International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-1 Issue-9 March-2013

Extracting Person Name, Date and Place from Text Documents Using LEX Tool

Roohi Sharma

Abstract

This paper contains the details of how one can extract person name, date and place from a text document using finite state automata and LEX tool. If we search a text document for some important information manually, the process is slow, tedious and error prone. The regular expressions are used to parse textual data to match patterns and extract variables. The lexical analyzer is used in this research, which scans the input program character by character and groups them together to form tokens. This paper describes a technique to perform identification and extraction of information by using LEX tool. It finds the names, date and places that appear in machine-readable text document. Regular expressions through which required information is extracted are also discussed.

Keywords

Regular Expressions, Finite State Automata, Information Extraction, Pattern Matching, Lexical Analyzer.

1. Introduction

Before the beginning of the computer and the internet, searching was time consuming and tedious. History research consists primarily of searching for source documents and extracting relevant information and was considered boring [1]. We are now into an era where searching for documents is becoming easier. With the rapid growth in the number of electronic resources over the internet, automatic mining data from the web or retrieving information across the Internet becomes even more important than ever. Driven by the need to search required information, Natural Language Processing (NLP) and Information Extraction (IE) have been used. We use finite state automata in NLP and IE because of its speed and compactness [2].

FSA provides efficient and convenient tools to represent the linguistic phenomena. The study of theory of computations is focused on answering fundamental questions about what can be computed and what amount of resources are required to perform those computations. Information Extraction (IE) consists in automatically filling databases out of texts written in natural languages. Two activities can be discriminated,

- Extracting information in the text.
- Extracting information about the text.

Finite State Automata (FSA)
Finite State Automata is a mathematical model which has following properties [3],

- Finite State System has some inputs and outputs.
- The system should always be in one of its Finite State.
- One state is start state and one or more states are the final states.

Finite state automat has two types: DFA and NDFA. In DFA, for each state, and for each symbol of its input alphabet exactly one edge with that symbol leaving that state. In NDFA, there is no restriction on the labels of their edges. But deterministic and non-deterministic finite automata are capable of recognizing the same languages.

Regular Expressions (RE)
An expression that defines simple string patterns is called regular expression. The use of regular expression for text searching is widely and well understood [4]. The following are some regular expressions over alphabet A = {a, b}, and the corresponding regular languages.

Table 1: Languages and their Regular expressions [8]

Language	Regular Expression
{a}	a
{a, b}	a+b
{a, aa, aaa,……..}	a^+
{^, a, aa, aaa,…..}	a^*

R is a regular expression if it is,

Roohi Sharma, IT, MIET College, Jammu, India.

- For some a in the alphabet å, standing for the language {a}.
- ε, standing for the language {ε}.
- Ø, standing for the empty language.
- If r1 and r2 are regular expressions then,
- r1+r2 represent the union of two regular expressions.
- r1r2 represent the concatenation of two regular expressions.
- r* signifies closure of a regular expression.
- (r) signifies parenthesis of a regular expression.

Lexical Analyzer

In computer science, lexical analysis is the process of converting a sequence of characters into a sequence of tokens. A program or function which performs lexical analysis is called a lexical analyzer. The problem that lexical analysis must solve is of transforming a linear character stream into a data item with a richer structure: a string of words, a record structure, the abstract syntax tree for a program, etc. There are various variables and functions used in LEX. By using these variables and functions, one can create its own program in LEX.

Table 2: Lex variables and Lex functions [1]

Lex variables & Lex functions	Description
yyin	Type: FILE*. This point to the current file being parsed by the lexer.
yyout	Type: FILE*. This point to the location where output of the lexer will write. By default, both yyin and yyout point to standard input and output.
yytext	The text of the matched pattern is stored in this variable (char*).
yyleng	Gives the length of the matched pattern.
yylineno	Provides current line number information.
yylex()	The function that starts the analysis. It is automatically generated by Lex.
yywrap()	This function is called when end of file is encountered. This function returns 1, the parsing stops. So, this can be used to parse multiple files.
yymore()	This function tells the lexer to append the next token to the current token.

LEX Tool

Lex is a tool for automatically generating a lexer or scanner given a lex specification (.l file). The input notation lex tool used is referred to as lex language. A lexer or scanner is used to perform lexical analysis or the breaking up of an input stream into meaningful units, or tokens.

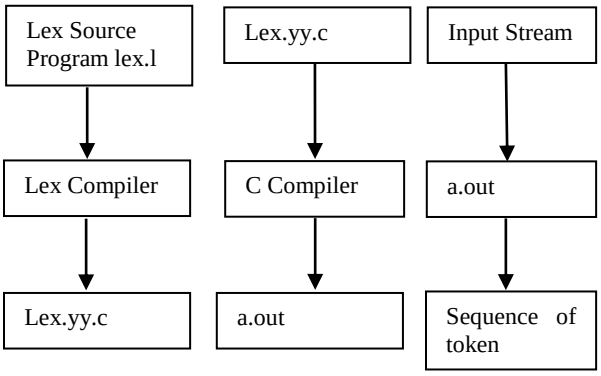

Figure 1: Creating a lexical analyzer with Lex [5]

The Figure 1 suggests how Lex is used. The lex compiler transforms the input patterns into a transition diagram and generates code, in a file called lex.yy.c, which simulates this transition diagram. The C-compiler output, a.out is the subroutine of parser.

2. Research Question

Even though we may be able to quickly find unstructured sources, it still takes time to extract relevant information. Inspecting the document for information requires that we read the entire document. It could take up to an hour.

We may read many paragraphs containing no direct information on dates or names. It is also possible, that as we read a lengthy document our mind wander and we may miss important information.

