α. Normally, the value of an adjusted count is somewhere between 1 and α. We accumulate the adjusted counts for each term and take the sum as the adjusted frequency for the term. By combining the frequency and position factors together, we can effectively locate the right terms we need for summarization. The terms with higher adjusted frequencies are considered important for content selection.

We rank all the terms recognized in the original web page by their adjusted frequencies, and select the top N terms to generate a summary. According to the empirical study on human-edited summaries (average length of the human written summaries in ODP data), it is found that most summaries are between 10-20 words. With 40 lexical units, we believe that there should be adequate information for summary generation.

4.3 Translation and Ordering

When human editors think about words to describe a web page, they mentally identify the key topics and use words or phrases that are more concise and comprehensive to form a summary. We can think this process as a kind of “translation” which maps a set of words from a web page into another set of words in a summary. For example, given a segment from a web-page about ASI accounting service (http://www.accountservices.com/): “We provide accounting, bookkeeping, payroll services and support, initial set up, data conversion, comprehensive problem solving, QuickBooks training.”, a human editor may describe it as “We offer a variety of accounting services”. We can see that word “provide” is translated to “offer”, and the long word sequence “accounting, bookkeeping, payroll services and support, initial set up, data conversion, comprehensive problem solving, QuickBooks training” is translated to “a variety of accounting services”. Based on these “translated” words and phrases, human editors can organize them into a proper order so that the summary can be more meaningful and readable.

Statistical machine translation has been studied for decades to provide a solution that translates one language to another by exploring the relationship between the two languages. For web-page summarization, we need to study the relationships between the words in web pages and the words in their summaries from a training set so that the parameters of the translation model can be optimized.

Given a web page d and a summary s, the translation can be seen as the process of finding s that maximizes $\Pr(s|d)$, which is the probability of having s as the summary for d. However, calculating the probability $\Pr(s|d)$ directly is often difficult, since many sets of words or MWTs could be the translations for each d, resulting in a large number of $\Pr(s|d)$ to estimate. On the other hand, the words and MWTs that tend to appear in human-edited summaries only make up a small fraction of a large vocabulary of words.

We could easily end up with a model that allocates little probability mass for the words and MWTs that actually appear in summaries. Thus, it is important that our model concentrates its possibility mass on the words and MWTs that we care about in order to reduce the search space to discover them.

We can turn the problem around by applying the Bayes' theorem as follows:

$$\Pr(s \mid d) = \frac{\Pr(s)\Pr(d \mid s)}{\Pr(d)} \qquad (4.3)$$

Here, Pr(*s*|*d*) is computed from three other distributions: Pr(*s*) is the prior probability of summary *s*; Pr(*d*) is the prior probability of web page *d*; and Pr(*d*|*s*) is the conditional probability of generating the web page *d* based on the summary *s*. The advantage of using Pr(*d*|*s*) is that it can be estimated by examining the individual pairs of *d* and *s* from training data. By focusing on the distribution Pr(*d*|*s*), a high quality translation can be achieved if we can estimate the distribution appropriately. Moreover, since our goal is to find *s* with the maximal Pr(*s*|*d*) for the same denominator Pr(*d*), we can simplify the problem by finding the *s* to make the product $\Pr(s)\Pr(d \mid s)$ as large as possible. This leads to the fundamental equation for statistical machine translation:

$$\hat{s} = \arg\max_{s} \Pr(s)\Pr(d \mid s) \qquad (4.4)$$

where $\hat{s}$ denotes the summary that maximizes the product $\Pr(s)\Pr(d \mid s)$. The first part Pr(s), also called the language model, captures how likely a summary is formed from all the possible summary candidates, and the second part Pr(d|s), also called the translation model, the Pr(*d*|*s*) part, illustrates how likely the words in a web page are aligned to the words in a summary. With this framework, the translation becomes the process of searching the best summary that maximize the product of Pr(s) and Pr(d|s), also called the decoder method for machine translation (Brown *et al*. 1990).

4.4 Building the Models for Summarization

Building the language and translation models requires an appropriate training data set. Fortunately, we can use the ODP data for this purpose. For each web page in the ODP data set, there is a human-edited summary for it. Thus, by applying machine learning techniques, we can train and refine the parameters for our models, and thus build a web-page summarization system.

In order to build the translation model, we need to compute the probabilities that align words in a web page to those in one of its summaries. The idea of alignments is introduced by (Brown *et al*. 1990) to indicate the word relationships in a translation between two languages. An example alignment can be shown graphically by the lines (connections) between the words of two sources in Figure 4.1.

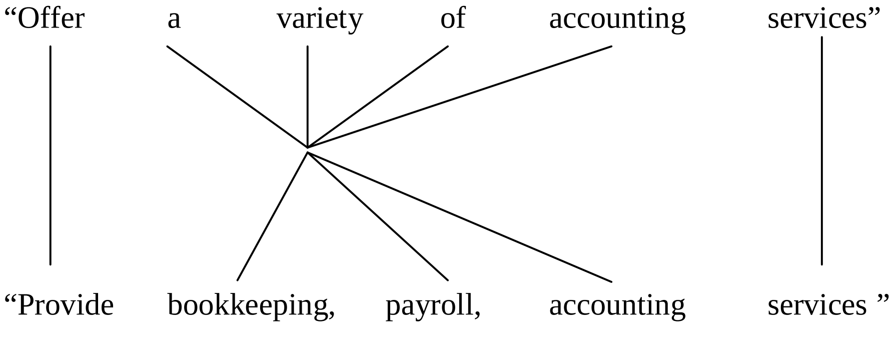


Figure 4. 1: Alignment for an Example Translation

Note that several words in a summary can be connected to several other words in a web page. For example, the three words "*bookkeeping*, *payroll*, *accounting*" together generate the four words "*a variety of accounting*".

The alignment process for summarization is less complex than that for language translation. As mentioned before, we only focus on the translations between single words and MWTs and do not need to worry about the order of these terms. As a result, we can restrict our alignments to one-to-one mappings where one group of words or MWTs in a summary connects with exactly one group of words or MWTs in a web-page, as shown in Figure 4.2.

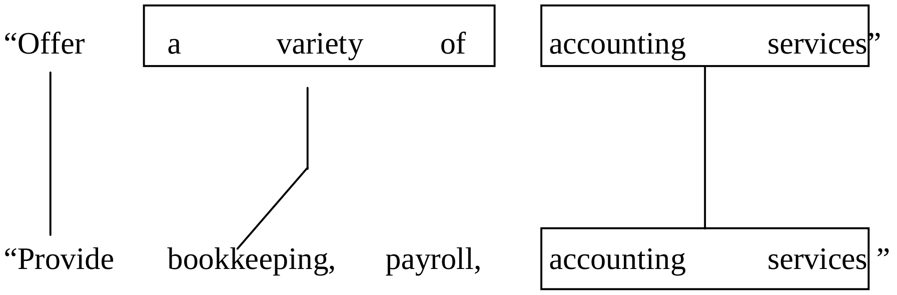


Figure 4. 2: One-to-one Alignments between Words and MWTs

With alignments, we can express the conditional probability Pr(d|s) in terms of Pr(d, a|s), which means the probability of generating web-page *d* under alignment *a* for summary *s*. More specifically, given a pair of (***d, s***) in the training data, if we parse web-page ***d*** into *m* terms (words or MWTs) and its summary ***s*** into *l* terms, then there are a total of *lm* possible connections between them since each of the *m* terms from ***d*** can be connected to any of the *l* terms in ***s***. Let A(***d***, ***s***) denote the set of all alignments between *d* and *s*. Then, A(***d***, ***s***) can be thought as an $l \times m$ matrix, where each element a_{ij} indicates the degree of connections between the *i*-th term in *s* and the *j*-th term in *d*.

We compute the strength of a connection between two terms by counting the number of times that the two terms connect to each other in the training data. Furthermore, we introduce the expected number of times that term s_j (the *j*-th term in ***s***) connects to term d_i (the i-th term in ***d***) in a translation as a measure of connection, which is denoted as $C(d_i \mid s_j; \vec{d}, \vec{s})$ and defined as follows:

$$C(d_i \mid s_j; \vec{d}, \vec{s}) = \frac{\Pr(d_i \mid s_j)}{\sum_{k=1}^{L} \Pr(d_i \mid s_k)} \sum_{k=1}^{M} \delta(d_k; \vec{d}) \sum_{k=1}^{L} \delta(s_k; \vec{s}) \quad (4.5)$$

where $\sum_{k=1}^{M} \delta\left(d_k, \vec{d}\right)$ is the count of term d_i in web-page ***d***, $\sum_{k=1}^{L} \delta\left(s_k; \vec{s}\right)$ is the count of term s_j in summary ***s***. $C(d_i \mid s_j; \vec{d}, \vec{s})$ multiples the observed number of times the two terms d_i and s_j in the summary and web-page pair (***d*****,** ***s***) with the probability $\Pr(d_i \mid s_j)$ that *d* happens in web-page ***d*** given summary ***s***. Thus, the strength of the connection between two terms in a web-page and summary pair is partially decided by the observations from the training data, and partially decided by the probability $\Pr(d_i \mid s_j)$.

