A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia

Fadillah Z Tala

0086975

Master of Logic Project
Institute for Logic, Language and Computation
Universiteit van Amsterdam
The Netherlands

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Stemming is a process which provides a mapping of different morphological variants of words into their base/common word (stem). This process is also known as conflation [10]. Based on the assumption that terms which have a common stem will usually have similar meaning, the stemming process is widely used in Information Retrieval as a way to improve retrieval performance. In addition to its ability to improve the retrieval performance, the stemming process, which is done at indexing time, will also reduce the size of the index file.

Various stemming algorithms for European languages have been proposed [10, 16, 17, 24, 28, 29, 31, 32]. The designs of these stemmers range from the simplest technique, such as removing suffixes by using a list of frequent suffixes, to a more complicated design which uses the morphological structure of the words in the inference process to derive a stem. These algorithms have also been evaluated in order to examine their effect on the retrieval performance. A good summary of these evaluation results can be found in [10, 19].

Results of stemming usage in information retrieval are inconsistent. Harman [12], in her experiments with three suffix stripping algorithms for English, reported inconsistent results. Whilst Krovetz [20] and Hull [15] both reported more favorable results of the stemming usage in English, especially for short queries. Popovic and Willett [28] reported a significant improvement in retrieval precision for Slovene language which is more complex than English [18]. He also reported that his control experiments confirmed the results in [12]. Experiments of stemmer usage for other European languages which are more complex than English, showed an improvement of retrieval precision and recall [13, 19, 27]. These studies support the hyphothesis in [18] and [27] namely, that the effectiveness of stemming in an IR systems also depends on the morphological complexity of the language.

In the case of Bahasa Indonesia, so far there is only one stemming algorithm which is developed by Nazief and Adriani [23]. This stemming algorithm was developed using a confix stripping approach with a dictionary look-up. The dictionary is very simple, it consists of a list of lemmas. The stemming process is done by stripping the shortest possible match of affixes. The dictionary look-up is performed before each stripping step and the stripping process itself is implemented recursively.

However, it is unfortunate that there is no experimental report about the effect of this stemmer on the retrieval performance. The morphological complexity of Bahasa Indonesia can be considered simpler than English because it does not recognize tenses, gender and plural forms. It is interesting to investigate whether the study of stemming effect in Information Retrieval in Bahasa Indonesia

will also support that hyphotesis. These are main reasons that motivated us to evaluate the stemming effect in Bahasa Indonesia.

Results of the experiments reported by Ahmad et al. [2] pointed out that dictionary plays an important role in the stemming process for Malay language. Since Bahasa Indonesia and the Malay language are very similar, we assume that dictionary also plays an important role in the stemming process of Bahasa Indonesia. However, based on the fact that resources such as a large digital dictionary for this language are expensive due to the lack of computational linguistics research, clearly, there is a practical need for a stemming algorithm without dictionary involvement. From a scientific point of view, it is also interesting to see whether stemming algorithm without involving a dictionary would also be effective for Bahasa Indonesia, such as it proved to be for Slovene [28] and Dutch [19].

This thesis is about a study of stemming algorithms in Bahasa Indonesia, especially their effect on the information retrieval. We try to evaluate the existing stemmer for Bahasa Indonesia [23] and compare it with a purely rule-based stemmer, which we created for this purpose. This rule-based stemmer is developed based on a study of morphological structure of Bahasa Indonesia words. A summary of the morphological structure of words in Bahasa Indonesia is introduced in Chapter 2. Chapter 2 also includes the design and the implementation of our rule-based stemmer.

Since the quality of the stemming algorithm in [23] has never been assessed, we conducted an experiment to evaluate its quality. In this experiment, we chose the Paice evaluation method [25] and results are given in Chapter 3. In this chapter, we also evaluated the effect of dictionary size to the quality of the stemmer. This hopefully will answer to what extent the dictionary-based stemmer can be approximated by a purely rule-based stemmer. It is especially relevant in the case of developing languages such as Bahasa Indonesia where new words are continuously being adopted.

The main task of this thesis is discussed in Chapter 4. In this chapter, the evaluation of stemming on the retrieval performance is explained in detail. In this evaluation, we used the traditional Precision/Recall measure. We also performed some detailed evaluations resulting in more concrete results. Finally, Chapter 5 describes the conclusion of our experiments.

Chapter 2

A Purely Rule-based Stemmer for Bahasa Indonesia

The purely rule-based stemmer we developed here is a *Porter-like stemmer* which is modified for Bahasa Indonesia. The Porter stemmer was chosen based on the consideration that its basic idea seems appropriate for the morphological structure of words in Bahasa Indonesia. First, a brief introduction to the morphological structure of words in Bahasa Indonesia is given, and second we will explain the mechanism of a Porter stemmer. Last, we will explain the modified Porter stemmer for Bahasa Indonesia which we used as the comparison in the retrieval evaluation.

2.1 Morphological Structure of Bahasa Indonesia Words

In this section, we discuss the morphological structure of Bahasa Indonesia words which is based on information in [6, 7, 23, 35]. We also looked at the morphological structure of Malay language words [1], since Bahasa Indonesia is very similar to Malay language. This discussion includes prefixes, suffixes, and combinations of them (confixes) in derived words. Although infixes do exist in Bahasa Indonesia, the number of derived words from these infixes is very small. Because of this and for the sake of simplicity, infixes will be skipped and ignored. Words that contain an infix will be considered as they are.

The morphology of Bahasa Indonesia words can comprise both *inflectional* and *derivational* structures. Inflectional is the simplest structure which is expressed by suffixes which do not affect the basic meaning of the underlying root word. These inflectional suffixes can be divided into two groups:

1. Suffixes *-lah, -kah, -pun, -tah.* These suffixes are actually the *particles* or functional words which have no meaning. Their occurrence in words is for emphasizing, examples:

dia	+	kah	⇒	*diakah*
(she/he)				(he/she - with emphasizing for questioning)
saya	+	lah	⇒	*sayalah*
(I)				(I - with emphasize)

2. Suffixes *-ku, -mu, -nya.* These suffixes, which are attached to the words, form the *possessive pronouns*, examples:

tas	+	mu	⇒	*tasmu*
(bag)		(you)		(your bag)
sepeda	+	ku	⇒	*sepedaku*
(bicycle)		(me)		(my bicycle)

Each suffix of groups 1 and 2 may occur in the same word. When they are both present, they follow a strict order: suffixes of the second group always precede the first group. This ordering motivates the following definition.

Definition 2.1 *The morphological structure of an inflectional word is:*

```
inflectional := (root + possessive_pronouns) |
                (root + particle) |
                (root + possessive_pronouns + particle)
```

The attachment of inflectional suffixes to a word/root will not change the spelling of the word/root in the derived word. In other words, no character in the root/original word is diluted in the derived word. The root/original word can still be located easily in the derived word.

Just like the Malay language, derivational structures of Bahasa Indonesia consist of prefixes, suffixes and a pair of combinations of the two [1, 6, 7, 34, 35]. The most frequent prefixes are: *ber-, di-, ke-, meng-, peng-, per-, ter-* [34, 35]. The following list shows an example of each prefix:

ber + *lari* (to run) ⇒ *berlari* (to run, running)
di + *makan* (to eat) ⇒ *dimakan* (to be eaten - passive form)
ke + *kasih* (to love) ⇒ *kekasih* (lover)
meng + *ambil* (to take) ⇒ *mengambil* (taking)
peng + *atur* (to arrange) ⇒ *pengatur* (arranger)
per + *lebar* (wide) ⇒ *perlebar* (to make wider)
ter + *baca* (to read) ⇒ *terbaca* (can be read, readable)

Some prefixes such as *ber-, meng-, peng-, per-, ter-* may appear in several different forms. The form of each of these prefixes depends on the first character of the attached word. Unlike the inflectional structure, the spelling of the word may be changed when these prefixes are attached. As an example, take the words *menyapu* (to sweep, sweeping) which is constructed from the prefix *meng-* and the root word *sapu* (broom, sweeping-brush). The prefix *meng-* is changed to *meny-* and the first character of the root word is diluted. Rules for various forms of these prefix attachments can be found in Appendix A, Table A.1.

Derivational suffixes are: *-i, -kan, -an.* Examples of words with these suffixes are:

gula (sugar) + i ⇒ *gulai* (to put sugar to)
makan (to eat) + an ⇒ *makanan* (food, something to be eaten)
beri (to give) + kan ⇒ *berikan* (to give to)

In contrast to prefixes, the attachment of suffixes never changes the spelling of the root in the derived word.

As mentioned earlier, the derivational structure also recognizes confixes, where a combination of prefix and suffix attaches together in a word to derive a new word. For example:

per + *main* (to play) + an ⇒ *permainan* (toy, game, thing to be played)
ke + *menang* (to win) + an ⇒ *kemenangan* (victory)
ber + *jatuh* (to fall) + an ⇒ *berjatuhan* (falling)
meng + *ambil* (to take) + i ⇒ *mengambili* (taking repeatedly)

Not all combinations of prefixes and suffixes can be joined together to form a confix. There are some combinations of prefix and suffix which are not permitted. Table 2.1 shows all of the illegal confixes.

Table 2.1: Illegal confix pairs.

Prefix	Suffix
ber	i
di	an
ke	i|kan
meng	an
peng	i|kan
ter	an

A prefix/confix can be added to an already confixed/prefixed word, which results in a *double prefix* structure. Just like the construction of confixes, not all prefixes/confixes can be added to a certain confixed/prefixed word to form a *double prefix*. There exist rules which govern the ordering of these double prefixes, but there are some exceptions to this rule. Table 2.2 shows these ordering rules.

Table 2.2: Double prefixes order.

Prefix 1	Prefix 2
meng	per
di	ber
ter	
ke	

Definition 2.2 *The morphological structure of a derivational word:*

```
derivational  := prefixed | suffixed | confixed | double_prefix
```

where

```
prefixed      := prefix + root
suffixed      := root + suffix
confixed      := prefix + root + suffix
double_prefix := (prefix + prefixed) | (prefix + confixed) |
                 (prefix + prefixed + suffix)
```

The last possibility to derive a new word is by adding the inflectional suffixes to an already prefixed, suffixed, confixed and even double prefixed word. These forms are the most complex structure in Bahasa Indonesia. Nazief and Adriani in [23] called these structures *multiple suffixes*.

From Definition 2.1 and Definition 2.2, the general morphological structure of words in Bahasa Indonesia can be simplified by Definition 2.3.

Definition 2.3 *The morphological structure of words in Bahasa Indonesia:*

```
[prefix1] + [prefix2] + root + [suffix] + [possesive_pronoun] + [particle]
```

where `[...]` *means an optional occurrence.*

2.2 The Porter Stemming Algorithm

The Porter stemming algorithm is a conflation stemmer which was proposed by Porter [29]. The algorithm is based on the idea that suffixes in English are mostly made-up of a combination of smaller and simpler suffixes [26]. The stripping process is performed on a series of steps, specifically five steps, which simulates the inflectional and derivational process of a word. At each step, a certain suffix is removed by means of substitution rules. A substitution rule is applied when a set of conditions/constraints attached to this rule hold. One example of such a condition is the minimal length (the number of vowel-consonant sequences) of the resulting stem. This minimum length is called *measure* [29]. Other simple conditions on the stem can be whether the stem ends with a consonant, or whether a stem contains a vowel.

When all conditions of a certain rule are satisfied, the rule is applied, which causes the removal of the suffix and the control moves to the next step. If the conditions of a certain rule in a current step cannot be met, the conditions of the next rule in that step are tested, until a rule is fired or the rules in that step are exhausted. This process continues for all five steps.

2.3 Porter Stemmer for Bahasa Indonesia

As mentioned at the beginning of this chapter, the Porter stemmer was chosen based on the consideration that its main idea fits the morphological structure of words in Bahasa Indonesia. The morphological structure of words in Bahasa Indonesia consists of a combination of smaller and simpler inflectional and derivational structure, where each is made-up of simpler and smaller suffixes and/or prefixes. This seems to fit the basic idea of the Porter algorithm. The series of linear steps in the Porter stemmer, which simulate the inflectional and derivational process of words in English also fits the derivational and inflectional structure of Bahasa Indonesia (Definition 2.3). These series of linear steps hopefully will reduce a word with complex structure in Bahasa Indonesia to a correct stem. The basic design of the Porter stemmer in Bahasa Indonesia is illustrated in Figure 2.1.

2.3.1 Implementation

Our implementation of the Porter algorithm is based on the English Porter Stemmer developed by Frakes [10]. This version is more readable because Frakes made a clear separation between substitution rules and procedures for testing the attachment conditions.

Because English and Bahasa Indonesia come from two different class of languages, some modifications had to be performed in order to make Porter's algorithm suitable for Bahasa Indonesia. The modifications consist of modifications in the cluster of rules and the *measure* condition. Since Porter's algorithm can only do suffix stripping, some additions have to be done also for handling

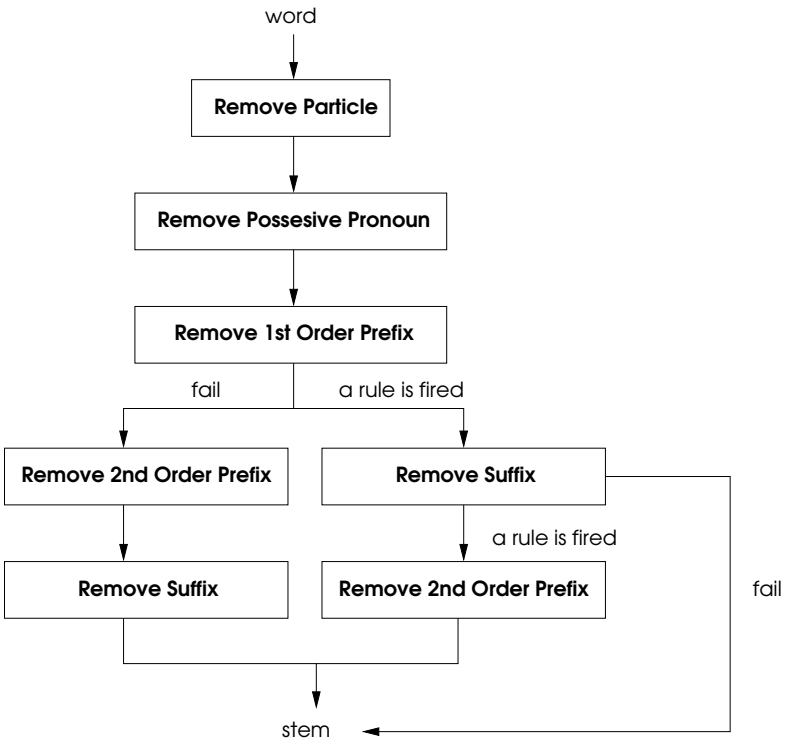


Figure 2.1: The basic design of a Porter stemmer for Bahasa Indonesia.

prefix stripping, confix stripping, and also spelling adjustment in the case where dilution of the first character of the root word had occurred.

