Stemming Indonesian: A confix-stripping approach.

Article · January 2007

Source: DBLP

CITATIONS
67

READS
1,542

5 authors, including:

Jelita Asian
Surya University
8 PUBLICATIONS **123** CITATIONS
SEE PROFILE

Bobby Nazief
University of Indonesia
23 PUBLICATIONS **236** CITATIONS
SEE PROFILE

S.M.M. Tahaghoghi
Microsoft
43 PUBLICATIONS **657** CITATIONS
SEE PROFILE

Hugh Williams
University of Melbourne
82 PUBLICATIONS **1,902** CITATIONS
SEE PROFILE

Stemming Indonesian: A Confix-Stripping Approach

MIRNA ADRIANI, University of Indonesia
JELITA ASIAN, RMIT University
BOBBY NAZIEF, University of Indonesia
S.M.M. TAHAGHOGHI, RMIT University
and
HUGH E. WILLIAMS, Microsoft

Stemming words to (usually) remove suffixes has applications in text search, machine translation, document summarization, and text classification. For example, English stemming reduces the words "computer," "computing," "computation," and "computability" to their common morphological root, "comput-." In text search, this permits a search for "computers" to find documents containing all words with the stem "comput-." In the Indonesian language, stemming is of crucial importance: words have prefixes, suffixes, infixes, and confixes that make matching related words difficult.

This work surveys existing techniques for stemming Indonesian words to their morphological roots, presents our novel and highly accurate CS algorithm, and explores the effectiveness of stemming in the context of general-purpose text information retrieval through ad hoc queries.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language models*; *Language parsing and understanding*; *Text analysis*; H.3.1 [**Information Systems**]: Content Analysis and Indexing—*Linguistic processing*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Indonesian, information retrieval, stemming

ACM Reference Format:
Adriani, M., Nazief, B., Asian, J., Tahaghoghi, S. M. M., and Williams, H. E. 2007. Stemming Indonesian: A confix-stripping approach. ACM J. Educ. Resour. Comput. 6, 4, Article 13 (December 2007), 33 pages. DOI = 10.1145/1316457.1316459. http://doi.acm.org/10.1145/1316457.1316459.

1. INTRODUCTION

Stemming is a basic text processing tool often used for efficient and effective text retrieval [Frakes 1992], machine translation [Bakar and Rahman 2003],

Correspondence address: S.M.M. Tahaghoghi, School of Computer Science and Information Technology, RMIT University, GPO Box 2476V, Melbourne, 3001, Australia; email: seyed@tahaghoghi.rmit.edu.au.

document summarization [Orăsan et al. 2004], and text classification [Gaustad and Bouma 2002]. Stemmers remove affixes to cluster words derived from a common *stem* or *root*; for example, the words "opens," "opened," and "opener" are clustered with the stem "open." Identifying words from a common root increases the sensitivity of retrieval by improving the ability to find relevant documents, but is often associated with a decrease in selectivity, where the clustering causes useful meaning to be lost. For example, mapping the word "stranger" to the same cluster as "strange" is likely to be desirable if the former is used as an adjective, but not if it is used as a noun. Stemming is expected to increase recall but possibly decrease precision.

However, for languages such as English [Hull 1996], French [Savoy 1999], Slovene [Popovič and Willett 1992], and Arabic [Larkey et al. 2002], stemming leads to increased precision. Popovič and Willett [1992] claim that morphologically complex languages—such as Slovene—are more likely to benefit from stemming. For the same reason, we suspect that Indonesian might benefit as well.

For the English language, stemming has been extensively investigated [Paice 1996], with techniques such as those of Lovins [1968] and Porter [1980] in widespread use. However, despite the growing volume of written content in other languages [Ballesteros and Croft 1997; Hollink et al. 2004], stemming for other languages is less well-known; while there are several approaches available for languages such as French [Savoy 1993], Spanish [Xu and Croft 1998], Malaysian [Ahmad et al. 1996; Idris 2001], and Indonesian [Arifin and Setiono 2002; Nazief and Adriani 1996; Vega 2001], there is almost no consensus about their effectiveness.

Despite its large population and the intense geopolitical interest Indonesia has attracted in recent years, there has been relatively little interest in the Indonesian language—also known as *Bahasa Indonesia*[1]—from the linguistic processing and information retrieval communities. Stemming is essential to support effective Indonesian information retrieval, and has uses as diverse as defense intelligence applications, document translation, and Web search. We explore stemming for this language.

Quinn [2001] states that the Indonesian language has its origin in the Malay language. It belongs to the Austronesian language family, which includes Tagalog, Javanese, Balinese, Malagasy, and Maori. During the spread of Islam between the 14th and the 15th centuries CE, Javanese and Arabic scripts were used to write Malay. From the second half of the 19th century, due to the influence of European missionaries,[2] Latin script came into widespread use. By the early 20th century, all Malay words were written in Latin script.

Some foreign words have been assimilated into Indonesian with their original spelling intact. Examples of such loan words include "moderator" ⟨moderator⟩ from English; "tempo" ⟨tempo⟩ from Italian; and "rekening" ⟨account⟩ and "tante" ⟨aunt⟩ from Dutch. Foreign words or phrases that are

[1]"Bahasa" literally means "language" in Indonesian.
[2]http://www.alhewar.com/habeeb_salloum_arabic_language.htm

not assimilated are shown italicized, for example "*ad hoc*," "*curriculum vitae*," and "*pianissimo*." Other foreign words may also be transliterated [Dwipayana 2001]. Dwipayana [2001] recommends that transliterated names be Romanized according to ISO standards, common English spelling, or Chinese Pinyin. Different languages have different ISO standards for transliteration: for example, Japanese uses ISO 3602:1989, Korean uses ISO/TR 11941:1996, and Arabic uses ISO 233:1984.[3]

"Teknologi" ⟨technology⟩, "kompas" ⟨compass⟩, and "narkotika" ⟨narcotics⟩ are examples of transliterated words.

Indonesian prefixes and suffixes have been influenced by foreign languages, especially Indo-European languages [Dwipayana 2001; Widyamartaya 2003; Wilujeng 2002]. These prefixes and suffixes can either be retained unchanged or transliterated into Indonesian. Examples of foreign prefixes adopted in Indonesian include "mono-" ⟨mono-⟩, "ekstra-" ⟨extra-⟩, "hiper" ⟨hyper⟩, "sin-" ⟨syn-⟩, and "ultra-" ⟨ultra-⟩. Likewise, suffix examples include "-si" ⟨-sion and -tion⟩, "-asme" ⟨-asm⟩, "-bel" ⟨-ble⟩, "-ikel" ⟨-icle⟩, and "-or" ⟨-or⟩. We do not consider these to be native affixes.

Indonesian does not have accented characters, and so accents are removed during transliteration. For example, "déjà vu" and "naïve" are typically written as "deja vu" and "naive" respectively. Indonesian verbs do not change with tense; instead, tense is implied by the context of the sentence and the presence of words specifying time, such as "kemarin" ⟨yesterday⟩ and "besok" ⟨tomorrow⟩ [Woods et al. 1995].

Indonesian employs affixes more heavily than English, and the application and order of stemming rules requires careful consideration. In addition to prefixes and suffixes, it has infixes (insertions) and confixes—also referred to as circumfixes—that are combinations of prefixes and suffixes. Consider the root word "perintah" ⟨rule, order⟩. Examples of words derived from this stem include the words "perintah<u>nya</u>," "<u>di</u>perintah," and "<u>pe*m*</u>erintah." Here, affixes are shown underlined and the recoding character—explained in the next paragraph—is shown in italics.

A prefix may change letters of the word it is added to. For example, the prefix "meny" added to the root word "sapu" ⟨broom⟩ produces "menyapu" ⟨to sweep⟩; the letter "s" of the root word does not appear in the derived form. Hence, an automatic stemming algorithm must be able to restore (or *recode*) such letters during the stemming process.

Some derived words are made from a repeated root with a confix spanning the constituent words. For example, the word "*se*baik-baik*nya*" ⟨as good as possible⟩ is derived from the word "baik" ⟨good⟩. Finally, countable words are repeated to indicate the plural. Hence, "buku-buku" ⟨books⟩ is the plural form of "buku" ⟨book⟩.

Stemming Indonesian is clearly a more complex endeavor than stemming English, but also more important for effective information retrieval. In this article, we explore the major stemming approaches for Indonesian, including

[3]http://www.nssn.org/

two for the related Malaysian language. We introduce the novel CS stemmer, which uses detailed rules of the Indonesian language. Against a baseline derived from multiple human subjects, we show that it is the best-performing stemmer for Indonesian. We also explore whether stemming can improve retrieval performance, and observe that—as with English—stemming has little effect on accuracy. Finally, we describe extensions to the CS stemmer that use *n*-grams to find proper nouns that should not be stemmed, and in this way help to improve retrieval performance. We find that stemming all words except proper nouns by using our CS stemmer and 4-grams produces higher average precision and R-precision than not stemming at all. This is statistically significant at the 95% confidence level using the one-tailed Wilcoxon signed ranked test. While this combination produces a precision at 10 that is slightly worse than for no stemming, the difference is not statistically significant. The recall value for the unstemmed collection is 0.728. Our CS stemmer without using *n*-grams and proper noun identification increases the recall value to 0.781, while the CS stemmer with the optimal combination of techniques produces the slightly lower recall value of 0.779.

The remainder of this aricle is organized as follows. In Section 2, we review how affixes are used in Indonesian. In Section 3, we describe several existing stemming algorithms that can be applied to Indonesian text. We introduce our new CS stemmer in Section 4, and continue in Section 5 with a description of how we investigate the effectiveness of our approach using stemming and information retrieval experiments. We present extensions to our method in Section 6, and conclude in Section 7 with a discussion of our findings and consideration of future work in the area.

2. AFFIXES IN INDONESIAN

Affixes can be *inflectional* or *derivational* [Payne 1997]. In English, inflectional affixes add context such as plurality or tense, whereas derivational affixes modify the lexical form of the word. For example, the English verb "teach" can take the inflectional suffix "-es" to form the present singular verb "teaches"; it can also take the derivational suffix "-er" to convert the original verb to the noun "teacher."

In Indonesian, inflectional affixes are limited to suffixes only, whereas derivational affixes may be prefixes, suffixes, or a combination of both (confixes).[4] We now examine each category in further detail.

Inflectional suffixes. These are of two types [Moeliono and Dardjowidjojo 1988]:

(1) Particles (P) {"-kah," "-lah," "-tah," "-pun"}.
A particle is an uninflected item of speech that is neither a verb nor a noun. For example, the suffix "-lah" added to the stem "duduk" ⟨sit⟩ produces the word "duduklah" ⟨please sit⟩.

[4]We cater for only native Indonesian affixes, and do not consider foreign affixes such as "pro-" ⟨pro-⟩ and "anti-" ⟨anti-⟩ that can form words with completely different meanings.

Moeliono and Dardjowidjojo [1988] add that the particle "-pun" can only be used in a declarative sentence. This particle emphasizes the noun or the noun phrase it follows, and should be written separately except when used as a conjunction, as in "walaupun" ⟨although⟩ and "apapun" ⟨no matter what⟩. However, we have observed that in practice, the particle "-pun" is often attached to the word it follows. Our CS stemming algorithm described in Section 4 deals with this common mistake.

(2) Possessive pronouns (PP) {"-ku," "-mu," and "-nya"}.
These express an ownership relationship. For example, the suffix "-nya" added to "buku" produces "bukunya" ⟨his or her book⟩.

