Asian Journal of Information Technology 5 (1) : 61-64, 2006

Phrase Based Approach for Document Representation

[1]S. Srinivasan [2]P. Thambidurai
[1]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology,Chennai – 600 119, [2]Department of Computer Science and Engineering and Information Technology, India Pondicherry Engineering College, Pondicherry – 605 014, India

Abstract: Many systematic approaches have been studied to represent documents. One of the approaches, called Phrase based Technique (PHT) uses phrases to represent documents, aiming at capturing the main phrases present in the document. A set of phrases is represented as an ATN. One of the main problems in this approach is to construct ATN that can capture all possible patterns. This study provides a frame work that is essential for capturing the most of the patterns used in English Language and proposes a way to automatically represent documents. Experiments have been performed on small set of documents and shown that phrases are more effective than keywords in terms of content indicators.

Key words: Phrases, Augmented Transmission Network (ATN), recall, precision, rank, syntax analyser

INTRODUCTION

The extensive and widespread use of World Wide Web raises the need for devising Effective Information Retrieval Techniques. Most of the search engines for web search use syntactical evaluation of a user query. We think that the text retrieval approaches should be user and goal oriented. In such cases, network traffic due to flood of irrelevant material could be avoided. In this study we have proposed a Phrase based Approach for representing text documents. Phrases can be considered as good content indicators that are very useful for document classification. One of the important studies to integrate information retrieval and artificial intelligence techniques is PHT which uses phrases to define measures for document retrieval.

However, PHT still has some drawbacks when used for real world applications. In this study we point out at some problems in PHT. One problem in PHT is about acquisition of knowledge. Users may pose queries without having to include phrases.

The rest of the study is organized as follows: In section 2 we have discussed some ongoing work in the area of information retrieval. Section 3 gives the system architecture. Here ATNs for handling Noun Phrases (NP), Prepositional Phrases (PP), Adjective Phrases (AP) and Verb Phrases (VP) are given. Ranking method is also discussed. Section 4 gives the conclusion and suggests directions for future work.

Related work: A set of extracted words could be considered to be good document representative. Certain amount of evidence was shown as effective in Barber[1]. Luhn[2] used sequence counts of words in the document text to determine which words were sufficiently significant to represent or characterize the document in the system. The use of statistical information about distributions of words in documents was further exploited by Maron and Kuhns[3] who obtained statistical association between keywords. These associations provided a basis for the construction of a thesaurus as an aid to retrieval. Spark Jones[4] has carried on this work using measures of association between keywords based on their frequency of co-occurrence. She has shown that this measure can be effectively used to improve the recall. Similar work is carried out for effectively representing and retrieving multimedia data types. Golshani[5] has given a method for multimedia data retrieval.

System architecture

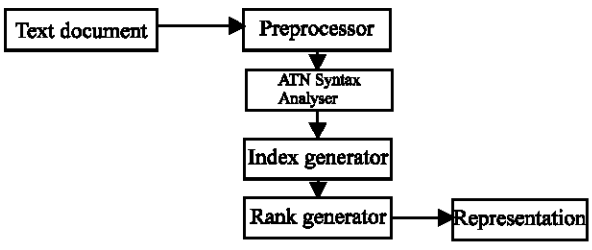

Fig. 1: System architecture

Corresponding Author: S. Srinivasan, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai – 600 119, India

