Machine readable grammar for optimizing automatic retrieval in text corpus

Prihantoro Prihantoro

MACHINE READABLE GRAMMAR FOR OPTIMIZING AUTOMATIC RETRIEVAL IN TEXT CORPUS: A Comparison of Regular Expression and Local Grammar Graph

Prihantoro
Universitas Diponegoro

Abstract

Machine Readable Grammar (MRG) is aimed at supporting the computer to perform Natural Language Processing (NLP) tasks. As for this paper, it discusses the one of the essences of MRG, which is to perform automatic retrieval in a text corpus. In automatic retrieval, the MRG serves a crucial importance in performing queries of the target expressions. In this research, I use Unitex, a Java based corpus processing software. It can manage texts from languages with their own alphabets such as Chinese, Greek, Japanese, Korean and etc. This software allows two different methods of queries. The first one is by using regular expression. This method resembles queries used in search engines like Google, Yahoo, Naver and some other well known search engines. The second one is by using Local Grammar Graphs (LGGs), which are the representation of finite state transducers (FSTs). When queries are performed by using LGGs, users can set various constraints and perform more complex retrievals. In terms of speed, regular expression works faster. But the advantage of LGGs is that it allows users to perform multiple retrievals and generate outputs at the same time. However, it depends completely to decide which MRG that suits their goal.

Keywords: Machine Readable Grammar, Corpus, Regular Expression, Local Grammar Graphs

1. CORPUS LINGUISTICS AND NATURAL LANGUAGE PROCESSING

Corpus linguistic deals with linguistic data and how to perform processing on the linguistic data. The classification of corpora (Baker, Hardie, & McEnery, 2006) might vary, such as from the size (large-small), text variation (general-specific), language (monolingual-multilingual-paralell). From the accessibility, corpora are often distinguished by open access or closed access corpus. The data is machine readable, and organized in a structured way to allow further processing (Adolphs, 2006). The advancement of computer technology allows robust processing with computer, improving speed of processing and reducing space and time at the same time. As corpus linguistics tends to approach the data, Natural Language Processing is most likely about corpus data processing (Kao & Poteet, 2007). What this section emphasises is the crucial importance of corpus data and corpus processing. The advencement of computer technology allows the automation of all manual processes to resullt on several applications from the simplest one such as concordance in the research of collocation for language learning (Nesselhauf, 2005), text mining (Clark, Fox, & Lapin, 2010), dialogue (Mitkov, 2003) for automatic question and answering system.

2. MACHINE READABLE GRAMMAR

Grammars are used to describe the behavior of a natural language expressions, which is considered as characters string. In computational term, users implement the grammar to capture or generate natural language expressions (Mitkov, 2003). For instance, the following ‘S => NP VP’ grammar can be implemented to capture sentences and generate sentences in English. However, this grammar does not apply for Fijian as the typology of this language is S=>VP NP.

This paper discusses grammar from the perspective of capturing natural language expressions by using queries, namely information retrieval (Tzoukermann, Klavans, & Stralkowski, 2003). There are two types of queries discussed in this paper. The first one is regular expression. Queries by regular expression is composed by using existing text characters on computer keyboard. This resembles browsing via search

engines like Google, Yahoo or other existing search engines, when we input text characters as a query, and we are most likely to get the results of the retrieval on the basis of character similarity. In regular expression, the query is sent to the corpus data and users will get the result, which can be equal or more than character based retrieval. In regular expression based query, besides alphabet, we can use any other characters to amplify the retrieval. Consider the following regular expression [^a-z]. It is used to retrieve all characters that are not alphabet. COCA or Contemporary American English Corpus (Davies, 2010) is one of the corpora that makes use of regular expression.