Where the two types of inflectional suffixes appear together, the particle is always added after the possessive pronoun. For example, "buku" ⟨book⟩ may be appended with the possessive pronoun PP "-mu" to give "bukumu" ⟨your book⟩, followed by the particle P "-lah" to give "bukumulah" ⟨it is your book that⟩.

Derivational prefixes {"be-," "di-," "ke-," "me-," "pe-," "se-," and "te-"}.[5] These may have variants; for instance, the prefix "me-" can appear as "me-," "mem-," "men-," "meng-," or "meny-" according to the first letter of the root word. Up to three derivational prefixes may be added to a root word. For example, the derivational prefixes "se-," "peng-" (a variant of "pe-") and "ke"[6] may be prepended to "tahu" ⟨know⟩, and the suffixes "-an" and "-ku" appended to this word, to produce "sepengetahuanku" ⟨as far as I know⟩ ("se-"+"peng-"+ "[k]e-"+ "tahu" "-an" + "-ku").[7]

Derivational suffixes {"-i," "-kan," "-an"}. Only one derivational suffix may be applied to a root word. For example, the word "lapor" ⟨to report⟩ can be suffixed by the derivational suffix "-kan" to become "laporkan" ⟨go to report⟩.

Derivational confixes, for example, {"be-an," "me-i," "me-kan," "di-i," "di-kan"}. Some derivational prefix-suffix pairs form derivational structures. For example, the prefix "ke-" and the suffix "-an" added to the root word "dalam" ⟨inside, deep⟩ form the word "kedalaman" ⟨depth, profundity⟩.

Not all prefix and suffix combinations form a confix [Moeliono and Dardjowidjojo 1988], but we choose to treat them as such during stemming. There is no official and complete list of Indonesian confixes; from Moeliono and Dardjowidjojo [1988], we conclude that the most common Indonesian confixes are "be-an" and "ke-an." These pairs of prefixes and suffixes can form either confixes or affix combinations depending on the root word to which they are appended.

---

[5]In Indonesian grammar books, the prefixes "pe-" and "me-" are written using different variants including "peng-" and "peN-" for "pe-" and "meng-" and "meN-"; similarly, the prefixes "be-" and "te-" are sometimes listed as "ber-" and "ter-". For clarity and compactness, we write them as "pe-," "me-," "be-," and "te-," respectively.

[6]When the prefix "pe-" is prepended to a word or a prefix that starts with the letter "k," the "pe-" becomes "peng-" and the letter "k-" is removed.

[7]"The brackets enclosing the letter "k" indicate that it is a recoding character.

Table I. Disallowed Prefix and Suffix Combinations

Prefix	Disallowed suffixes
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan
te-	-an

A word can have at most one particle and one possessive pronoun. These may be applied directly to root words or to words that have a derivational suffix. For example, the root word "lapor" ⟨to report⟩ may take the derivational suffix "-kan" to become "laporkan" ⟨go to report⟩, and then take the inflectional suffix "-lah" to become "laporkanlah" ⟨please go to report⟩.

Some prefixes never appear with certain suffixes. Table I shows these invalid affix pairs. There is one exception to this list: the prefix "ke-" and the suffix "-i" can appear together on the root word "tahu" ⟨to know⟩ to form "ketahui" ⟨to know⟩. Consider the word "disarikan" ⟨to paraphrase⟩. An automated algorithm may stem this as "*di*-sari-*kan*" to give the root word "sari" ⟨essence⟩, or as "*di*-sarik-*an*" to give the root word "sarik" ⟨thief⟩. During automatic stemming, we can perform *confix restriction* to rule out incorrect affix combinations, in this case "di-. . . -an."

3. RELATED WORK

We now examine existing stemming approaches that can be applied to Indonesian text. With the exception of the VEGA algorithm of Section 3.1, all the algorithms described here use a dictionary to verify word stems; none specifies a particular Indonesian dictionary. To provide a uniform baseline for all algorithms that use a dictionary, we choose to use the University of Indonesia dictionary of 29,337 root words described in Nazief and Adriani [1996]; we refer to this dictionary as DICT-UI.

3.1 Vega

[Vega 2001] describes an approach—that we refer to as VEGA—that uses rule sets to determine whether affixes can be removed from a word. The rules are accessed in the order they are presented in the code. When one rule fails, the algorithm proceeds to the next. Consider processing the word "kedatangan" ⟨arrival⟩ using the following set of rules:

Rule 1: *word(Root)* → *circumfix(Root)*
Rule 2: *word(Root): StemWord*
Rule 3: *circumfix(Root)* → *ber-(Root), (-kan | -an)*
Rule 4: *circumfix(Root)* → *ke-(Root), -an*
Rule 5: *ber-(Root)* → *ber-, stem(Root)*
Rule 6: *ke-(Root)* → *ke-, stem(Root)*

Processing starts with Rule 1, which requires us to test for a circumfix. We look up the first rule having *circumfix* on the left-hand side (Rule 3). This tests

for the prefix "ber-" by applying Rule 5. Since this prefix does not appear in the word "kedatangan," Rule 5 fails, and consequently the calling rule (Rule 3), fails as well.

The next rule listing *circumfix* on the left-hand side is Rule 4, which in turn calls Rule 6. This tests whether the word starts with "ke-." Since this is true for "kedatangan," we remove the prefix "ke-" to leave "datangan." On returning to Rule 4, we check whether "datangan" ends with "-an," and since it does, we remove the suffix to obtain the stem "datang" ⟨arrive⟩.

Had Rule 1 not been satisfied, Rule 2 would have been triggered, indicating that the input word is a stem word. The algorithm allows for explicit listing of exceptions; for example, we can prevent stemming "megawati" (the name of a former Indonesian president) even though it contains the confix "me-. . . -i."

This algorithm has *iterative* versions that stem recursively, and *extended* variants that—uniquely among Indonesian stemming algorithms—consider nonstandard colloquial affixes. In our results, we report results with only the first scheme, which we refer to as VEGA-1; the other variants are ineffective, performing between 10%– 25% worse than VEGA-1.[8]

A major shortcoming of the VEGA approach is the absence of a lookup stage where words are compared to a dictionary of known root words; stemming continues as long as the word contains affix letters, often leading to overstemming. Moreover, the algorithm does not cater for cases where recoding is required. Finally, the reliance on strict rules necessitates that the rules be correct and complete, and prevents ad hoc restoration of affix combinations.

3.2 Arifin and Setiono

Arifin and Setiono [2002] describe a stemming scheme—which we refer to as ARIFIN—that first removes up to two prefixes, and then removes up to three suffixes. After removal of each prefix or suffix, a dictionary lookup is performed, and stemming stops if the word in its current form appears in the dictionary.

If the word has not been found in the dictionary by the time the maximum number of prefixes and suffixes have been removed, the algorithm progressively restores different combinations of prefixes and suffixes in order, and checks against the dictionary at each step. For example, the word "kesendirianmu" ⟨your solitude⟩ has the prefix "ke-" and the suffixes "-an" and "-mu." The algorithm removes these three affixes, and also the apparent affixes "se-" and "-i" to produce "ndir," which is not a valid word. The prefixes and suffixes are then progressively replaced. Restoring the prefixes "ke-" and "se-" to "ndir" produces "kesendir," which is not a valid root word. The algorithm then restores only the prefix "se-" to "ndir," producing "sendir." This is still not valid. Similarly, the algorithm would first restore the suffix "-i," and then the suffix "-an" and "-mu." It would then restore the suffix "-i" together with the prefixes "ke-" and "se-" to produce the invalid word "kesendiri." The algorithm then tries to add only the prefix "se-" with the suffix "-i" to produce "sendiri" ⟨self, own⟩, which is the correct root. Had the dictionary lookup failed, the restoration

[8]All percentage differences quoted in this article are absolute percentage points.

process would have stepped through "kesendirian" ⟨solitude⟩ to "sendirian" ⟨being alone⟩ (both are valid words but not stems).[9]

This algorithm also tries recoding during prefix removal. If the new word is not found in the dictionary, a lookup is performed using the recoded form. If this also fails, the word is returned to the prerecoding form before proceeding to the next removal. Consider the word "penyendirian" ⟨isolation⟩. This has the root word "sendiri" ⟨self⟩. The algorithm removes the prefix letters "pe-" to obtain "nyendirian." This is not a root word, so the suffix letters "-an" are also removed to give "nyendiri." Not finding the word "nyendiri" in the dictionary, the algorithm tries combinations of the removed prefixes and suffixes including "nyendir," "penyendir," and "penyendiri" ⟨loner⟩. If this is unsuccessful, the algorithm then considers the letter "pe-" to be part of the prefix "peny-," and so removes the letters "-ny" to obtain "endiri."[10] Adding the recoding letter "s-" results in "sendiri"; this appears in the root word dictionary, and so the operation ends.

This scheme has two main shortcomings. First, it removes repeated affix letters even though affixes are never repeated in Indonesian; this leads to overstemming. For example, in the word "peranan" ⟨role, part⟩, the suffix letters "-an" appear twice. ARIFIN removes these in succession to obtain the valid word "per" ⟨spring⟩ instead of the correct root word "peran" ⟨to play the role of⟩.

Second, it is sensitive to affix removal order. For example, it incorrectly processes the word "memberikan" ⟨to give away⟩ by removing first "mem-" to obtain "berikan" ⟨please give away⟩, which is not a root word, and then "ber-" to obtain "ikan" ⟨fish⟩. The word "memberikan" is actually formed from the root word "beri"⟨to give away⟩ and the confix pair "me-" and "-kan."

3.3 Ahmad, Yusoff, and Sembok

The approach described by Ahmad et al. [1996] is designed for the Malaysian language, also known as *Bahasa Malaya*, rather than for Indonesian. However, the two languages are closely related and share many stemming rules.

The algorithm uses a root word dictionary and a list of valid affix combinations in the form of templates. Ahmad et al. [1996] report that the original algorithm uses a Malaysian dictionary called Kamus Dewan [Dewan Bahasa dan Pustaka 1991] containing 22,293 root words. Since we deal with Indonesian, we replace this with the University of Indonesia dictionary, DICT-UI described at the beginning of Section 3. At the start of the stemming process and at each step, a dictionary lookup is performed with the current form of the word, and stemming concludes if the word appears in the dictionary. After each unsuccessful lookup, the word is compared to the next matching affix template,

[9]Not all words produced by restoration steps are shown because some of these repeat the same patterns as the removal process. For example, during the removal process the stems "sendirianmu," "dirianmu," "dirian," "diri," and "dir" are produced; none of these stems are in the dictionary. These same stems are encountered again during the restoration process, but are not listed as they have already been considered and checked against a dictionary during removal.

[10]The prefix "pe-" has the variant "peny-," with the recoding character "s-."

and, where possible, affixes are removed. If all matching templates are exhausted without a successful dictionary lookup, the original word is returned unstemmed.

Consider the affix template "me-. . . -kan." The word "memberikan" ⟨to give away⟩ matches this template, and removing the letters corresponding to the prefix and suffix leaves "beri" ⟨to give away⟩, which is the correct stem. A word may match several affix templates, and so this algorithm is sensitive to the order in which the templates appear in the list. For example, the word "berasal" ⟨to come from⟩ can match both the templates ". . . -er-. . . " and "ber- . . . ." Applying the first produces the incorrect stem "basal" ⟨basalt⟩, whereas the second template produces the correct stem "asal" ⟨origin, source⟩.