Affix-rules

Based on the morphological analysis in Section 2.1, five affix-rule clusters were created for our Porter stemmer for Bahasa Indonesia. These five clusters are defined by reversing the order in which the affixes occur in the word formation process (see Definition 2.3). This means that the inflectional suffixes, i.e. particles and possessive-pronouns, are removed first before the derivational affixes. The five affix-rule clusters are shown in Table 2.3, Table 2.4, Table 2.5, Table 2.6 and Table 2.7.

Table 2.3: The first cluster of rules which covers the inflectional particles.

Suffix	Replacement	Measure Condition	Additional Condition	Examples
kah	NULL	2	NULL	bukukah → buku
lah	NULL	2	NULL	adalah → ada
pun	NULL	2	NULL	bukupun → buku

Measure Condition

In order to cope with the spelling of Bahasa Indonesia, the *measure* condition, which is used in Porter's algorithm, is modified. In Bahasa Indonesia, the smallest unit of a word is *suku kata*

Table 2.4: The second cluster of rules which covers the inflectional possessive pronouns.

Suffix	Replacement	Measure Condition	Additional Condition	Examples
ku	NULL	2	NULL	bukuku → buku
mu	NULL	2	NULL	bukumu → buku
nya	NULL	2	NULL	bukunya → buku

Table 2.5: The third cluster of rules which covers the first order of derivational prefixes

Prefix	Replacement	Measure Condition	Additional Condition	Examples
meng	NULL	2	NULL	mengukur → ukur
meny	s	2	V...*	menyapu → sapu
men	NULL	2	NULL	menduga → duga menuduh → uduh
mem	p	2	V...	memilah → pilah
mem	NULL	2	NULL	membaca → baca
me	NULL	2	NULL	merusak → rusak
peng	NULL	2	NULL	pengukur → ukur
peny	s	2	V...	penyapu → sapu
pen	NULL	2	NULL	penduga → duga penuduh → uduh
pem	p	2	V...	pemilah → pilah
pem	NULL	2	NULL	pembaca → baca
di	NULL	2	NULL	diukur → ukur
ter	NULL	2	NULL	tersapu → sapu
ke	NULL	2	NULL	kekasih → kasih

* *This notation means that the stem starts with a vowel.*

Table 2.6: The fourth cluster of rules which covers the second order of derivational prefixes

Prefix	Replacement	Measure Condition	Additional Condition	Examples
ber	NULL	2	NULL	berlari → lari
bel	NULL	2	ajar	belajar → ajar
be	NULL	2	K*er...	bekerja → kerja
per	NULL	2	NULL	perjelas → jelas
pel	NULL	2	ajar	pelajar → ajar
pe	NULL	2	NULL	pekerja → kerja

* *This notation means that the stem starts with a consonant.*

(syllable). A syllable comprises at least one vowel. Some examples of the adapted measure for Bahasa Indonesia can be seen in Table 2.8

The word measure which is designed here cannot capture all the actual measure of words in Bahasa Indonesia. This is because Bahasa Indonesia also recognizes *diphthongs*, that is a sequence of two vowels which is considered as a non-separable vowel. There are *ai, au, oi* diphthongs, e.g: *pantai* (beach) consists of two syllables *pan* and *tai*.

These diphthong forms are problematic, especially for the diphthongs *ai* and *oi* when they occur at the end of a word. It is difficult to separate it automatically from derivational words with suffix *-i*, such as *tandai* (to give a sign), which consists of three syllables, i.e. *tan*, *da* and *i*. Since the

Table 2.7: The fifth cluster of rules which covers the derivational suffixes

Suffix	Replacement	Measure Condition	Additional Condition	Examples
kan	NULL	2	prefix $\notin$ {ke, peng}	tarikkan → tarik (meng)ambilkan → ambil
an	NULL	2	prefix $\notin$ {di, meng, ter}	makanan → makan (per)janjian → janji
i	NULL	2	V|K$\ldots c_1c_1$, $c_1 \neq s, c_2 \neq i$ and prefix $\notin$ {ber, ke, peng}	tandai → tanda (men)dapati → dapat pantai → panta

Table 2.8: Examples of syllables in Bahasa Indonesia words.

Measure	Examples	Syllables
0	kh, ng, ny	kh, ng, ny
1	ma, af, nya, nga	ma, af, nya, nga
2	maaf, kami, rumpun kompleks	ma-af, ka-mi, rum-pun, kom-pleks
3	mengapa, menggunung, tandai	meng-a-pa, meng-gu-nung, tan-da-i

number of words with diphthong is smaller than the number of words with suffix -*i*, diphthongs are ignored. This causes words with diphthongs *ai*/*oi* to be treated as derivational words. The last character (-*i*) will be removed as the result of stemming process.

Based on the raw data collected by Nazief [22] and data from our own experiment of stopwords, an analysis on syllables in Bahasa Indonesia had also been conducted. This analysis is performed automatically with manual correction and checking. The result of the analysis showed that most of root words in Bahasa Indonesia consist of minimum of two syllables. This is the reason that the minimum length of the stemmed word is two.

Prefix-stripping and Spelling Adjustment

Prefix stripping is handled by treating it just like suffix stripping, with reverse replacement, that is at the beginning of the word. Since the prefix attachment might in some cases change the spelling of the attached word, spelling correction/adjustment must be performed. There is a difficulty in the implementation of the spelling correction since some rules in the derivational structure of Bahasa Indonesia themselves lead to ambiguity (see Appendix A). For example, take the prefix *meng-*, a derived word *mengubah* (changing) may originate from *ubah* (to change) or *kubah* (dome). Meanwhile the word *mengalah* (to give up) may originate from *kalah* (to loose), or *alah* (to dry out). Therefore the spelling adjustment for these ambiguity rules are neglected. We realize that this may lead to overstemming/understemming errors.

The spelling adjustment for non-ambiguous rules are done directly by substituting the prefix with the proper character for that prefix and its stem. The rules in Table 2.5 and Table 2.6 are ordered in such a way that the spelling adjustment for each prefix removal can be accommodated properly.

Confix and Double Prefix Stripping

The confix stripping case is handled in the main algorithm by arranging a consecutive sequence of prefix and suffix replacements. The prefix stripping is always prior to the suffix stripping. An additional condition is added to check the possibility of a suffix to form a legal confix combination

with the previously removed prefix. A suffix rule cannot be applied if its additional requirement is not fulfilled.

By neglecting the inflectional suffixes, there are five possible forms of a derived word, i.e. prefix only, suffix only, confix word, prefix of an already confixed word, or confix of an already prefixed word. The first three possibilities actually can be handled by a sequence of prefix and suffix replacement and the additional condition of legal confixes. The last two possibilities are actually double prefixes. They can be handled by adding another prefix stripping or confix stripping, which is dependent on the previous prefix and suffix replacement.

Chapter 3

Evaluation of the Stemming Algorithm

Before stemming is used for retrieval purposes, we want to evaluate the quality of the two stemming algorithms. The purely rule-based stemmer often yields a stem which cannot be considered to be comprehensible words, especially in Malay [2], while the linguistically-motivated (dictionary-based) stemmer can eliminate most of these errors [2, 17]. Therefore we need to perform an experiment to compare the quality of those two stemmers. This evaluation will hopefully give some information of how "good" or "bad" a purely rule-based stemming algorithm is, compared to a linguistically-motivated stemmer.

3.1 Stemmer Quality Evaluation

Out of various methods to evaluate the quality of a stemmer [10, 24, 25], we chose the Paice evaluation method [25]. In this evaluation method, the quality of the stemmer is assessed by counting the number of identifiable errors during the stemming process. The input words from various samples of texts have to be semantically grouped.

Ideally, a good stemmer will stem all words from the same semantic group to the same stem. But due to the irregularities which are prominent in all natural languages, all stemmers unavoidably make mistakes, including the ones which use vocabulary lists. In other words, we might say that no stemmer can be expected to work perfectly correct.

There will always exist error cases where words which ought to be merged will not be merged to the same stem (*understemming*) or cases where words are merged to the same stem while they are actually distinct (*overstemming*). A good stemmer should obviously produce as few overstemming and understemming errors as possible. By counting these errors for a sample of texts, we can gain some insight in the functioning of a stemmer. A comparison between two different stemmers is then possible.

3.1.1 The Paice Evaluation Method

Paice [25] defined three classes of relationship between pairs of words. These classes are defined as follows:

Type 0 Two words are identical in forms, they are already conflated. By ignoring the possibility of homographs, this kind of word is of no interest.

Type 1 Two words are different in form, but are semantically equivalent.

Type 2 Two words are different in form and are semantically distinct.

Using this relationship definition, a good stemming algorithm is defined as one which can conflate Type 1 pairs as many as possible, whilst conflating as few Type 2 pairs as possible. Paice then quantified the understemming and overstemming error using two new parameters called *Understemming Index* (UI) and *Overstemming Index* (OI). The Understemming Index (UI) is defined as the proportion of Type 1 pairs which are unsuccessfully merged by the stemming algorithm. The Overstemming Index (OI) is defined as the proportion of Type 2 pairs which are merged by the stemming procedure.

If all words from the sample texts are grouped semantically (as demanded by the definition of word relationship) then for a certain semantic group g, the desire to merge all of the words within that group is defined as

$$DMT_g = 0.5\ N_g\ (N_g - 1) \tag{3.1}$$

where N_g is the number of words in the group g. For a group which contains only one form, the DMT value for that group is 0 since no pairs can be formed. The desired merge value for all of the groups in the sample texts is called the *Global Desired Merge Total* and is defined as:

$$GDMT = \sum_{i \in n_g} DMT_{g_i} \tag{3.2}$$

where n_g is the total number of semantic groups in the sample texts.

After the stemming process, all of the words will have been reduced to their stems. In a non-fully conflated group, there will be more than one form of stem within the group. This means that not all of the words in that group are conflated to the same stem, the stemming algorithm is unable to merge those words. The inability of a stemmer to merge all of the words in a certain semantic group g to the same stem is quantified by a parameter which is called the *Unachieved Merge Total*,

$$UMT_g = 0.5 \sum_{i \in [1..f_g]} n_{g_i}\ (N_g - n_{g_i}), \tag{3.3}$$

where f_g is the number of distinct stems in the semantic group g, and n_{g_i} is the number of words in that group which are reduced to stem i.

The unachieved merge total value for all groups in the sample text is called the *Global Unachieved Merge Total*,

$$GUMT = \sum_{i \in n_g} UMT_{g_i} \tag{3.4}$$

Using Eq. 3.2 and Eq. 3.4, the Understemming Index (UI) can be redefined as follow:

$$UI = \frac{GUMT}{GDMT} \tag{3.5}$$

A stemmer might transform many pairs of words which originated from different semantic groups into identical stems. Every stem now defines a stem group whose members might be derived from a number of different semantic groups. If all items of a certain stem group were derived from the same original semantic group, then the stem group contains no error; conversely if a certain stem group contains members which are derived from different semantic groups, this means that "wrongly-merged" has occurred. The number of these wrongly-mergeds within a certain stem group s, which contains stems that are derived from f_s different original semantic groups, is called the *Wrongly-Merged Total*,

$$WMT_s = 0.5 \sum_{i \in [1..f_s]} n_{si} \ (N_s - n_{si}), \tag{3.6}$$

where N_s is the total number of items in the stem group s, n_{si} is the number of stems which are derived from the i^{th} original semantic group. The number of wrongly-merged for all words in the sample texts after the stemming process is called *Global Wrongly-Merged Total*,

$$GWMT = \sum_{i \in n_s} WMT_{s_i} \tag{3.7}$$

where n_s is the number of stem groups as the results of the stemming.

Every word within a certain semantic group has a possibility to be conflated (by a stemming algorithm) with words from a different semantic group, which should be avoided. For a certain group g, this number is called the *Desired Non-merged Total*,

$$DNT_g = 0.5 \ N_g(W - N_g) \tag{3.8}$$

where W is the total number of words in the sample texts. The possible number for the whole words in the sample texts is called *Global Desired Non-Merge Total* and defined as:

$$GDNT = \sum_{i \in [1..N_g]} DNT_g \tag{3.9}$$

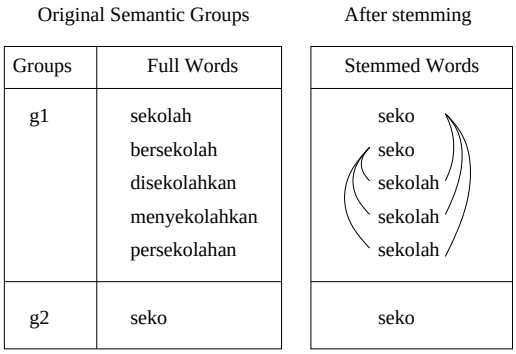
Just like the Understemming Index (UI), the Overstemming Index (OI) can be redefined using Eq. 3.7 and Eq. 3.9 as:

$$OI = \frac{GWMT}{GDNT} \tag{3.10}$$

The ratio between OI and UI is called the *Stemming Weight* (SW), which is used as the parameter to measure the strength of a stemmer. This parameter ranges from weak (indicated by a low value) to strong (indicated by a high value). Figure 3.1 illustrates how this evaluation method works.

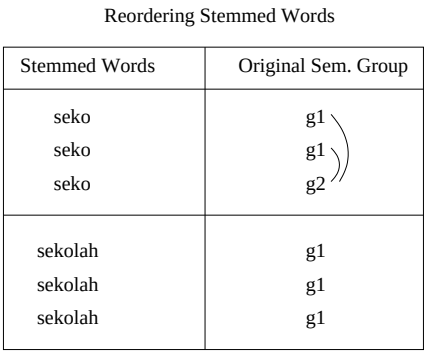
3.1.2 The Paice Experimental Results

The evaluation used sample texts taken from *Kamus Elektronik Bahasa Indonesia* (KEBI), an online digital dictionary built by *Badan Penelitian dan Pengembangan Teknologi* (BPPT), an Indonesian government organization which is responsible for research and technology development (http://nlp.aia.bppt.go.id/kebi/). This dictionary is chosen because it fulfills the prerequisite of the Paice evaluation method. In this dictionary, words are linguistically grouped according to their roots, and the assessment of grouping is done manually. This dictionary consists of 8550 root words and 14200 derivational words. Repetitions, e.g. *berlari-lari*, were removed because



(a) semantically grouped words

(b) after stemming process, UI=0.6



(c) reordered stemmed words into stem group, OI=0.4

Figure 3.1: Illustration of Paice evaluation methods.

they contain a non-word character ('-'). Homographs were also removed to fulfil the prerequisite of Paice's evaluation method, i.e., the input words do not contain duplicates [25].

For a linguistically-motivated stemmer, the dictionary size plays an important role in the stemming process. Therefore we would also like to know what the effect of the size of the dictionary is compared to the purely rule-based stemmer, especially in the case of a developing language such as Bahasa Indonesia, where new words are continuously being adopted from regional or foreign languages.