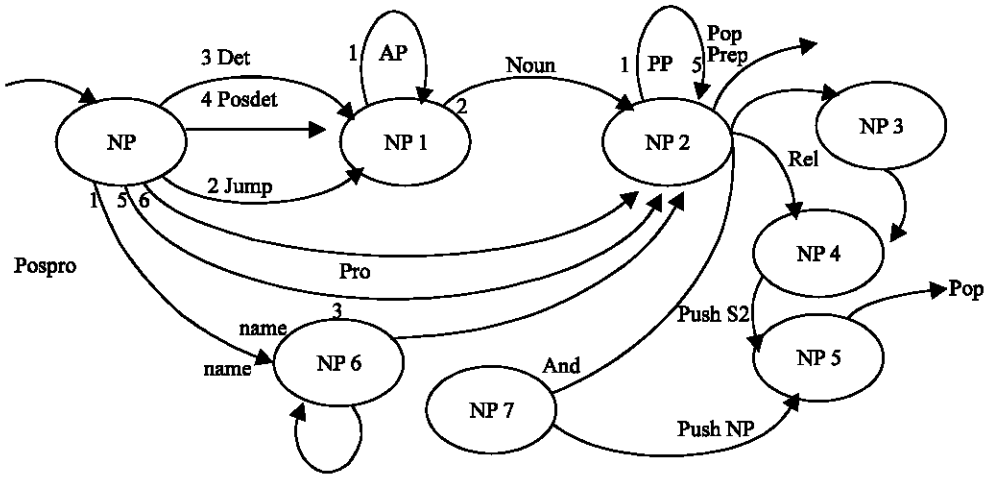

Fig. 2: ATN for NP

Table 1: Action Table

ARC	TEST	ACTION	REGISTER
NP/1	None	AddNAME(*)	
			NP.NAME <== AddNAME
			NP.Num <== Num(*)
NP/3	None	NP.Type <== Type(*)	NP.DET <== *
			NP.Num <== Num(*)
NP/4	Check if next word	becomes Possessive	NP.POSDET <== *
	is noun ignoring	Determiner so skip	NP.Num <== Num(*)
	Adjectives TRUE*	POSPRO Parse	
	FALSE*	becomes Possessive	NP.POSPRO <== *
		Pronoun	NP.Num <== Num(*)
NP/5	None		NP.PRO <== *
			NP.Num <== Num(*)
NP/6	None		NP.POSPRO <== *
NP1/1	Check vowel	Remember Last	
	match with	Adjective	NP.ADJ <== *
	respect to NP.Type		
	FALSE	Display Message	
NP1/2Fails	If Last Adjective	Make Last Adjective as	NP.MAIN <== LastAdj(ADJ)
	and Last Adjective	main noun.Remove last	NP.REF <== Ref(ADJ)
	is noun	adjective from ADJ	
	TRUE	Check Agreement between	NP.Num <== Agreement()
		NP.Num and *.Num	
FALSE 1		Parse FAILED!	
		Return NULL	
NP1/2	Check Vowel if ADJ	Check Agreement between	NP.MAIN <== *
Success	is NULL	NP.Num and *.Num	NP.REF <== Ref(*)
	FALSE	Display Message	NP.Num <== Agreement()
NP2/1			NP.PP <== *
if FLAG			
NP2/2	If next word is		
if FLAG	Relative	Set code = 0	
NP2/3		Set code = 1	
if FLAG			
NP4/1	If code = 0 TRUE	PUSH S(1) : S1	NP.EMBED <== *
	FALSE	PUSH S(2) : S2	
NP6/1		Concatenate NP.NAME	
NP5/1		Parsing Successful	
		Return NP	
NP6/1	If the next word is a	Skip NAME Parsing	NP.ADJ <== *
	noun, and the last		
	NAME is an Adjective		
	TRUE		
	FALSE	AddNAME (*)	NP.NAME <== *
NP7/1		PUSH NP:NP	NP.ADD <== *

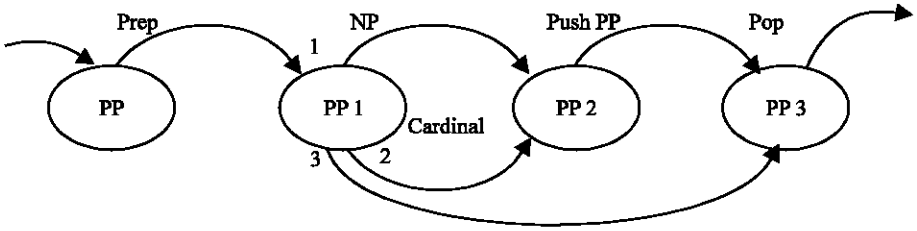

Fig. 3: ATN for PP

`Table 2: Action Table

ARC	TEST	ACTION	REGISTERS
PP/1			PP.PREP <== *
			PP.REF <== Ref(*)
			PP1/1PP.PNP <== *
PP1/2	Check Numeral	Get Numeral	PP.CARDINAL <==
		numeral	
PP1/3	Check Verb	Fail Parse/Set Error	
PP2/1			PUSH PP1 PP.NEXT <== *
PP3/1		Return PP/Error	

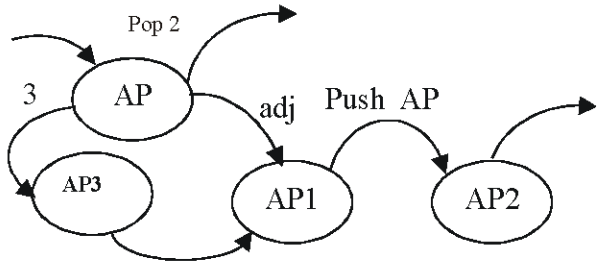

Fig. 4: ATN for AP

System architecture is given in Fig. 1. Preprocessor removes special characters like comma, semicolon and splits the text into sentences. ATN Syntax Analyser analyses the text and checks the validity or invalidity of individual sentences. It also extracts the noun phrases, verb phrases, prepositional phrases, and forms them according to the ATN given in section 3.1.

Noun Phrase NP: An ATN for handling NP is given in Fig.2.

Prepositional Phrase PP: An ATN for handling PP is given in Fig. 3.