Ahmad et al. use three different template sets referred to as A, B, and C. They state that template A with its 121 rules is a direct implementation of the work of Othman [1993]. Template B, which consists of 432 rules, extends template A with rules derived from the Qur'an, and this is in turn extended by template C with an additional 129 rules to cater for modern Malay words adapted from foreign languages, such as the prefix "infra-" as in "inframerah" ⟨infared⟩ and the suffix "-tual" as in "konseptual" ⟨conceptual⟩. All three sets have a single list of suffixes and infixes, sorted alphabetically, followed by a similarly sorted list of prefixes and confixes. The authors list the rules added for B and C, but do not specify how each incorporates the rules of the previous set. We explore three orderings for each of the B and C template sets: B_1, B_2, B_3, C_1, C_2, and C_3. In the B_1 and C_1 variants, the additional rules are appended to the previous rules as shown in Ahmad et al. [1996]. The B_2 and C_2 variants order the rules alphabetically without considering the affix types. In the B_3 and C_3 variants, the suffix and infix rules are listed alphabetically first, and are followed by the prefix and confix rules, also listed alphabetically. In preliminary experiments using several orderings, we have observed that they exhibit very similar performance. In this article, we describe results for the ordering (B_2) that we have found to perform the best in earlier experiments [Asian et al. 2005]. We suspect that the better performance of this variant—which we refer to as AHMAD-B_2—is due to its catering for general affixes before considering more specific affixes such as those from the Qur'an and those found only in modern Malaysian usage.

3.4 Idris

Idris [2001] extends the scheme of Ahmad et al. [1996] to perform progressive stemming and recoding. The algorithm alternates between removing prefixes and suffixes until the root word is found in a dictionary or a removal limit is reached. Since Idris does not specify recommended limits, we adopt the assumption of Arifin and Setiono [2002] that Indonesian words can have at most two prefixes and three suffixes.

A feature of this algorithm is that it uses two dictionaries: one general, and another specific to the document content, for example containing medical or legal terms. For Web retrieval applications, it is unlikely that the document

content will be known beforehand, and so we use only the general dictionary in the experiments we report.

Two variants of this algorithm exist: one changes prefixes before recoding, and the other performs the reverse. The first assumes that a word may have different prefixes. For example, the word "memasukkan" ⟨to enter something in⟩ with the root "masuk" ⟨to be present⟩ could be "*mem*-asuk-*kan*" or "*me*-masuk-*kan*." Removing the prefix "mem-" results in "asuk," which is invalid; the algorithm then adds the letter "m" back to obtain the valid stem "masuk."

The second variant attempts recoding first. For our example, after removing the prefix "mem-," we obtain "asuk" which is not in the dictionary. From recoding rules, we know that for the prefix "mem-," the letter "p" could have been dropped, so we prepend this letter to "asuk" to obtain the valid root word "pasuk" ⟨troop⟩.

In this way, the variants arrive at different root words—"masuk" and "pasuk"—for "memasukkan." We have found that the latter variant—that we call IDRIS-2—performs slightly better, and we report only experiments using this variant.

Incorrect affix removal order can lead to overstemming. Consider the word "medannya" ⟨his or her field, plain or square⟩, with the root "medan" ⟨field, plain or square⟩. Since IDRIS tries to first remove prefixes, it will remove the prefix letters "me-," to obtain the invalid candidate root word "dannya." Since this does not appear in the dictionary, the suffix "-nya" is then removed to produce "dan" ⟨and⟩. This is a valid root word, but not the correct one. Being designed for Malaysian, this algorithm uses a set of prefixes and suffixes that are slightly different from those used in Indonesian, and this can also contribute to overstemming.

4. THE CS STEMMER

We now present our confix-stripping approach to stemming Indonesian.[11] This scheme, which we refer to as CS, is based on a thorough understanding of the underlying rules of the language.

From the discussion of Section 2, we can view the affix order of use to be as shown below, with the square brackets indicating that an affix is optional.

[[[DP+]DP+]DP+] root-word [[+DS][+PP][+P]]

We apply this order and knowledge of basic rules of the Indonesian language as the foundation of our stemming approach:

(1) Words of three or fewer characters cannot contain affixes, so no stemming is performed on such short words encountered at any stage during stemming.
(2) Affixes are never repeated, so a stemmer should remove only one of a set of seemingly repeating affixes.

[11]A preliminary version of this algorithm appeared in Nazief and Adriani [1996] and in Asian et al. [2005].

(3) We can use confix restriction during stemming to rule out invalid affix combinations.
(4) When restoring characters after prefix removal, we perform recoding if necessary. We explain this in Step 5 of the next section.

4.1 Detailed Approach

We now describe our stemming technique in detail.

(1) At the start of processing, and at each step, check the current word against the root word dictionary; if the lookup succeeds, the word is considered to be a stem, and processing stops.
(2) Remove inflectional suffixes. As described earlier, inflectional suffixes do not affect the spelling of the word they attach to, and multiple inflectional suffixes always appear in order. We first remove any inflectional particle (P) suffixes {"-kah," "-lah," "-tah," "-pun"}, and then any inflectional possessive pronoun (PP) suffixes {"-ku," "-mu," or "-nya"}. For example, the word "bajumulah" ⟨it is your cloth that⟩ is stemmed first to "bajumu" ⟨your cloth⟩, and then to "baju" ⟨cloth⟩. This exists in the dictionary, so stemming stops.
According to our affix model, this leaves the word stem or the stem with derivational affixes, indicated as:

[[[DP+]DP+]DP+] root-word [+DS]

(3) Remove any derivational suffixes {"-i," "-kan," and "-an"}. In our affix model, this leaves

[[[DP+]DP+]DP+] root-word

Consider the word "membelikan" ⟨to buy for⟩; this is stemmed to "membeli" ⟨to buy⟩. Since this is not a valid dictionary root word, we proceed to prefix removal.
(4) Remove any derivational prefixes {"be-," "di-," "ke-," "me-," "pe-," "se-," and "te-"}:
   (a) Stop processing if:
      —the identified prefix forms an invalid affix pair with a suffix that was removed in Step 3;
      —the identified prefix is identical to a previously removed prefix; or
      —three prefixes have already been removed.
   (b) Identify the prefix type and disambiguate if necessary. Prefixes may be of two types:
   Plain. The prefixes {"di-," "ke-," "se-"} can be removed directly.
   Complex. Prefixes starting with {"be-," "te-," "me-," or "pe-"} have different variants, and must be further disambiguated using the rules described in Section 4.2. For example, the prefix "me-" could become

"mem-," "men-," "meny-," or "meng-" depending on the letters at the beginning of the root word.[12]

In the previous step, we partially stemmed the word "membelikan" to "membeli." We now remove the prefix "mem-" to obtain "beli." This is a valid root, and so processing stops.

If none of the prefixes above match, processing stops, and the algorithm indicates that the root word was not found.

(c) If a dictionary lookup for the current word fails, we repeat Step 4 (this is a recursive process). If the word is found in the dictionary, processing stops.

(5) If, after recursive prefix removal, the word has still not been found, we check whether recoding is possible by examining the last column of Table II. This column shows the prefix variants and recoding characters to use when the root word starts with a certain letter, or when the first syllable of the root word ends with a certain letter or fragment. A recoding character is indicated by a lowercase letter following the hyphen and outside the braces. Not all prefixes have a recoding character.

For example, the word "menangkap" ⟨to catch⟩ satisfies Rule 15 for the prefix "me-" (the initial prefix "men-" is followed by a vowel "a-"). After removing "men-" as in Step 4, we obtain "angkap," which is not a valid root word.

For Rule 15, there are two possible recoding characters, "n" (as in "men-*n*V. . . ") and "t" (as in "men-*t*V. . . "). This is somewhat exceptional; there is only one recoding character in most cases. The algorithm prepends "n" to "angkap" to obtain "nangkap," and returns to Step 4. Since this is not a valid root word, "t" is prepended instead to obtain "tangkap" ⟨catch⟩, and we return to Step 4. Since "tangkap" is a valid root word, processing stops.

(6) If all steps are unsuccessful, the algorithm returns the original unstemmed word.

Although confixes are not explicitly removed in the above steps, they are removed indirectly during prefix and suffix removal. There are some exception cases; for example, the confix "pe-an" in the word "pengusutan" could mean either "entanglement" if derived from "kusut" ⟨tangled⟩, or "examination, investigation" if derived from "usut" ⟨examine⟩. Without context, neither an automatic stemmer nor humans can tell which is the correct stem.

In the following subsections we describe features unique to the CS stemmer.

4.2 Prefix Disambiguation

When we encounter a complex prefix, we determine the prefix limits according to the rules shown in Table II. Consider the word "menangkap" ⟨to catch⟩. Looking at the rules for the prefix letters "me-," we see that the third letter of our word is "n" instead of "m," and so we exclude Rule 10, Rule 11, Rule 12, and Rule 13. We also exclude Rule 14 since the fourth letter "a" is not "c," "d,"

[12]According to Table II, we consider up to five letters.

Table II. Template formulas for derivation prefix rules. The letter "V" indicates a vowel, the letter "C" indicates a consonant, the letter "A" indicates any letter, and the letter "P" indicates a short word fragment such as "er." The prefix is separated from the remainder of the word at the position indicated by the hyphen. A lowercase letter following a hyphen and outside braces is a recoding character. If the initial characters of a word do not match any of these rules, the prefix is not removed. These rules do not strictly follow the affix rules defined in grammar books such as Moeliono and Dardjowidjojo [1988] and Sneddon [1996].

Rule	Construct	Return
1	berV...	ber-V... | be-rV...
2	berCAP...	ber-CAP... where C!='r' and P!='er'
3	berCAerV...	ber-CAerV... where C!='r'
4	belajar...	bel-ajar...
5	beC_1erC...	be-C_1erC ... where C_1!={'r'| 'l'}
6	terV...	ter-V... | te-rV...
7	terCP...	ter-CP... where C!='r' and P!='er'
8	terCer...	ter-Cer... where C!='r'
9	teC_1erC_2...	te-C_1erC_2... where C_1!='r'
10	me{l|r|w|y}V...	me-{l|r|w|y}V...
11	mem{b|f|v}...	mem-{b|f|v}...
12	mempe...	mem-pe...
13	mem{rV|V}...	me-m{rV|V}... | me-p{rV|V}...
14	men{c|d|j|z}...	men-{c|d|j|z}...
15	menV...	me-nV... | me-tV...
16	meng{g|h|q|k}...	meng-{g|h|q|k}...
17	mengV...	meng-V... | meng-kV...
18	menyV...	meny-sV...
19	mempV...	mem-pV... where V!='e'
20	pe{w|y}V...	pe-{w|y}V...
21	perV...	per-V... | pe-rV...
22	perCAP...	per-CAP... where C!='r' and P!='er'
23	perCAerV...	per-CAerV... where C!='r'
24	pem{b|f|v}...	pem-{b|f|v}...
25	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
26	pen{c|d|j|z}...	pen-{c|d|j|z}...
27	penV...	pe-nV... | pe-tV...
28	peng{g|h|q}...	peng-{g|h|q}...
29	pengV...	peng-V... | peng-kV...
30	penyV...	peny-sV...
31	pelV...	pe-lV...; Exception: for "pelajar", return ajar
32	peCP...	pe-CP... where C!={r|w|y|l|m|n} and P!='er'
33	peCerV...	per-CerV... where C!={r|w|y|l|m|n}

"j," or "z." We finally settle on Rule 15, which indicates that the prefix to be removed is "me-." The resultant stem is either "nangkap," which is not in the dictionary, or "tangkap" ⟨to catch⟩, which is in the dictionary.