Clearly, the probability $\Pr(d_i \mid s_j)$ is affected by the connection between terms. If two terms d_i and s_j have a strong connection, the probability $\Pr(d_i \mid s_j)$ will be large, and vice verse. Thus, we can say that the probability $\Pr(d_i \mid s_j)$ is directly proportional to $C(d_i \mid s_j; \vec{d}, \vec{s})$. We can write $\Pr(d_i \mid s_j)$ as:

$$\Pr\left(d_i \mid s_j\right) \equiv \sum_{t=1}^{T} C(d_i \mid s_j; D_{T,} S_T)$$

Equation 4. 1: Translation Probability

where (D_T, S_T) is a training set that contains T web-page and summary pairs. Thus, the probability $\Pr(d_i \mid s_j)$ is estimated from the expected number of times that s_j connects to d_i in the training set. By combining Formulas 4.5 and 4.6, we find that $\Pr(d_i \mid s_j)$ appears on both sides of the equation. This suggests an iterative procedure for finding the solution. We need to first give the probability $\Pr(d_i \mid s_j)$ an initial guess so that we can evaluate Formula 4.5 and use the result to estimate a new value for $\Pr(d_i \mid s_j)$ with Formula 4.6. This iterative process is a form of the EM algorithm, which is introduced by (Baum, 1972).

Based on the above solution for the probability $\Pr(d_i \mid s_j)$, we can adjust its values so that they will subject to the constraint for each s_j :

$$\sum_d \Pr\left(d \mid s_j\right) = 1 \qquad (4.7)$$

For this purpose, we introduce the normalization constant λ, which is defined as:

$$\lambda = \sum_d \sum_{t=1}^{T} C(d \mid s_j; D_T, S_T) \qquad (4.8)$$

After that, we can use λ to get the normalized probability $\Pr(d_i \mid s_j)$;

$$\Pr\left(d_i \mid s_j\right) = \lambda^{-1} \sum_{t=1}^{T} C(d \mid s; D_{T,} S_T) \qquad (4.9)$$

Formula 4.9 is the final equation for calculating the probability $\Pr(d_i \mid s_j)$, and we can write the whole process in the following steps:

Step 1: Choose initial values for each $\Pr(d_i \mid s_j)$.

Step 2: For each web page and summary pair (***d***, ***s***) in the training set (D, S), use Formula 4.5 to calculate the expected counts of s_j connected to d_i.

Step 3: Compute λ in Formula 4.8; and use the result to normalize $\Pr(d_i \mid s_j)$ with Formula 4.9.

Step 4: Repeat steps 2 and 3 until the values of $\Pr(d_i \mid s_j)$ are converged to a desired degree.

The initial values for each $\Pr(d_i \mid s_j)$ are not a significant issue to the final results because $\Pr(d_i \mid s_j)$ will reach a local maximum in the iteration process. In our implementation, we initialize all of the $\Pr(d_i \mid s_j)$ with an equal value.

For the language model, which corresponds to Pr(*s*) and indicates the order of terms in a summary, we calculate it with a bigram language model. A bigram language model assumes that the probability of a word appearing in a specific position only depends on the word before it. For example, the probability of the word sequence "I like red apple" under a bigram model is approximated as:

$$\Pr(I, like, red, apple) \approx \Pr(I \mid \Theta)\Pr(like \mid I)\Pr(red \mid like)\Pr(apple \mid red) \quad (4.10)$$

where $\Pr(I \mid \Theta)$ stands for the probability that "I" appear at the beginning. Thus, we only need to calculate every $\Pr(w_i \mid w_{i-1})$ from the training set as follows:

$$\Pr(w_i \mid w_{i-1}) = \frac{count(w_{i-1}w_i)}{count(w_{i-1})} \qquad (4.11)$$

where $count(.)$ is the frequencies of the corresponding terms in the training set.

The decoder method helps rank different summaries, but searching for all possible summaries is too expensive in practice. For example, if there are 20,000 words and MWTs for summaries, generating a summary of 20 terms will have up to $20{,}000^{20}$ combinations for all possible summaries. To cope with the complexities, we need to set limits on both the number of terms N for the summary and the number of terms M from the web page through content selection. For our experiments in Chapter 5, we set N = 20 and M = 100.

During the process of finding up to N terms in a summary, we further apply the local beam search method to reduce the cost in time and space. The beam refers to the number of candidates to consider each time we add a new term for the summaries. In our experiments, we set this number to be 40. At the beginning, we look at all the summary terms that are related to the top M terms from a web page through content selection. Then, we use the decoder method to rank all these terms and only keep the top 40 candidates in the beam. After that, we expand each candidate with all possible summary terms and again use the decoder method to rank and keep the top 40 candidates for the next summary term. This process is repeated until up to N terms are selected for the summaries and the best summary according to the decoder method will be selected for the final summary. By controlling the size of the beam, we are able to reduce the cost for both time and space, making the search process manageable for summarization.

4.4 Summary Generation

We get a subset of words and MWTs that contain the important information about a web page through content selection. With the decoder solution of machine translation, we are able to map these terms to the related words and MWTs from the summaries of the training data so that we can form an abstractive summary for the given web page. As stated earlier, the mappings only capture the important relationships between the terms, not the order of the mapped terms, since a summary is much shorter than a web page, unlike the language translation where the mapped sentence is more or less the same length as the source sentence. Summary generation is to optimize the order of the mapped terms in a summary and we intend to use the *Viterbi* algorithm (Forney, 1973) for this purpose.

The *Viterbi* algorithm can find a state sequence with the maximum likelihood. It is usually expressed with the partial probabilities δ :

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} p\{q_{1,} q_2, ..., q_{t-1,} q_t = i, o_1, o_2, ..., o_{t-1} \mid \lambda\} \qquad (4.12)$$

where $\delta_t(i)$ is the maximum probability of all sequences ending at state *i* at time *t*, and the partial best path is the sequence which achieves this maximal probability. *Viterbi* algorithm starts the calculation from *t*=1: $\delta_1(i) = \pi_j b_j(o_1)$. Based on the first order Markov assumption that the probability of a state in a sequence depends only on its previous state, *Viterbi* algorithm uses $\delta_{t-1}(i)$ to compute each $\delta_t(i)$ as follow:

$$\delta_{t+1}(i) = b_j(o_{t+1})[\max_{1 \le i \le N} \delta_t(i) a_{ij}] \qquad 1 \le i \le N, \quad 1 \le t \le T-1 \qquad (4.13)$$

where state *j* is the only factor that makes difference for each $\delta_t(i)$. Thus, the problem reduces to finding the state j which maximizes the δ function, that is, $j^* = \arg\max_{1 \le j \le N} \delta_t(j)$.

Using the *Viterbi* algorithm, we expect to find the best order of the mapped terms that are both meaningful and readable for the user and serve as a summary for the given web page.

