**ON MONITORING LANGUAGE CHANGE**

**WITH THE SUPPORT OF CORPUS PROCESSING**

Prihantoro

Universitas Diponegoro

prihantoro2001@yahoo.com

**Abstract**

One of the fundamental characteristics of language is that it can change over time. One method to monitor the change is by observing its corpora: a structured language documentation. Recent development in technology, especially in the field of Natural Language Processing allows robust linguistic processing, which support the description of diverse historical changes of the corpora. The interference of human linguist is inevitable as it determines the gold standard, but computer assistance provides considerable support by incorporating computational approach in exploring the corpora, especially historical corpora. This paper proposes a model for corpus development, where corpus are annotated to support further computational operations such as lexicogrammatical pattern matching, automatic retrieval and extraction. The corpus processing operations are performed by local grammar based corpus processing software on a contemporary Indonesian corpus. This paper concludes that data collection and data processing in a corpus are equally crucial importance to monitor language change, and none can be set aside.

Keywords: Corpus, pattern matching, automatic retrieval, local grammar.

1. **INTRODUCTION**

The use of computer affects diverse aspects of life, not only technology, but also language. The idea is that the computer can be utilized to support language analysis in various branch of linguistics. A variety of terms is suggested to describe computer support in performing linguistic analysis ranging from phonology, morphology, syntax, semantics and to other branches of linguistic as well. However, two most commonly used terms are Natural Language Processing (NLP) and computational linguistics (CL). These two terms suggest an interdisciplinary study of computer science and linguistics.

 Among linguistic branches, one that benefit from NLP/CL is corpus linguistics that focuses on the documentation of language, both spoken and written data. Computer assistance in this study can significantly enhance the documentation function. Besides avoiding large space to store the data, it can structure the data as required by the users. It also supplies users with ease of data retrieval without having to manually index the data. Consider the data of Malay classical literature corpora on Malay concordance project (<http://mcp.anu.edu.au/>) presented by figure 1:

**Figure 1. Sample of Structured Corpora**



Figure 1 illustrates a structured data on Malay classical literature corpora. There, the data is indexed by the alphabetical order: however, one can also index the data on chronological order. Another function is automatic retrieval, where it can retrieve lexicogrammatical pattern as required by the user. Figure 2 illustrates the display of automatic retrieval (morphology), still on Malay concordance corpus.

**Figure 2. Malay Concordance Corpus**



This paper seeks to describing a mechanism for performing automatic retrieval. This retrieval will benefit users in terms of lexicogrammatical pattern matching on the corpora, including the monitor corpora. Monitor corpora is meant to monitor language shift, which to some extent, is inevitable (Davies, 2010).

1. **LANGUAGE DEVELOPMENT AND MONITOR CORPUS**

One of the corpora allowing chronological language observation is COCA, which stands for the Corpus of Contemporary American English (Davies, 2010). Davies (2010) also claimed this corpora as the first reliable monitor corpora. Consider figure 3:

**Figure 3. Query Box for ‘Google’ in COCA (**<http://corpus.byu.edu/coca/>)



Available on line, COCA are composed of various genres of spoken and written English. The corpora are also equipped with various computational operations. One of the most basic operations is information retrieval; the retrieval of information from stored data (on/off line) via queries formulated by the user (Tzoukermann et al, pp. 529-541). Consider the demonstration of COCA retrieving a target word ‘google’ a search engine. The aim of this demonstration is to observe the changing trend of the target word from year to year. Figure 4 presents the result:

**Figure 4. Historical Chart for ‘Google’ in COCA**



Figure 4 is divided into two large sections. The left section indicates the identification of target words on several genres such as: spoken corpus, fiction, magazine, newspaper and academic corpus. Two genres where the target words are less frequently used are fiction and academic. Magazine and newspaper sections dominate the use of the target word.

