Improving the Syntax-based Retrieval System Using Collocation Indexing

Ruey-Jinng Chen, Chin-Hwa Kuo, Nai-Lung Tsao, and Tsung-Fu Hung
CAN Lab., Dept. of CSIE,
Tamkang University, Taiwan,
695410596@s95.tku.edu.tw, chkuo@mail.tku.edu.tw, beaktsao@mail2000.com.tw,
kidd@mail.iwillnow.org

Abstract-*The purpose of this paper is to design a syntax search system and to apply it to a movie search system. The concepts applied include those in the field of linguistics and collocation, to increase the speed of the syntax search system. First, we must process the keywords in the database by labeling them according to their part of speech. From the results of the process, we will construct a K-gram index and Collocation index. In this proposal we bring out a few examples of common English syntax rules and sentence structures as test models. After the run through, the K-gram index and the Collocation index are compared. We have found that part of the sentence, after having gone through the Collocation index search, has a far smaller sample space that the K-gram index alone, which is to say that the Collocation index is able to find the most correct result from fewer samples, thus minimizing the time cost in Query Match.*

Kerywords：POS tagging, Lemmatizing, Collocation, k-gram, Indexing

1. Introduction

Due to the swift expansion of the Internet, whereas words and text were the main medium of transmitting information, they are replaced with multimedia products. The combination of the multimedia and the Internet represents the future trend and model. In general, when talking about multimedia, people think about images, music, and visual aids, and entertainment. In recent years, research has shown that if multimedia can greatly aid education [1][2]. One of the liveliest discussions involves the combination of movies in the teaching of English and digital learning.

We have designed a system in order to provide teachers and students to search for syntax through movies dialogues. Like regular keyword search systems, we included POS tagging and lemmatizing to each of the keywords in the dialogues. This information is packaged into XML mode, and through the use of Regular expressions, we can easily obtain the information we need. Also, we used K-gram indexing [10][11]13] and Query match to obtain an basic syntax search result.

The most important step in a search process is obtaining the information. A significant topic of research targets how to increase the efficiency of the search process and to increase the accuracy of the results.

Therefore this proposal uses the concepts of

language learning to increase the quality of these parts. Michael McCarthy [4] states that Collocation is a group of words that commonly appear together. Often, this group encounters many restrictions in terms of its application. Collocation needs to calculate the frequency that two words appear together. Whether this type of combination is coincidental or not is determined by the analysis of the database. Most of the grouping of the Collocations is very meaningful. Therefore the degree of the word-to-word relationship is a significant step.

Gledhill [5] also points out three different viewpoints concerning the Collocation: (i) statistical: from a statistical perspective, Collocation have been shown to commonly appear in certain positions [6][7]. (ii) construction, which sees collocation either as a correlation between a lexeme and a lexical-grammatical pattern, or as a relation between a base and its collocative partners (iii) expression, a pragmatic view of collocation as a conventional unit of expression, regardless of form.

The Collocation concept led us to constructing a Collocation index [14]. Through the Collocation index system, we can effectively decrease the number compared in the Query match, which improves the efficiency.

2. System Overview

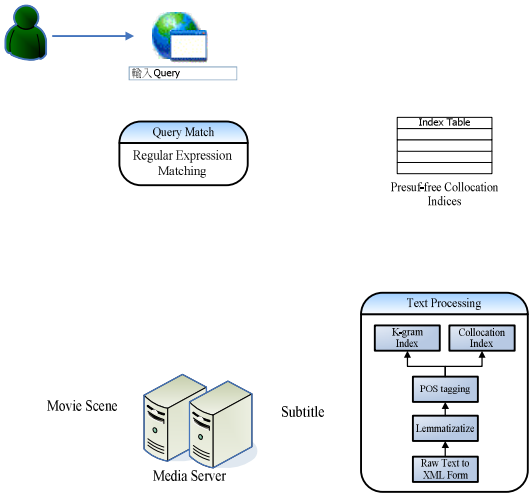

Figure 1. System diagram

The structure of the system is as shown in Figure 1. Media Serve is the system database, and contains the video clips and dialogue. After the dialogue has gone through text processing, it constructs a system query chart. The user can enter the keywords on the interface and the system will look for movie scenes and conversations.

2.1. Text Processing

In this system, text processing can be distinguished into three parts:

(1) Part of speech tagging - the purpose of part of speech tagging is to find the corresponding part of speech for the keyword. This way, the user may search for keywords according to the parts of speech.