Another type of query is by using Finite State Automata or Transducer (FSA or FST). FSA or FST (Ludeling & Kyto, 2009) is an abstract machine used to generate or to recognize language. These are very basic concepts in several NLP based applications such as information extraction, opinion mining, automatic summarizing and information retrieval. Unitex makes use of FSA in the form of Local Grammar Graphs (Gross, 1997). LGGs can be applied offline as the works of Prihantoro (2011b & 2011c), though it is possible to integrate LGGs and Unitex online as well. See the documentation on Unitex manual (Paumier, 2008). As for now, consider the visualization of FSA and LGGs as illustrated by figure 1:

Figure 1. Finite State Automaton (FSA) and Local Grammar Graphs (LGGs)

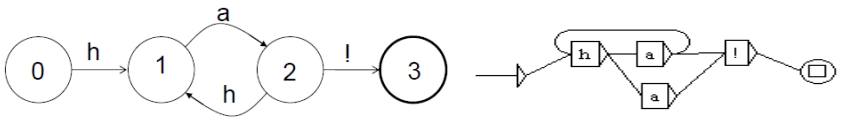


Figure one contains some characters such as numbers <0,1,2,3>, letters <a,h>, and symbol <!>. The circles are called states. Number <0> is the start state, and number <3> is the final state. Each character is inserted between states. This FSA allows the recognition or the generation of recursive <ha>strings such as : <ha!>, <haha!>, <hahaha!>. FST is FSA that has an output. By using FSA, strings with output such as <hahaha!/LAUGH> is possible to create or recognize. As for the LGGs, the application of recursive states [ha] will obtain the same result.

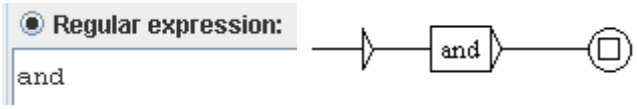
3. CORPUS PROCESSING SOFTWARE AND DATA

The corpus processing software used in this research is UNITEX. It is capable of performing some NLP tasks such as creating machine readable dictionary and grammar, information retrieval, information extraction, text annotation and etc. The corpus in this research is obtained from the default corpus of UNITEX, which is a classic work of Sir Walter Scott, Ivanhoe. In this research, I evaluate both input and output (Mc Enery & Hardie, 2012) of two different machine readable grammars (regular expression and local grammar graph) to perform automatic retrieval on the corpus. The result is evaluated on the basis of its speed and concordance generation. The procedures can be summarized as follow: 1) text is uploaded to Unitex, 2) The text is annotated by existing lexical resource (for English), 3) Queries are applied by using both regular expression and LGGs, 4) the concordances are compared, 5) analyzing the advantages and disadvantages of the two MRGs.

4. APPLICATION

The accuracy of the automatic retrieval can be effective both by using regular expressions or Local Grammar Graphs (LGGs). Both are effective to perform automatic retrieval on the basis of character similarity, or morphological or semantic feature. However, regular expression works faster than LGGs. This is because, regular expression works on the surface text, while the LGGs work deep on the automaton. Regardless of the difference, to some extent, equal result can be gained. As for now, let us consider the interfaces of regular expression and LGGs are expressed by figure 1:

Figure 1. The Interface for Retrieval by Regular Expressions and Local Grammar Graph



For regular expression, Unitex user must entry the query on the query box, as presented by the left side of figure one. It retrieves the string of [and]. The same result is obtained by applying the LGG on the right side of figure 1. The retrieval here is simply character based. Therefore, it retrieves only the string, equal to the query. The next sub-section describes more retrieval under different grounds such as morphological or semantic feature. Even the character based retrieval performed by the next sub-section is more complex.

4.1 Character based

We sometimes are required to retrieve words containing specific character strings. For instance, we are requested to retrieve all words that begin with {anti-}, or ends {-al}. Figure 2 shows the result of a retrieval by using query <<ess$>>.