 The right section of the result, which is the target of this demonstration, indicates the distribution of the target word on the chronological order. The system identified no result from 1990-1994. There is, however, one target word indentified on the corpora, between 1995 – 1999, not significant though. Later, It significantly improves from 2002 to 2004, 2005 to 2009, and 2010 to 2011. This is not surprising since Google is officially incorporated as a private company in 1998. The fame came a bit later than that, around 2000. Now, we might be interested in understanding the context, on what context this word is identified on 1994-1999 corpora. Again, the retrieval makes this operation easily executable. Consider the result on figure 5:

**Figure 5. Target Word in Context**



The context display in figure 5 indicates that the use of the target word is not the search engine ‘google’, but instead, it is the last name of a person (Barney Google), a cartoon character.

1. **DATA PROCESSING**

This section attempts to discuss data processing, focusing on data annotation that applies for automatic retrieval of Indonesian prefixed verbs. It briefly describes the crucial importance of data annotation, and its practice in an experiment corpus of contemporary Indonesian.

* 1. **Data Annotation**

Data annotation is very essential in corpus processing. Without annotation, computer allows only some basic statistical operations (character based) such as word and character count, basic retrieval and extraction, with poor linguistic examination on the data. These character based pose many challenges for linguists. For instance, what constitute a word in basic computing are merely space-separated strings of characters. This causes non-canonical words, such as words with affix(es), are treated as distinct lexemes.

It is possible for two space-separated strings of characters, to root on one canonical form. However, without proper annotation, the computer will likely fail recognizing them. For instance, in Indonesian, *memukul* (to hit) and *dipukul* (being hit) root on the same canonical form *pukul*. In order to perform lexeme based statistical count, annotation is required. Without annotation, these two strings, *memukul* and *dipukul*, are counted as two distinct units, whereas it is commonly known for Indonesian that they are the variants of *pukul*.

Some linguistic features are shared among languages in the world, but some others are specific to individual language. Without proper language module computer is most likely to fail to perform character recognition. English is usually the default language module in a computer. The question is, how will this module recognize other language, such as Chinese, Japanese or Korean, for instance, that employ different writing system?

Character recognition is not the only challenge. Another challenge is segmentation. In English, and some other languages, space is the separator that segment one word to another. However, we need to understand that this segmentation method does not apply to all languages, for instance; Chinese, where a string of characters might be composed of a complex sentence. Without a proper module, the computer will recognize this string as one word only, disregarding the fact that it is composed of several words.

* 1. **Application**

Previous sub section has dealt with some annotation problems. One of the problems is how the computer can recognize two words that root on the same canonical form. The solution to this problem must be executed in pre-processing stage, namely annotation. The data must be annotated carefully; otherwise, the computer will recognize the derived forms of a canonical form as distinct lexemes (types).

There are some formalism methods to perform data annotation. In this paper, the writer employ entry line formalism, used in UNITEX (2008), a Local Grammar based corpus processing software. Local Grammar (Gross, Local Grammars and Their Representations by Finite State Automata, 1993) is used to support computational process in some programming languages. Its graphical representation, Local Grammar Graph (LGG), is designed to powerfully describe linguistic phenomena in computational sense. Besides English, LGGs have successfully been used in computational research of various languages such as French (Gross, 1984)**,** Korean (Nam & Choi, 1997), Arabic (Traboulsi, 2009), Indonesian (Prihantoro, 2011) and some other languages

Entry line formalism used in UNITEX attributes lexicogrammatical and semantic properties to the canonical forms. To perform precise recognition of lexeme and its derived forms, annotation must first be performed morphologically. However, the first procedure is to structure the canonical forms lexical resource. These forms are then to be inflected morphologically to generate their inflected forms. Consider the lexical resource of some Indonesian verbs canonical forms presented by figure 6:

**Figure 6. Some Indonesian Verbs Canonical Forms Lexical Resource**



This resource attributes each entry with at least a grammar code, which reflects the parts of speech. Most of the entries are attributed with <V> indicating that the canonical form is a verb. However, it is possible for the same canonical form to appear twice. These doublets indicate that they have distinct grammar code and hence distinct part of speech. For instance, the word *bor* (*drill*) has two grammar codes: <N> which indicates *drill* as a noun, and <V>, which indicates *drill* as a verb. This distinction will benefit in terms of disambiguation. The number in adjunct to the grammar code is the morphology inflection code. The code is not always the same for each entry as some entry might be inflected by distinct morphological LGG. As an instance, prefix {meN-} in Indonesian takes distinct forms for distinct entry. Table 1 presents some canonical forms in Indonesian and their possible surface representations:

**Table 1. Some Canonical Forms and their Surface Representations**



Table 1 presents distinct concatenation of prefix {meN-} to some entries. In *larang*, the prefix concatenates to the entry line by deleting the {N-}, which stands for possible nasal orthographical forms. In *sapu*, the {N-} is converted to <ny>, the orthographical form of [ɳ], accompanied by the deletion of the first orthographical letter of the entry <s>. In *tulis*, the deletion of <t> takes place, succeeding the concatenation of {men-} to the lexeme *tulis* where the surface representation of {N-} is <n>. These instances illustrate how distinct canonical forms prefer distinct concatenation method, and hence, result on distinct surface representation. They require distinct morphological inflection methods. In response to this issue, several morphological LGGs (Local Grammar Graph) are built. These LGGs are inflected to the lexemes with reference to the morphology inflection code annotated on the canonical forms lexical resource.

**Figure 7. Morphological LGG to Perform Direct and Indirect Concatenation**



Figure 7 from Prihantoro (2011) on the left shows one of the inflectional LGGs applied on UNITEX (Paumier, 2008), a corpus processing software. Canonical lexical resource illustrated on figure 6 is inflected by this LGG (and also some other inflection LGGs) with reference to Indonesian lexicogrammatical patterns. The inflection results on current lexical resource (showed by figure 8 on the right) which comprises both of inflected and canonical wordlist. This word list is a formalization of Indonesian lexicogrammatical pattern for prefix {meN-} and {di-}. Therefore, when this lexical resource is applied to a particular corpus, the user can identify the inflected forms in condition that the inflection module (in this case LGGs) is completed. Consider the result of automatic retrieval of lexeme <tanam> on the contemporary Indonesian corpus, presented by figure 8.

**Figure 9. Inflected Forms of <tanam>**



In this corpus, with the application of the lexical resource, the computer retrieves two inflected forms of <tanam> (*ditanam, menanam*) and one canonical form *tanam* as well as prefixed forms on verb entries with eight different morphological concatenation methods. Comprehensive recognition absolutely requires LGGs that comply to all affixes in Indonesian. However, the LGGs designed for Indonesian verbs in this research have managed to recognize prefix {meN-} and {di-} for Indonesian verbs.

1. **CONCLUSION**

The application of LGGs has demonstrated automatic retrieval of both inflected and canonical entries on a contemporary Indonesian corpus. The method can also be applied on monitor corpora so allowing users to observe the inflection system of a language from one point of time to another. This, of course, requires sufficient and valid data collection as well as careful annotation method: a scheme for further research. With the existence of Corpus processing software and LGGs, some natural language operations can be performed on the monitor corpora and in turn it can benefit its users on observing how language change over time, not only for affix, but to the extent of more linguistic complexities.

# Bibliography

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. (1994). Constructing Lexicon Grammars. In A. Zampolli, & B. Atkins, *Computational Approaches to the Lexicon* (pp. 213-263). Oxford: Oxford University Press.

Gross, M. (1984). Lexicon Grammar and the Syntactic Analysis of French. *Proceedings of the 10th International Conference on Computational Linguistics* (pp. 275-282). Stanford: Association for Computational Linguistics.

Gross, M. (1993). Local Grammars and Their Representations by Finite State Automata. In M. Hoey, *Data, Description, Discourse: Papers on the English Language in Honour of John Mc.Sinclair* (pp. 26-38). London: Harper Collins.

Nam, J. S., & Choi, K. S. (1997). A Local Grammar Based Approach to Recognizing Proper Names in Korean Texts. *Workshop on Very Large Corpora: ACL*, (pp. 273-288). Hongkong.

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

Prihantoro. (2011). 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.

Traboulsi, H. (2009). Arabic Named Entity Extraction: A Local Grammar Based Approach. *Proceedings of the International Multiconference Science and Information Technology* (pp. 139-143). Mragowo, Poland: Polish Information Processing Society.

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.