Chapter 5

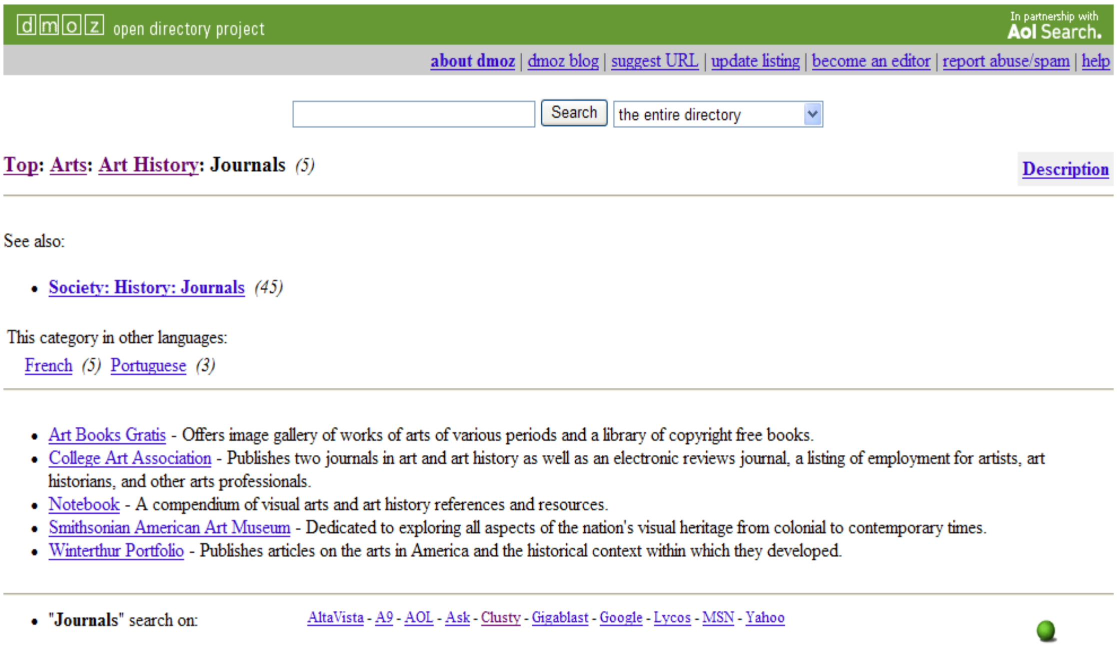
Experimental Results and Discussions

In this chapter, we describe the methods used for evaluating our systems for multi-word term extraction and web-page summarization. We first conduct a series of experiments to compare and evaluate our multi-word term extraction methods on a data set from Open Directory Project (OPD). Both human and automatic evaluations are used to measure the performance of our methods in order to understand the advantages and the problems of our approaches.

We then perform two separate sets of experiments to evaluate our web-page summarization method. The first set of experiments focuses on the content selection part of the summarization system on 2007 DUC (Document Understanding Conferences) data set. The second set of experiments measure the overall performance of our web-page summarization system on the ODP data set. We apply the automatic ROUGE measure (a package for the automatic evaluation of summaries) to evaluate the performance of our web-page summarization method.

5.1 Data Sets

The first data set is taken from the Open Directory Project (ODP), which has the largest and most comprehensive human-edited directory of the Web. It is constructed and maintained by a global community made of a vast number of volunteer editors. The ODP dump we used contains over 4 millions of edited websites from 74,719 editors all over the world. Figure 5.1 is a sample web page which shows the basic structure of an ODP website.


dmoz open directory project
In partnership with Aol Search.
about dmoz | dmoz blog | suggest URL | update listing | become an editor | report abuse/spam | help
Search
the entire directory
Description

Top: Arts: Art History: Journals *(5)*

See also:

- **Society: History: Journals** *(45)*

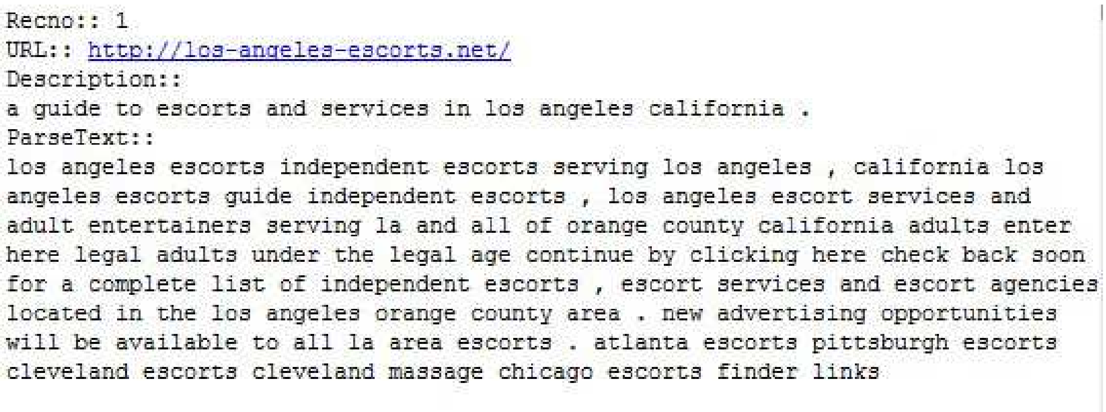
This category in other languages:
French *(5)* Portuguese *(3)*

- Art Books Gratis - Offers image gallery of works of arts of various periods and a library of copyright free books.
- College Art Association - Publishes two journals in art and art history as well as an electronic reviews journal, a listing of employment for artists, art historians, and other arts professionals.
- Notebook - A compendium of visual arts and art history references and resources.
- Smithsonian American Art Museum - Dedicated to exploring all aspects of the nation's visual heritage from colonial to contemporary times.
- Winterthur Portfolio - Publishes articles on the arts in America and the historical context within which they developed.

- "**Journals**" search on: AltaVista - A9 - AOL - Ask - Clusty - Gigablast - Google - Lycos - MSN - Yahoo

Figure 5. 1: Sample Web-page from Open Directory Project

Each website link in ODP is paired up with its human written summary that is stored in RDF (Recourse Description Frame http://www.w3.org/RDF/) format. We use the dump file ***content.rdf.u8.gz*** (http://www.dmoz.org/rdf.html) as our source of experiment data. After splitting the URLs and the human written summaries of each website out of the dump file, we apply Nutch (http://nutch.apache.org/), which is open source web-search software to get the textual content of these web-pages. This allows us to match the web-page content with its human-written summary, an example of which is shown in Figure 5.2 below:



```
Recno:: 1
URL:: http://los-angeles-escorts.net/
Description::
a guide to escorts and services in los angeles california .
ParseText::
los angeles escorts independent escorts serving los angeles , california los
angeles escorts guide independent escorts , los angeles escort services and
adult entertainers serving la and all of orange county california adults enter
here legal adults under the legal age continue by clicking here check back soon
for a complete list of independent escorts , escort services and escort agencies
located in the los angeles orange county area . new advertising opportunities
will be available to all la area escorts . atlanta escorts pittsburgh escorts
cleveland escorts cleveland massage chicago escorts finder links
```

Figure 5. 2: Matched Web-page Content and Summary Pair

where “Description” tag indicates the human-edited summary and “ParseText” tag indicates the original web-page content.

From over 4 millions edited websites, we successfully generate more than 3.8 millions of web-page and summary pairs since some websites no longer exist for Nutch to fetch. After that, we filter out all non-English pairs and remove those with little contents and/or descriptions. The remaining 2 millions of web page and summary pairs are used for our experiments.

In addition to the ODP data, we also use the data set from Document Understanding Conferences (DUC) 2007 to validate the content selection part of our web-page summarization system. DUC is a series of summarization evaluation conferences that were conducted by the National Institute of Standards and Technology (NIST) between 2001 and 2007. After 2008, DUC became a summarization track of the Text Analysis Conferencs (TAC). DUC seeks to promote the development of automatic text summarization, and provide researchers the resources and opportunities to carry out experiment for both the development and evaluation. The main task of DUC 2007 contains 45 topics and a set of 25 relevant documents for each topic. For each topic and its document cluster, there are four 250-word summaries that are given by four different NIST assessors as references.

5.2 Evaluation Methods

5.2.1 Measures for Multi-word Term Extraction

Evaluating the extraction of multi-word terms (MWTs) is a challenging task in that there is no well-accepted standard. MWTs are usually domain and language dependent, and for different corpora, evaluation methodologies and testing scopes are often different, leading to varied results for a given approach.

Current evaluations can be broadly divided into human-based, dictionary-based and standard-based. Human-based evaluations are made by humans, usually with linguistic knowledge. Human judges can be more accurate and fair, but they are both expensive and inefficient to perform the evaluation. Dictionary-based and standard-based evaluations are made by computers, typically based on the comparisons of the results from an extraction system with existing dictionaries or measurements for some pre-defined standards. These evaluation methods can process a large amount of results with high efficiency, but both dictionaries and standards can be diverse, making the evaluation results less comparable. In addition, these methods are not always objective since no dictionary and standard can be fully complete and unbiased. Finally, the scopes of the evaluations are often varied: some evaluations cover all the results, while others only focus on part of the results such as the top-N best results.

