

Widiarti-Winarko Algorithm for Grouping Syllables Result from the Javanese Literature Document Image Recognition

ANASTASIA RITA WIDIARTI
Faculty of Science and Technology
Sanata Dharma University

Kampus III Paingan Maguwoharjo Depok Sleman Yogyakarta 55282
INDONESIA
rita_widiarti@usd.ac.id <http://www.usd.ac.id>

EDI WINARKO
Faculty of Mathematics and Natural Sciences
Gadjah Mada University

Gedung SIC, Lantai 3 Sekip Utara Fakultas MIPA Universitas Gadjahmada Yogyakarta
INDONESIA
ewinarko@ugm.ac.id <http://www.mkom.ugm.ac.id>

Abstract: - Document image recognition can be used to help translate ancient documents written in Javanese character. If the ancient documents mentioned written in Latin character, it can be read by young people in Indonesia today for various purposes and in an effort to help preserve the rich culture especially in Javanese literature. One of the problems in the document image recognition for Javanese literature is how can we make words from syllable sequence which are result of Javanese character recognition into the correct words in the Javanese language rules, because there are not space in the rules of the Javanese character written. This paper describes Widiarti-Winarko algorithm that can be used to grouping syllables Javanese language combined with the ability of Lucene as a software to create a dictionary of Javanese words. The dictionary used to check whether the output words are the correct form. Results from the test in the output of document image recognition in the two pages Hamong Tani book, with the source data dictionary maker from all of pages Hamong Tani book, the system gave a words found by the truth of the formation of words in context sentences of 62.96%, and 75% found the word correctly in the Javanese language. By looking at the magnitude of the percentage of the truth of the formation of words in context sentences are still below 70%, it still needs to be improvements in the algorithm.

Key-Words: - Document Image Recognition, Information Retrieval, Javanese Literature Document, Lucene

1 Introduction

The culture city of Yogyakarta and Surakarta in Indonesia has many museums that stores have a range of things past heritage and heritage sites scattered in various places. According to Riboet who is the experts epigraphy, Sonobudaya museum in Yogyakarta saved a lot of the richness of the cultural treasures of literature, among other things consists of 172 *keropak lontar*, where about half of it is damaged. Likewise, a similar condition that affects about 1200 manuscripts with paper media [1]. Whereas the manuscripts storing wealth history or others. If this damage continues, it could be possible that the cultural heritage contained in the manuscripts destroyed.

Document image analysis science developments, namely the analysis on visual representations such as paper documents, faxes, journal papers, sheets, field offices and others [2], opening up great opportunities to benefit the preservation of ancient manuscripts found in Yogyakarta and Surakarta. O'Gorman and Kasturi [3] giving the stages a document image analysis processes can be modified for the Javanese literature document image recognition. Starting with the data capture phase where data from paper documents will be read by the optical scan tool and the result is saved as an image file. Continued processing stage level pixels which aims to prepare the document's image, as well as make intermediary to help recognize the image. The third stage is the stage of the introduction of the alphabet in order to translate a series of characters

that have a variety of shapes and sizes. The fourth stage is the processing of the text so that the result of the introduction of a more useful.

2 Problem Formulation

One of the problems that appear in the text-processing stage is the difficulty in grouping syllables from the results of the Javanese character recognition into Javanese words. This happens because the final results of the process in the processing stage in the pixel-level image of text documents is a row of images of Javanese literature syllables, while in the rule of writing a Javanese document with Javanese character there aren't found symbol space to divide the word as shown in Fig. 1. For it need to built an application in which the syllable group or this word is a true word in the rule of Javanese language.