Some ambiguity remains. For example, according to Rule 17 for the prefix "me-," the word "mengaku" ⟨to admit, to stiffen⟩ can be mapped to either "meng-aku" with the root "aku" ⟨I⟩ or to "meng-kaku" with the root "kaku" ⟨stiff⟩. Both are valid root words, and we can only determine the correct root word from the context.[13]

[13]Currently, the CS stemmer stems "mengaku" to "aku" since it checks whether a resulting stem is in the dictionary before performing recoding.

The same ambiguity can also occur for words that can be a stem or an affixed word. For example, the word "mereka" can be a stem, with the meaning "they," or an affixed word, which could be stemmed to "reka" ⟨to invent, to devise⟩. This is a common stemming problem not unique to Indonesian [Xu and Croft 1998]. To resolve these ambiguities, we must also consider the context surrounding the word.

4.3 Rule Precedence

The order in which rules are applied affects the outcome of the stemming operation. Consider an example where inflectional suffix removal fails. The word "bertingkah" ⟨to behave⟩ is formed from the prefix "be-" and the root word "tingkah" ⟨behavior⟩. However, the algorithm will remove the suffix "-kah" to obtain the word "berting," and then remove the prefix "be-" to obtain the valid word "ting" ⟨lamp⟩. This particular problem arises only in limited cases with specific prefixes and particles.

We may also encounter ambiguity when encountering specific prefixes paired with certain derivational suffixes; for the word "dinilai" ⟨to be marked⟩, we may obtain the construct "*di*-nilai" with the correct stem "nilai" ⟨mark⟩, or the construct "*di*-nila-*i*" with the incorrect (but valid) stem "nila" ⟨indigo⟩.

Stemming may also fail when part of a word is the same as a valid affix. For example, a word "beli" ⟨to buy⟩ can be mistaken as the word "bel"⟨bell⟩ with the suffix "-i."

To minimize the most common ambiguities, we remove the prefix before the suffix when encountering the confix pairs "be-. . . -lah"; "be-. . . -an"; "me-. . . -i"; "di-. . . -i"; "pe-. . . -i"; and "te-. . . -i." In rare instances, however, the suffix should be removed before the prefix. For example, the word "mengalami" ⟨to experience⟩ is derived from "*meng*-alam-*i*," and the correct stem is "alam" ⟨experience⟩. Under our rule precedence, this is treated as "*meng*-alami," producing the valid but incorrect stem "alami" ⟨natural⟩.

Interestingly, the "di-. . . -i" precedence rule can handle misspellings where the locative preposition "di" ⟨in, at, on⟩ appears mistakenly attached to a following word ending with an inflection or derivation suffix such as "-i." For example, the phrase "di sisi" ⟨at the side⟩—with the correct stem "sisi" ⟨side⟩—is sometimes misspelled as "disisi." If we were to first remove the derivation suffix "-i" and then the derivation prefix "di-," we would obtain the stem "sis" ⟨hissing sound⟩. Using the "di-. . . -i" precedence rule, we first remove the prefix "di-." Stemming stops here, since "sisi" appears in the dictionary.

4.4 Hyphenated Words

In Indonesian, hyphenated words are of two types:

Plurals. These are always formed by repeating the singular form of the word. For example, "buku" ⟨book⟩ becomes "buku-buku" ⟨books⟩.

Composite words. These have distinct component words—sometimes derived from a common root—separated by a hyphen.

For example, in "sebaik-baiknya" ⟨as good as possible⟩ the words on either side of the hyphen are formed from the common root, "baik" ⟨good⟩. In contrast,

the components of "bolak-balik" ⟨to and fro⟩ stem to "bolak" (a valid word with no independent meaning) and "balik" ⟨come back⟩.

When the words on either side of the hyphen have the same root, we return this root as the stem. Thus, for example, "buku-buku" is stemmed to "buku," and "sebaik-baiknya" is stemmed to "baik." Where the component stems are different, as with "bolak-balik," we assume that the original hyphenated word is the stem.

This approach does not always succeed. For example, the word "benar-tidaknya" ⟨the right or wrong of⟩ should be stemmed to "benar-tidak" ⟨right or wrong⟩. However, our algorithm stems and compares the words on either side of the hyphen to get "benar" ⟨right⟩ and "tidak" ⟨wrong⟩. Since they are not the same, the stem is assumed to be the original word "benar-tidaknya." It is difficult to avoid this type of problem without incorporating explicit lookup lists.

5. EXPERIMENTAL EVALUATION

To determine the effectiveness of our approach, we conduct two experiments. First, we examine how well it—and other state-of-the-art stemmers—perform on basic stemming against a human baseline. Second, we explore how stemming affects information retrieval from Indonesian text.

For these, we require data collections and appropriate ground truth. The absence of suitable existing testbeds led us to create our own; we describe these in the following sections.

5.1 Comparison with Humans

The first experiment we report assesses the effectiveness of automated stemmers against a baseline created from user experiments.

5.1.1 *Collection.* We formed a collection of words to be stemmed by extracting every fifth word from a set of 9,898 news stories[14] from the online edition of the Kompas[15] newspaper published between January and June 2002. We define a word as a sequence of characters enclosed by white space, with a letter as the first character.

The mean word length (including short words) in this list is 6.15, while the mean word length in DICT-UI is 6.75. We have found that words shorter than six characters are generally root words and so rarely require stemming. For our list containing words with five or fewer characters, only about 0.04% of such words (39 unique words) from 1,419,383 nonunique words were not root words, and so we decided to omit words with fewer than six characters from our collection. Note that, by design, our algorithm does not stem words shorter than three characters; this is an orthogonal issue to the collection creation process.

[14]In previous work, we have stated that the collection contains 9,901 news items [Asian et al. 2005]. We later discovered that three items were test documents, and excluded them.
[15]http://www.kompas.com

Table III. Results of manual stemming by four Indonesian native speakers, denoted as A to D. The values shown are the number of cases out of 3,986 where participants agree, with the percentage indicated in parentheses.

	B	C	D
A	3,674 (92%)	3,689 (93%)	3,564 (89%)
B		3,588 (90%)	3,555 (89%)
C			3,528 (89%)

We obtain 1,807 unique words forming a final collection of 3,986 nonunique words, reflecting—we believe—a good approximation of their frequency of use. We chose to extract nonunique words to reflect the real-world stemming problem encountered in text search, document summarization, and translation. The frequency of word occurrence in normal usage is highly skewed [Williams and Zobel 2005]; there are a small number of words that are very common, and a large number of words that are used infrequently. In English, for example, the word "the" appears about twice as often as the next most common word; a similar phenomenon exists in Indonesian, where "yang" (a relative pronoun that is similar to "who," "which," or "that," or "the" if used with an adjective) is the most common word. It is important that an automatic stemmer processes common words correctly, even if it fails on some rarer terms.

We use the collection in two ways. First, we investigate the error rate of stemming algorithms relative to manual stemming for the nonunique word collection. This permits quantifying the overall error rate of a stemmer for a collection of real-world documents, that is, it allows us to discover the total errors made. Second, we investigate the error rate when stemming unique words only. This allows us to investigate how many different errors each scheme makes, that is, the total number of unique errors. Together, these allow effective assessment of stemming accuracy.

5.1.2 *Ground Truth for Stemming.* Humans do not always agree on how a word should be stemmed, nor are they always consistent. When producing our ground truth, we deliberately cater for these characteristics. We asked four native Indonesian speakers to provide the appropriate root for each of the 3,986 words in the list.[16] The words were listed in their order of occurrence, that is, a word could be repeated at different points in the collection, and words were not grouped by prefix. Table III shows the level of agreement between the assessors: as expected, there is no consensus as to the root words between the assessors, and indeed, the agreement ranges from around 93% (for assessors A and C) to less than 89% (for assessors C and D). For example, the word "bagian" ⟨part⟩ is left unstemmed in some cases and stemmed to "bagi" ⟨divide⟩ in others, and similarly "adalah" ⟨to be⟩ is sometimes stemmed to "ada" ⟨exists⟩ and sometimes left unchanged. Indeed, the latter example illustrates another problem: in some cases, an assessor was inconsistent, on some occasions stemming "adalah" to "ada," and on others leaving it unchanged.

[16]Three of the assessors were undergraduate students, while the fourth was a PhD candidate.

Table IV. Consensus and majority agreement for manual stemming by four Indonesian native speakers, denoted as A to D. The values shown are the number of cases out of 3,986 where participants agree.

	ABCD	ABC	ABD	ACD	BCD	Any three
Number	3,292	3,493	3,413	3,408	3,361	3,799
Percentage	(82.6%)	(87.6%)	(85.6%)	(85.5%)	(84.3%)	(95.3%)

Having established that native speakers disagree and also make errors, we decided to use the majority decision as the correct answer. Table IV shows the number of cases where three and four assessors agree. All four assessors are in agreement on only 82.6% of all words, and the level of agreement between any set of three assessors is only slightly higher. The number of cases where any three or all four assessors agree (shown as "Any three") is 95.3%. We use this latter case as our first baseline to compare to automatic stemming: if a majority agree, we keep the original word in our collection and note its answer as the majority decision. We refer to this as the MAJORITY baseline; it contains 3,799 words. Words that do not have a majority stemming decision are omitted from the collection.

Clearly, the majority decision is not necessarily the correct one. First, the majority may make a mistake. For example, the word "gerakan" ⟨movement⟩ can be correctly stemmed to either the root word "gera" ⟨to frighten⟩ or "gerak" ⟨to move⟩. For this particular word, all four assessors stemmed "gerakan" to "gerak."

Second, the majority may confuse words. For example, the word "penebangan" ⟨cutting down⟩ should be correctly stemmed to "tebang" ⟨to cut down⟩. However, the majority misread this as "penerbangan" ⟨flight⟩, and so stemmed it to "terbang" ⟨to fly⟩.

Third, the lack of consistency of individual assessors means that the majority decision for individual words may in fact vary across the occurrences of that word. For example, the word "adalah" was stemmed by three assessors to "ada" in some cases, and left unstemmed in others. From our collection of 3,799 words, the 1,751 unique words map to 1,753 roots according to the majority decision.

These problems are rare, and the majority decision is a good baseline. We complement this with two further baselines. One is the set of 1,753 unique roots reported by the users. We refer to this set as UNIQUE and use it to assess algorithm performance on unique words. We also use a third baseline formed from the answers provided by at least one assessor; this set contains the original 3,986 nonunique words, and we refer to this set as SUBJECTIVE.