Following the common practice in information retrieval, we use recall and precise along with a combination of the two to measure the comparisons between the computed results and the desirable results for multi-word term extraction. More specifically, we classify the results into four categories: true positive, false positive, false negative, and true negative. If an n-gram is recognized as a MWT by a particular approach and is also a desirable MWT, it is considered as true positive; if an n-gram is recognized as a MWT but is not a desirable MWT, it is considered as false positive; if an n-gram is not recognized as a MWT but is indeed a desirable MWT, it is considered as false negative; and if an n-gram is not recognized as a MWT and it is indeed not a desirable MWT, it is considered as true negative. Based on these four categories, recall and precision are then defined as follows:

$$\begin{aligned} recall &= \frac{truepositive}{truepositive + falsenegative} \\ precision &= \frac{truepositive}{truepositive + falsepositve} \end{aligned} \qquad (5.1)$$

Recall is a measure of completeness, while precision is a measure of exactness. A high recall score means that the extraction system can identify many desirable MWTs, but it does not tell how many n-grams are incorrectly recognized as MWTs. A high precision score means that the extraction system can identify MWTs with a high accuracy, but it may miss many desirable MWTs. Often, there is an inverse relationship between recall and precision: increasing recall often decreases precision, and vice versa. In order to measure the overall performance, recall and precision need to be combined and one such composite measure is called the F measure, defined as follows:

$$F = 2 \cdot \frac{recall \times precision}{recall + precision} \qquad (5.2)$$

In addition to recall, precision, and F measure, researchers have been trying to find other measures in order to make disparate extraction approaches comparable. The work (Evert and Krenn, 2001) carries out a series of tests to find a qualitative evaluation of lexical association measures. The evaluation method includes: best N-list, recall and precision graph, frequency strata, significance testing, and statistical estimation on low-frequency terms. (Zhang *et al*. 2008) proposes an evaluation method based on the voting algorithm, which combines many extraction approaches together to decide the rank of a selected MWT for a total of 100 evaluated terms. Then the evaluation method judges the performance of a certain extraction approach by comparing the ranking order offered by this approach against that provided by the voting algorithm. If the approach gives a similar ranking order for the 100 evaluated MWTs as the voting algorithm, the approach is considered good; otherwise, poor.

5.2.2 Measures for Web-page Summarization

The evaluation of automatically generated summaries is also a difficult task since even for the manually generated summaries, there is no agreement on what makes a good summary. Furthermore, summaries for different forms of documents can differ greatly. For example, a summary for a newspaper article is often different from that for a web page. Human evaluations may get relatively accurate and unbiased results, but it is not feasible since it is both time consuming and expensive.

The current prevailing evaluation method is to compare the machine generated summaries with the human edited ones for the same target document. (Lin and Hovy, 2002) proposes such an evaluation approach and weight the comparison results by a recall value which is defined as the ratio of the number of the model unit mark at or about a threshold *t* with the number of the model units in the model summary. The threshold *t* is set to a number between 1-4, where 1 means hardly; 2 means some; 3 means most; and 4 means all. Such an evaluation method measures the completeness of a summary and assumes that the more information a summary preserves from the original text, the better the quality.

(Lin, 2004) later introduces Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measures which have now become the standards for the evaluation of automatically generated summaries. ROUGE offers 4 kinds of measures at different levels: N-gram co-occurrence (ROUGE-N), longest common subsequence (ROUGE-L), weighted longest common subsequence (ROUGE-W), and Skip-bigram co-occurrence (ROUGE-S). Each measure has its own advantages and is good for some summarization tasks.

N-gram co-occurrence (ROUGE-N) measures n-gram co-occurrences between reference and candidate summaries and is defined as follows:

$$ROUGE-N=\frac{\sum_{S\in\{\mathrm{Re}\ fernce\}}\sum_{gram_n\in S}Count_{match}(gram_n)}{\sum_{S\in\{\mathrm{Re}\ ference\}}\sum_{gram_n\in S}Count(gram_n)} \qquad (5.3)$$

where n stands for the length of the n-gram, and $Count_{match}(gram_n)$ is the number of n-grams co-occurring in a reference summary and a candidate summary. When there are more than one reference summaries, we calculate the ROUGE-N between a candidate summary and each of the reference summaries and among these ROUGE-N scores; then we pick the maximum as the final ROUGE-N score.

While ROUGE-N counts the co-occurrences of n-grams between candidate and reference summaries, longest common subsequence (ROUGE-L) focuses on the maximum length of the word sequence that a candidate and a reference have in common. The longer the maximum length word sequence, the higher the ROUGE-L score. When both the candidate and the reference summaries are made up of sentences, it can be defined as follows:

$$ROUGE - L_{sentence} = \frac{LCS(S,R)}{length(R)} \quad (5.4)$$

where S stands for a candidate summary sentence, R stands for a reference summary sentence, *LCS(S.R)* is the length of the longest common subsequence between S and R, and $length(R)$ is the length of the reference sentence.

Compared with ROUGE-N, ROUGE-L cares more about the sentence structure and the readability. For the following example,

Reference: Joe likes hot dog.

Candidate 1: Joe like hot dog.

Candidate 2: hot dog like Joe.

Both candidates have the same ROUGE-2 score since they all have bigram “hot dog”, but candidate 1 has a higher ROUGE-L score than candidate 2 because it has a longer common subsequence “Joe hot dog” than candidate 2. Thus, ROUGE-L is useful for evaluation tasks at the sentence level.

Weighted longest common subsequence (ROUGE-W) is based on ROUGE-L, but favors a word sequence with consecutive matches between a candidate and a reference. It records the lengths of consecutive matches by a dynamic programming table, and is defined as follows:

$$ROUGE-W = f^{-1}\left[\frac{WLCS(S,R)}{f(length(R)}\right] \qquad (5.5)$$

where W*LCS(S.R)* is the weighted longest common subsequence score between sentences S and R, and *f* is a weighting function which must meet the requirement *f*(*x*+*y*) > *f*(*x*) + *f*(*y*) to ensure that consecutive matches receive higher scores.

Skip-bigram co-occurrence (ROUGE-S) counts skip-bigram co-occurrences between a candidate and a reference. Skip-bigram is any pair of words in the sentence order, allowing for arbitrary gaps. For example, the sequence “Joe likes hot dog” has 6 skip-bigrams: {“Joe likes”, “Joe hot”, “Joe dog”, “likes hot”, “likes dog”, “hot dog”}. ROUGE-S considers long distance dependencies, and allows gaps in matches by counting all in-sequence pairs. More specifically, it is defined as follows:

$$ROUGE-S = \frac{Skip2(S,R)}{C(length(R),2)} \qquad (5.6)$$

where *Skip2(S.R)* is the weighted longest common subsequence score of S and R, and C is the combination function. When we deal with long sequences, we need to limit the skip distance to reduce the skip-bigrams we can get. Normally, we set the skip distance to 4, that is, we match bigrams with skip distance up to 4 words.

In addition, when ROUGE-S cannot find any skip-bigram matches due to the word order of the candidate, we add unigram matches so that we can differentiate the candidates that do have common words from the ones that do not have matched words with the references. Such an extension is called ROUGE-SU.

Generally speaking, ROUGE-N is useful for all kinds of evaluation tasks. ROUGE-W is suited for short (e.g., 100 words) single document summarization. ROUGE-SU and ROUGE-L works well for tiny (e.g., 10 words) headline summarization. ROUGE-S and ROUGE-SU are also good for multi-document summarization. For our evaluation of content selection with the DUC data, we use ROUGE-N (1-4), ROUGE-L and ROUGE-SU4. For our evaluation of web-page summarization, we take the human edited summaries as references, and use ROUGE-N (1-4), ROUGE-W, ROUGE-L and ROUGE-SU4 to measure the experimental results.

5. 3 Results on Multi-word Term Extraction

In Chapter 3, we described four versions of our implementation for multi-word term extraction. The baseline system relies on Sequence Frequencies (SF), which is a re-implementation of the symmetrical conditional probability method from (Silva *et al*. 1999) but uses the sequence frequencies directly as recommended by the work (Aires *et al*, 2008). The second version replaces the joint probabilities in the calculations of symmetrical conditional probabilities with our Smoothed Probabilities (SP) as discussed in section 3.3. The third version explores the potential of our Normalized Sequence Probabilities (NSF) computed from the sequence frequencies as described in version one above. Finally, the fourth version is based on our Normalized Smoothed Probabilities (NSP) which combines the normalized sequence probabilities with the smoothed conditional probabilities of n-grams. We expect that the smoothed probabilities helps address the sparse data problem and the normalization makes it possible to find MWTs made of high-order n-grams,

5.3.1 Experiments and Results

Our multi-word term extraction methods need the frequencies of n-grams. We divide the processed ODP data into two subsets: 1,200,000 for training and 800,000 for testing. We count the frequencies of 1-grams to 5-grams from the 1,200,000 web-page and summary pairs of the training subset.