Figure 2. The Concordance for <<ess$>> Retrieval

om accident, whereas the former evinced awkwardness and want of management of the weapon and
s.{S} In his seat he had nothing of the awkwardness of the convent, but displayed the easy a
short and thick curled hair of a raven blackness, corresponding to his unusually swart comp
ssive, had been burnt almost into Negro blackness by constant exposure to the tropical sun,
"has punished us all, in chastising the boldness of my friend.{S} Let me hope she will be le
with his train, and eyeing with all the boldness of royal criticism the beauties who adorned
f at the disposal of every one claiming business with him.{S} Isaac at once replaced on the
ior of Jorvaulx, declared the principal business of the day had been forgotten.{S} "By my ha
a mind, which was a strange mixture of carelessness and presumption with low artifice and c
me old; but they shall find, alone and childless as I am, the blood of Hereward is in the v
id her hand heavy upon his strength and comeliness?"{S} "He was darker," said the Palmer, "a

By complying to the query, Unitex retrieves all strings that are 1) separated by spaces, 2) ended by {-ess}. As figure 2 shows, the concordance shows all words ended by [–ess], where mostly are nouns derived from adjectives. However, this retrieval does not have morphological constraint such as part of speech (POS). Therefore, verb strings such as ‘impress’, ‘harness’, quantifier like ‘less’, or nouns that are not derived from adjectives such as ‘actress’, or ‘waitress’ are included in the concordance result. The next sub section describes how to set morphological constraint on automatic retrieval.

4.2 Morphological Feature

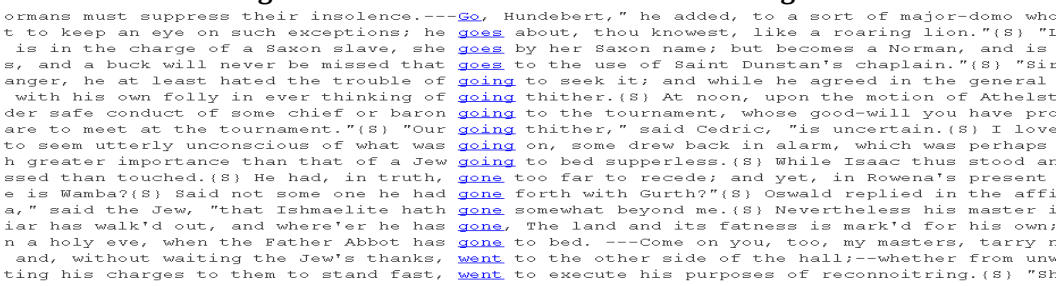
Retrieving under the constraint of morphological feature is more complex than character based retrieval as it requires the corpus annotated. When the corpus is annotated, the retrieval beyond character based, such as morphological features based retrieval is made possible (Meyer, 2004). One of the functionalities of Unitex is to perform annotation by using lexical resources (often referred as ‘machine readable dictionary’). When the corpus is already annotated, it is possible to retrieve the strings with morphological constraint by using annotation code. Consider the retrieval of prepositions as shown by figure 3.

Figure 3. The Retrieval of Strings with <PREP> Prepositional Constraint

mark the existence of the Anglo-Saxons as a separate people subsequent to the reign of Wi
chains with which they were loaded.{S} At court, and in the castles of the great nobles,
was so dear to every English bosom, and at the certain hazard of being involved as a party
their power, to place themselves each at the head of such forces as might enable him to
his extensive wood are still to be seen at the noble seats of Wentworth, of Warncliffe Par
beautiful hills and valleys which lie between Sheffield and the pleasant town of Doncast
.ll, however, the necessary intercourse between the lords of the soil, and those oppressed
lual formation of a dialect, compounded betwixt the French and the Anglo-Saxon, in which t
, and to maintain a line of separation betwixt the descendants of the victor Normans and
d; yet the great national distinctions betwixt them and their conquerors, the recollectio
ight indeed purchase temporary repose; but it must be with the sacrifice of that independ
Normans and Anglo-Saxons, or to unite, by common language and mutual interests, two hosti

By writing a simple query <PREP>, Unitex has managed to retrieve all prepositions, even to the extent of archaic preposition ‘betwixt’. This made possible by the annotation code on lexical resource that annotate all prepositions with PREP code. By applying this query, however, we cannot specify word forms. The retrieval of word forms requires the query written by lowercases as presented by figure 4.