(2) Lemmatizing - In this step, the word undergoes a lemmatizing process. For instance, a noun may be changed from plural to singular form, or present participle to present, as in "Paying" becomes "pay". Both part of speech tagging and lemmatizing offer an alternative way to examine the sentence structure.

(3) Change the original movie subtitles to XML form: Before a language search is performed, the movie subtitles must be converted to XML form. XML form includes the information obtained from POS tagging and Lemma and can be compared to Regular Expression.

2.2. Index Construction

An index is constructed for each keyword as the basis of searching, so that within a large amount of subtitles, one can quickly find the most suitable search. This system includes K-gram indexing and Collocation indexing, which will be described in greater detail in the next section.

3. Index Structure

In this part we will describe in detail how the K-gram index and Collocation index are constructed and the algorithms behind the index.

3.1. K-gram indexing

What K-gram means is that the K in the K-gram is exchanged for different kinds of values, so that each keyword is divided into many grams. Then the index terms make up the search. The process is mainly composed of two steps, which are: (i) Multi-gram division (ii) index term search.

(i) A string of words *X*, of length n. A substring, where i is between 1 to n, length *k*, is the K-gram of the string.

(ii) The string is divided through Multi-gram, so there are different lengths of index terms. Therefore these new index term is the original String's index value, as shown in Table 1..

Table 1. Index of index term

Index term	Index term's value
met	material、comet、helmet、method
tor	torrent、sector、torch、motor
wor	word、work、worry、worst
rel	release、rely、sorrel、relax
ease	release、please、ease、cease

3.2. Useful index

After the Multi-gram indexing is constructed, we will find many index terms in the Multi-gram index are not discriminative. Therefore, from the Useful index mentioned by Cho and Rajagopalan [10]'s research and Chen Fu Chang[11]'s research, we can decrease the number of index terms in the Multi-gram index.

Definition 3.1 *x* is any gram (index term) of M . M is a Data unit (word). There are N number of Data Units (word). M(*x*) is the Data Unit variable which includes gram *x*. Filter fact (*ff*) decides whether or not gram *x* should eliminate or retain a filter value. The algorithm is:

$$ff(x) = 1 - \frac{M(x)}{N}$$

From each gram x, we can calculate and select one. When it is greater, this means that the gram (index term) are useful, therefore it must be kept. From this concept, we can eliminate the gram (index terms) that do not matter and thereby reduce the number of index. The remaining gram (index term) are collected and become a "useful index".

Example 3.1 If we set $\min ff = 0.98$, it will filter 98% of data units, and only retain 2% of data units. These index terms are "useful."

3.3. Presuf free set

In order to retain the most useful index terms amidst a large number, we find that each index term that stems from a useful index term is also useful. Therefore we use the concept of presuf (prefix and suffix) free set to solve this problem.

Example 3.2 In the sentence "I am one of the CAN Lab member." CAN is useful, so any gram (index term) that comes from CAN is also useful, such as "e CAN" or "CAN La". Actually, there is no need; if we find this sentence by searching using the index term "CAN", we can also find the same sentence from searching "e Can" or "CAN La". However, these two index terms are unable to find more sentences containing "CAN".

Definition 3.2 Presuf free set: in this index set, an index term that does not exist will be the prefix or suffix of another index term.

3.4. Collocation indexing

In the introduction we often mention that collocation can be seen from a syntax viewpoint, a statistical viewpoint as well. From a syntax perspective, the decision is made from the part of speech. On the other hand, from a statistical perspective, the word-to-word relationship and position makes the decision. The two following steps help use search of the collocations within the sentences: (i) Collocation construct (ii) Collocation filtering.

Collocation Construct：We focus on the Part of Speech

tagging for each term. The basis of this process is from BNC (British National Corpus) [12]. The part of speech definitions are varied in the BNC, so we have simplified them into verbs, nouns, adjectives, adverbs, prepositions, conjunctions, and WH-word. We need to verify the positional relationship between the words. To do this, we created a standard where when the two words are part of the above categories and separated by no more than five words, we call it a Collocation.

Collocation Filtering：After Collocation is constructed through the syntax perspective, many Collocation index terms are produced, but not all of them are commonly used in English. First, for the positional relationship, we use a look-ahead method, that is, if the Pre_word_idx (the index of the preceding word) is greater than the Next_word_idx (the index of the following word), this index terms is eliminated. To determine how useful the collocation is, we still need to see how often it appears in the essay. If it only appears a few times in the entire database , this means that people do not normally use the words in this manner, so it is also eliminated.