5.1.3 *Results and Analysis.* Table V shows the results of automatic stemming for the MAJORITY, UNIQUE, and SUBJECTIVE collections. Our scheme produces the best results, correctly stemming 94.8% of word occurrences in MAJORITY, 95.3% of UNIQUE, and 97.0% of SUBJECTIVE. For MAJORITY, this is some 6% better than the best-performing other scheme (AHMAD-B_2). Using McNemar's one-tailed test [Sheskin 1997], we have found this difference to be statistically significant at the 99% confidence level. We observe that the only nondictionary scheme, VEGA-1, is less effective than even the IDRIS-2

Table V. Automatic Stemming Performance Compared to the MAJORITY, UNIQUE and SUBJECTIVE Baseline Collections

	MAJORITY		UNIQUE		SUBJECTIVE	
Stemmer	Correct (%)	Errors (words)	Correct (%)	Errors (words)	Correct (%)	Errors (words)
CS	94.8	196	95.3	82	97.0	119
AHMAD-B_2	88.8	424	88.3	205	91.4	344
IDRIS-2	87.9	458	88.8	197	89.8	405
ARIFIN	87.7	466	88.0	211	90.0	397
VEGA-1	66.3	1,280	69.4	536	67.7	1,286

and AHMAD-B_2 schemes designed for stemming Malaysian. It makes almost five times as many errors on the MAJORITY collection as CS, illustrating the importance of validating decisions using an external word source.

On the UNIQUE collection, IDRIS-2 is 0.5% or eight words better than the AHMAD-B_2 scheme on which it is based. However, on the MAJORITY collection, it is 0.9% or 34 words worse. This illustrates an important characteristic of our experiments: stemming algorithms should be considered in the context of word occurrences and not unique words. While IDRIS-2 makes fewer errors on rare words, it makes more errors on more common words, and is less effective overall.

The particle "-tah" is dated and rarely used in modern Indonesian, and incorporating it in our stemmer causes errors. For example, the word "pemerintah" ⟨government⟩—derived from the root "perintah" ⟨rule, order⟩—is incorrectly stemmed to "perin" (a valid word with no independent meaning). Similarly, the words "dibantah" ⟨to be denied⟩ and "membantah" ⟨to deny⟩—derived from the root "bantah" ⟨to argue, deny⟩—are incorrectly stemmed to "ban" ⟨wheel⟩. Not catering for this prefix actually improves effectiveness from 94.7% (with 201 errors) to 94.8% (with 196 errors) for MAJORITY, and from 95.2% (with 85 errors) to 95.3% (with 82 errors) for UNIQUE. This particle is not implemented by other Indonesian stemming algorithms but is handled by templates B and C of the Malaysian stemming algorithm of Ahmad et al. [1996]. All experiments we report here exclude this prefix.

The stemming approaches exhibited comparable performance across collections, with the CS stemmer performing the best. As expected, performance on SUBJECTIVE is slightly better than for MAJORITY or UNIQUE, since an automated approach need only agree with a single assessor. ARIFIN produces slightly better results than IDRIS-2 for SUBJECTIVE, but the difference is very small (0.2%).

We note that some prefixes are mutually exclusive. For example, neither the prefix "me-" nor the prefix "ke-" can ever appear with the prefix "di-." The word "mendidik" ⟨to educate⟩ is derived from the prefix "me-" and the stem "didik" ⟨to educate⟩. However, none of the Indonesian stemming algorithms that use a dictionary—except for the algorithm of which uses template rules—restrict the combination and order of derivational prefix removal; this could lead to overstemming. If the word "didik" is not in the dictionary, the fragment "di-" at the beginning of the word is considered to be a prefix, and the determined stem is "dik" ⟨a younger sibling⟩. Since all the algorithms check the dictionary

Table VI. Classified failure cases of the CS stemmer on the MAJORITY collection. The total shows the total occurrences, not the number of unique cases.

	Example			
Fault Class	Original	Error	Correct	Cases
Nonroot words in dict.	sebagai	sebagai	bagai	92
Incomplete dictionary	bagian	bagi	bagian	31
Misspellings	penambahanan	penambahanan	tambah	21
Peoples' names	Abdullah	Abdul	Abdullah	13
Names	minimi	minim	minimi	9
Composite words	pemberitahuan	pemberitahuan	beritahu	7
Recoding ambiguity-dict.	berupa	upa	rupa	7
Acronyms	pemilu	milu	pemilu	4
Understemming	mengecek	ecek	cek	4
Hyphenated words	masing-masing	masing-masing	masing	3
Recoding ambiguity-rule	peperangan	perang	erang	3
Foreign words	mengakomodir	mengakomodir	akomodir	1
Human error	penebangan	terbang	tebang	1
Total				196

after each removal, this problem is rare, and occurs only when the dictionary is not complete.

Table VI shows the main categories where our stemming approach fails to produce the same result as the majority baseline.[17] The largest portion (46%) of problems are caused by a nonroot word appearing in the dictionary, causing stemming to end prematurely. Another 16% of failures occur when the relevant root word is not in the dictionary, causing the algorithm to backtrack unnecessarily. Other difficulties are caused by the absence in the dictionary of proper nouns, composite words, acronyms, and foreign words.

The CS stemmer does not solve all stemming problems, as ambiguity is inherent in human languages. The understemming problem is also indirectly related to ambiguity. If we include the prefix "menge-" to correctly stem the word "mengecek" ⟨to check⟩ to "cek" ⟨to check⟩, the word "mengenang" ⟨to reminisce about⟩ will be wrongly stemmed to "nang" (a proper noun existing in the dictionary) instead of the correct stem "kenang" ⟨to think of⟩. To solve problems such as word-sense ambiguity and homonymity, we must incorporate more detailed knowledge of the language to be stemmed. Furthermore, disambiguation tasks require the context surrounding the words to be stemmed, and a large data collection to allow collection of statistical data.

We discuss other problems not specific to any one algorithm in Section 7.

5.2 Ad Hoc Queries

A major application of text processing techniques is information retrieval, where documents are retrieved in response to a user query. These documents are typically provided to the user as a list, sorted by decreasing estimated likelihood of being relevant to the user's information need as expressed in the query [Jones et al. 2000].

[17] We classify human errors and misspellings as two separate issues. Misspellings occur when the words are written, while the human errors occur when assessors wrongly stem a word.

The effectiveness of a retrieval system is commonly expressed in terms of its *precision*—the fraction of retrieved items that are relevant—and its *recall*—the fraction of relevant items that are retrieved. The related *R-Precision* measure computes the precision at the *N*th retrieved item, where *N* is the number of relevant items in the collection.

To explore whether stemming is useful in the context of Indonesian text retrieval, we must evaluate whether each retrieved document is relevant to a query. To automate what is essentially a manual task, we require a testbed comprising a common document collection, a set of queries, and a list of documents in the collection that are considered to be relevant to each query.[18]

5.2.1 *Collection.* The Text REtrieval Conference (TREC) series provides researchers with appropriate testbeds for evaluating information retrieval (IR) techniques for several retrieval paradigms [Harman 1992]. The original TREC track was named the *ad hoc* track, and aimed to investigate searches for new topics in archived data. This is the approach used by most users of Web search engines, where the typical query is a phrase or set of keywords that describe an information need.

The TREC data collections and relevance judgments are widely used by IR researchers. The Linguistic Data Consortium (LDC) [Liberman and Cieri 1998] also provides data collections for use by the IR community. However, there is no publicly available testbed for Indonesian IR. The Indonesian document collections that do exist [Fahmi 2004; Tala 2003; Vega 2001] either do not have topics and relevance judgments, or are not published. To allow rigorous evaluation of IR techniques for Indonesian, we formed a collection and associated ground truth using the first 3,000 news articles from the 9,898 articles described in Section 5.1.1. The resulting collection is around 750 KB in size and contains 38,601 distinct words. The rest of the articles are used as training data for the experiments described in Section 6.2.

Compared to text collections typically used for English text IR, this collection is relatively small. Two examples of collections commonly used for English text IR research are the *Wall Street Journal* articles (1987–1989) of size 276 MB with 98,732 documents, and the Associated Press newswire dispatches (1989) of size 254 MB with 84,678 documents [Voorhees and Harman 1999]. Nevertheless, it is a starting point for further development of Indonesian IR research. The small size of our collection also allows detailed ground truth to be prepared; with TREC document collections, not every document is judged, and a pooling method is used [Voorhees and Harman 1999]. As [Zobel 1998] points out, if the pool is not deep enough, pooling may favor newer systems that combine and improve the retrieval techniques of old systems. Effectiveness measurement may also discount actual relevant documents that have not been seen by the reviewers during the relevance judgment process.

Following the TREC approach [Voorhees and Harman 1999], we kept the data as close to the original as possible, and did not correct any faults such as spelling mistakes or incomplete sentences. The collection is stored in a single

[18]An initial description of this collection and the ad hoc queries appeared in [Asian et al. 2004].

```
<DOC>
<DOCNO>news10513-html</DOCNO>
Mayjen Syafrie Samsuddin akan Jadi Kapuspen TNI JAKARTA (Media):
Mantan Pangdam Jaya Mayjen Syafrie Samsuddin akan menjadi
Kapuspen TNI menggantikan Marsekal Muda Graito Husodo.  Menurut
informasi yang diperoleh Antara Jakarta Kamis, Syafrie Samsuddin
menjadi Kapuspen TNI dan serah terima jabatan akan dilakukan
pada akhir Februari 2002.  Namun kebenaran informasi tersebut
hingga kini belum dapat dikonfirmasikan ke Kapuspen TNI.
( M-1 )
</DOC>
```

Fig. 1. An example Kompas newswire document from our test collection, marked up in the TREC format.

```
<top>
<num> Number: 14
<title> nilai tukar rupiah terhadap
dolar AS
<desc> Description: Dokumen harus
menyebutkan nilai tukar rupiah
terhadap dolar AS.
<narr> Narrative: Asalkan dokumen
ada menyebutkan nilai tukar rupiah
terhadap dollar tanpa indikasi
menguat atau melemah sudah dianggap
relevan.  Prediksi nilai tukar
dianggap tidak relevan.
</top>
```

```
<top>
<num> Number: 14
<title> The exchange rate between
rupiah and US dollar
<desc> Description: Document shall
mention the exchange rate of
Indonesian rupiah against US dollar.
<narr> Narrative: The document is
relevant as long as it mentions the
exchange rate of rupiah against USA
dollar, even without indication
whether rupiah strengthened or
weakened. Exchange rate prediction
is not relevant.
</top>
```

Fig. 2. An example topic (left) and its English translation (right).

file, marked up using standard TREC tags. An example document is shown in Figure 1; the tags <DOC> and </DOC> mark the beginning and end of a document respectively, and each document has a document identifier delimited by the <DOCNO> and </DOCNO> tags.

5.2.2 *Ground Truth for IR Queries.* After examining the documents, we compiled 20 query topics that would have relevant answers in the collection. The topics are of two types: *general*, where many documents meet the information need, and *specific*, where the set of relevant documents is small. We define general topics as those containing ten or more relevant documents; an example of a general query on our collection is, "World Cup Report" (topic 13). Specific topics have fewer than ten relevant documents; for example, the query, "What are the symptoms and causes of asthma?" (topic 10) is specific when applied to our collection. An example of an Indonesian topic and its English translation is shown in Figure 2. The queries follow the TREC format [Voorhees and Harman 2000], with a number, title, description, and narrative.

Table VII. Precision Measures Before and After Stemming

Measure	Without Stemming	With Stemming
Mean Average Precision	0.4605	0.4842
Mean Precision at 10	0.3750	0.3550
Mean R-Precision	0.4210	0.4534

To determine which documents are relevant to the information needs expressed by each query, each of the 3,000 documents was read and judged manually. This resulted in an exhaustive tabulation of $20 \times 3{,}000 = 60{,}000$ relevance assessments. Additional information and the testbed itself are available online.[19]

A topic set of size 20 may not be significant, and the results presented here are not necessarily reproducible with other collection sets [Sanderson and Zobel 2005]. However, we believe that the preliminary results they offer can aid further research in Indonesian IR.