To see to which extent the size of the dictionary will effect the quality of a dictionary-based stemmer, we have done several experiments by reducing the dictionary size of the Nazief stemmer [23]. Instead of finding some new words, that are not listed in the list of the Nazief stemmer lemmas, we preferred to reduce the size of that list of lemmas. The deleted lemmas are considered as new words, which are not recognized by the dictionary. These deleted lemmas are chosen randomly. The minimum reduction is 10% and the maximum is 90% of the original dictionary size.

The results of the experiments can be seen in Table 3.1. As can be expected, the Nazief stemmer [23], which uses a full dictionary list, performed better than the Porter stemmer. And as can

be seen from Figure 3.2, the Nazief stemmer resides closer to the origin[1] than our Porter stemmer. This is of course an acceptable result, since the Nazief stemmer comes with dictionary with 30528 words, which is larger than the size of the KEBI dictionary. But this size is only 39% of the size of the complete printed dictionary [8].

Table 3.1: Comparison of two Bahasa Indonesia stemmers.

Stemming Algorithm	UI	OI$\times 10^{-6}$	SW$\times 10^{-6}$	Time (ms)
Porter	0.262	8.44	32.27	486
Nazief	0.09	3.60	40.85	741
Nazief (10% reduced)	0.165	3.92	23.75	715
Nazief (20% reduced)	0.31	4.46	14.41	696
Nazief (30% reduced)	0.384	4.52	11.78	668
Nazief (40% reduced)	0.47	4.73	10.17	642
Nazief (50% reduced)	0.55	4.74	8.70	650
Nazief (60% reduced)	0.62	4.71	7.61	635
Nazief (70% reduced)	0.72	3.52	4.89	583
Nazief (80% reduced)	0.82	3.24	3.97	589
Nazief (90% reduced)	0.91	2.13	2.35	564

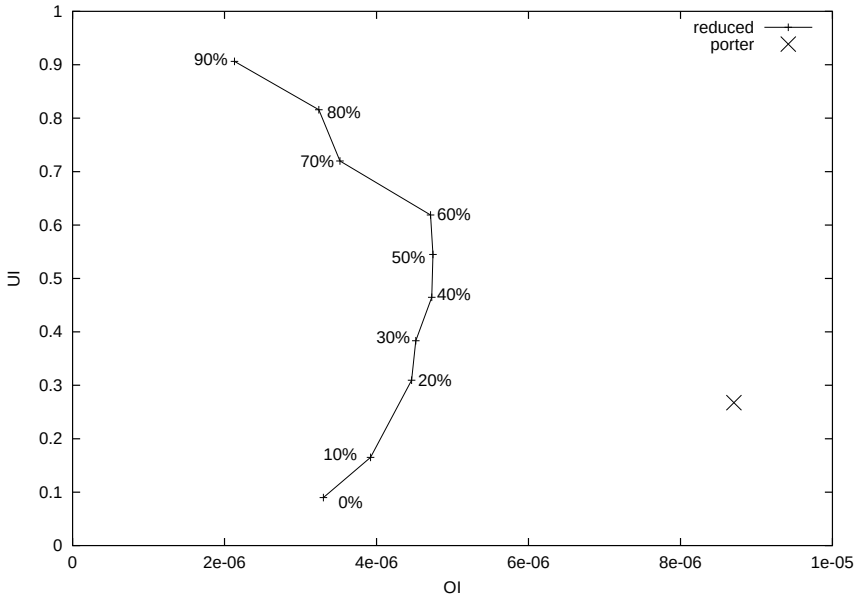


Figure 3.2: UI x OI plot

Obviously, there are hardly overstemming errors in the linguistically-motivated stemmer (Nazief). The experimental results show that the OI values are still low even when the dictionary is already being reduced. In contrast, our Porter stemmer for Bahasa Indonesia tends to make more overstemming errors. This tendency can be explained by the characteristic of Porter stemmer, i.e., it removes the first longest matched string at each step. Meanwhile, most of the prefixes and suffixes forms are substring of each others. For example, the prefix *me-* is a substring of the prefix *mem-* in the word *memakan* (to eat). Our Porter for Bahasa Indonesia will remove the prefix *mem-* from that word and leaves the words *akan* as a stem, although the correct stem is *makan* (to eat). This will be explained in more detail in the Section 3.2.

As the size of the dictionary became smaller, many words were not stemmed by the Nazief stemmer.

[1] A "good" stemmer will lie closely to the origin.

Many understemming errors have been made, as shown by the increasing values of UI in Table 3.1. This means that the Nazief stemmer really depends on the completeness of the dictionary. Since Bahasa Indonesia is in its development such that it keeps adapting new words, a complete digital dictionary can be considered as something expensive. Considering this fact and the result of this experiment, the usage of linguistically-motivated stemmer, such as Nazief stemmer, in Bahasa Indonesia for practical purposes is questionable. We still need to find out whether the linguistically-motivated stemmer can be useful such that it can improve the retrieval performance.

3.2 Error Analysis

Our error analysis is conducted by analyzing the results of both stemmers for each type of word structures, i.e., the inflectional structure and the derivational structure.

3.2.1 Inflectional Structure

Both stemmers perform well for stripping the inflectional suffixes from a word. In most cases, they stripped inflectional words correctly. Table 3.2 shows some results of stripping inflectional suffixes.

Table 3.2: Results of stripping inflectional suffixes.

	Words	Stems	Inflectional Suffix 1	Inflectional Suffix 2
Porter	bukunya	*buku* (book)	nya	-
	bukukah	*buku*	-	
	bukunyakah	*buku*	nya	kah
	dibukukannya	*dibukukan*	nya	-
Nazief	bukunya	*buku* (book)	nya	-
	bukukah	*buku*	-	
	bukunyakah	*buku*	nya	kah
	dibukukannya	*dibukukan*	nya	-

There were some cases when errors emerged in our Porter stemmer. These cases arose because there exist a word w, which comprises of two substrings w_1 and w_2. The substring w_1 consists of more than two syllables and $w_2 \in \{\text{inflectional suffixes}\}$. The stemmer mistakenly stemmed the substring w_2, which is actually part of the root word of w. The most frequent cases especially happened if there was a prefix attached to the root word of w. In other word, the substring w_1 contains a prefix. Examples of these cases are shown in Table 3.3.

The Nazief stemmer may also produce the same kind of error stems, although correct stems were listed in its dictionary. Similar with the Porter, these errors occurred when a word comprise of substrings w_1 and w_2, where $w_2 \in \{\text{inflectional suffixes}\}$ and w_1 contains prefix and a word which is included in the dictionary (see Table 3.3).

3.2.2 Derivational Structure

For this structure, our Porter stemmer produces more errors compared to the Nazief stemmer for the same input words. Some examples are shown in Table 3.4. The causes of these errors can be divided into three categories.

Table 3.3: Errors in the inflectional suffix stripping.

	Words	Prefix	Stem	Inflectional Suffix	Actual Root
Porter	*bersekolah* (school)	ber	*seko* (spy)	lah	*sekolah* (school)
	majalah (magazine)	-	*maja* (kind of tree)	lah	*majalah* (magazine)
Nazief	*bersekolah* (school)	ber	*seko*	lah	*sekolah* (school)

The first error category is occurred if there is a substring w in a root, such that $w \in \{\text{prefixes}\} \cup \{\text{derivational suffixes}\}$ and the root consists of more than two syllables. Examples of this type of error are listed in the first two rows of Porter part in Table 3.4.

The second error category is caused by the stripping mechanism, i.e., the removal of the longest possible match. This mechanism causes errors since most of the prefixes and suffixes are substrings of each other. For example, the prefix *meng-* with its various forms viz. *me-*, *men-*, *mem-*, *meny-*, and *meng-*, are substrings of each others. Suffixes *-kan* and *-an* are substrings one of each other even though one of them is not the various form of the other. The last three rows of Porter part in Table 3.4 show this kind of error. The Nazief stemmer also suffers from this kind of error, but it is because of its shortest possible match. In the case of the Nazief stemmer, this case happened especially with the infixes *-an and -kan* (the last rows of Table 3.4).

The last type of errors occurred because of the difficulty in the implementation of derivational rules for Bahasa Indonesia, that contain ambiguities. Both stemmers suffer from this kind of errors, but of course the Porter stemmer suffers more than the Nazief stemmer. Some examples of these errors are shown in Table 3.5.

Table 3.4: Results of derivational prefix and suffix stripping.

Stemmer	Words	Prefix	Stem	Suffix	Actual Root
Porter	*naluri* (instinct)	-	nalur	i	naluri
	perahu (boat)	per	ahu	-	perahu
	bentrokan (clash)	-	bentro	kan	*bentrok* (to clash)
	perbaikan (improvement)	per	bai	kan	*baik* (good)
	berkedudukan (located)	ber	*kedudu*	kan	*duduk* (to sit)
Nazief	*naluri* (instinct)	-	naluri	-	naluri
	perahu (boat)	-	perahu	-	perahu
	bentrokan (clash)	-	bentrok	an	*bentrok*
	perbaikan (improvement)	per	*baik*	an	baik
	berkedudukan (located)	ber	*keduduk* (a kind of plant)	an	*duduk*

Table 3.5: Spelling adjustment errors in stripping suffixes.

Words	Prefix	Stem	Derivational Suffix	Actual Root
mengalahkan (defeating)	meng	alah	kan	*kalah* (to defeat)
mengobarkan (to fire someone up)	meng	*obar*	kan	*kobar* (to inspire)
mengupas (peeling)	meng	*upas* (security guard)	-	*kupas* (to peel)

Chapter 4

Stemmer Performance Evaluation for Information Retrieval

In this chapter we evaluate the performance of the two stemmers introduced before in the setting of Information Retrieval. We used a non-stemming system as the baseline of this evaluation. In the next four sections, we explain the environments of the experiments. Results and an analysis of the evaluation are given in Section 4.5.

4.1 The Test Collections

4.1.1 The Document Collections

Since there is no document collection in Bahasa Indonesia available in standard collections such as the TREC collection and the CLEF collection, we setup our own collections. We took our documents from two sources, *Kompas*, an Indonesian daily online newspaper (`http://www.kompas.com`), and *Tempo*, an Indonesian daily online news (`http://www.tempo.com`). From these two sources we created two document collections, viz. `kompas` and `tempo`. The `kompas` collection is a two-years headline edition (that is from January 2001 until December 2002). And the `tempo` collection is also a two-years edition (that is from June 2000 until July 2002). Table 4.1 shows the statistics of each collection.

Table 4.1: Test-Collection Statistics

	`kompas`	`tempo`
size (MB)	27.52	45.57
♯ of documents	5449	22944
avg. docs length (byte)	4031.00	1549.59
avg. unique words (terms)	326.09	155.00

Both document collections have been parsed in order to remove all of the HTML tags. These collections have also been transformed into an SGML-like structure. The format follows the overall TREC document structure with two main considerations, viz. easy parsing, so that these documents can easily be used for the purpose of this experiment and for the future expectation,

such that these document collections can help further IR research in Bahasa Indonesia. An example of a document from the `kompas` collection can be seen in Figure 4.1. Manual correction has also been performed to all these document collections.

```
<DOC>
<DOCID> KOMPAS−HL2001−310101−PRES01 </DOCID>
<TITLE> Presiden Bantah Terlibat </TITLE>
<TEXT>
Presiden Abdurrahman Wahid membantah terlibat dalam penyelewengan dana
Yayasan Bina Sejahtera (Yanatera) Badan Urusan Logistik (Bulog)
    ⋮

</TEXT>
</DOC>
```

Figure 4.1: Document example: `kompas` document KOMPAS-HL2001-310101-PRES01.

4.1.2 The Information Requests (Queries)

Both document collections are accompanied with a set of information requests (queries) that are used for the evaluation purposes. Each query is a description of an information need, which is constructed in natural language. The queries construction was done manually by two University of Amsterdam students whose native language is Bahasa Indonesia.

The queries covers widely known events, which had happened in Indonesia during the year each collection covers. Some queries in the `kompas` and `tempo` collections are about the same topics. Queries in the `kompas` are created such that they are longer than queries in the `tempo`. We also fixed the number of queries for each document collection to 35, since this number exceeds the minimum number of topics which are needed for an experiment within the TREC general consensus [5]. The statistics of the queries can be found in Table 4.2.

The queries were also written in an SGML-like format to allow easy access for the purpose of the experiments and for the purpose of defining the relevant sets. The format, which includes a clear description of each query, should help the assessors determine the relevant documents. Figure 4.2 shows an example of a query from the `kompas` collection.

4.1.3 Relevant Documents for Every Information Request

The set of relevant documents for each query is constructed manually by the student that created the query. This manual way is chosen because the collections size are not huge, which makes it possible to do so.

The set of these relevant documents are assessed again by the second student which resulted in a double checking relevant sets. In the case that these two assessors have different opinions about the relevance of a certain document for a certain query, the document is then considered as

<QRY>
<QRYID> KOMPAS2001−Q−2 </QRYID>
<TITLE> TKI ilegal di Malaysia </TITLE>
<RQST>
Masalah Tenaga Kerja Indonesia (TKI/TKW) ilegal di Malaysia
</RQST>

<DESC>

Dokumen berisi berita seputar masalah tenaga kerja Indonesia (TKI/TKW) ilegal di Malaysia yang mencuat karena adanya pemberlakuan hukum baru bagi para tenaga kerja ilegal tersebut. Berita pemulangan tenaga kerja ilegal dan usaha Pemerintah RI dalam hal pemulangan tenaga kerja ilegal asal Indonesia. Termasuk juga catatan pengamat tentang masalah tenaga kerja Indonesia (TKI/TKW), terutama masalah TKI di Malaysia akibat pemberlakuan hukum baru tersebut.
</DESC>
</QRY>

Figure 4.2: Query example: query KOMPAS-HL2001-Q-2.

non-relevant. The statistics of the relevant sets for `kompas` and `tempo` collections can be seen in Table 4.2.

Table 4.2: Test-Query Statistics

	`kompas`	`tempo`
♯ of queries	35	35
avg. queries length (word)	8.777	5.2
avg. ♯ unique words	8.63	5.17
avg. ♯ of relevant docs per query	22.657	66.971

4.2 The FlexIR System

All our experiments used the FlexIR information retrieval system. FlexIR is an automatic information retrieval system built at the Universiteit van Amsterdam. This system is based on the vector space model [3] and implemented in Perl [21]. It supports many types of scoring, such as *Precision/Recall*, *Average Precision* and *R-Precision* which were used in this evaluation.

The original design of this system is dedicated for Western-European languages such as English. Therefore some modifications have been performed in order to use the system for Bahasa Indonesia. The weighting scheme is the `Lnu.ltc` scheme [4, 33], fixing the slope to 0.2 and setting the pivot to the average number of unique words per document as in [21].