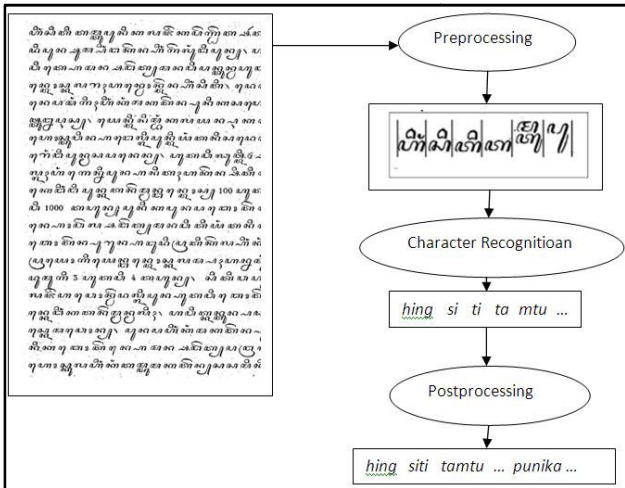


Fig.1. Overview of the problems

From the study on the procedures for writing Javanese character in Javanese language, there are several problems associated with the process of making word. The following results are outlined in the writing problems of closely associated with document image segmentation results, as well as the reading of literary documents of Java. The first problem found delineated in the following example. If there are pieces of a single line of the image of the document as shown in Fig. 2. below.

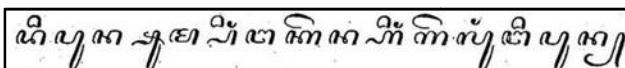


Fig. 2. Illustration of an image that contains the first issue

Transliteration of the image in Fig. 2 above is as follows.:

di pu na _su ma _ping nga ke na _hing ge lung ngi pu n

Transliteration that begin with the translation ‘_’ shows that in Javanese script writing rules, if there is a consonant alphabet letters side by side in addition to *h*, *ng*, and *r* then the first consonant is written i.e. the script essentially intact, while the next consonant is written with the symbol of the Javanese couples. Example after a consonant *pu* in the second image is part of the word *dipun*. Then the *pu* will be rendered intact, likewise with the suffix *n*. However, because the word *dipun* are not as the end of the sentence, then the writing rules that apply to the suffix *n* is still denoted by *n* characters intact *n*, while the next syllable *su* will be rendered instead of the character *sa* with *sandhangan swara sa* *su*, but rather will be written with the symbol of a given pair *sandhangan swara sa* *ꦱꦱꦱꦱ*.

Then if continued with translation results, will be get output is follows.:

di pu n su m ping nga ke n ing ge lung ngi pu n

But reading the above is still not perfect, because the reading of the results is expected to be *dipun sumpingngaken ing gelungngipun*, which means it is pinned on the bun.

As for the second problem, if there is line of pieces as shown in the image in Fig. 3. below:

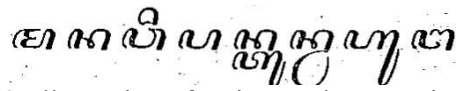


Fig. 3. Illustration of an image that contains the second issue

If reads the image one by one, then the image in Fig. 3. will read as follows:

ma na wi pa ntu nwa hu nga

with the suppositions that the correct reading of the results was: *manawi pantun wahu nga*.

The issue that arises is on syllable *nwa* from image *ꦒꦤ꧀*, script *n* became a part of the previous word i.e. *pantun*, whereas tribes of *wa* to be part of the next word i.e. *wahu*. On this issue, then an alternative solution is any combination of two consonant side by side except for *dh*, *th*, *ng*, *ny* needs to be made a function to verify whether the script first became a part of the previous word or words afterwards (pairs of consonants i.e. *dh*, *ny*, *th*, *ng* are basic syllables in Javanese written originally as *dha*, *nya*, *tha*, and *nga*). Examples of syllable *ntu*, and *nwa* above.

The next question is how do we check if the word is a true word. A simple way to check whether

a particular word valid in a specific language is to match whether the word is in the dictionary of that language. Currently there are open source software popular can be used to build applications to search for a word in a document, such as lucene, lingpipe, and gate [4] easily and tested its ability in the process of information retrieval. This paper suggested a simple idea to grouping the syllables from the results of the Javanese literature document image recognition in order to become a correct words in Javanese language. To declare a constructed word is a correct word, then appear the idea to first build the kind of Javanese language dictionary. With a dictionary, then every time get a new word from results grouping syllbles, the system will search if there is word in the dictionary. Lucene will became the only tool which will be used to build the index words from document data source, and to search the word.