As stated earlier, we need to compare the computed results with the desirable MWTs identified by human editors to measure the performance. To control the cost, we randomly select 500 web pages from the testing subset, and among the 500 web pages, we further remove 100 web pages due to inappropriate content (mostly related to porn and adult websites). From the remaining 400 web pages, we manually select MWTs based on the rules detailed in the Appendix. Four people with NLP background are invited to do the selection, and in the end, they agree on 737 MWTs from the 400 web-pages along with the related summaries.

For the smoothed conditional probabilities, we need to tune the weighting parameters for n-grams of different orders as described in Chapter 3. Due to limited time and resources, we focus on finding optimal values for λ_2 and λ_3, but empirically assign λ_4=0.6 and λ_5=0.8. This is because a majority of MWTs are 2-grams and 3-grams and only a small portion of MWTs are longer than 3-grams. Thus, optimizing λ_2 and λ_3 may have a great impact on the performance of systems, while λ_4 and λ_5 may not be as important.

We perform a relatively crude method to find optimal parameter values. We assume the range of [0, 0.3] for λ_2 and the range of [0, 0.6] for λ_3, but within these ranges, we try all possible combinations of the small interval values. More specifically, we try 28 combinations for λ_2 and λ_3, using intervals (0, 0) (0, 0.1) (0. 0.2), and so on, and find that when λ_2=0 and λ_3=0.3, the SP and NSP methods give the best performance. More detailed results on the tuning of these parameters are given in the Appendix.

Our best results for the four versions of our implementation for multi-word term extraction are shown in Table 5.1.

Systems	Recall	Precision	F-Measure
SF	0.592	0.167	0.261
SP	0.707	0.194	0.305
NSF	0. 937	0. 429	0.588
NSP	0. 940	0. 445	0.604

Table 5. 1: Performance of Extraction Models before filtering

In addition, we added the post-processing filter for all these four systems and the improved results are shown in Table 5.2.

Systems	Recall	Precision	F-Measure
SF	0.491	0.523	0.506
SP	0.610	0.573	0.591
NSF	0. 812	0. 661	0.728
NSP	0. 814	0. 662	0.730

Table 5. 2: Performance of Extraction Models after filtering

As can be seen from Tables 5.1 and 5.2, the baseline system based on Sequence Frequencies (SF) has lower performance than our system based on Smoothed Probabilities (SP), while the systems based on Normalized Sequence Probabilities (NSF) and Normalized Smoothed Probabilities (NSP) achieve better results than those without using the normalization. These results indicate that our proposed normalization method is effective in improving the performance, and the smoothed probabilities are also working well as we expected. When combined with the normalization, the smoothed probabilities help achieve the best results (NSP over other methods). The post-processing filter, as shown in Table 5.2, is particularly effective for English and helps improve the performance significantly for all versions of our systems.

Besides recall, precision, and F-Measure, we count the number of MWTs across different lengths that are extracted by the four systems and the results are shown in Table 5.3.

Systems	2-gram	3-gram	4-gram	5-gram
SF	554	0	0	0
SP	554	0	0	0
NSF	624	166	19	0
NSP	625	167	22	1

Table 5. 3: Length Distributions of the N-grams Extracted

Systems based on SFs and SPs can mostly extract MWTs of two words, while systems based on NSFs and NSPs can also extract longer MWTs. From Table 5.3, we can see that around 166 3-grams and 20 4-grams are captured by systems based on NSFs and NSPs. These results illustrate another advantage of our proposed normalization method in that it can successfully extract long MWTs.

5.3.2 Discussion

From the experiments above, we find that our proposed normalization method works well as expected. Systems based on this normalization outperform those without the normalization significantly when used with the LocalMaxs algorithm. The core concept of the LocalMaxs algorithm is that a MWT must have a stronger glue value than both its antecedents and its successors. Our normalization helps shrink the gaps between n-grams of different lengths effectively, making these n-grams more comparable for the glue values. As a result, we can select more meaningful MWTs and at the same time reduce the unwanted ones.

Furthermore, our normalization method effectively “enlarges” the glue values for long MWTs, making it possible to extract terms longer than two words. Long MWTs are also valuable in capturing the content: they contain detailed information and sometimes represent intact concepts that cannot be replaced. However, recognizing them can often be difficult. First, long MWTs seldom repeat themselves frequently in a context, leading to low frequency and probability values. Secondly, the longer the terms, the weaker the glue values within them. One problem with the SCP measure is that the glues of bigrams are so strong that only MWTs of two words can be selected. Our normalization method overcomes the issue. We selected over 160 trigrams and 19 4-grams in our experiments for multi-word term extraction.

Our smoothing method for conditional probabilities also brings significant improvements in our experimental results. Due to the time and resource constraints, we only did coarse-grained tuning for the weighting parameters, but there should be rooms for further improvements. Overall, smoothing is useful to address the sparse data problem for natural language data and we will leave it as future work to explore more effective ways of smoothing n-gram probabilities.

When extracting MWTs, we intentionally keep the stop words such as “of”, “as”, and “in” for readability. Certainly, these stop words can have interferences for our results, especially on precision. This is because stop words tend to have high frequencies and accordingly, the glues between the stop words and their adjacent words are usually high as well. For example, terms such as “of the” and “in the” usually have very high glue values, but they are not so meaningful for representing concepts. Our post-processing filter simply remove stop words at the start or the end of MWTs and, as demonstrated by our experiments, can achieve a huge boost in the performance for all four versions of our implementation for multi-word term extraction.

5.4 Results on Web-page Summarization

In Chapter 4, we introduced our method for web-page summarization. The system is based on the decoder solution of machine translation and MWTs will be used for both content selection and summary generation. In this section, we take a two-step approach to verify our web-page summarization system. We first apply the ROUGE toolkit to validate the content selection part. It is a necessary prerequisite to make sure that this part can locate the key information in a web page effectively. Otherwise, we cannot expect that the translation will lead to a meaningful summary. After confirming the performance of content selection, we move on to the second step of evaluating the overall performance of our web-page summarization system.

5.4.1 Experiments and Results

The quality of content selection affects the following step and the performance of the entire summarization system. A high quality output from the content selection part can sometimes be taken as a summary directly. We compare our results with the well-known summarization system called SumBasic (Nenkova and Vanderwende, 2005). SumBasic produces generic multi-document summaries. Its design is motivated by the observation that words occurring frequently in a document cluster also occur with a high probability in the human-edited summaries. One reason we choose SumBasic is that it selects sentences for a summary so that we can measure how our selection of MWTs compares with SumBasic in terms of ROUGE scores and readability. Intuitively, MWTs may not be as readable as sentences, but they should contain as much information as the sentences with less redundancy for stop words. If our content selection part can achieve a better recall performance than SumBasic, it should be effective for text summarization and even more useful for web-page summarization due to the lack of structures in web pages.

In addition, we also compare our results with another summarization system that simply selects words based on their frequencies. We expect that summaries based on MWTs should be more readable and at the same preserve as much as or even more information than summaries made of individual words.

We use the DUC 2007 data set that contains 45 topics, each of which has a set of 25 relevant documents. For each document cluster, there are four well-organized 250-word summaries given by four different human assessors. To make the lengths comparable, we select 9 sentences for the summary of SumBasic and about 150 MWTs for the summary of our content selection part.

The results based on ROGUE-1, ROGUE-2, ROGUE-3, ROGUE-4, and ROGUE-SU4 are shown in Table 5.4 below.

	SumBasic	MWTs	Individual Words
ROUGE-1	0.4983	0.5235	0.2739
ROUGE-2	0.1001	0.1374	0.0504
ROUGE-3	0.0281	0.0346	0.0074
ROUGE-4	0.0119	0.0125	0.0026
ROUGE-L	0.4480	0.4198	0.1651
ROUGE-SU4	0.1733	0.1994	0.0602

Table 5. 4: Recall Performance of Three Different Systems

In Table 5.4, we can see that our content section part gives the best recall values, followed by SumBasic. The system based on individual words gives the worst results among the three systems. This proves that our content section part is more effective for extracting key content from a target document than SumBasic and the system based on individual words. Although the readability of NWTs is not as good as sentences, we find that they still provide meaningful information about the related document, as shown in the following sample summary generated by our content selection part.