4.3 Performance Measurements

4.3.1 Precision/Recall

The traditional *Average Precision-Recall* measure is used because it is the standard measurement and it is used extensively in the literature [3, 10, 30]. *Recall* is the proportion of relevant items retrieved, while *precision* is the proportion of retrieved items that are relevant. Equation 4.1 gives the specific definition of these two measurements.

$$\begin{aligned} \text{recall} &= \frac{N_{\text{rr}}}{N_{\text{rel}}} \\ \text{precision} &= \frac{N_{\text{rr}}}{N_{\text{ret}}} \end{aligned} \tag{4.1}$$

where N_{rr} is the number of relevant items retrieved, N_{rel} is the number of relevant items and N_{ret} is the number of retrieved items.

The P-R measurement is based on the average precision at certain recall levels. By assuming that a certain recall level must be attained for every query, the best retrieval system is the one that attains this recall level with the fewest number of non-relevant documents (the highest precision). Although there are some critics of using this measurement [3, 14], from our point of view, this measurement is a nice tool for macro-evaluation of the retrieval systems.

4.3.2 Average Precision

The *Average Precision* is a single value summary. For a certain query, it is computed by averaging all precision values for that query at the relevant document position in the ranking. From its definition, it can be seen that this measurement represents the entire area underneath the recall-precision curve. It is also recommended to be used as a measurement in the general purposes retrieval evaluation [5].

4.3.3 R-Precision

The *R-Precision* is also a single value summary. It is calculated by computing the precision after R documents have been retrieved, where R is the total number of relevant documents for the current evaluated query. This measurement is used, because our document collections consist of a large variety in the number of relevant documents [19].

4.4 Stoplists

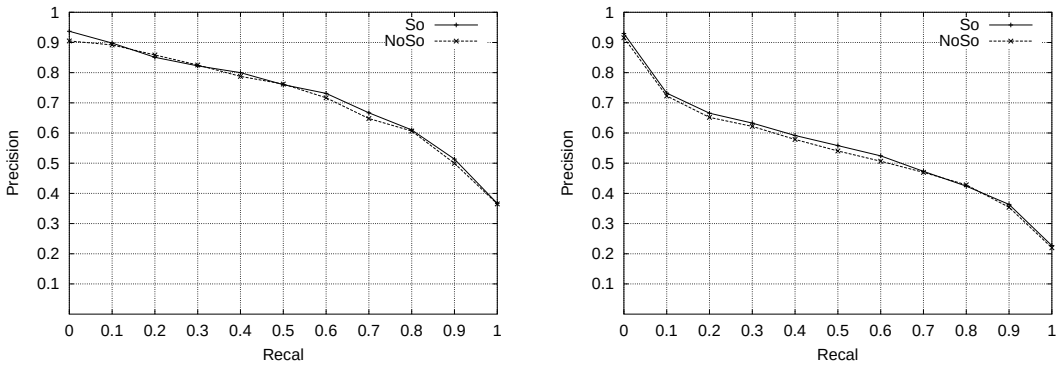
To complete the IR environment, we also propose in this thesis, a new stoplist for Bahasa Indonesia (see Appendix D), because there is no available stoplist for Bahasa Indonesia which can be used. The proposed stoplist is derived from the results of the analysis of word frequencies in Bahasa Indonesia (see Appendix C). It is compared to the result of computational linguistics research in Bahasa Indonesia [22] and with the stoplist in [9].

Before using the the proposed stoplist in the evaluation of stemming effect to retrieval performance, we conducted some experiments to evaluate our stoplist. In these experiments, two systems (for each document collection) were evaluated, viz. the IR system without using either stemmer and stoplist (**NoSo**) and the IR system with stoplist only (**So**).

The results of these experiments are depicted in Figure 4.3 and Table 4.3. For both document collections, the results show that the removal of these stopwords can enhance the precision, especially at low recall levels although not significant. Therefore we can say that the proposed stoplist can be used in the further retrieval evaluation.

Table 4.3: Average Precision and R-Precision results of system without and with stoplist (**NoSo** and **So**)

	NoSo		**So**	
	`kompas`	`tempo`	`kompas`	`tempo`
non-interpolated avg. precision	0.7015	0.5251	0.7101	0.5329
R-Precision	0.6542	0.5168	0.6649	0.5252



(a) **NoSo** vs. **So** for `kompas`

(b) **NoSo** vs. **So** for `tempo`

Figure 4.3: Comparison of Recall-Precision between non stopwords vs. stopwords filtering system.

4.5 Evaluation Results

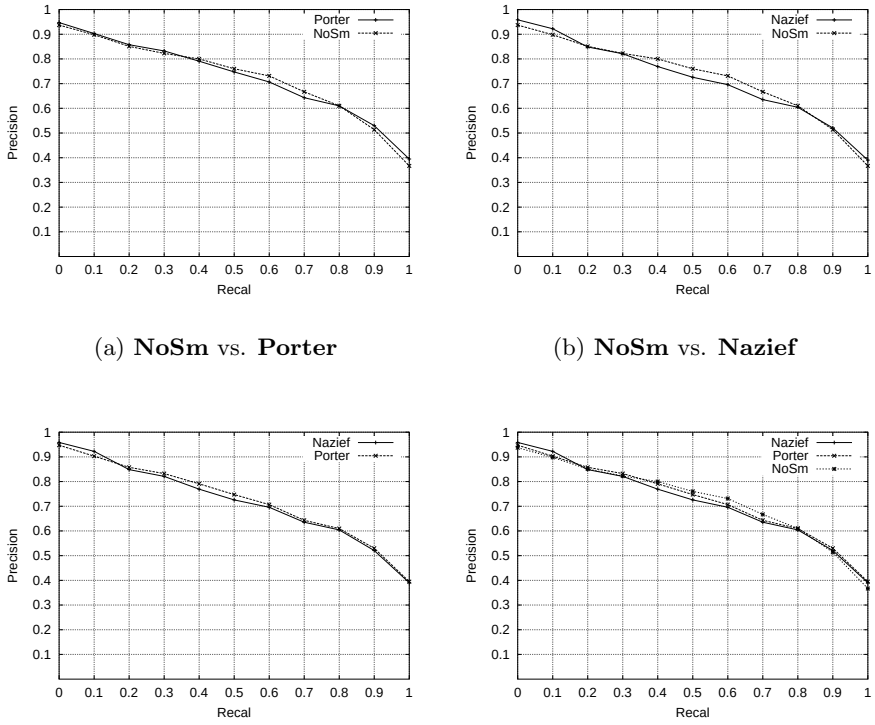
This section describes the experiments which we conducted for evaluating the stemmers effect on the retrieval performance in Bahasa Indonesia. We used all the retrieval environments which have been described in the previous sections. In this evaluation, we contrast the two stemmers with a baseline of no stemming at all. Therefore there are three systems which were evaluated, viz. no stemming at all (**NoSm**), the Nazief stemmer (**Nazief**), and our Porter stemmer for Bahasa Indonesia (**Porter**).

The result of the experiments (for each document collection) can be seen in Figure 4.4, Figure 4.5, and Table 4.4. As can be examined from those two figures and table, the differences of the performance values between the three systems are very small and the results for both document collections show an inconsitency between the stemming systems and the non-stemming system. Therefore at this point we cannot make any conclusion based on the difference of these values.

Table 4.4: Average Precision and R-Precision results over all queries for the three systems

	NoSm		Porter		Nazief	
	`kompas`	`tempo`	`kompas`	`tempo`	`kompas`	`tempo`
non-interpolated avg. precision	0.7101	0.5329	0.7086	0.5456	0.7026	0.5464
R-Precision	0.6649	0.5252	0.6574	0.5403	0.6563	0.5383

In this situation, Hull [15] suggested to perform a statistical testing, which can provide valuable evidence about whether the experimental results have a more general significant differences. The statistical testing and its results are described in Sub Section 4.5.1.



(a) **NoSm** vs. **Porter**

(b) **NoSm** vs. **Nazief**

(c) **Porter** vs. **Nazief**

(d) **NoSm** vs. **Porter** vs. **Nazief**

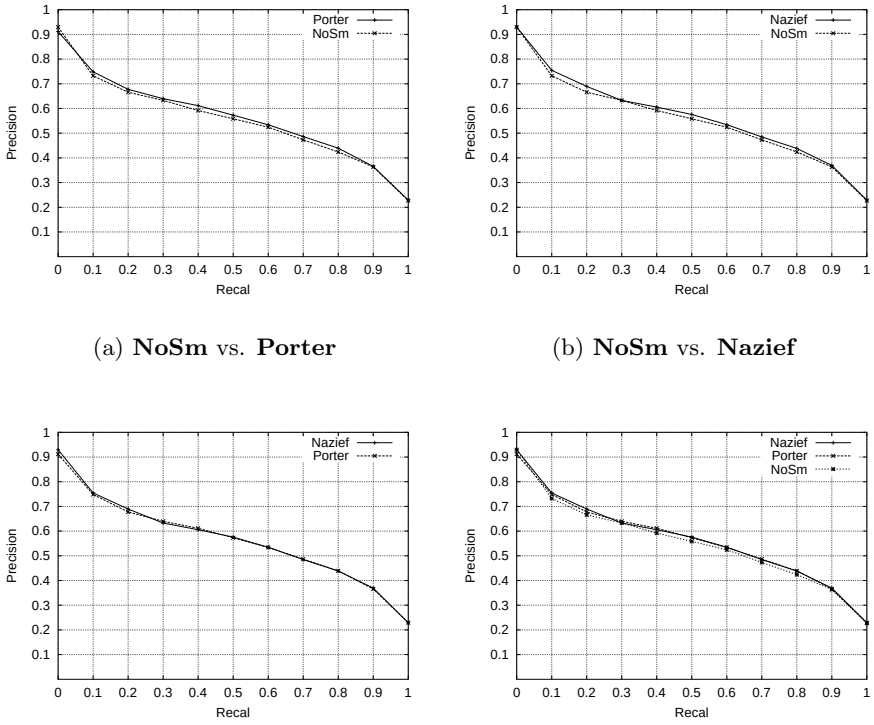
Figure 4.4: PR-Curves for `kompas` Collection.

4.5.1 Statistical Testing

The statistical testing which we conducted here follows the procedure in [14]. The standard statistical model for a certain query and a certain retrieval method is defined as

$$y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \tag{4.2}$$

where y_{ij} is the observed data corresponds to the retrieval performance for query i and method j, μ is the true mean performance, α_i is the query effect, β_j is the retrieval method effect, and



(a) **NoSm** vs. **Porter**

(b) **NoSm** vs. **Nazief**

(c) **Porter** vs. **Nazief**

(d) **NoSm** vs. **Porter** vs. **Nazief**

Figure 4.5: PR-Curves for `tempo` Collection

ϵ_{ij} is the error. The query effect α_i and the method effect β_j are assumed to be independent and additive.

The *Null Hypothesis* (H_0), which is tested, is that the observed stemmer methods are in equal performances. If the p-value is very small, then evidence suggests that the observed statistics reflect an underlying difference between the stemmers.

Because there were three stemmers to be evaluated, the *two-way ANOVA* is used [14, 15, 36]. In the ANOVA, the F-test for $\beta_j = 0$ for all j is defined as

$$F = \frac{MS_{\text{bet.stem}}}{MS_{\text{residual}}} = \frac{\left(\dfrac{n\sum(\bar{y}_j - y)^2}{m-1}\right)}{\left(\dfrac{\sum_{i,j}(y_{i,j} - \bar{y}_i - \bar{y}_j + \bar{y})^2}{(n-1)(m-1)}\right)} \tag{4.3}$$

where $y_{i,j}$ is the observed data, which corresponds to the retrieval performance of method $j = [1 \dots m]$ for query $i = [1 \dots n]$ (m and n are the number of stemmers and queries respectively). The value $\bar{y}_i$ is the average performance of a query i over all methods, while $\bar{y}_j$ is the average performance of a method j over all queries. $MS_{\text{bet.stem}}$ and MS_{residual} are the mean square between stemmers and the residual errors.

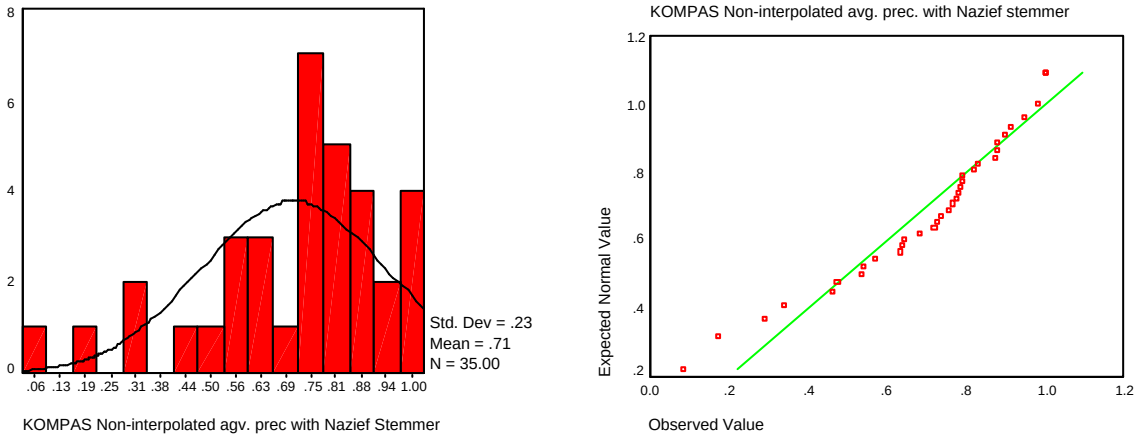
If the F-test is significant (identified by the very small p-value), then the ANOVA multiple comparison is used. The ANOVA multiple comparison is Tukey's studentized range test distribution

under H_0 which is defined as:

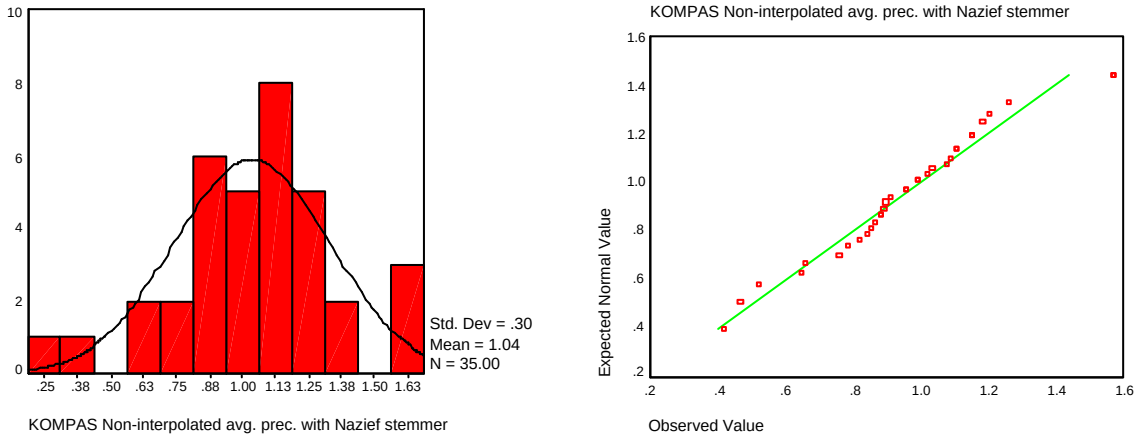
$$|\bar{y}_k - \bar{y}_l| \sim \frac{q^{\alpha}_{m,v}\ s}{\sqrt{n}} \tag{4.4}$$

where $q^{\alpha}_{m,v}$ is the studentized range statistic for $v = (n-1)(m-1)$ at significant level α and $s = \sqrt{MSE}$ (Mean Squared Error). All mean differences between method k ($\bar{y}_k$) with method l ($\bar{y}_l$) that are greater than the value at the right-hand side of Eq. 4.4 are assumed to be significant.