5.2.3 *Results and Analysis.* To explore whether stemming aids retrieval for Indonesian text, we evaluated its effect over the 20 queries on our collection of 3,000 documents. We used the `zettair`[20] search engine to query the stemmed and unstemmed data with the topic titles, returning 100 answers per query. `Zettair` has native support for the `TREC` format for collections, topics, and relevance assessments.

Table VII shows the results of this experiment. We calculate average precision for each query by taking the sum of the precision values at each answer document for the first 100 answers, assuming that answers not in the top 100 to have precision values of 0, and divide the sum by the number of relevant documents in the collection. The mean average precision (MAP) is the average of precision values across all queries. This MAP measure indicates that stemming improves retrieval performance by around 2%. The second row shows average precision after processing 10 documents, averaged over all queries; this shows a 2% drop in performance when stemming is performed. The R-Precision results in the final row favor stemming by around 3%.

The Wilcoxon signed ranked test indicates that these differences in performance are not statistically significant at the 95% confidence level. Indonesian words have many more variants than those in English, and we expected that the removal of prefixes, infixes, and suffixes should improve retrieval performance. However, these results are consistent with those observed in English text retrieval [Hull 1996].

Figure 3 shows the per-query performance. For each topic, three bars are shown: (1) to the left, the total number of relevant documents; (2) in the middle, the number of relevant documents found without stemming; and, (3) on the right, the number of relevant documents found with stemming. The results show that—with the exception of topics 2, 9, and 19—there is little difference between recall with and without stemming. The overall recall value without stemming is 0.728; with stemming, this increases to 0.781.

[19] http://www.cs.rmit.edu.au/~jelita/corpus.html
[20] http://www.seg.rmit.edu.au/zettair/

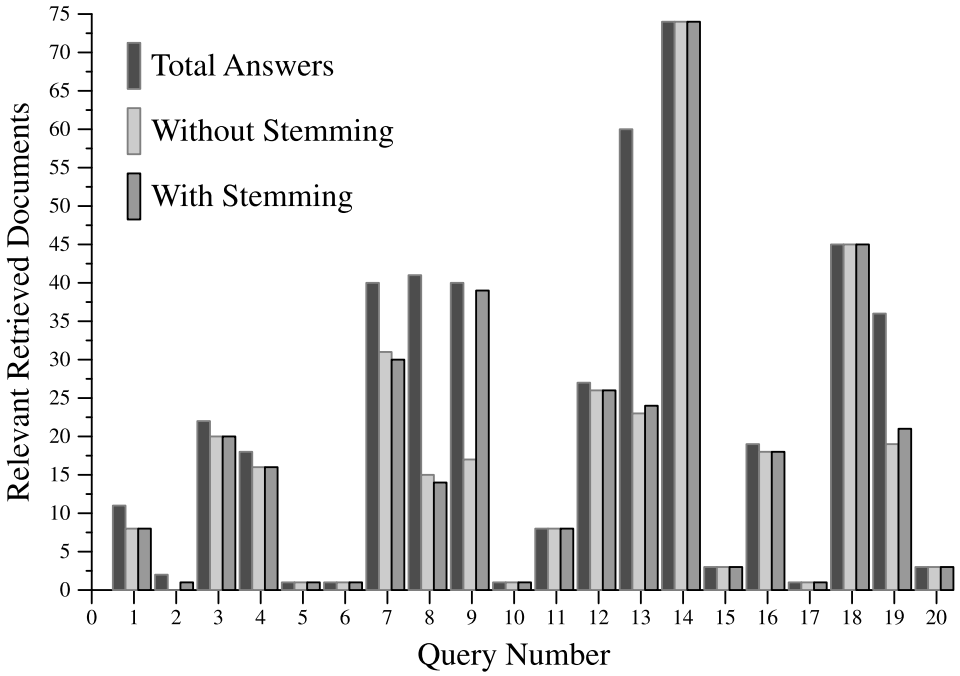

Fig. 3. Topic-by-topic performance with and without stemming. For each topic, the left column shows the number of relevant documents, the middle column represents the number retrieved without stemming, and the right column shows the number retrieved with stemming. Only the query title text is used for querying.

6. EXTENSIONS

The dictionary plays a very important role in algorithm effectiveness. From Table VI, we see that 51 of 196 stemming errors (26%) can be traced to an incomplete dictionary or to misspellings. In this section, we describe three extensions to the stemmer; two address the dictionary issue, while the third aims to prevent stemming of proper nouns.

6.1 Dictionary Augmentation Using *n*-grams

To offset problems caused by words not appearing in the dictionary, we propose a scheme based on *n*-grams, a common mechanism for evaluating similarity between strings [Bakar et al. 2000; Ng et al. 2000]. A character string of length N can be decomposed into $N - n + 1$ substrings or *n*-grams of length n. For example, the string "perintah" can be decomposed into the 2-grams "pe," "er," "ri," "in," "nt," "ta," and "ah."

Consider the case where this word appears misspelled as "perimtah"; the corresponding *n*-grams would be "pe," "er," "ri," "im," "mt," "ta," and "ah." Of these seven *n*-grams, five—or 71.4%—are identical to those of the correctly spelled word.

Table VIII. Comparison of stemming performance for original CS plus CS with dictionary extension using *n*-grams and the *Q*-gram distance measure for MAJORITY and UNIQUE collections

	MAJORITY		UNIQUE	
Stemmer	Correct (%)	Errors (words)	Correct (%)	Errors (words)
CS	94.8	196	95.3	82
CS, 4-grams	83.5	626	81.4	324
CS, 5-grams	94.8	197	95.2	83
CS, 6-grams	94.8	196	95.3	82
CS, 7-grams	94.8	196	95.3	82

We can employ this trend to augment the word list. If, at the conclusion of all stemming operations, the final dictionary lookup step fails, we can return the closest dictionary word using the approach of Zobel and Dart [1996]. To determine the closest dictionary word, we can use one of the measures described in the literature; these include *Q*-grams [Ukkonen 1992], Priority [Zobel and Dart 1996], and Onechar [Hall and Dowling 1980].

We have found that of the nine approaches we examined, the *Q*-gram approach, using an *n*-gram length of between 5 and 7 characters, produces the best results. Here, the distance between two strings s and t is defined as: $|G_s|+|G_t|-2|G_s \cap G_t|$ where G_s is the number of *n*-grams in string s and $|G_s \cap G_t|$ is the number of identical *n*-grams in the two strings. For the strings "perintah" and "perimtah," we compute the distance to be $7 + 7 - (2 * |5|) = 4$.

Using *n*-grams does not always result in improved stemming precision. Table VIII shows that 6-grams and 7-grams produce results that are similar to the CS stemmer without any extension, and slightly better than for 5-grams. With longer *n*-gram sizes, we can better avoid incorrectly stemming proper nouns. For example, with 5-grams, the proper nouns "hidayatullah" and "ismatullah" are stemmed to "mullah," while 6-grams and 7-grams avoid this problem. However, in some cases, it is better to use shorter *n*-grams. For example, with 5-grams we correctly stem "tetabuhan" ⟨percussion instrument⟩ to "tabuh" ⟨to hit a percussion instrument⟩, whereas we fail to find a stem when using 6-grams or 7-grams. While we aimed to use *n*-grams to correct misspellings, we found that none of the 19 misspellings that appeared in SUBJECTIVE are correctly handled. The misspelling errors produced by *n*-grams are similar to CS without any extension, which implies that they do not address the problem. Overall, we consider that 5-grams offers the best trade-off; we present a scheme to handle proper nouns later in this section.

Interestingly, the best stemming effectiveness is not accompanied by the best retrieval effectiveness. From Table IX, we see that the best average precision and R-Precision results are actually produced when using 4-grams. This is similar to the situation described by McNamee and Mayfield [2004] for European languages such as English, French, German, Italian, Spanish, and Swedish. Despite the increase in precision, combining the CS stemmer with *n*-grams does not help recall either. We conclude that using *n*-grams does not necessarily correct misspellings, but it is a useful technique for increasing average precision and R-Precision in an information retrieval environment.

Table IX. Performance comparison for original CS and CS with *n*-gram extension using different gram lengths and the *Q*-gram distance measure. The best values for each measure are in bold.

Measure	CS	CS 3-grams	CS 4-grams	CS 5-grams	CS 6-grams	CS 7-grams
Average Precision	0.4842	0.4950	**0.5111**	0.4842	0.4842	0.4842
Precision at 10	**0.3550**	0.3400	0.3400	**0.3550**	**0.3550**	**0.3550**
R-Precision	0.4534	0.4825	**0.5010**	0.4534	0.4534	0.4534

Table X. Comparison of stemming performance (precision) for CS and CS with dictionary extension using *n*-grams and the *Q*-gram distance measure for the MAJORITY and UNIQUE collections using the DICT-UI, DICT-KBBI, and DICT-KEBI dictionaries.

	MAJORITY			UNIQUE		
	CORRECT (%)			CORRECT (%)		
Stemmer	DICT-UI	DICT-KBBI	DICT-KEBI	DICT-UI	DICT-KBBI	DICT-KEBI
CS	94.8	90.4	95.45	95.3	89.8	95.15
CS, 4-grams	83.5	79.9		81.41	75.0	
CS, 5-grams	94.8	90.4		95.24	89.9	
CS, 6-grams	94.8	90.4		95.29	89.9	
CS, 7-grams	94.8	90.4		95.29	89.9	

6.2 Other Issues

In our earlier experiments, we have relied on the DICT-UI dictionary with 29,337 words. We now describe experiments that instead use the "Kamus Besar Bahasa Indonesia" ⟨Greater Indonesian Dictionary⟩ (KBBI) with 27,828 entries, and also the KEBI online dictionary with 22,500 root words.[21] We refer to these dictionaries as DICT-KBBI and DICT-KEBI respectively.

The results shown in Table X are consistent with those for the CS stemmer without any *n*-gram extensions. The performance obtained using the original DICT-UI dictionary is consistently better than that obtained using the DICT-KBBI dictionary.

We did not use the KEBI online dictionary for the ad hoc queries experiment due to the high latency of lookup requests. We were also unable to use it for experiments with *n*-grams, since we require access to the complete dictionary word list when creating the grams.

Table XI shows that the DICT-KBBI produces better results with 3-grams. We hypothesize that this is caused by the larger number of proper nouns, such as "Jakarta" and "Indonesia," contained in the DICT-KBBI. However, longer *n*-grams provide a higher probability of leaving a proper noun unstemmed even if it is not in the dictionary, and so offer a more robust choice.

6.3 Identifying Proper Nouns

Thompson and Dozier [1997] state that proper nouns make up between 38.8% and 67.8% of queries. Proper nouns are considered to be root words, and should not be stemmed. In Table VI, around 13.27% of stemming errors are shown to be caused by improperly stemming proper nouns. To address this problem, we

[21] http://nlp.aia.bppt.go.id/kebi/

Table XI. Comparison of IR performance (average precision) for the CS stemmer, and CS with n-gram extensions with and without Proper Noun Identification using the merged-nouns lists of AU, PIU, BEST-IU, BEST-OIU, and WAT. Experiments using the Q-gram distance measure with DICT-UI and DICT-KBBI.