Table 3: Action Table

Arc	Test	Action
Registers		
AP/1	None	AppendAP(*)
AP.ADJ <== *		
AP.REF <== Ref(*)		
AP/3		Remember
Question		
AP3/1	Check Next Word	
AP.ADJ <== "<Questioned>"	for a Noun	
AP.REF<== Ref(ADJ)		
AP/2	None	Return AP
AP1/1	None	PUSH AP
AP.NEXTADJ <== *		
AP2/1	None	Return AP

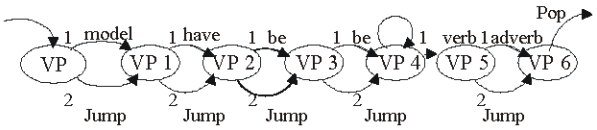

Fig. 5: ATN for VP

Adjective Phrase AP: An ATN for handling AP is given in Fig. 4.

Verb Phrase VP: An ATN for handling Verb Phrase VP is given in Fig. 5.
The Action Table is given in Table 4.

Index generator: Index Generator produces index terms, which are derived from the text of the document to be described or may be derived independently. Index Generator generates index terms for the given document. It is formed in terms of various phrases viz., NP, VP, PP and AP.

Rank generator: The two most important factors governing the effectiveness of the indexing namely exhaustive indexing and specificity indexing are considered. It is used to increase the recall and precision.

If p_1 is probability of a random document belonging to one of the subsets and x1, x2 are the mean occurrences in the two classes, and then statistical behavior of content – bearing word over two classes is given by

$$f(n) = \frac{p_1 e^{-x1} x_1^n}{n_!} + \frac{(1-p_1) e^{-x2} x_2^n}{n!}$$

The ratio

$$\frac{p_1 e^{-x1} x_1^k}{p_1 e^{-x1} x_1^k + (1-p_1) e^{-x2} x_2^k}$$

gives the probability that the particular document belongs to the class which treats w to an average extent of x1 given that it contains exactly k occurrences of w.

Rank Generator examines the distribution of index terms in the collection and assign ranks according to their frequency.

Table 4: Action Table

ARC	TEST	ACTIONS	REGISTERS
VP/1	None	Set Modal Flag	VP.MODAL <== *
VP1/1	If Modal Flag is set and Infinite Form of Verb -TRUE	Set Aux Flag	VP.AUX <== AppendAux(*) V.P.Num <== AuxAgreeCheck()
	FALSE	Verb Phrasing Error"	
	If Aux Flag not set, perform.	Set Aux Flag	VP.AUX <== AppendAux(*)
VP2/1	the the same test TRUE		VP.Num <== AuxAgreeCheck()
VP3/1	FALSE		Verb Phrasing Error"
	If Aux Flag is set, Check for verb form to be Infinite TRUE		Return Verb Phrasing Error
	FALSE	Set Aux Flag	VP.AUX <== AppendAux(*) VP.Num <== AuxAgreeCheck()
VP4/1	If Aux Flag is set, check for * to be a verb TRUE	MAIN-V <== *	VP.MAIN-V <== * (Last(AuxList) (Last(AuxList)
		Check agreement between VP.Num and *.Num	VP.REF<==Ref(*) VP.Num <== AuxAgreeCheck()
	FALSE	MAIN-V<== Last(AuxList) Check Agreement between VP.Num and MAIN-V.Num	VP.MAIN-V<== Last(AuxList) VP.REF<== VP.Num <== AuxAgreeCheck() VP.MAIN-V <== *
	If Aux Flag not set, check for * to be a verb TRUE	MAIN-V <== * Check Agreement between	VP.REF<==Ref(*) VP.Num and MAIN_V.Num VP.Num<==
AuxAgreeCheck()			
	FALSE	Return "Verb Phrasing Error"	
VP4/2		Adv <== *	VP.ADV <== Adv
VP5/1	None	Adv <== *	VP.ADV <== *
VP6/1	NoneReturn VP		

CONCLUSIONS

Phrase identification is an important task of text representation. Most retrieval techniques are designed and based on keywords, and its occurrence. But our PHT approach emphasizes on phrase occurrence in the document. So, research on document representation and its retrieval is a meaningful work in reality.

Future work includes presenting similar works and emphasis on contents. Since text documents get updated every day in many applications, formation of various phrases is also a practically important task.

REFERENCES

1. Barber, A.S., E.D. Barraclough and D.A. Gray, 1973. On-line information retrieval as a scientist's tool'. Information Storage and Retrieval, 9: 429-44.
2. Luhn, H.P., 1957. A statistical approach to mechanized encoding searching and library information. IBM J. Res. Develo, 1: 309-317.
3. Maron, M.E. and J.L. Kuhns, 1960. On relevance, probabilistic indexing and information retrieval', Journal of the ACM, 7: 216-244.
4. Sparck J.K.,1971. Automatic keyword classification for Information Retrieval, Butterworths, London.
5. Golshani, F. and N. Dimitrova, 1998. Language for content Based Video Retrieval. Multimedia tool and Applications, 6:289-312