In order to convince ourselves of using the two-way ANOVA, we tested the data by making a quantile plot. One of the example of the quantile plot for data from the Non-interpolated average precision values of the Nazief stemmer can be seen in Figure 4.6(a). The quantile plot shows that the data are skewed. Therefore we transformed the data using the function $f(x) = \arcsin(\sqrt{x})$ as suggested in [14]. As can be seen from Figure 4.6(b), the transformed data already follows the normal distribution, therefore these data could be used for the ANOVA analysis.



(a) The original Non-interpolated average precision values



(b) The transformed Non-interpolated average precision values

Figure 4.6: Quantile Plots from Non-interpolated average precision values of Nazief for the `kompas` collection

Table 4.5: ANOVA Table for Average Precision Measurement

Doc. Coll	Source	Sum of Sq.	df	Mean of Sq.	F	p
`kompas`	Stemmer	0.0016	2	0.0008	0.3481	0.7072
	Query	8.6742	34	0.2551	112.2208	$6.99485E-48$
	Residual	0.1546	68	0.0022		
`tempo`	Stemmer	0.0053	2	0.0027	2.3298	0.1050
	Query	6.4746	34	0.1904	167.4948	$1.13734E-53$
	Residual	0.0773	68	0.0011		

Table 4.6: ANOVA Table for R-Precision Measurement

Doc. Coll	Source	Sum of Sq.	df	Mean of Sq.	F	p
`kompas`	Stemmer	0.0021	2	0.0010	0.4455	0.6424
	Query	7.5064	34	0.2208	94.6643	$1.93728E-45$
	Residual	0.1586	68	0.0023		
`tempo`	Stemmer	0.0054	2	0.0027	2.6641	0.0769
	Query	5.1979	34	0.1529	151.2662	$3.41583E-52$
	Residual	0.0687	68	0.0010		

The ANOVA tables of our experiments for average precision and R-precision performance can be seen in Table 4.5 and Table 4.6. By taking p-value less than 0.05 for rejecting the H_0, from the ANOVA results of the stemmers effect for both document collections, we can say that we accept the H_0. This means we accept that the three systems are equal in the average precision and R-Precision performance.

Since the statistical test was unable to detect the significant difference, we conducted a detailed analysis by examining a number of individual queries and their stems to gain more information about why the significance tests failed. This detailed analysis is explained in Sub Section 4.5.2

4.5.2 Detailed Analysis

At this point, it was not clear to us why the statistical test was unable to detect a significant difference between the three systems. Therefore we conducted a detailed analysis of some queries by examining their stem results and some of the retrieved documents within those queries. We used the queries from both document collections and discuss several cases where the performance of one system is equals or outperforms the other two systems.

`kompas` **Q.1: "Kasus penyalahgunaan dana Yanatera Bulog dan Memorandum DPR"**
The fraud of Yanatera Bulog Budget and Parliament's Memorandum

The Nazief stemmer did not recognize that the words *penyalahgunaan* (fraud), *disalahgunakan* (misuse - passive form), and *menyalahgunakan* (misuse) are related. It also did not recognize that these words are also related to the phrase/compound *salah gunakan* and *salah guna*. *Penyalahgunaan* is a compound word. This compound is originated from the phrase/compound *salah guna*. This kind of compound is a phenomenon that might exist in Bahasa Indonesia, because there is a rule which governs the phrase with both prefix and suffix to be written unseparately [6]. Since the stem *salahguna* is not in its lexicon, it did not stem any words in this query, such that its

performance is equal to the non-stemming system.

Our Porter stemmer converted the derivational compound *penyalahgunaan*, *menyalahgunakan*, and *disalahgunakan* to the same stem *salahguna*. The usage of this stem made more relevant documents being retrieved so that its performance is better than that of the two other systems. But without further splitting, it also failed to recognize that those compounds are also related to the phrase *salah gunakan* and *salah guna*.

`kompas` Q.2: "Kasus kerusuhan antar agama di Poso dan penanganannya"
Religious conflict in Poso and its solution

Both Nazief and Porter stemmer converted the word *penanganannya* (the process of handling) to *tangan* (hand). This conversion is a bad decision since the word *tangan* is a common word in Bahasa Indonesia. This word can be combined with various words to form compound words, such as *campur tangan* (involvement), *tanda tangan* (signature), which are contained in many documents in this collection. Both stemmers also converted the word *kerusuhan* (turmoil) to *rusuh* (restless). This conversion is unhelpful since the resulting word *rusuh* is an adjective which is considered as non-important term for information retrieval. Therefore the performance of both stemming systems are below the non-stemming system.

`kompas` Q.3: "Pelaksanaan sidang istimewa MPR meminta pertanggungjawaban Presiden Abdurrahman Wahid. Dekrit Presiden"
The extraordinary session of People's Consultative Assembly (MPR) to ask President's responsibility. The Decree of President

Just in the case of `kompas` Q.1, the Nazief stemmer failed to recognize that the compound words *pertanggungjawaban* (responsibility), *mempertanggungjawabkan* (to account for) and *dipertanggungjawabkan* (to account for - passive form) are related to the phrase *bertanggung jawab* (to be responsible) and *tanggung jawab* (responsibility). Since the stem *tanggungjawab* is not in its lexicon, it left the word *pertanggungjawaban* as it is. However, it successfully recognized that the words *pelaksanaan* (implementation), *melaksanakan* (to perform), *dilaksanakan* (being performed), and *pelaksana* (executor) are related and stemmed them to one stem that is *laksana*. The usage of this stem could pull out more relevant documents such that its performance is better than the non-stemming systems.

The Porter stemmer successfully recognized that the words *pertanggungjawaban*, *mempertanggungjawabkan*, *dipertanggungjawabkan* are related. It stemmed them to the same stem *tanggungjawab*, but without further splitting process, it failed to recognized that those compounds are also related to the phrase *bertanggung jawab* and *tanggung jawab*. Therefore it gained only small benefit. Just as the Nazief stemmer, our Porter stemmer also recognized that words *pelaksanaan*, *pelaksana*, *melaksanakan* and *dilaksanakan* are related and stemmed them to *laksana*. The usage of these two stems made its performance outperform the other two systems.

`kompas` Q.4: "Konflik bersenjata Aceh, Gerakan Aceh Merdeka (GAM) dan penanganannya"
Armed conflict in Aceh, Free Aceh Movement and its solution

The Nazief stemmer converted the word *gerakan* (movement) to *gerak* (to move), where actually the word *gerakan* is part of an organization name (proper name) and should not be converted.

Similar to the case of `kompas` Q.2, it converted *penanganan* to *tangan*. The Porter stemmer also converted the part of the organization name *Gerakan* to *gera*, and *merdeka* to *erdeka* which should not be converted. It also converted the word *penanganan* to *tangan*. The usage of these stems decreased the performance of both stemmers.

`kompas` Q.5: "Konflik antar etnis Madura-Dayak di Kalimantan"
Madura-Dayak ethnics clashes in Kalimantan

All three systems have the same retrieval performance. The Nazief stemmer did not stem any of the words in this query. The Porter stemmer incorrectly stemmed the word *Kalimantan* (the name of Borneo island) to *Kalimant*, but since the word *Kalimant* does not exist in Bahasa Indonesia, this did not hurt its performance.

`kompas` Q.7: "Kasus penyalahgunaan dana nonbudgeter Bulog yang melibatkan Akbar Tandjung"
The fraud of nonbudgeter budget of Bulog which involves Akbar Tanjung

As in the case of `kompas` Q.1, the Nazief stemmer did not stem the derivational compound word *penyalahgunaan*, whilst our Porter stemmer stemmed it to *salahguna*. The Nazief stemmer wisely recognized that the word *melibatkan* (involving), *terlibat* (involved) and *keterlibatan* (involvement) are related and stemmed them to the same stem *libat* (to involve). Just like Nazief stemmer, our Porter stemmer also recognized that those words except *keterlibatan* are related. In this case, it suffered from understemming error by converting the word *keterlibatan* to *terlibat*. But the Nazief stemmer did not gain great benefit since the resulting stem *libat* is a verb. As verbs cannot be considered as important terms, therefore its performance is slightly lower than our Porter stemmer, but it is still better than the non-stemming system.

`kompas` Q.9: "Kasus penculikan dan pembunuhan ketua Presidium Dewan Papua (PDP) Theys Eluay"
The kidnapping and the murder case of the head of Papua Council Presidium, Theys Eluay

Both the Nazief and Porter stemmer relate the word *penculikan* (kidnapping) with the words *menculik* (to kidnap), *diculik* (being kidnapped) and *penculik* (kidnapper) and stem them to *culik* (to kidnap) which is a verb. Both stemmers also relate the word *pembunuhan* (murder) with the words *membunuh* (to kill), *dibunuh* (killed), and *pembunuh* (killer/murderer) and stem them to *bunuh* (to kill).

This conversion turned out to be an unwise decision, due to the fact that both resulting stems are verbs which cannot be considered as important term for information retrieval. Also the document collection contains many stories about murder and kidnapping which were done by the Free Papua Movement (OPM). Therefore the performance of both stemming systems are below the non-stemming system.

`kompas` Q.18: "Kasus pengambilalihan Semen Padang"
The take over of Semen Padang

This is the same case as `kompas` Q.3, where the word *pengambilalihan* (process of taking over) is supposed to be converted to *ambil alih* (to take over) such that it can be related to other words, i.e.

mengambil alih (taking over), *diambil alih* (took over - passive form). The Nazief stemmer could not stem this query, which caused the stemmed query to be equal to the non-stemmed version. The Porter stemmer converted it to *ambilalih* which is not recognized in Bahasa Indonesia except if it is splitted. Therefore the performance of the three systems are equal.

kompas Q.23: "Perubahan Undang Undang Dasar 1945 menyangkut pemilihan presiden langsung"
The amendment of 1945 State Constitution in the view of direct presidential election act

Both Nazief and Porter stemmers converted the word *perubahan* (alteration) to *ubah* (to change), which is a very common term. The word *perubahan* in the phrase "Perubahan Undang Undang" has a specific meaning in the domain of law, which usually associated to the word *amendment*. Therefore their performance were lower than the non-stemming system.

kompas Q.25: "Kasus penangkapan tiga warga negara Indonesia Tamsil Linrung, Agus Dwikarna dan Abdul Jamal Balfas di Filipina"
The arrest of three Indonesians, Tamsil Linrung, Agus Dwikarna and Abdul Jamal Balfas, in Philippine

All the three systems have the same performance for this query. As we can see, that this query contains many specific names. The usage of specific names might be the cause that the three systems have equal performance.

kompas Q.31: "Kasus peledakan bom yang terjadi di Bali"
The Bali bombing blast

Both Nazief and Porter stemmer converted the word *peledakan* (blast, explotion) to *ledak* (to blast, to explode). Both stemmers do not get benefit from this conversion, since the resulting stem is a verb which cannot be considered as important term for information retrieval. Even, it seems to decrease their performance. The Nazief stemmer also converted the word *Bali* to *bal* (ball). Since *bal* is in its lexicon, it also converts *Balkan* to the same stem. Therefore some of the retrieved documents also contain the story of Balkan. This worsens its performance.

kompas Q.32: "Kasus kontak senjata antara anggota TNI AD dengan Kepolisian di Binjai"
The armed conflict between Indonesian army and Indonesian police force in Binjai

The Nazief and our Porter stemmer both converted the word *Kepolisian* (police force) to *polisi* (police). The word *kepolisian* is a specific word which is related to the police organization. This conversion made the performance of Nazief and Porter stemmer both lower than the non-stemming system.

`kompas` **Q.34: "Kasus sengketa Sipadan dan Ligitan antara Indonesia-Malaysia"**
The dispute between Indonesia and Malaysia over Sipadan and Ligitan islands

This is the same case as queries `kompas` Q.5 and `kompas` Q.6, except that our Porter stemmer incorrectly converted the word *Sipadan* to *sipad* (ear) and the word *Ligitan* to *ligit*. But these two terms did not decrease the performance of our Porter stemmer since the word *sipad*, which comes from Javanese, is rarely used, and *sipad* is not recognized in Bahasa Indonesia.

`tempo` **Q.1: "Kenaikan harga dan subsidi BBM"**
The increase of oil prices and oil's subsidy

Both Nazief and our Porter stemmer stemmed the word *kenaikan* (increase) to *naik* (to increase). This resulting stem is a verb and is a very common term in Bahasa Indonesia. Therefore the usage of this stem made the performance of both Nazief and our Porter stemmer lower than the non-stemming system.

`tempo` **Q.2: "Konflik bersenjata di Aceh"**
Armed conflict in Aceh

Both Nazief and our Porter stemmer converted the word *bersenjata* (armed) to *senjata* (weapon) which is a noun. The usage of this stem can pulled out many relevant documents such that the performance of both stemming systems are better than the non-stemming system. Compare to the Query `kompas` Q.4 from the `kompas` collection, this query is shorter and contains less proper name. The results of both queries show that both stemming systems have better performance for this query than for the Query `kompas` Q.4.

`tempo` **Q.3: "Penyelewengan dana nonbudgeter Bulog"**
The fraud of nonbudgeter budget of Bulog

Both Nazief and our Porter stemmer converted the word *penyelewengan* (fraud) to *seleweng* (to deceive). This conversion turned out to be an unwise decision, since the documents collection contains many stories about corruption in Indonesia.

`tempo` **Q.4: "Kasus Buloggate (dana Yanatera) dan Bruneigate"**
Buloggate (Yanatera budget) and Bruneigate

All the three systems have the same retrival performance. Both Nazief and our Porter stemmer did not stem any words in this query.

`tempo` **Q.18: "Kecelakaan pesawat Cassa NC-212 di Irian Jaya"**
Airplane Cassa NC-212 crashed in Irian Jaya

Both Nazief and our Porter stemmer converted the word *kecelakaan* (crash, accident) to *celaka* (unfortunate). This conversion can be considered as unwise decision, since the stem *celaka* is an adjective which cannot be considered to be helpful in pulling-out more relevant documents. Even it makes the performance of the two stemming systems are below the non-stemming system.