Definition3.3 A collocation index term C_k, to be part of a perfect collocation set, must fulfill the following two requirements:

1. Pre_word_idx < Next_word_idx
2. $Freq\ (C_k) > T$

When the Collocation is greater than the filter value *T*, it becomes an index-term. These index terms are not only helpful to language learning, but more importantly, it can decrease the number of index terms that need to be compared in regular expression, therefore increasing the efficiency.

3.5. Term Search System

In order for the user to quickly search for syntax within a large collection of dialogue, we use the K-gram indexing mechanism, then decrease the index number through presuf set and useful index, as shown Figure 2. Then the Collocation index adds the filtered part of the presuf set and useful, so that the system's index becomes more complete.

From 3.1, 3.2, 3.3 we can obtain the following:

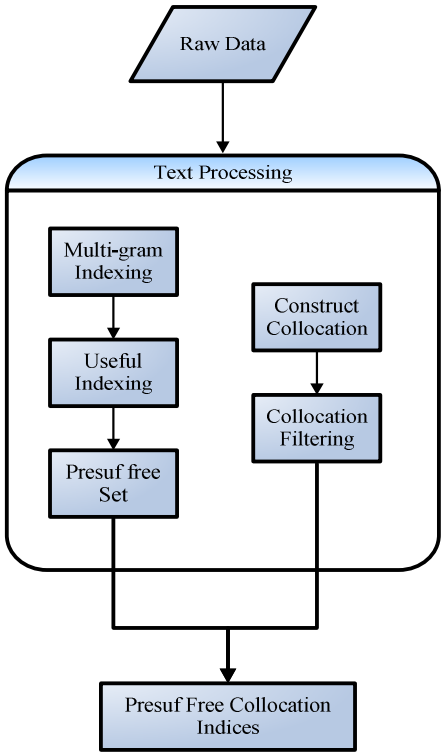

Figure 2. Architecture of subsystem

```
Input：text collection
Output：index

[1]  k=1，Useless={.} // . is a zero-length string
[2]  while (Useless is not empty)
[3]   k-grams：＝ all k-grams in text collection
            whose (k-1)-prefix∈Useless
                or (k-1)-suffix∈Useless
[4]   Useless：＝{}
[5]   For each x in k-grams
[6]        If ff(x) ≥ minff  Then
[7]            insert(x,index) //the gram is useful
[8]        Else
[9]            Useless：＝Useless∪{x}
[10]  k：＝k+1
```

Figure 3. K-gram indexing algorithm

```
Input：Sentence collection
Output：index

[11]   Collocation = {.}
[12]   For each word in Sentence
                  //construct collocation
[13]  If (Pre_word_idx - Next_word_idx) < 1
[14]       (Pre_word + Next_word) ∈ Collocation
[15]   For each y in Collocation
[16]       If (Frequency(y) < min_threshold)
[17]            Delete y from Collocation
                  //the collocation y is not significant
```

Figure 4. Collocation indexing algorithm

4. Experiment

The syntax search system in this paper is a web-based system, which means the browser is used both to search and to display the search results. Figure 5 shows a screenshot of the interface:

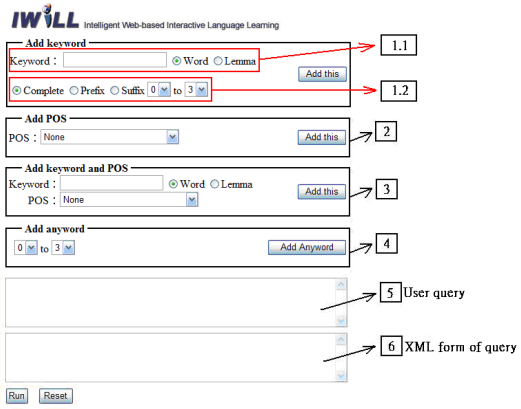

Figure 5. The designed user interface of query generator

From Figure 5, the Add Keyword portion can be divided into two parts. In Figure 5-(1.1), the user can enter a Keyword under "Keyword", and then select Word or Lemma method. If the user chooses word, the Prefix, Suffix, and Complete options are also available (Figure 5-(1.2)). If the user chooses prefix, the Suffix of the search results, and the two values can control the preceding character area of the Keyword. Lemma will return all the types of sentences from that word back to the user. After confirming the search requirements, the user clicks "Add this", and the search keyword will be converted to ser search syntax and regular expression, as shown in Figure 5-5 and Figure 5-6.