	DICTIONARY	
Approach	DICT-UI	DICT-KBBI
CS	0.4842	0.4787
CS, 3-grams	0.4950	0.5137
CS, 4-grams	0.5111	0.5095
CS, 5-grams	0.4842	0.4787
CS, 6-grams	0.4842	0.4787
CS, 7-grams	0.4842	0.4787
CS, 4-grams, Proper Nouns	0.5252	0.5181
CS, 5-grams, Proper Nouns	0.4934	0.4856

use the training collection of 6,898 documents from the Kompas Web site to draw up a list of common proper nouns to use alongside our dictionary.

We approach proper noun identification from four aspects that we now describe; Curran and Clark [2003] describe similar aspects of Proper Noun Identification. First, we identify words that are likely to be acronyms and should not therefore be stemmed. Acronyms are typically written in uppercase (all-uppercase, or AU). However, it is common to find acronyms written with only the initial letter in uppercase (OIU), or all lowercase (AL). In some cases, acronyms appear mid-sentence within parentheses: for example, in Narkoba ⟨drugs⟩, which is the acronym for "narkotika dan obat-obatan terlarang."[22] We treat words containing only alphabetical characters, appearing between parentheses, and with at least the initial letter in uppercase, to be acronyms. We represent such words with the symbol PIU.

Second, words that appear mid-sentence with the initial letter predominantly in uppercase are likely to be proper nouns. We may require that at least the initial letter to be in uppercase (IU), or that only the initial letter be in uppercase (OIU). The first rule would match all of the words "Jakarta," "Indonesia," "ABRI" (the acronym for the Indonesian army), and "MetroTV" (a private Indonesian television station), whereas the second rule would match only the first two.

Initially, we believed that words that appear with the initial letter in uppercase (either IU or OIU), and do not appear in the beginning of sentences should be considered to be proper nouns. However, this assumption includes words appearing in titles of documents or in organization or committee names, such as "keterlibatan" ⟨involvement⟩, which can be stemmed to "libat" ⟨involve⟩. Not stemming such words decreases the average precision of ad hoc retrieval. The average precision of the CS stemmer without extension is 0.4842, increasing to 0.5111 and 0.4842 with 4-grams and 5-grams respectively. Stemming is still useful; not stemming this category of words resulted in precision values of 0.4522 and 0.4533 respectively.

[22]The acronym Narkoba appears with initial letter in uppercase (IU) in only 22.7% of instances, in all lowercase (AL) in 75% of instances, and in all uppercase (AU) in 2.3% of instances.

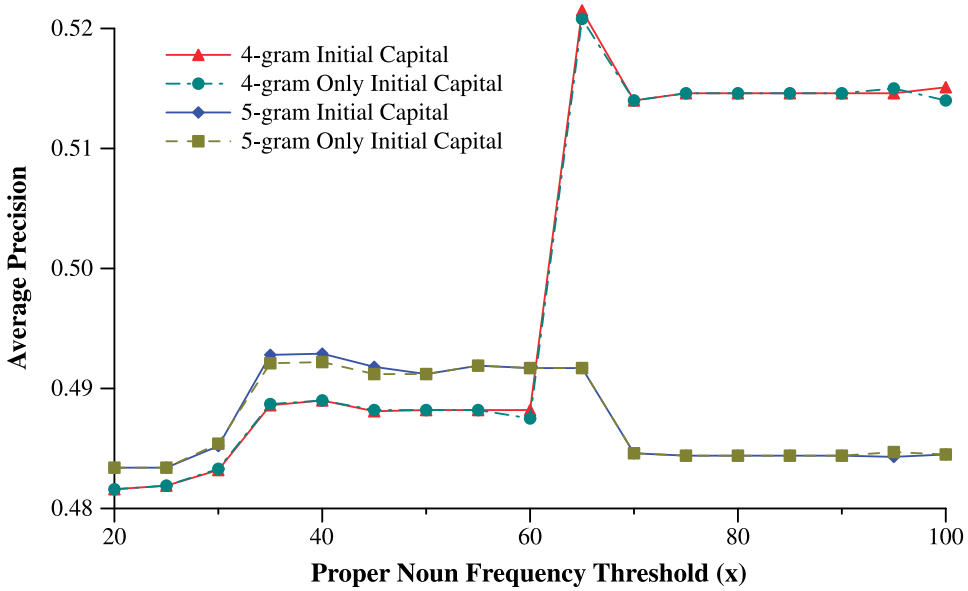

Fig. 4. Mean average precision for the CS stemmer using *n*-grams and proper noun identification. Here, proper nouns are words that appear mid-sentence at least *x* times with the initial letter in uppercase (IU) or only initial letter in uppercase (OIU); *x* is the number corresponding to the x-axis: for example, the MAP value of 0.4882 is achieved by not stemming the proper nouns identified by the 4-gram Initial Capital scheme for appearing at least 60 times in the collection. The 4-gram variants overlap, as do the 5-gram variants.

This result, and the fact that proper nouns do not always appear with consistent capitalization, led us to believe that we should apply a required ratio between the capitalization types. For example, we could require that to be considered a proper noun, a word should appear overwhelmingly with at least the first letter in uppercase.

This inspired further experimentation with different threshold frequencies of words appearing in a particular way in the training collection. Figure 4 shows the retrieval effectiveness for varying thresholds. For 4-grams, the average IU and OIU precision values are quite similar, and the same is the case for 5-grams; this leads to two overlapping lines in the figure. When the threshold is too low, too many words are considered to be proper nouns and are not stemmed (false positives). If the threshold is too high, some proper nouns may be wrongly stemmed (false negatives). We need to determine the threshold that affords the highest average precision. From the figure, we see that average precision peaks for a threshold of 65 for 4-grams and at 40 for 5-grams in both IU and OIU. These numbers translate to approximately 66% of words with the initial letter in uppercase, IU, (62% for 4-grams and 69% for 5-grams) and approximately 69% for only the initial letter in uppercase, OIU, (75% for 4-grams and 63% for 5-grams). We determine that a threshold of 65 is good for both IU and OIU for our collection, and use this ratio, along with *n*-grams

Table XII. Average precision (P), precision at 10 (P@10), and R-Precision (RP) of queries using proper noun identification for unstemmed (US), stemming with CS, and stemming with CS extended with 4-gram dictionary augmentation. Multiple acronyms indicate technique combination, where proper noun lists of component techniques are merged. Key: {AU=All Uppercase, PIU=Parentheses, Initial Uppercase; BEST-IU=Best Initial Uppercase; BEST-OIU=Best Only Initial Uppercase; WAT=Words After Titles; EW=English Words; ALL-EW=All Combinations except English Words}.

Measure	US	CS	CS AU PIU	CS BEST-IU	CS BEST-OIU	CS EW	CS WAT	ALL-EW
P	0.4605	0.4842	0.5169	0.5215	0.5208	0.4746	0.5100	0.5252
P@10	0.3750	0.3550	0.3550	0.3650	0.3650	0.3550	0.3400	0.3700
RP	0.4210	0.4534	0.5002	0.5136	0.5136	0.4347	0.5010	0.5136

of size 4 (which produce the best results for IR experiments) for further experiments as BEST-IU and BEST-OIU.

Third, we consider words from the Indonesian text that also appear in English documents to be proper nouns. We formed a list of these "English words" (EW) from the documents of volumes 1 to 5 of the TREC Research Collection.[23] These documents comprise content from the Associated Press (AP), the *San Jose Mercury* News (SJM), the *Wall Street Journal* (WSJ), the *Financial Times* (FT), and the *Los Angeles Times* (LATimes).

Fourth, we consider words that appear after titles (WAT) such as "Dr." to be probable proper nouns. We produced a list of such words by extracting words of 2-4 letters from our training data, and manually selecting valid titles that are always followed by a proper noun. The resultant list contained the following words (all shown in lower-case): {dr., dra., drs., inf., ir., jl., kec., kol., mr., mrs., nn., ny., p.m., pol., prof., purn., rep., sdr., tn., yth.}. Words that follow these titles are considered to be proper nouns, with two exceptions. First, multiple titles may appear together, as in "Prof. Dr. Ibrahim." Second, single letters may follow titles, as in "Dr. A. Salam"; these are likely to be initials. For such exception cases, we do not consider the word following the title to be a proper noun.

Table XII shows the best average precision these methods achieve. As explained earlier, the BEST-IU and BEST-OIU results were obtained by using the threshold 65. The last column is the combination of all methods that can increase average precision over CS without any extension. Lists are combined by merging the proper nouns in one list with those of another, and removing duplicates.

The combination method excludes the approach of identifying words from English word documents (EW) as proper nouns. Using CS with 4-grams and the EW extensions actually decreases performance from 0.4842 for CS without any extension to 0.4746. This is surprising, since the list itself appears to contain many valid proper nouns. These include personal nouns such as {"abraham," "thatcher," "diponegoro"}, place nouns such as {"afghanistan,"

[23]http://trec.nist.gov/data/docs_eng.html

"thailand," "zimbabwe"}, and also English words that should not be stemmed, such as {"treasury," "frontier," "foreman," "dialogue"}. Nevertheless, the list of words may in fact contain some Indonesian words such as "permodalan" ⟨capitalization⟩ (of which the root word is "modal" ⟨capital⟩) and "pendidikan" (of which the root word is "didik" ⟨to educate⟩). Clearly, not all these Indonesian words are root words. We believe that determination of whether or not to stem nonroot words for improved retrieval effectiveness merits further investigation.

The only technique that produces a statistically significant improvement in the average precision and R-precision over unstemmed words is the combination of AU+PIU+BEST-IU+BEST-OIU+WAT, as shown in the last column of Table XII. This combination technique produces a significantly better average precision and R-precision than no stemming, at a 95% confidence level using the Wilcoxon one-tailed signed ranked test. For precision at 10, this combination technique produces a slightly worse value than no stemming, but the difference is not statistically significant at the 95% confidence level. The basic CS stemmer continues to produce good results. While the remaining techniques do increase average precision, the improvement is not statistically significant.

Adding a proper noun identification component does not increase recall values. The best recall value of 0.781 is similar to the original recall value produced by the CS stemmer. For the other variants that incorporate proper noun identification, recall values range from 0.730 to 0.779. We conclude that using *n*-grams and proper noun identification may increase precision, but not necessarily improve recall.

7. DISCUSSION AND FUTURE WORK

In this work, we have described the principal structure of derived words in the Indonesian language, and introduced a new confix-stripping approach for automatic and highly accurate stemming of derived words to their morphological roots. We have compared the precision of this algorithm with other automated stemming approaches for Indonesian using a baseline created from human experiments, and have shown that it is the most accurate.

We have also reported on results of ad hoc queries on a testbed for Indonesian text retrieval that includes 3,000 newswire documents, 20 topics, and exhaustive relevance judgments. We have shown that stemming does not significantly aid retrieval performance on this collection. We suspect that this is because some relevant documents answer the query implicitly, and do not contain the query terms. For instance the query for "nama bos Manchester United" ⟨the name of the Manchester United boss⟩ does not retrieve a document that discusses "the MU manager." A human assessor understands that "manager" is a synonym of "boss" and "MU" is the acronym of "Manchester United"; automated retrieval systems generally use words directly from the query, and stemming is ineffective here.

There are other possible reasons why stemming might fail to increase precision. Krovetz [1993] and Savoy [1999] suggest that short documents

benefit more from stemming than longer documents. Krovetz suggests that a stemmer that caters for different meanings and disambiguates them might improve precision. From experimentation on French data, Savoy conjectures that more complex stemmers that remove derivational suffixes may cause conflation errors.