4.5.3 Summary of the Detailed Analysis

We found that the linguistically correct stems, which are produced either by the linguistically-motivated stemmer or by the rule-based stemmer, may not be optimal for retrieval purposes. In this case, the stemming process is harmful.

Similar to what happened in English and Dutch [15, 17, 20], we found many examples where the rule-based stemmer, such as the Porter stemmer, produced non-comprehensible words. Because the morphological rules in Bahasa Indonesia contain many ambiguities, the rule-based stemmer without using any additional knowledge might produce many more non-comprehensible words than rule-based stemmers for other languages. Here, our Porter stemmer produced 11.8% non-comprehensible words in stemming all of words in the queries of `kompas` collection.

From the query analysis, we found examples where the linguistically-motivated stemmer, such as the Nazief stemmer, undesireably stems some words to a word with a very different meaning, even though it is already accompanied by a lexicon.

From our detailed analysis of queries, we found that words which were stemmed have very small number of variations. The average number of derivational variations of a certain word (excluding proper name) from all queries is only about 4.135. This is very small compared to Slovene language [27]. We also found that the number of inflectional and derivational affixes which are handled with these two stemmers are very small compared to the number of affixes which are handled in the Slovone stemmer [28], the Dutch stemmer [17] and the Porter stemmer [29]. Recall to the purpose of stemming as morphological normalization, a stemmer which handles a small number of affixes should also gain a small number of benefit in retrieval performance. This may be the cause of the non-significant difference in performance of both stemming systems compared with the non-stemming system in our experiments.

From analysis of the results of stemming all words in the queries of both documents collections, we see that most of the resulting stemmed words are verbs and adjectives. As verbs and adjectives are less important term for index or search keys than nouns in information retrieval, this might also be the cause that our experiment results show non-significant differences between stemming systems and the non-stemming system.

We can also see that some of the queries consist of derivational compound words.[1] These derivational compound words are not recognized by the Nazief stemmer, hence it did not stem them and left them as they were. In this case the performance of the Nazief stemmer equals to that of the non-stemming system. Whilst for the Porter stemmer, although it could stem these words correctly, it could not get any benefits from it, unless they were further splitted.

Similar to what happened in English [15], stemming process seems to give more benefit for short queries. This can be seen from the results of both documents collections. Although the results of the experiments of stemming and non-stemming systems are not statistically significant, the performance of stemming experiments with the `tempo` collection which has shorter queries is better than the experiments with the `kompas` collection.

We found that some of the queries do not need to be stemmed at all. For the `kompas` collection, the number of this kind of queries are about 29% of all queries. We also found that many of the queries consist of many proper names which are left untouched by both stemmers. These also made the performance of all three systems are equal.

[1] These derivational compound words are exist in Bahasa Indonesia. As stated before, there is a rule which governs the compound words (phrase) to be written unseparated if there is a prefix attached to the first word and a suffix attached to the second word. If only prefix or suffix attached to the first or second word, then they should be written separately.

Chapter 5

Conclusions

After several evaluations of the effet of stemming on retrieval performance in Bahasa Indonesia, we reach a number of conclusions:

1. Our Porter stemmer for Bahasa Indonesia produces many non-comprehensible words which are caused by the ambiguity in the morphological rules of Bahasa Indonesia. In some cases the errors do not hurt performance, but in other cases they decrease the performance. Extending it with a digital dictionary is somewhat a dilemma since a digital dictionary is expensive. In further research, extending the rule-based stemmer with words co-occurence may give better results.
2. Such as in English, the linguistically-motivated stemmer which is developed by Nazief for Bahasa Indonesia, posses two main problems. First, the ability of the stemmer depends on the size of the dictionary. It cannot stem a word which is not in its lexicon. Second, a linguistically correct stems which is produced by this kind of stemmer does not always optimal for the purpose of information retrieval application. Therefore, if it is to be used for IR, this linguistically motivated stemmer should be enhanced with other tool such as domain linguistic analysis or adding a domain specific lexicon.
3. Derivational compounds in Bahasa Indonesia seems to need special treatment in order to get benefit from stemming. Further morphological research needs to be conducted to see whether compound splitting is needed for information retrieval. The derivational compound in Bahasa Indonesia is not as common as in Dutch, and it can only be useful if it is combined with a stemmer. And also a more complex IR system, which recognizes phrases, is required.
4. From our detailed analysis of queries, we found that words which were stemmed have very small number of variations. The average number of derivational variations of a certain word (excluding proper name) from all queries is only about 4.135. This is very small compared to Slovene language [27]. We also found that the number of inflectional and derivational affixes which are handled with these two stemmers are very small compared to the number of affixes which are handled in the Slovone stemmer [28], the Dutch stemmer [17] and the Porter stemmer [29]. Recall to the purpose of stemming as morphological normalization, a stemmer which handles a small number of affixes should also gain a small number of benefit in retrieval performance. This may be the cause of the non-significant difference in performance of both stemming systems compared with the non-stemming system in our experiments.
5. Failure of the statistical significance test in our experiment to detect significant difference does not necessarily mean that there is no difference between systems [14]. We realized that

our corpora are far from perfect due to the fact that these corpora are created and judged only by two persons. We also know that our queries were formulated such that they contain many proper names. Therefore tests on a number of different other corpora (collections) are needed to be performed further.

Appendix A

Derivational Rules of Prefix Attachment

Table A.1: Rules and Variation Forms of Prefixes

Prefix	Variation Form	Rules
meng	meng	+ Vowel|k|g|h. . . , e.g: *ambil* (to take) → *mengambil* (taking) *embun* (vapor) → *mengembun* (to condense) *ikat* (to tie) → *mengikat* (to tie/to bind) *olah* (process) → *mengolah* (processing) *ukur* (to measure) → *mengukur* (measuring) *kurus* (slim) → *mengurus* (become slimmer) *urus* (to take care) → *mengurus* (to take care) *ganggu* (to disturb) → *mengganggu* (disturbing) *hilang* (to lose) → *menghilang* (to dissapear)
	meny	+ s. . . , e.g: *sisir* (comb) → *menyisir* (to comb something)
	mem	+ b|f|p. . . , e.g: *beku* (frozen) → *membeku* (to become frozen) *fitnah* (to accuse) → *memfitnah* (accusing) *pukul*(to hit) → *memukul* (hitting)
	men	+ c|d|j|t. . . *cuci* (to wash) → *mencuci* (washing) *darat* (land) → *mendarat* (landing/docking) *jual* (to sell) → *menjual* (selling) *tukar* (to change) → *menukar* (changing)
	me	+ l|m|n|r|y|w. . . , e.g: *lintas* (to cross) → *melintas* (crossing) *makan* (to eat) → *memakan* (eating) *nikah* (marriage) → *menikah* (to get married) *rusak* (to break) → *merusak* (breaking) *wabah* (epidemic) → *mewabah* (outbreak) *yakin* (sure) → *meyakin(kan)* (to convince someone)
peng	peng	+ Vowel|k|g|h. . . , e.g: *ikat* (to tie) → *pengikat* (something that is used to tie)

continue to next page

continued from previous page

Prefix	Variation Form	Rules
		olah (to process) → *pengolah* (processor) *ukur* (to measure) → *pengukur* (measurement) *urus* (to take care) → *pengurus* (person who take cares) *ganggu* (to disturb) → *pengganggu* (person who disturbs) *halus* (soft) → *penghalus* (softener)
	peny	+ s. . . , e.g: *saring* (to filter) → *penyaring* (filter)
	pem	+ b|f|p. . . , e.g: *baca* (to read) → *pembaca* (reader) *fitnah* (to accuse) → *pemfitnah* (people who accuse) *pukul*(to hit) → *pemukul* (things that is used to hit)
	pen	+ c|d|j|t. . . *cuci* (to wash) → *pencuci* (laundress/laundryman) *datang* (to come) → *pendatang* (the comer) *jual* (to sell) → *penjual* (seller) *tukar* (to change) → *penukar* (changer)
	pe	+ l|m|n|r|y|w. . . , e.g: *lintas* (to cross) → *pelintas* (passerby) *makan* (to eat) → *pemakan* (eater) *rusak* (to break) → *perusak* (destroyer) *warna* (color) → *pewarna* (dye)
ber	bel	+ ajar, eg: *ajar* (to teach) → *belajar* (to study/to learn)
	be	+ r|KVr. . . , e.g: *rencana* (plan) → *berencana* (to have a plan)
		kerja (to work) → *bekerja* (working)
	ber	+ any word which violates conditions of the alomorphs bel and be *tukar* (to change) → *bertukar* (to change, changing)
per	pel	+ ajar, e.g: *ajar* (to teach) → *pelajar* (student)
	pe	+ r|KVr. . . , e.g: *ramal* (to predict) → *peramal* (fortune-teller)
	per	+ any word which violates conditions of the alomorphs pel and pe *kaya* (rich) → *perkaya* (to make richer)
ter	te	+ r. . . , e.g: *rasa* (to feel) → *terasa* (to be felt)
	ter	+ K|V. . . , where K ≠ r, e.g: *atur* (to arrange) → *teratur* (to be properly arranged)

Appendix B

The Meaning of Affixations

Table B.1: The meaning of affixations

Affix	Functions	Examples
meng-	verb to verb form noun to verb form	*makan* (to eat) → *memakan* (to eat, eating) *sisir* (comb) → *menyisir* (to comb)
di-	verb to passive verb form noun to passive verb form	*makan* → *dimakan* (to be eaten) *sisir* → *disisir* (to be combed)
ter-	verb to passive accidental verb form noun to passive accidental verb form	*makan* → *termakan* (to be eaten accidently) *paku* (nail) → *terpaku* (to get nailed accidently)
peng-	noun to noun form verb to noun form adjective to noun form	*tani* (farm) → *petani* (farmer) *baca* (to read) → *pembaca* (reader) *rusak* (damaged, destroyed) → *perusak* (destroyer)
ber-	verb to active verb form noun to active verb form adjective to active verb form	*main* (to play) → *bermain* (to play, playing) *sepeda* (bicycle) → *bersepeda* (to bike/cycling) *gembira* (happy) → *bergembira* (to be excited)
per-	verb to noun form noun to causative verb form adjective to causative verb form	*kerja* (to work) → *pekerja* (worker) *istri* (wife) → *peristri* (to take someone as a wife) *halus* (soft) → *perhalus* (to make softer)
ke-	adjective to noun form	*tua* (old) → *ketua* (leader)
-kan	verb to command verb form noun to command verb form adjective to command verb form	*ambil* (to take) → *ambilkan* (asking someone to take) *sisir* → *sisirkan* (asking someone to comb something) *jauh* (far) → *jauhkan* (asking someone to move something further)
-i	verb to intensive/repetitive verb form noun to intensive/repetitive verb form adjective to commmand verb form	*ambil* → *ambili* (taking something several times) *sisir* → *sisiri* (combing something several times) *jauh* → *jauhi* (asking someone to move further from something)
-an	verb to noun form	*makan* → *makanan* (food, something to be eaten)

Appendix C

Word Frequency Analysis

Word frequency analysis was conducted by performing experiment on Bahasa Indonesia corpus. This experiment used online Indonesia newspapers as text source. One year editions are collected from Online Kompas, http://www.kompas.com, one of the most widely read newspaper in Indonesia. These editions are taken in a consecutive of every day in a year (started from January 2001 until December 2001) with the total of 3160 documents. These documents are only the daily headlines of the newspaper.

The corpus, which is built from this analysis, consists of 50.000 unique words, after removing the names of peoples, cities, organisations, countries, etc. The results of the most frequently occur words can be seen in Table C.1. This list consists of root words and derived words. The number of root words and derived words are still under investigation. The further investigation will be done by using Indonesian Dictionary.

Table C.1: Most frequently occur words

No	Word	Freq.	No.	Word	Freq.	No.	Word	Freq	No	Word	Freq
1	yang	55971	51	lalu	3495	101	dana	2191	151	kali	1587
2	dan	41286	52	kita	3467	102	pukul	2184	152	umum	1584
3	itu	24768	53	kalau	3438	103	bukan	2174	153	ujar	1564
4	tidak	18723	54	belum	3422	104	tetap	2169	154	terus	1539
5	dengan	18281	55	terjadi	3417	105	jika	2128	155	jelas	1528
6	dari	17632	56	besar	3346	106	semua	2122	156	sedang	1502
7	untuk	16886	57	terhadap	3284	107	sama	2110	157	diri	1495
8	dalam	15681	58	kepala	3243	108	waktu	2109	158	memberikan	1488
9	ini	14707	59	masyarakat	3211	109	sejumlah	2086	159	juta	1482
10	akan	12433	60	sampai	3211	110	bank	2083	160	sebelumnya	1473
11	juga	9343	61	sementara	3197	111	polisi	2073	161	masuk	1469
12	pada	9212	62	politik	3197	112	memang	2062	162	hasil	1454
13	ada	8592	63	setelah	3183	113	hingga	2042	163	adanya	1450
14	presiden	8310	64	tak	3177	114	sejak	2019	164	maupun	1447
15	karena	7935	65	antara	3149	115	partai	2017	165	berada	1445
16	bisa	6703	66	lagi	3145	116	baik	2012	166	per	1445
17	sudah	6690	67	ketua	3093	117	sekarang	1991	167	pernah	1442
18	tersebut	6121	68	melakukan	2989	118	sendiri	1965	168	meminta	1433
19	pemerintah	5963	69	dilakukan	2897	119	tim	1959	169	bangsa	1423
20	tahun	5766	70	saja	2894	120	apa	1955	170	kini	1419
21	oleh	5675	71	katanya	2866	121	menyatakan	1951	171	jadi	1416
22	saya	5643	72	persen	2858	122	tentang	1930	172	menurut	1409
23	atau	5429	73	dapat	2839	123	korban	1890	173	soal	1404
24	mereka	5392	74	daerah	2820	124	pihak	1889	174	segera	1397
25	kepada	5336	75	jalan	2798	125	sehingga	1860	175	aksi	1397
26	menjadi	5219	76	anggota	2796	126	dunia	1856	176	perlu	1391
27	harus	5184	77	sangat	2785	127	demikian	1855	177	mulai	1388
28	hari	5163	78	pun	2756	128	lainnya	1847	178	sebelum	1379
29	kata	5148	79	hal	2749	129	masalah	1815	179	bersama	1372
30	sebagai	5068	80	rumah	2736	130	rakyat	1807	180	termasuk	1371
31	adalah	4922	81	warga	2676	131	salah	1803	181	seluruh	1370
32	lebih	4818	82	beberapa	2639	132	kasus	1796	182	pusat	1364
33	para	4686	83	seorang	2618	133	tempat	1793	183	agung	1358
34	mengatakan	4650	84	banyak	2613	134	kemudian	1792	184	milyar	1357
35	hanya	4457	85	atas	2593	135	berbagai	1790	185	sidang	1349
36	orang	4430	86	ekonomi	2546	136	keamanan	1788	186	kenaikan	1334
37	telah	4363	87	agar	2525	137	harga	1785	187	akibat	1333
38	masih	4283	88	serta	2517	138	tengah	1767	188	melalui	1315
39	bahwa	4266	89	bagi	2467	139	pertemuan	1763	189	rapat	1303
40	tetapi	4180	90	kota	2439	140	bulan	1755	190	setiap	1297
41	namun	4134	91	kembali	2410	141	langsung	1748	191	empat	1296
42	saat	4105	92	ketika	2394	142	wakil	1696	192	tanpa	1287
43	seperti	4080	93	hukum	2369	143	selain	1664	193	pemerintahan	1257
44	negara	4027	94	selama	2367	144	membuat	1652	194	begitu	1256
45	sekitar	4009	95	tiga	2347	145	pasar	1640	195	pesawat	1254
46	secara	3959	96	merupakan	2340	146	malam	1623	196	kerja	1243
47	lain	3797	97	sebuah	2306	147	pertama	1623	197	kemarin	1241
48	kami	3785	98	kedua	2277	148	nasional	1619	198	apakah	1234
49	satu	3750	99	negeri	2256	149	sebesar	1612	199	ujarnya	1216
50	baru	3591	100	luar	2225	150	bahkan	1607	200	datang	1211