> “monday turkey would be admitted into the european union but the human rights situation with the kurds and its relations with greece said kinkel eu membership relations with the relations between turkey and eu have been soured decided last december not to accept turkey as did not mesut yilmaz turkey's entry into the eu ankara nato today turkish european union turkey has the european commission membership turkey is turkey in by the in a turkish president suleyman demirel said that turkey will enlargement the eu summit brussels european countries applicant”

With the encouraging results of content selection, we move on to evaluate the performance of the entire web-page summarization system. Based on the MWTs extracted from content selection, we translate them to the MWTs in the summary space and use the Viterbi algorithm to order them for the generation of the abstractive summaries. We use the ODP data set, which provides the human-edited summaries for each web page. By comparing our generated summaries with those edited by humans, we can easily evaluate our web-page summarization system with ROUGE scores.

We use OCLET system (Berger and Mittal, 2000) as the baseline which relies on individual words for web-page summarization. The results based on ROGUE-1, ROGUE-2, ROGUE-3, ROGUE-L, ROUGE-W-1.2, ROGUE-SU4 are shown in Table 5.5. Note that we randomly select 396 web-page and summary pairs from the testing subset and the result below are generated from this smaller subset.

	Our System	OCLET System
ROUGE-1	0.09508	0.06258
ROUGE-2	0.01489	0.00699
ROUGE-3	0.00146	0.00058
ROUGE-W	0.06344	0.02547
ROUGE-L	0.08994	0.05190
ROUGE-SU4	0.03111	0.01389

Table 5. 5: Performance of Web-page Summarization Systems

From Table 5.5, we can see that our web-page summarization system achieve higher ROUGE scores than OCLET. This demonstrates that a summarization system based on MWTs has advantages over that based on individual words.

5.4.2 Discussion

Content selection is the first step of our web-page summarization system. As shown in Table 5.4, our content selection part based on MWTs is even more effective than a popular sentence-based text summarization system. Compared to sentences, MWTs can capture more useful information about a document by reducing the redundant information such as stop words. Compared to single words, MWTs can be more precise in representing concepts and have better readability. Given the lack of structure of web pages, we believe that MWTs are suitable information units for web-page summarization.

To evaluate the overall performance of our translation-based method for abstractive web-page summarization, we test our implementation on the ODP data set and compare the results against the word-based OCLET system. Our summarization system can be seen as an extension of OCLET in that we both use machine translation approach for web-page summarization, but instead of using individual words; we rely on MWTs for abstractive web-page summarization. As shown in Table 5.5 our system does significantly better than OCLET for all the ROUGE scores, indicating that MWTs are suitable for web-page summarization for both capturing the key information and maintaining the readability.

Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions

In this thesis we surveyed the area of multi-word term extraction and proposed several new word association measures based on the smoothing of conditional probabilities and the normalization of sequence probabilities. Our experiments on the ODP data set showed that systems with the normalization significantly outperform those without it when used with the LocalMaxs algorithm. The main reason is that the normalization helps shrink the gaps between n-grams of different lengths, making long n-grams more comparable with short n-grams for the glue values. As a result, we can select more meaningful multi-word terms and at the same time reduce the unwanted ones. In addition, the smoothing of conditional probabilities also works well to bring some significant improvements in our experiments. However, due to the limited time and resources, we only did coarse-grained search for optimizing the parameter values. We leave it as future work to explore more effective ways of smoothing n-gram probabilities. In addition, we introduced a post-processing filter to remove stop words at the two ends of multi-word terms, which leads to a huge boost in the performance for all versions of our implementation for multi-word term extraction.

We further applied our approach for multi-word term extraction to web-page summarization and explored the use of multi-word terms as an effective representation for generic web-page summaries. We proposed a new summarization system based on the decoder solution of machine translation that involves content selection of multi-word terms from a web page, alignment with multi-word terms in the summary space, and generation of an optimal order for the aligned multi-word terms for the final summary. Such a process is necessary since a web page may not be well structured and the information may be scattered and by using terms from the summary space, we are able to produce abstractive summaries for the related web pages. We showed in our experiments that multi-word terms are suitable information units that can not only capture meaningful contents but also preserve readability for web-page summarization.

6.2 Future Work for Multi-word Term Extraction

Our experimental results showed that multi-word terms are useful information units for representing documents, especially those that lack of coherent structures. We believe that approaches based on word association measures will have a wide range of applications since they are both domain and language independent. One possible improvement is to find effective ways of optimizing the smoothing parameters for our word association measures. Most word association measures are based on the probabilities of related n-grams. However, due to the sparse data problem, many n-grams are either missing or have very low frequencies in a training dataset. In our experiments, we tried to smooth the n-gram probabilities by the shrinkage method, which combines the probability of a high-order n-gram with those of its low-order n-grams through weighted sums. Although our results showed reasonable improvements for our smoothed methods, we feel that there are still rooms for further tuning or different ways of searching the optimal parameter values in future work.

Another future direction is to keep searching for a better association measure for the glue values of multi-word terms, especially those with more than two words. Although our normalized association measures help identify some useful long multi-word terms, there are still some desirable long terms that are missed in the extraction process. In addition, the interference of stop words is another challenge. If we get rid of stop words, some long multi-word terms will be disconnected. On the other hand, if we keep them, they will certainly add noise for the extraction process. Therefore, how to address stop words appropriately in word association measures is worth more research for multi-word term extraction.

6.3 Future Work for Web-page Summarization

As demonstrated in our experiments, multi-word terms are an effective representation for web-page summarization due to the lack of structures in web pages. With the popularity of new Internet protocols, multi-word terms can also be used for various natural language processing tasks that deal with new forms of documents such as blogs, instant messages, and social network postings.

Most existing work on text summarization focuses on selecting sentences or words or multi-word terms for extractive summaries. In this thesis, we explored the feasibility of generating abstractive summaries based on the decoder solution of machine translation method. In the future, we can search for more effective methods to generate abstractive summaries based on advanced techniques in machine translation.

In addition, we can also do cross-language summarization where the source web page may be in one language but the target summary is in another language, a truly machine-translation application. This is particularly useful for countries like Canada where there are multiple official languages used in daily life. By extending the "alignment step" and changing the multi-word terms in the summary space to another language, we can effectively produce the desired summaries in the other language. Clearly, we will need a bilingual dataset to train our system, but it will be easy to find such a corpus since many official documents are written in both English and French in Canada and other multilingual countries.

References Cited

Ananiadou, S., 1994. A methodology for Automatic Term Recognition. In Proceedings of the 15th International Conference on Computational Linguistics (COLING94), pp. 1034-1038. Kyoto, Japan.

Abracos, J. and Lopes, G. P., 1997. Statistical methods for retrieving most significant paragraphs in newspaper articles. In ACL/EACL Workshop on Intelligent Scalable Text Summarization, pages 51-57.

Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B., 1999. A trainable summarizer with knowledge acquired from robust nlp techniques. In Mani, I. and Maybury, M. T., editors, Advances in Automatic Text Summarization, pages 71-80. MIT Press.

Aires, J., Lopes, G. P., and Silva, J. F., 2008. Efficient multi-word expressions extractor using suffix arrays and related structures. In Proceeding of the 2nd ACM Workshop on Improving Non English Web Searching (Napa Valley, California, USA, October 30 - 30, 2008). iNEWS '08. ACM, New York, NY, 1-8.

Amitay, E. and Paris, C., 2000. Automatically summarizing Web sites: is there a way around it?. *In Proceedings of the Ninth international Conference on information and Knowledge Management* (McLean, Virginia, United States, November 06 - 11, 2000). CIKM '00. ACM, New York, NY, 173-179.

Baxendale, P., 1958. Machine-made index for technical literature - an experiment. IBM Journal of Research Development, 2(4):354-361.

Baum, L. E., 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1-8.

Bourigault, D., 1992. Surface Grammatical Analysis for the Extraction of Terminological Noun Phrases. Proceedings of 14th International Conference on Computational Linguistics, Nantes, France, pp. 977-981.

Brown, P. F., Pietra, V. J., Pietra, S. A., and Mercer, R. L. 1993. The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. 19, 2 (Jun. 1993), 263-311.