3. Evaluation

In the first step one we convert any text document in any format into plain text. Step two performs lexical analysis. It converts a sequence of characters into a sequence of tokens with appropriate token type. It does this in two phases. First it uses a finite state machine to separate the sequence of characters into a sequence of tokens. This is done quickly.

In the second phase we categorize every token as to whether it is a given name, city, country, state, or

none-of-the-above. Step three uses a grammar to parse the tokens and extract complete names, dates and place.

4. Work Done

Regular expression used in Lex tool for extracting name, date and place from the file as follow,

- ([0-2] [0-9] | [3] [0-1]) / \ ((0 (1|3|5|7|8)) | (10|12)) / \ ([1-2] [0-9] [0-9] [0-9]) :

This regular expression is for the extraction of a month contains 31 days. It will check for the date first, when the look ahead '/' symbol comes it start reading the next character. Next sub-expression is for checking month. The sub-expression ((0(1|3|5|7|8)) | (10|12) is for the months having 31 days, like 01(January) or 03(March), so on. The next sub-expression is for checking the year.

- ([0-2] [0-9] | 30) / \ ((0(4|6|9)) | 11) / \ ([1-2] [0-9] [0-9] [0-9]):

This regular expression is for the extracting the information of the months which have 30 days. The first sub-expression is for checking the dates. The day contain number only upto 30. After the look ahead '/' symbol, next character will be read. The next sub-expression ((0(4|6|8)) | 11) is for months contains 30 days i.e, 04(April) or 06(June), so on. The next character read is for checking the year. The year will be in between [1000-2999].

- ([0-1] [0-9] | 2[0-8]) / \ 02 / \ ([1-2] [0-9] [0-9] [0-9]):

The above two regular expressions are for the month of 31 and 30 days. The next regular expression is for the month of February, having 28 or 29 days. This regular expression is for the 28 days month. The date will be anything among [01-19] and [20-28]. The last sub-expression is for the year.

- 29 / \ 02 / \ ([1-2] [0-9] [0-9] [0-9]):

This regular expression is only for leap year. The only month having 29 days is February. It first checks for the date, then month and at last year.

- (yr % 4 = = 0 || (yr % 100 = = 0 && yr % 400 != 0)):

This regular expression will check whether the date extracted in above regular expression is a leap year or not.

- [keyword] {letter (letter | digit) *}

This regular is used for extracting person name or place. The name will be a letter followed by another letter or digit.

Table 3: Result

Content in input file (ppd.l)	Output
12/12/2011	It is a valid date.
32/10/2009	It is not a valid date.
29/02/2008	It is a valid date.
31/04/1990	It is not a valid date.
29/02/2001	It is not a valid date.

The Figure 3, show whether the date given in a document is valid or not after extracting date from the given document.

5. Conclusion

Regular expressions are being used for most of the popular web sites and only 4% of the regular expressions are unique [5]. LEX tool is used for generating lexical analyzer. This paper presents regular expressions that are created in LEX tool, used for extracting information person name, place and date from a document. Lex tool generate a C program which contains the specification specified by the lex program. Following are the future directions in which work can be carried out:

- Use of regular expression for identification of information using lex tool in other natural languages.
- Lex output can be integrated with YACC for checking whether the grammar used in the language is correct or not.

Table 4: Comparison

Previous Approach	New Approach
1. Till now the methods used was the guessing from surnames. 2. No regular expressions had been made for extracting any kind of data. 3. Previous methods use a large corpus of tagged documents for statistical training.	1. No such guessing is involved in this new approach. Database has been created. 2. Regular expressions have been made for extracting person name, place and date. 3. In this new approach no such tagged documents are needed.

Acknowledgement

I wish to thank the referee for the careful reading of the paper and many valuable suggestions including recent references.

References

[1] M. Upadhyaya, "Simple Calculator Compiler Using Lex and YACC," International Conference on Electronics and Computer Technology, pp. 182-187, 2011.

[2] Y. Roman, G. Ralph, "Transforming Examples into Patterns for Information Extraction," Proceeding of a workshop TIPSTER, pp. 97-103, 1998.

[3] A. V. Aho, M. S. Lam, R. Sethi, J.D. Ullman, "Lexical Analysis" in Compilers Principles, Techniques and Tools, Pearson Education and Dorling Kindersley Publishers, pp.109-155, 2008.

[4] Clarke. A. L. Charles, C. V. Gordon, "On the use of Regular Expressions for Searching text," Journal ACM Transactions on Programming Languages and Systems, Vol. 19, pp. 413-426, 1997.

[5] S. S. Sane, "Theory of Computer Science," Technical Publications, pp. 88-135, 2002.

[6] Y. S. Han and D. Wood, "Obtaining Shorter Regular Expressions from Finite-State Automata," Proceeding of the Conference on Theoretical Computer Science, pp. 110-120, 2006.

[7] H. Larkin, "Object Oriented Regular Expressions," 8th International Conference on Digital Object Identifier, pp. 491-496, 2008.

[8] Z. S. Bing, Y. Chunfa, "Person Name Identification in Chinese Document Using Finite State Automata," Proceeding of IEEE International Conference on Intelligent Agent Technology, pp. 478-481, 2003.

[9] C. A. Chahine, N. Chaiguard, J. Kotowicz, J. Peeuchat, "Context and Keyword Extraction in Plain Text using a Graph Representation," Proceeding of IEEE International Conference on Signal Image Technology and Internet based Systems, pp. 692-696, 2008.

Roohi Sharma, received her M.Tech degree in Computer Science and Application from Thapar University Patiala, India. Currently she works as a Lecturer in MIET College Jammu. Her research interests include Network Management and Security. She published one paper in the Conference.