Appendix D

A Stoplist for Bahasa Indonesia

Table D.1: Suggested stoplist for Bahasa Indonesia

Word	Root	Part of Speech	Word	Root	Part of Speech
ada	ada	verb	lah	lah	particle
adanya	ada	noun	lain	lain	adjective
adalah	adalah	verb	lainnya	lain	adjective
adapun	adapun	particle	melainkan	lain	verb
agak	agak	adverb	selaku	laku	particle
agaknya	agak	adverb	lalu	lalu	verb
agar	agar	particle	melalui	lalu	verb
akan	akan	particle	terlalu	lalu	adverb
akankah	akan	particle	lama	lama	adjective
akhirnya	akhir	noun	lamanya	lama	noun
aku	aku	pronomia	selama	lama	noun
akulah	aku	pronomia	selama-lamanya	lama	adjective
amat	amat	adverb	selamanya	lama	adjective
amatlah	amat	adverb	lebih	lebih	adjective
anda	anda	noun	terlebih	lebih	adverb
andalah	anda	noun	bermacam	macam	adjective
antar	antar	particle	bermacam-macam	macam	adjective
diantaranya	antar	verb	macam	macam	noun
antara	antara	noun	semacam	macam	adverb
antaranya	antara	particle	maka	maka	particle
diantara	antara	verb	makanya	maka	particle
apa	apa	pronomia	makin	makin	adverb
apaan	apa	pronomia	malah	malah	adverb
mengapa	apa	pronomia	malahan	malah	adverb
apabila	apabila	particle	mampu	mampu	adjective
apakah	apakah	pronomia	mampukah	mampu	adjective
apalagi	apalagi	pronomia	mana	mana	pronoun
apatah	apatah	pronomia	manakala	manakala	particle
atau	atau	particle	manalagi	manalagi	particle
ataukah	atau	particle	masih	masih	adverb
ataupun	atau	particle	masihkah	masih	adverb
bagai	bagai	noun	semasih	masih	adverb
bagaikan	bagai	particle	masing	masing	pronomia

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
sebagai	bagai	particle	masing-masing	masing-masing	pronomia
sebagainya	bagai	particle	mau	mau	adverb
bagaimana	bagaimana	pronomia	maupun	mau	particle
bagaimanapun	bagaimana	pronomia	semaunya	mau	adverb
sebagaimana	bagaimana	particle	memang	memang	adverb
bagaimanakah	bagainamakah	pronomia	mereka	mereka	pronomia
bagi	bagi	particle	merekalah	mereka	pronomia
bahkan	bahkan	adverb	meski	meski	particle
bahwa	bahwa	particle	meskipun	meski	particle
bahwasanya	bahwasannya	particle	semula	mula	adverb
sebaliknya	balik	adverb	mungkin	mungkin	adverb
banyak	banyak	adjective	mungkinkah	mungkin	adverb
sebanyak	banyak	numeralia	nah	nah	particle
beberapa	beberapa	numeralia	namun	namun	particle
seberapa	beberapa	numeralia	nanti	nanti	adverb
begini	begini	pronomia	nantinya	nanti	adverb
beginian	begini	adjective	nyaris	nyaris	adverb
beginikah	begini	pronomia	oleh	oleh	particle
beginilah	begini	pronomia	olehnya	oleh	particle
sebegini	begini	numeralia	seorang	orang	noun
begitu	begitu	adverb	seseorang	orang	noun
begitukah	begitu	adverb	pada	pada	particle
begitulah	begitu	adverb	padanya	pada	particle
begitupun	begitu	adverb	padahal	padahal	particle
sebegitu	begitu	numeralia	paling	paling	adverb
belum	belum	adverb	sepanjang	panjang	noun
belumlah	belum	adverb	pantas	pantas	adjective
sebelum	belum	adverb	sepantasnya	pantas	adjective
sebelumnya	belum	adverb	sepantasnyalah	pantas	adjective
sebenarnya	benar	adverb	para	para	particle
berapa	berapa	pronomia	pasti	pasti	adjective
berapakah	berapa	pronomia	pastilah	pasti	adjective
berapalah	berapa	pronomia	per	per	paticle
berapapun	berapa	pronomia	pernah	pernah	adverb
betulkah	betul	adjective	pula	pula	particle
sebetulnya	betul	adverb	pun	pun	particle
biasa	biasa	adjective	merupakan	rupa	verb
biasanya	biasa	adjective	rupanya	rupa	noun
bila	bila	particle	serupa	rupa	verb
bilakah	bila	particle	saat	saat	noun
bisa	bisa	verb	saatnya	saat	noun
bisakah	bisa	verb	sesaat	saat	noun
sebisanya	bisa	adverb	saja	saja	adverb
boleh	boleh	particle	sajalah	saja	adverb
bolehkah	boleh	particle	saling	saling	adverb
bolehlah	boleh	particle	bersama	sama	verb
buat	buat	particle	bersama-sama	sama	verb
bukan	bukan	adverb	sama	sama	adjective
bukankah	bukan	pronomia	sama-sama	sama	adjective
bukanlah	bukan	adverb	sesama	sama	noun
bukannya	bukan	adverb	sambil	sambil	particle

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
cuma	cuma	adverb	sampai	sampai	verb
percuma	cuma	adverb	sana	sana	noun
dahulu	dahulu	adverb	sangat	sangat	adverb
dalam	dalam	particle	sangatlah	sangat	adverb
dan	dan	particle	saya	saya	pronomia
dapat	dapat	adverb	sayalah	saya	pronomia
dari	dari	particle	se	se	particle
daripada	daripada	particle	sebab	sebab	particle
dekat	dekat	adjective	sebabnya	sebab	particle
demi	demi	particle	sebuah	sebuah	numeralia
demikian	demikian	pronomia	tersebut	sebut	verb
demikianlah	demikian	pronomia	tersebutlah	sebut	verb
sedemikian	demikian	pronomia	sedang	sedang	particle
dengan	dengan	particle	sedangkan	sedang	particle
depan	depan	noun	sedikit	sedikit	adjective
di	di	particle	sedikitnya	sedikit	adverb
dia	dia	pronomia	segala	segala	adjective
dialah	dia	pronomia	segalanya	segala	adjective
dini	dini	adjective	segera	segera	adverb
diri	diri	noun	sesegera	segera	adverb
dirinya	diri	noun	sejak	sejak	particle
terdiri	diri	verb	sejenak	sejenak	noun
dong	dong	particle	sekali	sekali	adverb
dulu	dulu	adverb	sekalian	sekali	numeralia
enggak	enggak	adverb	sekalipun	sekali	particle
enggaknya	enggak	adverb	sesekali	sekali	adverb
entah	entah	adverb	sekaligus	sekaligus	adverb
entahlah	entah	adverb	sekarang	sekarang	adverb
terhadap	hadap	particle	sekarang	sekaranglah	adverb
terhadapnya	hadap	particle	sekitar	sekitar	noun
hal	hal	noun	sekitarnya	sekitar	noun
hampir	hampir	adverb	sela	sela	adverb
hanya	hanya	adverb	selain	selain	particle
hanyalah	hanya	adverb	selalu	selalu	adverb
harus	harus	adverb	seluruh	seluruh	numeral
haruslah	harus	adverb	seluruhnya	seluruh	numeral
harusnya	harus	adverb	semakin	semakin	adverb
seharusnya	harus	adverb	sementara	sementara	particle
hendak	hendak	particle	sempat	sempat	adverb
hendaklah	hendak	adverb	semua	semua	numeralia
hendaknya	hendak	particle	semuanya	semua	adverb
hingga	hingga	particle	sendiri	sendiri	adverb
sehingga	hingga	particle	sendirinya	sendiri	adverb
ia	ia	pronomia	seolah	seolah	verb
ialah	ialah	particle	seolah-olah	seolah	adverb
ibarat	ibarat	particle	seperti	seperti	particle
ingin	ingin	particle	sepertinya	seperti	particle
inginkah	ingin	verb	sering	sering	adverb
inginkan	ingin	verb	seringnya	sering	adverb
ini	ini	pronomia	serta	serta	particle
inikah	ini	pronomia	siapa	siapa	pronomia

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
inilah	ini	pronomia	siapakah	siapa	pronomia
itu	itu	pronomia	siapapun	siapa	pronomia
itukah	itu	pronomia	disini	sini	adverb
itulah	itu	pronomia	disinilah	sini	adverb
jangan	jangan	particle	sini	sini	adverb
jangankan	jangan	particle	sinilah	sini	adverb
janganlah	jangan	particle	sesuatu	suatu	pronomia
jika	jika	particle	sesuatunya	suatu	pronomia
jikalau	jikalau	particle	suatu	suatu	pronomia
juga	juga	adverb	sesudah	sudah	particle
justru	justru	adverb	sesudahnya	sudah	particle
kala	kala	noun	sudah	sudah	adverb
kalau	kalau	particle	sudahkah	sudah	adverb
kalaulah	kalau	particle	sudahlah	sudah	adverb
kalaupun	kalau	particle	supaya	supaya	particle
berkali-kali	kali	adverb	tadi	tadi	adverb
sekali-kali	kali	adverb	tadinya	tadi	adverb
kalian	kalian	pronomia	tak	tak	adverb
kami	kami	pronomia	tanpa	tanpa	adverb
kamilah	kami	pronomia	setelah	telah	adverb
kamu	kamu	pronomia	telah	telah	adverb
kamulah	kamu	pronomia	tentang	tentang	particle
kan	kan	particle	tentu	tentu	adjective
kapan	kapan	particle	tentulah	tentu	adjective
kapankah	kapan	particle	tentunya	tentu	adverb
kapanpun	kapan	particle	tertentu	tentu	adjective
dikarenakan	karena	verb	seterusnya	terus	adverb
karena	karena	particle	tapi	tetapi	particle
karenanya	karena	particle	tetapi	tetapi	particle
ke	ke	particle	setiap	tiap	numeralia
kecil	kecil	adjective	tiap	tiap	adjective
kemudian	kemudian	particle	setidak-tidaknya	tidak	adverb
kenapa	kenapa	pronomia	setidaknya	tidak	adverb
kepada	kepada	particle	tidak	tidak	adverb
kepadanya	kepadanya	particle	tidakkah	tidak	adverb
ketika	ketika	noun	tidaklah	tidak	adverb
seketika	ketika	adverb	toh	toh	particle
khususnya	khusus	adverb	waduh	waduh	particle
kini	kini	adverb	wah	wah	particle
kinilah	kini	adverb	wahai	wahai	particle
kiranya	kira	adverb	sewaktu	waktu	noun
sekiranya	kira	verb	walau	walau	particle
kita	kita	pronomia	walaupun	walau	particle
kitalah	kita	pronomia	wong	wong	pronomia
kok	kok	particle	yaitu	yaitu	particle
lagi	lagi	adverb	yakni	yakni	particle
lagian	lagi	adverb	yang	yang	particle

Table D.2: Most common words in Bahasa Indonesia newspapers

Word	Root	Part of Speech	Word	Root	Part of Speech
berada	ada	verb	masa	masa	noun
keadaan	ada	noun	semasa	masa	adverb
akhir	akhir	noun	masalah	masalah	noun
akhiri	akhir	verb	masalahnya	masalah	noun
berakhir	akhir	verb	termasuk	masuk	verb
berakhirlah	akhir	verb	semata	mata	adverb
berakhirnya	akhir	noun	semata-mata	mata	adverb
diakhiri	akhir	verb	diminta	minta	verb
diakhirinya	akhir	verb	dimintai	minta	verb
mengakhiri	akhir	verb	meminta	minta	verb
terakhir	akhir	adjective	memintakan	minta	verb
artinya	arti	noun	minta	minta	verb
berarti	arti	verb	mirip	mirip	adverb
asal	asal	particle	dimisalkan	misal	verb
asalkan	asal	particle	memisalkan	misal	verb
atas	atas	noun	misal	misal	noun
awal	awal	noun	misalkan	misal	verb
awalnya	awal	noun	misalnya	misal	noun
berawal	awal	verb	semisal	misal	noun
berbagai	bagai	verb	semisalnya	misal	noun
bagian	bagi	noun	bermula	mula	verb
sebagian	bagi	noun	mula	mula	noun
baik	baik	adjective	mulanya	mula	verb
sebaik	baik	adjective	dimulai	mulai	verb
sebaik-baiknya	baik	adverb	dimulailah	mulai	verb
sebaiknya	baik	adverb	dimulainya	mulai	noun
bakal	bakal	adverb	memulai	mulai	verb
bakalan	bakal	verb	mulai	mulai	verb
balik	balik	noun	mulailah	mulai	verb
terbanyak	banyak	adjective	dimungkinkan	mungkin	verb
bapak	bapak	noun	kemungkinan	mungkin	noun
baru	baru	adjective	kemungkinannya	mungkin	noun
bawah	bawah	noun	memungkinkan	mungkin	verb
belakang	belakang	noun	menaiki	naik	verb
belakangan	belakang	noun	naik	naik	verb
benar	benar	adjective	menanti	nanti	verb
benarkah	benar	adjective	menanti-nanti	nanti	verb
benarlah	benar	adjective	menantikan	nanti	verb
beri	beri	verb	menyatakan	nyata	verb
berikan	beri	verb	nyatanya	nyata	adjective
diberi	beri	verb	ternyata	nyata	verb
diberikan	beri	verb	pak	pak	pronomia
diberikannya	beri	verb	panjang	panjang	adjective
memberi	beri	verb	dipastikan	pasti	verb
memberikan	beri	verb	memastikan	pasti	verb
besar	besar	adjective	penting	penting	adjective
sebesar	besar	adjective	pentingnya	penting	adjective
betul	betul	adjective	diperlukan	perlu	verb
kebetulan	betul	adverb	diperlukannya	perlu	noun