Figure 4. The Retrieval of Word Forms of <go>

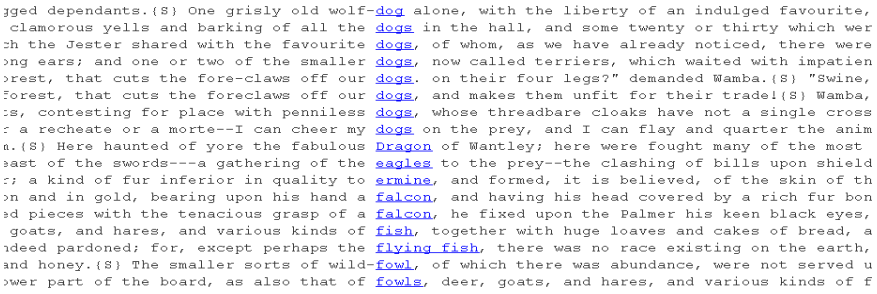


The use of lower cases, with single left-right brackets, indicated that it initiates the retrieval of word forms. Therefore, it retrieves all word forms of 'go' such as 'goes', 'gone', 'went', and 'go'. Some lexical items, simplex or compound, are also semantically annotated. It makes possible for the user to retrieve with additional constraint, which is semantic.

4.3 Semantic Feature

One of the semantic feature annotations is <+Anl> for 'animal'. By using this annotation code it is possible to retrieve all strings that are 1) nouns, and also 2) animal. Figure 5 shows the concordance as a result of the retrieval. Notice that it retrieve all animals, regardless of its quantity (singular or plural). It has also managed to retrieve the compound 'flying fish'.

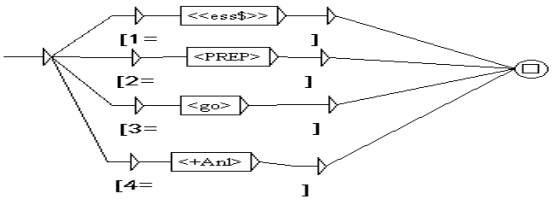
Figure 5. The Retrieval of <+Anl> Animal Strings

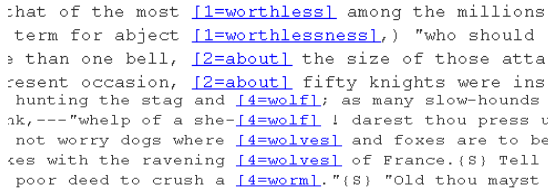


4.4 LGGs: Multiple Retrieval and Output Manipulation

The previous sub section has described retrieval that can be performed by both regular expression and LGGs. However, there are some limits to which regular expressions can apply. First, we cannot perform multiple retrieval by regular expression. However, it is possible by using LGGs. See the LGGs presented by figure 6 and its concordance:

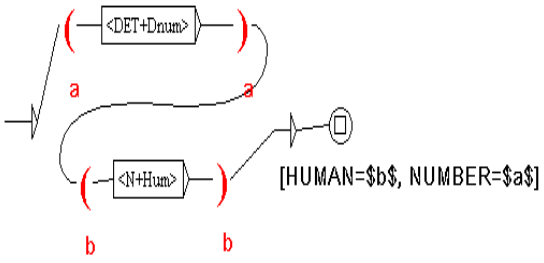
Figure 6. Multiple Retrieval and Output





We can see that all retrievals in the previous sub section can be summarized with LGGs as shown in figure 7. The LGGs also give output, in this case numbers, to label each retrieval (from 1 – 4). Another example of LGGs function, which is to manipulate output, is shown by figure 7. The figure shows the LGGs applied to retrieve compound noun, which involves determiner. In figure 7, a morphological constraint (numeral) is set to the determiner, so that it only retrieves determiners that are numerals. A constraint is also set to noun to retrieve only human nouns. See the LGGs and the concordance displayed by figure 7:

Figure 7. Output Manipulation (Permutation)



resent occasion, about fifty knights[HUMAN=knights, NUMBER=fifty] w
edged the blade Which fifty wives[HUMAN=wives, NUMBER=fifty] to wi
as follows: First, the five challengers[HUMAN=challengers, NUMBER=f
chosen colours of the five knights challengers[HUMAN=knights chall
riers were opened, and five knights[HUMAN=knights, NUMBER=five], ch
s fixed upon them, the five knights[HUMAN=knights, NUMBER=five] adv
could bear down these five knights[HUMAN=knights, NUMBER=five] in
announced to him that five men[HUMAN=men, NUMBER=five], each leadi
eath this tree four or five yeomen[HUMAN=yeomen, NUMBER=five] lay s

The brackets surround each token in figure 7 is labeled by ‘a’ and ‘b’. The output is the permutation of the two, therefore resulting in ‘b’ ‘a’. It reverses the order of DET N, to N Det. Another output is given, which is to name label a as NUMBER, and name label b as HUMAN.

5. CONCLUSION

In this paper, two machine readable grammars are discussed: regular expression and Local Grammar Graphs (LGGs). One has the advantage over another. Regular expression is a machine readable grammar that works faster than LGGs. The reason is it is more memory-efficient (Yu, Chen, Diao, & Katz, 2006). LGGs, on the other hand, allow the user to perform multiple retrievals and give output to the retrieval. Users can also perform phrasal retrieval more effectively by using LGGs. As each grammar has its own advantage, then it is the users’ decision to use one of which. The decision is made upon the aim of the retrieval.

REFERENCES

Adolphs, S. (2006). Introducing Electronic Text Analysis. London: Routledge.

Baker, P., Hardie, A., & McEnery, T. (2006). A Glossary of Corpus Linguistics. Edinburgh: Edinburgh University Press.

Clark, A., Fox, C., & Lapin, S. (2010). Natural Language Processing and Text Mining. Oxford: Blackwell.

Davies, M. (2010). The Corpus of Contemporary American English as the First Reliable Monitor Corpus of English. Language and Literary Computing , 447-464.

Gross, M. (1997). The Construction of Local Grammars. In E. Roche, & Y. Schabes, Finite State Language Processing (pp. 329-354). Massachusetts: MIT Press.

Kao, A., & Poteet, S. (2007). Natural Language Processing and Text Mining. London: Springer.

Ludeling, A., & Kyto, M. (2009). Corpus Linguistics: An International Handbook. Berlin and New York: Walter de Gruyter.

Mc Enery, T., & Hardie, A. (2012). Corpus Linguistics: Method, Theory and Practice. Cambridge: Cambridge University Press.

Meyer, C.-F. (2004). English Corpus Linguistics: An Introduction. Edinburgh: Cambridge University Press.

Mitkov, R. (2003). Handbook of Computational Linguistics. Oxford: Oxford University Press.

Nesselhauf, N. (2005). Collocation in a Learner Corpus. Amsterdam: John Benjamins Publishing.

Paumier, S. (2008). Unitex Manual. Paris: Universite Paris Est Marne La Valee & LADL.

Prihantoro. (2011b). Local Grammar Based Auto Prefixing Model for Automatic Extraction in Indonesian Corpus (Focus on Prefix MeN-). Proceedings of International Congress of Indonesian Linguists Society (KIMLI) (pp. 32-38). Bandung: Universitas Pendidikan Indonesia Press.

Prihantoro. (2011c). Transducer for Auto-Convert of Archaic to Present Day English: A Support for Computer Assisted Language Learning. Proceeding for The 16th English in Southeast Asia Conference. Yogyakarta: Sanata Dharma University Press.

Tzoukermann, E., Klavans, J., & Stralkowski, T. (2003). Information Retrieval. In R. Mitkov, The Oxford Handbook of Computational Linguistics (pp. 529-544). Oxford: Oxford University Press.

Yu, F., Chen, Z., Diao, Y., & Katz, R.-H. (2006). Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection. Architecture for Networking and Communication System , 93-102.