From early studies, it was found some research related to the analysis of the word. Suciadi conducting research on syntactic and semantic analysis on interpretation of NLP. Suciadi find that there is no method of parsing are ideal for all kinds of problems in NLP, thematic roles further clarify the role of each element of a sentence, and word-sense hierarchy used by the selectional restrictions by very helpful in doing the process wordsense disambiguation [5]. Margaretha, et al., found that the Latent Semantic Analysis (LSA) are able to understand the various semantic information with relatively little data on training [6]. Jurgens and Steven discover that approach to Word Sense Induction (WSI) based cluster is very sensitive to find differences in the sense of the word from context, compared with lexical feature based [7]. Krovetz and Croft do the analysis of lexical ambiguity in the information retrieval to see the usefulness of meanings of the words on the election process the relevance of documents found. They found that the granting of the weighting based on the number of the meaning of those words improve the effectiveness of information discovery data collection, although relatively few are used [8].

3 Problem Solution

From the analysis of the problems that arise, then the simplest way that can be done is take of the first syllable as new word that were found and will be a lucene search system input. An example is found words *hing* which is now considered a word. Then with the search system with lucene done comparison whether the word is in the dictionary of Javanese language which had been built with lucene. If the

word is found in the dictionary, the word is found correct, whereas if it is not, continue by combining one syllable next closest. Do a search with lucene, and if it is a new word created exists in the dictionary, then enter the word as the output of the system, but if it does not continue with the merge of the next syllable, etc., up to a maximum of 7 (seven) syllables are combined [9].

By considering the second problem in reading the document, then every time a merger is necessary to check syllables when two consonants are found side by side, unless consonants *dh*, *ny*, *th*, and *ng*. In detail, the proposed algorithm Widiarti-Winarko (WW) is as follows.:

```

1 read input file which is a result of the document
  recognition
2 Start merging syllables (syl):
3 for (int i = 0; i < syl.length - 1; i++)
4     { count++;
5       if (checkConsonant(syl[i + 1]))
6         { syl[i] = syl[i] + syl[i + 1].charAt(0);
7           syl[i + 1] = syl[i + 1].substring(1);}
8       word += syl[i];
9       if (en.search(word).length > 0 || count == 7
10        || syl[i].matches("\\W"))
11         { Print(word + "-" + count);
12           merge += word + " ";
13           count = 0;
14           word = ""; } }
```

The contents of the algorithm checkConsonant(String const) is the following:

```

1 if (const.length() > 2 dan !(const.substring(0,
2) 1).matches("[aiueo]") || const.substring(1,
3) 2).matches("[aiueo]"))
4 { for (int i = 0; i < constSideBySide.length; i++)
5   if (const.substring(0, 2).equalsIgnore
6   Case(constSideBySide[i]))
7     return false;
8   return true;}
9 else
10  return false;
```

with the note that data structure of consSideBySide variable is:

```
String[] constSideBySide = {"dh", "ny", "th", "ng"};
```

4 Experimental Result

As a data for experiment, used documents from one book Hamong Tani by Javanese literature experts to source materials making dictionaries, and a results

page reading of the book on page 2. Book Hamong Tani consists of 87 page [10], and all of the pages are used as sources for the making of a dictionary. Then from the testing of 1 page document image recognition results, i.e. a row of syllables that comply with the Javanese character, will count how many words a is true, so that the percentage of truth be obtained according to said establishment of the formula (1).

$$\% \text{ correctness} = \frac{\sum \text{correct word}}{\sum \text{words in a document}} \times 100 \quad (1)$$

Analysis on the results of the formation of the words issued by the system that produces the summary results of the analysis were constructed as shown in table 2. Table 2 presents the 94 words formed from a test file uji.txt.