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
dibuat	buat	verb	memerlukan	perlu	verb
dibuatnya	buat	verb	perlu	perlu	adverb
diperbuat	buat	verb	perlukah	perlu	adverb
diperbuatnya	buat	verb	perlunya	perlu	noun
membuat	buat	verb	seperlunya	perlu	adverb
memperbuat	buat	verb	pertama	pertama	numeralia
bulan	bulan	noun	pertama-tama	pertama	adverb
bung	bung	noun	memihak	pihak	verb
cara	cara	noun	pihak	pihak	noun
caranya	cara	noun	pihaknya	pihak	noun
secara	cara	particle	sepihak	pihak	noun
cukup	cukup	adjective	pukul	pukul	noun
cukupkah	cukup	adjective	dipunyai	punya	verb
cukuplah	cukup	adjective	mempunyai	punya	verb
secukupnya	cukup	adjective	punya	punya	verb
terdahulu	dahulu	adverb	merasa	rasa	verb
didapat	dapat	verb	rasa	rasa	noun
mendapat	dapat	verb	rasanya	rasa	noun
mendapatkan	dapat	verb	terasa	rasa	verb
terdapat	dapat	verb	rata	rata	adverb
berdatangan	datang	verb	berupa	rupa	verb
datang	datang	verb	disampaikan	sampai	verb
didatangkan	datang	verb	kesampaian	sampai	verb
mendatang	datang	adjective	menyampaikan	sampai	verb
mendatangi	datang	verb	sampai-sampai	sampai	verb
mendatangkan	datang	verb	sampaikan	sampai	verb
dua	dua	numeralia	sesampai	sampai	particle
kedua	dua	numeralia	tersampaikan	sampai	verb
keduanya	dua	numeralia	menyangkut	sangkut	verb
empat	empat	numeralia	satu	satu	numeralia
seenaknya	enak	adjective	disebut	sebut	verb
digunakan	guna	verb	disebutkan	sebut	verb
dipergunakan	guna	verb	disebutkannya	sebut	verb
guna	guna	noun	menyebutkan	sebut	verb
gunakan	guna	verb	sebut	sebut	verb
mempergunakan	guna	verb	sebutlah	sebut	verb
menggunakan	guna	verb	sebutnya	sebut	verb
hari	hari	noun	keseluruhan	seluruh	noun
berkehendak	hendak	verb	keseluruhannya	seluruh	noun
menghendaki	hendak	verb	menyeluruh	seluruh	verb
diibaratkan	ibarat	verb	sendirian	sendiri	pronomia
diibaratkannya	ibarat	noun	bersiap	siap	verb
ibaratkan	ibarat	verb	bersiap-siap	siap	verb
ibaratnya	ibarat	particle	mempersiapkan	siap	verb
mengibaratkan	ibarat	verb	menyiapkan	siap	verb
mengibaratkannya	ibarat	verb	siap	siap	verb
ibu	ibu	noun	dipersoalkan	soal	verb
berikut	ikut	adjective	mempersoalkan	soal	verb
berikutnya	ikut	adjective	persoalan	soal	noun
ikut	ikut	verb	soal	soal	noun
diingat	ingat	verb	soalnya	soal	noun

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
diingatkan	ingat	verb	diketahui	tahu	verb
ingat	ingat	verb	diketahuinya	tahu	noun
ingat-ingat	ingat	verb	mengetahui	tahu	verb
mengingat	ingat	verb	tahu	tahu	verb
mengingatkan	ingat	verb	tahun	tahun	noun
seingat	ingat	adverb	ditambahkan	tambah	verb
teringat	ingat	verb	menambahkan	tambah	verb
teringat-ingat	ingat	verb	tambah	tambah	verb
berkeinginan	ingin	verb	tambahnya	tambah	verb
diinginkan	ingin	verb	tampak	tampak	verb
keinginan	ingin	noun	tampaknya	tampak	verb
menginginkan	ingin	verb	ditandaskan	tandas	verb
jadi	jadi	verb	menandaskan	tandas	verb
jadilah	jadi	verb	tandas	tandas	adjectice
jadinya	jadi	noun	tandasnya	tandas	verb
menjadi	jadi	verb	bertanya	tanya	verb
terjadi	jadi	verb	bertanya-tanya	tanya	verb
terjadilah	jadi	verb	dipertanyakan	tanya	verb
terjadinya	jadi	noun	ditanya	tanya	verb
jauh	jauh	adjective	ditanyai	tanya	verb
sejauh	jauh	noun	ditanyakan	tanya	verb
dijawab	jawab	verb	mempertanyakan	tanya	verb
jawab	jawab	verb	menanya	tanya	verb
jawaban	jawab	verb	menanyai	tanya	verb
jawabnya	jawab	verb	menanyakan	tanya	verb
menjawab	jawab	verb	pertanyaan	tanya	noun
dijelaskan	jelas	verb	pertanyakan	tanya	verb
dijelaskannya	jelas	verb	tanya	tanya	verb
jelas	jelas	adjective	tanyakan	tanya	verb
jelaskan	jelas	verb	tanyanya	tanya	verb
jelaslah	jelas	adjective	ditegaskan	tegas	verb
jelasnya	jelas	verb	menegaskan	tegas	verb
menjelaskan	jelas	verb	tegas	tegas	verb
berjumlah	jumlah	verb	tegasnya	tegas	verb
jumlah	jumlah	noun	setempat	tempat	noun
jumlahnya	jumlah	noun	tempat	tempat	noun
sejumlah	jumlah	noun	setengah	tengah	numeralia
sekadar	kadar	adverb	tengah	tengah	adverb
sekadarnya	kadar	adverb	tepat	tepat	adjective
kasus	kasus	noun	terus	terus	adverb
berkata	kata	verb	tetap	tetap	adjective
dikatakan	kata	verb	setiba	tiba	particle
dikatakannya	kata	noun	setibanya	tiba	noun
kata	kata	verb	tiba	tiba	verb
katakan	kata	verb	tiba-tiba	tiba-tiba	adverb
katakanlah	kata	verb	tiga	tiga	numeralia
katanya	kata	noun	setinggi	tinggi	adjective
mengatakan	kata	verb	tinggi	tinggi	adjective
mengatakannya	kata	verb	ditujukan	tuju	verb
sekecil	kecil	adjective	menuju	tuju	verb
keluar	keluar	verb	tertuju	tuju	verb

continue to next page

continued from previous page

Word	Root	Part of Speech	Word	Root	Part of Speech
kembali	kembali	verb	ditunjuk	tunjuk	verb
berkenaan	kena	verb	ditunjuki	tunjuk	verb
mengenai	kena	particle	ditunjukkan	tunjuk	verb
bekerja	kerja	verb	ditunjukkannya	tunjuk	verb
dikerjakan	kerja	verb	ditunjuknya	tunjuk	verb
mengerjakan	kerja	verb	menunjuk	tunjuk	verb
dikira	kira	verb	menunjuki	tunjuk	verb
diperkirakan	kira	verb	menunjukkan	tunjuk	verb
kira	kira	noun	menunjuknya	tunjuk	verb
kira-kira	kira	adverb	tunjuk	tunjuk	verb
memperkirakan	kira	verb	berturut	turut	adverb
mengira	kira	verb	berturut-turut	turut	adverb
terkira	kira	verb	menurut	turut	particle
kurang	kurang	adverb	turut	turut	verb
sekurang-kurangnya	kurang	adverb	bertutur	tutur	verb
sekurangnya	kurang	adverb	dituturkan	tutur	verb
berlainan	lain	verb	dituturkannya	tutur	noun
dilakukan	laku	verb	menuturkan	tutur	verb
melakukan	laku	verb	tutur	tutur	verb
berlalu	lalu	verb	tuturnya	tutur	verb
dilalui	lalu	verb	diucapkan	ucap	verb
keterlaluan	lalu	adjective	diucapkannya	ucap	verb
kelamaan	lama	adjective	mengucapkan	ucap	verb
berlangsung	langsung	verb	mengucapkannya	ucap	verb
lanjut	lanjut	adjective	ucap	ucap	verb
lanjutnya	lanjut	verb	ucapnya	ucap	verb
selanjutnya	lanjut	adverb	berujar	ujar	verb
berlebihan	lebih	adjective	ujar	ujar	noun
lewat	lewat	particle	ujarnya	ujar	noun
dilihat	lihat	verb	umum	umum	adjective
diperlihatkan	lihat	verb	umumnya	umum	adverb
kelihatan	lihat	noun	diungkapkan	ungkap	verb
kelihatannya	lihat	noun	mengungkapkan	ungkap	verb
melihat	lihat	verb	ungkap	ungkap	verb
melihatnya	lihat	verb	ungkapnya	ungkap	verb
memperlihatkan	lihat	verb	untuk	untuk	particle
terlihat	lihat	verb	usah	usah	verb
kelima	lima	numeralia	seusai	usai	particle
lima	lima	numeralia	usai	usai	verb
luar	luar	noun	terutama	utama	adverb
bermaksud	maksud	verb	waktu	waktu	noun
dimaksud	maksud	verb	waktunya	waktu	noun
dimaksudkan	maksud	verb	meyakini	yakin	verb
dimaksudkannya	maksud	verb	meyakinkan	yakin	verb
dimaksudnya	maksud	verb	yakin	yakin	adjective
semampu	mampu	adjective			
semampunya	mampu	adjective			

Bibliography

[1] Tata Bahasa Melayu, 1998. URL `http://tatabahasabm.tripod.com`.

[2] F. Ahmad, M. Yusoff, and T. M. T. Sembok. Experiments with a Stemming Algorithm for Malay Words. *Journal of The American Society for Information Science*, 47:909–918, 1996.

[3] R. Baeza and B. Ribeiro. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] C. Buckley, A. Singhal, M. Mitra, and G. Salton. New Retrieval Approaches using SMART: TREC 4. In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48, 1995.

[5] C. Buckley and E. M. Voorhees. Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, Athena, Greece, 2000. ACM Press.

[6] Dept. of Cultural and Education, Republic of Indonesia, editor. *Pedoman Umum Ejaan Bahasa Indonesia yang Disempurnakan*. Pustaka Setia, 1987.

[7] Dept. of Cultural and Education, Republic of Indonesia, editor. *Tata Bahasa Baku Bahasa Indonesia*. Balai Pustaka, 1988.

[8] Dept. of Education, Republic of Indonesia, editor. *Kamus Besar Bahasa Indonesia*. Balai Pustaka, 2001.

[9] C. Fox. Lexical Analysis and Stoplists. In Frakes and Baeza [11], pages 102–130.

[10] W. B. Frakes. Stemming Algorithms. In Frakes and Baeza [11], pages 131–160.

[11] W. B. Frakes and R. Baeza, editors. *Information Retrieval, Data Structures and Algorithms*. Prentice Hall, 1992.

[12] D. Harman. How effective is suffixing? *Journal of The American Society for Information Science*, 42:7–15, 1991.

[13] V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monoligual Document Retrieval for European Languages. 2003.

[14] D. A. Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 329–338, Pittsburgh, Pennsylvania, 1993. ACM Press.

[15] D. A. Hull. Stemming Algorithms - A Case Study for Detailed Evaluation. *Journal of The American Society for Information Science*, 47, 1996.

[16] M. Kantrowitz, B. Mohit, and V. Mittal. Stemming and Its Effects on TFIDF Ranking. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 357–359, Athens, Greece, 2000. ACM Press.

[17] W. Kraaij and R. Pohlman. Porter's stemming algorithm for Dutch. In L. G. M. Noordman and W. A. M. de Vroomen, editors, *Informatiewetenschaap 1994: Wetenschappelijke Bijdragen aan de deede STINFO Conferentie*, pages 167–180, Tilburg, 1994.

[18] W. Kraaij and R. Pohlman. Evaluation of a Dutch Stemming Algorithm. In J. Rowley, editor, *The New Review of Document and Text Management*, volume 1, pages 25–43. Taylor Graham, 1995.

[19] W. Kraaij and R. Pohlman. Viewing Stemming as Recall Enhancement. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 40–48, Zurich, Switzerland, 1996.

[20] R. Krovetz. Viewing Morphology as an Inference Process. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 191–203, Pittsburgh, Pennsylvania, 1993. ACM Press.

[21] C. Monz and M. de Rijke. Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, Germany and Italian. In C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors, *Evaluation of Cross-Language Information Retrieval System. Second Workshop of the Cross-Language Evaluation Form, CLEF 2001*, LCNS 2406, pages 357–359, Darmstadt, Germany, Sept. 2001. Springer.

[22] B. Nazief. Spelling Checker Facility and The Analysis of the Word Frequency. *Proceedings of Computer and Arts Conference*, 1995.

[23] B. Nazief and M. Adriani. Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia. Technical report, Faculty of Computer Science, University of Indonesia, Depok, 1996.

[24] C. D. Paice. Another Stemmer. *ACM SIGIR Forum*, 24(3):56–61, 1990.

[25] C. D. Paice. Method for Evaluation of Stemming Algorithms Based on Error Counting. *Journal of The American Society for Information Science*, 47(8):632–649, Aug. 1996.

[26] C. D. Paice. What is Stemming?, 1996. URL `http://www.comp.lancs.ac.uk/computing/research/stemming/general/index.htm`.

[27] A. Pirkola. Morphological Typology of Languages fo IR. *Journal of Documentation*, 57: 330–348, May 2001.

[28] M. Popovic and P. Willett. The Effectiveness of Stemming for Natural-Language Access to Slovene Textual Data. *Journal of the American Society for Information Science*, 43(5): 384–390, June 1992.

[29] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[30] G. Salton and M. J. McGill, editors. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[31] J. Savoy. Stemming of French Words Based on Grammatical Catagories. *Journal of the American Society for Information Science*, 44:1–9, Jan. 1993.

[32] J. Savoy. Report on CLEF-2001 Experiments. In C. Peters, editor, *Results of the CLEF 2001, Cross-Language System Evaluation Campaign*, pages 11–19, Sophia-Antipolis, 2001.

[33] A. Singhal, C. Buckley, and M. Mitra. Pivoted Document Length Normalization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Zurich, Switzerland, 1996. ACM Press.

[34] S. Y. Tai, C. S. Ong, and N. A. Abdullah. On Designing an Automated Malaysian Stemmer for the Malay Language. In *Procedings of the Fifth International Workshop on Information Retrieval with Asian Languages*, pages 207–208, Hongkong, China, 2000. ACM Press.

[35] H. G. Tarigan. *Pengajaran Morfologi*. Angkasa, Bandung, 1995.

[36] R. J. Wonnacott and T. H. Wonacott. *Introductory Statistics*. John Willey & Son, fourth edition, 1985.