Boguraev B., Kennedy C., Bellamy R, Brawer S., Wong Y.Y., & Swartz J., 1998. Dynamic presentation of document content for rapid on-line skimming. In AAAI Spring 1998 Symposium on Intelligent Text Summarization

Berger, A. L. and Mittal, V. O., 2000. OCELOT: a system for summarizing Web-pages. In Proceedings of the 23rd Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Athens, Greece, July 24 - 28, 2000). SIGIR '00. ACM, New York, NY, 144-151.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G., 2005. Learning to rank using gradient descent. In ICML '05: Proceedings of the 22nd international conference on Machine learning, pages 89-96, New York, NY, USA. ACM.

Church, K.W., & Hanks, P., 1989. Word association norms, mutual information and lexicography. Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics, University of British Columbia, Vancouver, Canada, 26-29 June 1989, pp. 76-83.

Carenini, Giuseppe , Raymond T. Ng, Xiaodong Zhou. (1997) Summarizing email conversations with clue words. InProceedings of the 16th international conference on World Wide Web, page 91-100

Carbonell, J. and Goldstein, J., 1998. The use of MMR, diversity-based re-ranking for reordering documents and producing summaries. In Proceedings of SIGIR '98, pages 335-336, New York, NY, USA.

Conroy, J. M. and O'leary, D. P., 2001. Text summarization via hidden markov models. In Proceedings of SIGIR '01, pages 406-407, New York, NY, USA.

Chiang, David, 2005. A hierarchical phrase-based model for statistical machine translation, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, p.263-270, June 25-30, 2005, Ann Arbor, Michigan.

Cruz, Carlos Méndez and Urrea, Alfonso Medina, 2005. Extractive summarization based on word information and sentence position. In Proceeding of the 6th international conference on Computational Linguistics and Intelligent Text Processing. 653-656

Dagan, Ido, Church, Kenneth W. and Gale, William A. 1993. Robust Bilingual Word Alignment for Machine-Aided Translation. In Proceedings, Workshop on Very Large Corpora: Academic and Industrial Perspectives, Columbus, Ohio, 1-8. Association for Computational Linguistics.

Dunning, T., 1993. Accurate Methods for the Statistics of Surprise and Coincidence. In Association for Computational Linguistics, 19[1].

Daille, B., 1995 Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. The Balancing Act Combining Symbolic and Statistical Approaches to Language, MIT Press.

Dias, G., Gilloré, S., Lopes, G., 1999. Language Independent Automatic Acquisition of Rigid Multiword Units from Unrestricted Text corpora. In Proceedings of the TALN’99, p. 333-338. Corse, july 12-17.

Dias, G., S. Guillor, J-C. Bassano, and J.G. Pereira Lopes, 2000. Combining linguistics with statistics for multiword term extraction: A fruitful association? In Proc. of Recherche d'Informations Assistee par Ordinateur 2000 (RIAO'2000).

Delort, J.-Y., B. Bouchon-Meunier, and M. Rifqi. 2003 Enhanced web document summarization using hyperlinks. In Proceedings of the 14th ACM conference on Hypertext and hypermedia, pages 208–215, New York, NY, USA.

Edmundson, H.P., & Wyllys, W., 1961. Automatic abstracting and indexing- Survey and recommendations. Communications of the ACM, 4, 226-234.

Evert, S. and Krenn, B. 2001. Methods for the qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting on Association For Computational Linguistics* (Toulouse, France, July 06 - 11, 2001). Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ, 188-195.

Forney, G.D., Jr, 1973. The viterbi algorithm. Proceedings of the IEEE, Volume: 61 Issue: 3, page(s): 268 – 278.

Frantzi, K, S. Ananiadou, and H. Mima. 2000. Automatic Recognition of Multi-word term: the C-value/NC-value Method. International Journal on Digital Libraries, 3[2]:115–130, August.

Glover, E. J., Tsioutsiouliklis, K., Lawrance, S., Pennock, D. M., and Flake, G. W., 2002 Using Web Structure for Classifying and Describing Web-pages. Proceedings of the 11th International World Wide Web Conference, Honolulu USA 562-569

Grineva, M., Grinev, M., and Lizorkin, D., 2009. Extracting key terms from noisy and multitheme documents. In Proceedings of the 18th international Conference on World Wide Web (Madrid, Spain, April 20 - 24, 2009). WWW '09. ACM, New York, NY, 661-670.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M. & Lin, C., 2000. Question Answering in Webclopedia. In: 9th Text Retrieval Conference.

Justeson, J. S. and Katz, S. M., 1995. Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text, Natural Language Engineering. 1(1):9-27.

Jacquemin, Christian. 1999. Syntagmatic and paradigmatic representations of term variation. In *Proceedings of the 37rd* Annual Meeting of the Association for Computational Linguistics, pages 341.348. College Park, MD, USA, 20-26 June 1999.

Kupiec, J., Pedersen, J., and Chen, F., 1995. A trainable document summarizer. In Proceedings SIGIR '95, pages 68-73, New York, NY, USA.

Katragadda, Rahul, Prasad Pingali , Vasudeva Varma, 2009. Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm, Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies, p.46-52, June 04-04, Boulder, Colorado

Luhn, H. P., 1958. The automatic creation of literature abstracts. IBM Journal of Research Development, 2(2):159-165.

Li, Wentian. 1992. "Random Texts Exhibit Zipf's-Law-Like Word Frequency Distribution". *IEEE Transactions on Information Theory* 38 (6): 1842–1845

Lin, C.-Y., 1999. Training a selection function for extraction. In Proceedings of CIKM '99, pages 55-62, New York, NY, USA.

Lin, C.-Y. and Hovy, E., 2002. Manual and automatic evaluation of summaries. In Proceedings of the ACL-02 Workshop on Automatic Summarization, pages 45-51, Morristown, NJ, USA.

Lin, C.-Y., 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004.

Miller, G. A., R. Beckwith, C. D. Fellbaum, D. Gross, K. Miller. 1990. WordNet: An online lexical database. Int. J. Lexicograph. 3, 4, pp. 235-244.

Mani, I. and Bloedorn, E., 1997. Multi-document summarization by graph search and matching. In AAAI/IAAI, pages 622-628.

Monta, Elena, Irene Daz, Jos Ranilla, Elas F., 2005. Combarro, Javier Fernndez, "Scoring and Selecting Terms for Text Categorization," IEEE Intelligent Systems, vol. 20, no. 3, pp. 40-47.

Mihalcea, R. and Csomai, A., 2007. Wikify!: linking documents to encyclopedic knowledge. In CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 233–242, New York, NY, USA.

Nenadić, G., Spasić, I., and Ananiadou, S., 2002. Automatic discovery of term similarities using pattern mining. In Coling-02 on COMPUTERM 2002: Second international Workshop on Computational Terminology - Volume 14 International Conference On Computational Linguistics. Association for Computational Linguistics, Morristown, NJ, 1-7.

Nenkova, Ani and Vanderwende, Lucy, 2005. Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101

Pal, Santanu and Kumar Naskar, Sudip and Pecina, Pavel and Bandyopadhyay, Sivaji and Way, Andy. 2010. Handling named entities and compound verbs in phrase-based statistical machine translation. In: MWE 2010 - Workshop on Multiword Expressions: from Theory to Applications, 28 August 2010, Beijing, China.

Radev, D. R. and McKeown, K.,1998. Generating natural language summaries from multiple on-line sources. Computational Linguistics, 24(3):469-500.

Radev, D. R., Hovy, E., and McKeown, K., 2002. Introduction to the special issue on summarization. Computational Linguistics., 28(4):399-408.

Radev, D. R., Jing, H., Stys, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. Information Processing and Management 40 (2004), 40:919-938

Salton, G. and Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. 24[5]:513–523.

Smadja, F. 1993. Retrieving collocations from text: Xtract. *Comput. Linguist.* 19, 1 (Mar. 1993), 143-177.

Shimohata, S., 1997. Retrieving Collocations by Co-occurrences and Word Order Constraints. In Proceedings of ACL-EACL’97 (pp 476--481).

Strzalkowski T., Wand J., and Wise B., 1998. A robust practical text summarization. In AAAI 98 Spring Symposium on Intelligent Text Summarization, pages 26-33.

Silva, J.F, G.Dias, S.Guilloré, J.G.P.Lopes. 1999. Using Lo-calMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In P. Barahona, editor, Progress in Artificial Intelligence: 9th Portuguese Conference on AI, EPIA'99, Évora Portugal September 1999, Proceedings. LNAI series, Springer-Verlag, Vol. 1695, p. 113-132.

Sun, J., Shen, D., Zeng, H., Yang, Q., Lu, Y., and Chen, Z. 2005. Web-page summarization using clickthrough data. In *Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Salvador, Brazil, August 15 - 19, 2005). SIGIR '05. ACM, New York, NY, 194-201.