Table 2. List of words

no	words that are formed	no	words that are formed
1	hing	48	punapa
2	siti	49	boten
3	tamtu	50	howel
4	punika	51	sanget
5	lajeng	52	manawi
6	kapendhet	53	tiyang
7	sarta	54	tani
8	dipun	55	boten
9	sumpingngaken	56	purun
10	hing	57	hangudipratikel
11	gelungngipun	58	hing
12	hawit	59	kangprayogiyemngam os
13	eman	60	kala
14	sanget	61	mpahhannamungdu
15	manawi	62	mugi
16	pantun	63	5
17	wahu	64	hutawi
18	ngantos	65	4
19	kala	66	tahun
20	rahhanwontenhing	67	siti
21	siti	68	wahu
22	Dene	69	lajeng
23	pamanggih	70	hawon
24	hing	71	wedallipun
no	words that are formed	no	words that are formed
25	kang	72	hutawi
26	makaten	73	boten
27	punika	74	kenging

28	sahestu	75	katanem
29	leres	76	manmalih
30	yen	77	hawit
31	tinimbang	78	kantun
32	kala	79	padhas
33	yanrekahostu	80	kemawon
34	wihangellipunti	81	punapa
35	yangtanisaderengngi	82	hing
36	pun	83	kang
37	masa	84	makaten
38	panen	85	punika
39	hutawi	86	boten
40	Gusti	87	heman
41	Allah	88	sanget
42	hanggen	89	pangrahos
43	mpunhanita	90	kula
44	hhakensitike	91	hing
45	ngingngipunkatanem an	92	kang
46	ngantos100hutawi1	93	tamtu
47	000tahunpunika	94	makaten

From table 2 that contains the words a system, then the result can be obtained following analysis:

- Words that are not shaded suggests that the word is correct, either in the rules as well as the context of the sentence. With the correct number of words in the meaning or context of the sentence is as much as 68 words, then the percentage of corectnes is the introduction of the word correctly according to the meaning and context of the sentence is equal to = $68/106 \times 100 = 62.96\%$.
- There are 13 words that have a background with shades of gray, which indicates that the word is only true in the rules, but in the context of the sentence is not appropriate. From here the corectness percentage can be calculated correctly that word recognition there are 81 words properly according to the meaning or the dictionary registration = $81/108 \times 100 = 75\%$. Mismatch context sentences here meant is a word found is correct in the sense of those words, or indeed are in the dictionary, but if viewed from the context of the discussion of the sentence containing the word, the word is less appropriate. The discrepancy occurred because:
 - at the time of completion of the process of grouping your kind words from one syllable or Word, then the word is searched for compliance in the dictionary, the word is already there. An example for the word *hing*

which are listed in numbers 24, should have is the word *hingkang*. But due to the dictionary, the word *hing* will be found first, then the said *hing* to be taken, so that every time this word is followed by the word *kang*, word *hing* will certainly be found error in classification. Likewise with other words, i.e. words *kala* is supposed to is the word *kalarahhan*, *kalayan*, and *kalampahhan*, the word *hanggen* is supposed to *hanggennipun*, said *katanem* should be *katanemman*.

- ii. The process of grouping before the word is found in a grouping of new words become less precise. Example of a word in the number 36 is supposed to be part of the word *saderengipun*, but because the algorithms built led to the word *saderengi* became a part of the previous word.
- c. Words shaded with oblique indicates that the word is wrong. An example of the word *kangprayogiyenngantos*, *rahanwon-tenhing*, *yanrekahostu*, and *hhakensitike*. Error on 3 word first occurred due to the merger the previous syllable is not in accordance with the context of the sentence. For example the word *kangprayogiyenngantos*, which is supposed to be syllable *kang* joined the previous syllable *hing* being said *hingkang*. However as there are words in the dictionary the *hing* means 'where' [10], then the program will stop groping the said *hing*. As a result the program will continue the process of grouping syllable *kang* with the tribes next words. Because it turns out with merging into syllables to the next, followed by syllables the next until it reaches seven syllables, no word is right, then the merger that took place gave rise to the word *kangprayogiyenngantos*. This turned out to be able to bring up the impact of errors, i.e. if for carrom games the next words turned out to be a part of the word before, but already separated, then surely most likely will not appear in common a word formed with the rest of the formation of the next syllable.