Add POS (Figure 5-2) is based primarily on Parts of Speech search. We provided 11 kinds of common parts of speech for the user's selection. In addition, "Add Keyword" is available. For example, keep + V-ing shows how these two go together.

Add Keyword and POS portions (Figure 5-3) , the Keyword typed by the user can not only select Word or Lemma, but also select the part of speech of the keyword.

Add Anyword (Figure 5-4)'s main function, is that when the user hopes to find how words work together, this collocation often has a great distance from word-to-word, rather than following one after another. For instance, "Add keyword" and "Add anyword" can be applied to the words "do… favor" to obtain this effect.

4.1. Experiment and Discussion

This paper mainly researches on the collocation indexing technology and its ability to improve the original system's functions. Also we increased the number of search results. To further this discussion, we used 119 movies, 161,900 sentences from dialogues as our training data.

Through our training data process, we use 15 sentences that are commonly used. Figure 6 shows the sentences that we tested in Regular Expression.

1. used to
   <[^(<|>)]+>used</w><[^(<|>)]+>to</w>
2. (could | would | should) + have + p.p
   <[^(<|>)]+>\w{1,2}ould</w><[^(<|>)]+>have</w><[^>]*POS="VVN"[^<]*>[^(<|>)]+</w>
3. approach(es | ing) + to + Ving
   <[^(<|>)]+>approach\w{0,2}</w><[^(<|>)]+>to</w><[^>]*POS="VVG"[^<]*>[^(<|>)]+</w>
4. of + WH-Word
   <[^(<|>)]+>of</w><[^>]*POS="(AVQ|DTQ|PNQ

lAVQ-CJS)"[^<]*>[^(<l>)]+</w>

5. on + the + other + hand
<[^(<l>)]+>on</w><[^(<l>)]+>the</w><[^(<l>)]+>other</w><[^(<l>)]+>hand</w>
6. to + Ving
<[^>]*Lemma="to"[^<]*>[^(<l>)]+</w><[^>]*POS="VVG"[^<]*>[^(<l>)]+</w>
7. no + matter
<[^>]*Lemma="no"[^<]*>[^(<l>)]+</w><[^>]*Lemma="matter"[^<]*>[^(<l>)]+</w>
8. must + have + p.p
<[^(<l>)]+>on</w><[^(<l>)]+>the</w><[^(<l>)]+>other</w><[^(<l>)]+>hand</w>
9. so + Adj + that
<[^(<l>)]+>so</w><[^>]*POS="AJ0"[^<]*>[^(<l>)]+</w><[^(<l>)]+>that</w>
10. so + Adj + as to
<[^(<l>)]+>so</w><[^>]*POS="AJ0"[^<]*>[^(<l>)]+</w><[^(<l>)]+>as</w><[^(<l>)]+>to</w>
11. At + the + (sight l thought)+ of
<[^(<l>)]+>at</w><[^(<l>)]+>the</w><[^(<l>)]+>\w{2,4}ght</w><[^(<l>)]+>of</w>
12. as + <AnyWord>{0,3} + say*{0,4}
<[^(<l>)]+>as</w>(<[^>]+>[^>]+</w>){0,3}<[^(<l>)]+>say\w{0,4}</w>
13. Once + <AnyWord>{0,5} + -s form verb
<[^>]*Lemma="Once"[^<]*>[^(<l>)]+</w>(<[^>]+>[^>]+</w>){0,5}<[^>]*POS="VVZ"[^<]*>[^(<l>)]+</w>
14. if + only
<[^>]*Lemma="if"[^<]*>[^(<l>)]+</w><[^>]*Lemma="only"[^<]*>[^(<l>)]+</w>
*>[^(<l>)]+</w><[^(<l>)]+>as</w>
15. as + Adv + as
<[^(<l>)]+>as</w><[^>]*POS="AV0"[^<]*>[^(<l>)]+</w><[^(<l>)]+>as</w>

Figure 6. 15 sentences and its regular expression

Table 2 shows the results from our experiment shown in Figure 6. In particular, the Candidate column is added separately to the number of sentences before and after it is paired with the collocation index. Figure 7 is the statistical results of Table 2. The Final column represents the final number after Candidate and Query. From looking at Candidate (with collocation index) in Figure 7, it appears to be less than Candidate (without collocation index).