Seretan, V. and Wehrli, E., 2006. Accurate Collocation Extrac-tion Using a Multilingual Parser. Proceedings of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of ACL, Sydney, July 2006. pp.953-960

Svore, K., Vanderwende, L., and Burges, C., 2007. Enhancing single-document summarization by combining RankNet and third-party sources. In Proceedings of the EMNLP-CoNLL, pages 448-457

Witten, I., G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. In Proceedings of the Fourth ACM

Conference on Digital Libraries, pages 254–255, Berkeley, CA, USA, August 11–14.

Witschel, H.F., 2005. Terminology extraction and automatic indexing - comparison and qualitative evaluation of methods. In Proc. of Terminology and Knowledge Engineering [TKE].

Zechner K., 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In Proceedings of the International Conference on Computational Linguistics.

Zhang, Ziqi; Iria, Jos'e; Brewster, Christopher and Ciravegna, Fabio, 2008. *A Comparative Evaluation of Term Recognition Algorithms.* In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08), Marrakech, Morrocco.

Appendix

The Selecting proceeding for Multi-word term

1. Scan through each sentence, separate them into *sense groups*.

A *sense group* is a brief unit of information organized according to grammatical cues, conceptual extendedness, or to semantic similarity. The words in one sense group are strongly related to each other; they are inseparable to forming a piece of independent information together. Briefly speaking, a sense group is a combination of concepts which come from words or phrases in the group. The concept is not a combination of disorder, but according to certain combination relations.

"The government of South Africa said the Zambian President has grossly neglected the incidence of AIDS."

a. The government of South Africa / said the Zambian President / has grossly neglected / the incidence of AIDS /

b. The government of South Africa said / the Zambian President / has grossly neglected / the incidence of AIDS /

The sense units are introduced by grammatical words such as relative pronoun, conjunction and preposition.

General principles are summarized as follows:

1) article + noun, for example: a country

2) noun+ preposition+ noun, for example: the government of South Africa

3) noun + noun, for example: comrade Li

4) demonstrative pronoun + noun, for example: this book

5) adjective + noun, for example: natural science

6) adjective used as noun or participle + noun, for example: New Year’s Day

7) numeral + noun, for example: thirty-two note-books

8) numeral + numeral, for example :223 - two hundred and twenty-three

9) indefinite pronoun + noun, for example: some ink

10) prepositional phrase, verb+ preposition, for example: look at ; break into preposition + noun, for example : from now on; in the bag adjective + preposition, for example: be good at, be satisfy with

11) adverb phrase, example: first of all

12) adverb + prepositional phrase, for example: early in the morning; far into the night

13) adverb + verb, or verb + adverb, for example: quite understand; study hard

14) some fixed phrases of the verb, for example: to take a rest,; to get ready

15) link verb + predicate, for example: be at school; grow quite well

16) subject + predicate, short sentence is taken as one group, for example: He stands up. They are very happy.

17) Subject + predicate + object，short sentence is taken as one group, for example: I can speak English. He gave me a boo

18) Noun clause, including the subject clauses, predicative clauses and object clauses, for example: That he will come here / is certain. (Subject clause) This is / how he studies English. (Predicative clause) He told me / where I could find my book. (Object clause)

19) Short attributive clause, for example: This is a factory / that makes cloth. information.

20) Object complement, for example: Official website / that containing contact information.

21) Short adverbial clause, for example: I waited / till him come back. He can't come / because he is ill. We worked fast / so that we finish our plan.

There may be overlaps between the principles. For example:

“The government of South Africa” →**“**The government/ of South Africa**”**

“The government of South Africa/**”**

We apply the “greedy search” rule when we separate the sense groups.

We take one sense group as a piece of information. If we have more than one ways to separate sense groups from a sentence or a part of a sentence; we take the one which can make the sense groups contain more specific information (usually the longer one).

We take **“**The government of South Africa/**”** as one sense group instead **“**The government/ of South Africa**”**

2. Read through these sense groups we get, select the ones provide "usable (summariable)" information.

If the *sense group* has more than one word and form a intact and independent information, we take the entire sense group as a multi-word term candidate.

For example: "a guide to escorts and services/ in los Angeles California". We take "a guide to escorts and services" as a multi-word term. "in los Angeles California" as another one.

Intact means if we take the sense group from the text, the sense group can still provide a meaningful sense/information/concept.

Independent means the sense group can form a meaningful sense/information/concept without the support from the context.

"The government of South Africa /said the Zambian President /has grossly neglected /the incidence of AIDS."

"has grossly neglected" is not a **intact and independent** sense group.

3. Select the multi-word term from the candidates.

1) Simplify the candidates. (If the candidates become one word after removing the redundant part, we don't take it as a candidate)

a. Remove the article/indefinite/adverb/ numeral/conjunction word if they appear in front of a noun word/phase to modify them.

   For example: "a guide to escorts and services"→"guide to escorts and services"

b. Remove the preposition word if they appear in front of a noun word/phrase.

For example: “in los Angeles California”→”los Angeles California”

c. Remove the adjunct words/phrases if they don’t have high stickiness with the word they modify and they can be replaced by other adjuncts and the candidate will not lose key information without them. Use “Google” to help identify if the adjunct and the word are being modified has a close relationship.

   For example: “a bibliographical listing of other reference sources”, other is an adjunct word that modify “reference sources”. We can replace it with “any, literary….” and after it being removed, we still keep the key information “a listing of reference sources”.

Pay special attention to the structure “adjective+ noun/noun concept”, some adjective can be removed while others should be kept.

For example “recommended cds” recommended is kept.

“abridged dictionary of composers” abridged is removed since it can not provide useful information alone.

Use “Google” to help identify if the adjunct and the word are being modified has a close relationship.

2) Separate the long multi-word term candidate into shorter ones.

We try to keep all the information contained in a multi-word term candidate, but given our goal is to evaluate the multi-word term extracted by auto-system which tend to be short (most of them are bi-gram), we have to separate the long multi-word term into shorter one to get a better understanding how the performance is.

a. If the candidate consists of more than one independent concepts/information, and each of these concepts or information consists of more than 2 words, we separate them into shorter multi-word term candidates.

For example “guide to escorts and services” → “guide to” “escorts and services”

If the candidate only consists of 3 or 4 words, we tend to keep all the words. A typical structure is “noun+ preposition+ noun”. For example “dictionary of composer”

3) Identify the multi-word term from these candidates.

Most of the multi-word term candidates can be taken as multi-word term directly.

For the rest which can not be sure.

a. If the multi-word term candidate is a combination of two or more concepts with “and”. Google it to see its popularity, count it as a multi-word term if it’s often-used. 100,000 is a acceptable number. For example” day and night”

b. If the multi-word term candidate is not a popular term, check the concepts consist of more than two words separately, if they are fix collocation or regular used phrase, keep them as multi-word term.

For example “hot dog and pop” → “hot dog”

c. If the multi-word term candidate is a concept of people’s name or location or organization or specific time period. Google it to see its popularity, count it as a multi-word term if it’s often-used. For example ”tom hanks”

4) Detail the samples

“biographical data , recommended cds , books and sheet music , bibliography , and links to biographical essays from dr . estrella's incredibly abridged dictionary of composers .”

1) Separate it into sense groups.

"biographical data**/** , recommended cds/ , books and sheet music/ , bibliography/ , and links to biographical essays/ from dr . estrella's incredibly abridged dictionary of composers/ ."

2) Select the sense groups.

1) biographical data 2) recommended cds 3) books and sheet music 4) and links to biographical essays 5) from dr . estrella's incredibly abridged dictionary of composers

3) Select the multi-word term

a. simplify the candidates

biographical data → biographical data

recommended cds → (Google **84500 items)**→ recommended cds

books and sheet music → books and sheet music

and links to biographical essays → links to biographical essays

from dr . estrella's incredibly abridged dictionary of composer → dr . estrella's dictionary of composer

b. separate the long candidates

links to biographical essays → links to / biographical essays

dr . estrella's dictionary of composer → dr . estrella's / dictionary of composer

dictionary of composer → dictionary of composer

c. identify multi-word term

recommended cds → recommended cds

books and sheet music→ (Google, not a popular combination) → sheet music

links to → links to

biographical essays → biographical essays

dr . estrella's →(Google, not a popular combination) → dr . estrella's

dictionary of composer → dictionary of composer

So, we get “recommended cds; sheet music; links to; dictionary of composer; biographical essays.