5 Conclusion

There are two main conclusions that can be drawn from the results of testing a system that implements the algorithm Widiarti-Winarko (WW) for grouping syllables using lucene, namely:

- a. Lucene can be used to make the dictionary the word Javanese language obtained from a set of Javanese language document and can be used to

speed up the search process word appropriate Javanese language dictionary.

- b. Percentage yield truth algorithm WW for grouping Javanese syllable with the percentage of truth by the rules of word formation is 75%, and the percentage of truth appropriate word formation rule sentence of 62.96%. The results obtained were relatively good, yet it needs to be done as well as other methods of refinement of the algorithm so that the results of the grouping for the better again.

6 Future

From the results of the system analysis using algorithms WW and lucene for the creation of dictionaries and search word, retrieved the things that become the cause of shortage of algorithm WW developed, namely:

- a. the result of this algorithm depends very much with the completeness of the dictionary used. Then it would be a huge deal when a document that is processed by the larger because each time to see the dictionary to match the words. Things that might be done to overcome this is to involve the calculation of statistics, for example, by using a large probability of occurrence information of a word after Word of the other, so that the system does not need any time to see the dictionary.
- b. Grouping algorithm syllables Widiarti-Winarko developed yet include testing against the context of the sentence, so that the process of merging is not rash just by looking at the presence of the word in the dictionary. Need other efforts such as using algorithms-algorithms associated with processing the phrase sentence.

Acknowledgment:

I would like to thank my students Audris Evan Utomo has helped me to implement this algorithm.

References:

- [1] Kusuma, M., *Riboet Darmosoetopo Rujukan Pembacaan Prasasti*. URL: <http://indonesiabuku.com/?p=6608>, accessed date: April 10, 2011.
- [2] Srihari, S.N., Lam, S.W., Govindaraju, V., Srihari, R.K., and Hull, J.J., *Document Image Understanding*. New York: CEDAR, 1986.
- [3] O'Gorman, L., and Kasturi, R., *Executive briefing: documen image analysis*, USA: IEEE Computer Society Press., 1997.

- [4] Konchady, M., *Building Search Applications: Lucene, LingPipe, and Gate*, USA: Mustru Publishing, 2008.
- [5] Suciadi, J., *Studi Analisis Metode-Metode Parsing Dan Interpretasi Semantik Pada Natural Language*, 2001, URL: <http://puslit.petra.ac.id/journals/informatics/>, accessed date: Dec 16, 2011.
- [6] Margaretha, E., Franky, and Manurung, R., *English-to-Indonesian Lexical Mapping using Latent Semantic Analysis*, URL: <http://bahasa.cs.ui.ac.id/pub/malindo08lsa.pdf>, accessed date: Dec 16, 2011.
- [7] Jurgens, D., and Steven K., Measuring the Impact of Sense Similarity on Word Sense Induction, *Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing*, 2011, URL: <http://aclweb.org/anthology-new/W/W11/W11-2214.pdf>, accessed date: Dec 16, 2011.
- [8] Krovetz, R., and Croft WB., *Lexical Ambiguity and Information Retrieval*, URL: <http://www.lexicalresearch.com/tois-lex-ambiguity.pdf>, accessed date: Dec 16, 2011.
- [9] Prawiroatmodjo, S., *Bausastra Jawa – Indonesia*, Jakarta: Gunung Agung, 1981.
- [10] Holle, KP., *Hamong Tani*, Batavia: Landsdrukkerij, 1876.