After adding the filtering collocation indexing, it can effectively decrease the number of times needed for comparison during regular expression, and can increase the speed for searching. This can improve the effectiveness of language search, which is the main contribution of this paper.

Table 4.1 The different from (with / without) Collocation index

Query	without collocation		with collocation	
	Candidate	Final	Candidate	Final
1	3410	120	90	120
2	21190	119	21190	119
3	520	0	520	0
4	161900	0	61480	35
5	8160	8	8160	8
6	161900	0	99360	18
7	10200	64	10200	64
8	8580	39	270	39
9	92450	9	14620	9
10	111420	0	920	3
11	58720	3	15480	3
12	27420	31	530	31
13	7030	2	1450	2
14	10960	6	100	6
15	14400	49	4270	49

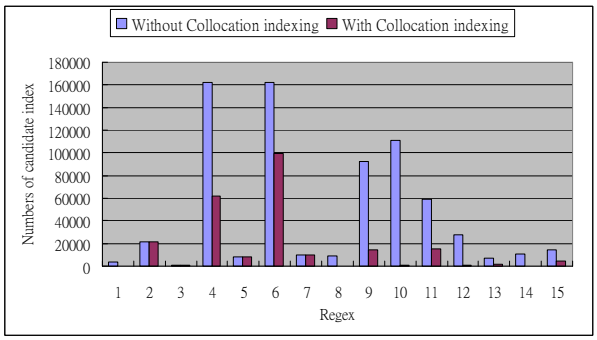

Figure 7. Total number of sentences between (with/without) Collocation index

5. Conclusion

This paper mainly explores whether or not the collocation indexing method can improve K-gram indexing search, and increase the number of search results. In the experiments, we find that the assistance of the collocation index enables sentences that could not originally be captured it to be easily found, as shown in Fig. 4.2. Therefore, collocation index's assistance greatly improves the retrieval system.

6. Acknowledgment

The work described in this paper was partially supported by the grants from the National Science Council, Taiwan (Project No. NSC 95-2520-S-032-004-MY3).

Reference

[1] Jane King, "Using DVD Feature Films in the EFL Classroom, " ,*Computer Assisted Language Learning*, Vol. 15, No. 5, pp 509-523, 2002.

[2] Erwin Tschirner, "Language Acquisition in the Classroom: The Role of Digital Video," *Computer Assisted Language Learning*, Vol. 14, No. 3-4, 2001.

[3] Thorsten.Brants, TnT-A Statistical Part-of-Speech Tagger. In Proceedings of the Sixth Applied Natrual Language Processing *Conference ANLP-2000*, Seatle,WA, 2000.

[4] McCarthy, Michael and O'Dell, Felicity, *English Collocations in Use : how words work together for fluent and natural English*. Cambridge University Press, 2005.

[5] Gledhill C., *Collocations in Science Writing*, Narr, Tübingen, 2000.

[6] Sinclair J. *The Search for Units of Meaning*, in Textus, IX, 75-106, 1996

[7] Smadja F. A & McKeown, K. R. "Automatically extracting and representing collocations for language generation", *Proceedings of ACL* 90, 252-259, Pittsburgh, Pennsylvania, 1990

[8] Moon R. Fixed Expressions and Idioms, a Corpus-Based Approach. Oxford, Oxford University Press, 1998

[9] Frath P. & Gledhill C. "Free-Range Clusters or Frozen Chunks? Reference as a Defining Criterion for Linguistic Units," In *Recherches anglaises et Nord-americaines*, vol.38：25-43, 2005

[10]Cho, Junghoo and Sridhar Rajagopalan, "A Fast Regular Expression Indexing Engine." In *Proceedings of 18th IEEE Conference on Data Engineering*, 2002

[11] Chin-Hwa Kuo, David Wible, Nai-Lung Tsao, and Chen-Fu Chang, "A Video Retrieval System for Computer Assisted Language Learning," *AI-ED 2005*, July 18-22, 2005.

[12] BNC http://www.natcorp.ox.ac.uk/.

[13] Ryoichi Ando, Koichi Shinoda, Sadaoki Furui, Takahiro Mochizuki "Robust Scene Recognition Using Language Models for Scene Contexts" In *Proceedings of the 8th ACM international workshop on Multimedia Information Retrieval*, 2006

[14]Petrovic, S. Snajder, J. Dalbelo-Basic, B. Kolar, M. "Comparison of Collocation Extraction Measures for Document Indexing" *Information Technology Interfaces ITI 2006*, June 19-22, 2006