We have reported on using *n*-grams to try to correct spelling errors; while using *n*-grams did not in practice handle misspellings well, we observed that it increased the average precision and R-Precision in a text retrieval setting. We have also described schemes for identifying proper nouns in Indonesian text to avoid incorrect stemming. Not stemming proper nouns identified using combinations of different methods, except for English words, does improve precision, although it does not necessarily increase recall.

Some of the stemming problems are not specific to Indonesian. Natural language is inherently ambiguous, and even the best stemming algorithm will still make mistakes. The context where the word appears is important to choosing the right stem. For example, the word "mengurus"⟨to take care of, to become thin⟩ can be stemmed to either "urus"⟨to take care of, to handle⟩ or "kurus" ⟨skinny⟩. Such natural language processing is beyond the scope of this article. The difference in stemming results between humans and an automatic stemmer could also be caused by human error. For example, we have found "adalah"⟨to be⟩ to be stemmed to "ada"⟨to exist⟩ at one time, and left unstemmed at another time, by the same person. These cases are difficult to address.

We plan to explore several extensions to this work. First, we have considered only generic stemming, but many words adopt different meanings in different contexts. Xu and Croft [1998] show that schemes that cater for different content perform better than a generic stemming scheme that stems words independently of the corpus content. As part of our ongoing investigation of stemming, we plan to explore the use of different general-purpose or domain-based dictionaries, including the CICC [1994] dictionary, on stemming effectiveness. Second, we intend to further study schemes of finding proper nouns to increase retrieval effectiveness; preliminary experiments have shown good results. Finally, we plan to expand the number of queries and documents in our testbed to improve the statistical significance of any results.

We believe that the techniques outlined in this work represent a considerable advancement in the literature on Indonesian stemming, and will aid more effective information retrieval for Indonesian text.

ACKNOWLEDGMENTS

We thank Falk Nicolas Scholer for his advice on formulas and statistics; Vinsensius Berlian Vega for the VEGA source code; Riky Irawan for the Kompas newswire documents; and Gunarso for the KBBI dictionary. We also thank Wahyu Wibowo for his help in answering our queries and Eric Dharmazi, Agnes Julianto, Iman Suyoto, and Hendra Yasuwito for their help in creating our human stemming ground truth.

REFERENCES

AHMAD, F., YUSOFF, M., AND SEMBOK, T. M. T. 1996. Experiments with a stemming algorithm for Malay words. *J. Amer. Soc. Inform. Sci. 47*, 12, 909–918.

ARIFIN, A. Z. AND SETIONO, A. N. 2002. Classification of event news documents in Indonesian language using single pass clustering algorithm. In *Proceedings of the Seminar on Intelligent Technology and its Applications (SITIA)*. Teknik Elektro, Sepuluh Nopember Institute of Technology.

ASIAN, J., WILLIAMS, H., AND TAHAGHOGHI, S. 2005. Stemming Indonesian. In *Proceedings of the 28th Australasian Computer Science Conference (ACSC'05)*. V. Estivill-Castro, Ed. Australian Computer Society, Inc., 307-314.

ASIAN, J., WILLIAMS, H. E., AND TAHAGHOGHI, S. 2004. A testbed for Indonesian text retrieval. In *Proceedings of the 9th Australasian Document Computing Symposium (ADCS'04)*. P. Bruza, A. Moffat, and A. Turpin, Eds. University of Melbourne, Department of Computer Science, Melbourne, Australia, 55–58.

BAKAR, Z. A. AND RAHMAN, N. A. 2003. Evaluating the effectiveness of thesaurus and stemming methods in retrieving Malay translated Al-Quran documents. In *Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access*, T. M. T. Sembok, H. B. Zaman, H. Chen, S.R.Urs, and S. Myaeng, Eds. Lecture Notes in Computer Science, vol. 2911. Springer-Verlag, 653–662.

BAKAR, Z. A., SEMBOK, T. M. T., AND YUSOFF, M. 2000. An evaluation of retrieval effectiveness using spelling-correction and string-similarity matching methods on Malay texts. *J. Amer. Soc. Inform. Sci. Technol. 51*, 8, 691–706.

BALLESTEROS, L. AND CROFT, W. B. 1997. Phrasal translation and query expansion techniques for cross-language information retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*. ACM, 84–91.

CICC.1994. Research on Indonesian dictionary. Tech. rep. 6-CICC-MT53, Center of the International Cooperation for Computerization, Tokyo, Japan.

CURRAN, J. R. AND CLARK, S. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of Conference on Natural Language Learning*. W. Daelemans and M. Osborne, Eds. Association for Computational Linguistics, Edmonton, Canada, 164–167.

DEWAN BAHASA DAN PUSTAKA. 1991. *Kamus Dewan (Council Dictionary)*. Dewan Bahasa dan Pustaka, Kuala Lumpur, Malaysia.

DWIPAYANA, G. 2001. *Sari Kata Bahasa Indonesia (The Essence of Indonesian)*. Terbit Terang, Surabaya, Indonesia.

FAHMI, I. 2004. Personal communication.

FRAKES, W. 1992. Stemming algorithms. In *Information Retrieval: Data Structures and Algorithms*, W. Frakes and R. Baeza-Yates, Eds. Prentice-Hall, Englewood Cliffs, NJ, Chapter 8, 131–160.

GAUSTAD, T. AND BOUMA, G. 2002. Accurate stemming of Dutch for text classification. *Lang. Comput. 45*, 1, 104–117.

HALL, P. A. V. AND DOWLING, G. R. 1980. Approximate string matching. *Comput. Surv. 12*, 4, 381–402.

HARMAN, D. 1992. Overview of the First TREC conference (TREC-1). In *Proceedings of the Text Retrieval Conference (TREC)*. NIST Special Publication 500-207, 1–20.

HOLLINK, V., KAMPS, J., MONZ, C., AND DE RIJKE, M. 2004. Monolingual document retrieval for European languages. *Inform. Retrieval 7*, 1-2, 33–52.

HULL, D. A. 1996. Stemming algorithms: A case study for detailed evaluation. *J. Amer. Soc. Inform. Sci. 47*, 1, 70–84.

IDRIS, N. 2001. Automated essay grading system using nearest neighbour technique in information retrieval. M.S. thesis, University of Malaya.

JONES, K. S., WALKER, S., AND ROBERTSON, S. E. 2000. A probabilistic model of information retrieval: Development and comparative experiments. *Inf. Process. Manag. 36*, 6, 779–808.

KROVETZ, R. 1993. Viewing morphology as an inference process. In *Proceedings of the ACM- SIGIR International Conference on Research and Development in Information Retrieval*. R. Korfhage, E. Rasmussen, and P. Willett, Eds. ACM, 191–202.

LARKEY, L. S., BALLESTEROS, L., AND CONNELL, M. E. 2002. Improving stemming for Arabic information retrieval: Light stemming and co-occurrence analysis. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*. M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Järvelin, Eds. ACM, 275–282.

LIBERMAN, M. AND CIERI, C. 1998. The creation, distribution and use of linguistic data. In *Proceedings of the First International Conference on Language Resources and Evaluation*. European Language Resources Association, Granada, Spain.

LOVINS, J. 1968. Development of a stemming algorithm. *Mechanical Transla. Computa. 11*, 22–31.

MCNAMEE, P. AND MAYFIELD, J. 2004. Character n-gram tokenization for European language text retrieval. *Inform. Retrieval 7*, 1-2, 7397.

MOELIONO, A. M. AND DARDJOWIDJOJO, S. 1988. *Tata Bahasa Baku Bahasa Indonesia (The Standard Indonesian Grammar)*. Departemen Pendidikan dan Kebudayaan, Republik Indonesia, Jakarta, Indonesia.

NAZIEF, B. A. A. AND ADRIANI, M. 1996. Confix-stripping: Approach to stemming algorithm for Bahasa Indonesia. Internal publication, Faculty of Computer Science, Univ. of Indonesia, Depok, Jakarta.

NG, C., WILKINSON, R., AND ZOBEL, J. 2000. Experiments in spoken document retrieval using phonetic n-grams. *Speech Comm* (Special issue on Accessing Information in Spoken Audio). 32, 12 61–77.

ORĂSAN, C., PEKAR, V., AND HASLER, L. 2004. A comparison of summarisation methods based on term specificity estimation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*. European Language Resources Association, 1037–1041.

OTHMAN, A. 1993. Pengantar perkataan Melayu untuk sistem capaian dokumen (introduction to Melayu words for document retrieval system). M.S. thesis, University Kebangsaan Malaysia.

PAICE, C. D. 1996. Method for evaluation of stemming algorithms based on error counting. *J. Amer. Soc. Inform. Sci. 47*, 8, 632–649.

PAYNE, T. E. 1997. *Describing Morphosyntax: A Guide for Field Linguists*. Cambridge University Press, Cambridge, UK.

POPOVIČ, M. AND WILLETT, P. 1992. The effectiveness of stemming for natural-language access to Slovene textual data. *J. Amer. Soc. Inform. Sci. 43*, 5, 384–390.

PORTER, M. 1980. An algorithm for suffix stripping. *Program 13*, 3, 130-137.

QUINN, G. 2001. *The Learner's Dictionary of Today's Indonesian*. Allen & Unwin, St. Leonards, Australia.

SANDERSON, M. AND ZOBEL, J. 2005. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*. G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, and N. Ziviani, Eds. ACM, 162–169.

SAVOY, J. 1993. Stemming of French words based on grammatical categories. *J. Amer. Soc. Inform. Sci. 44*, 1 (January), 1–9.

SAVOY, J. 1999. A stemming procedure and stopword list for general French corpora. *J. Amer. Soc. Inform. Sci. 50*, 10, 944–952.

SHESKIN, D. 1997. *Handbook of Parametric and Nonparametric Statistical Proceedures*. CRC Press LLC, Boca Raton, FL.

SNEDDON, J. N. 1996. *Indonesian: A Comprehensive Grammar*. Routledge, London, UK.

TALA, F. 2003. A study of stemming effects on information retrieval in Bahasa Indonesia. M.S. thesis, University of Amsterdam.

THOMPSON, P. AND DOZIER, C. 1997. Name searching and information retrieval. In *Proceedings of Second Conference on Empirical Methods on Natural Language Processing*. C. Cardie and R. Weischedel, Eds. Association for Computational Linguistics, 134–140.

UKKONEN, E. 1992. Approximate string-matching with q-grams and maximal matches. *Theor. Comput. Sci. 92*, 1, 191–211.

VEGA, V. B. 2001. Information retrieval for the Indonesian language. M.S. thesis, National University of Singapore.

VOORHEES, E. AND HARMAN, D. 1999. Overview of the 8th Text REtrieval Conference (TREC-8). In *Proceedings of the Text Retrieval Conference (TREC)*. E. Voorhees and D. Harman, Eds. TREC, NIST Special Publication 500-246, 1–23.

VOORHEES, E. M. AND HARMAN, D. 2000. Overview of the 9th TREC conference (TREC-9). In *Proceedings of the Text Retrieval Conference (TREC)*. E. Voorhees and D. Harman, Eds. NIST Special Publication 500-249, 1–14.

WIDYAMARTAYA, A. 2003. *Seni Menerjemahkan*, 13th Ed. Kanisius, Yogyakarta, Indonesia.

WILLIAMS, H. AND ZOBEL, J. 2005. Searchable words on the Web. *Int. J. Digit. Libr. 5*, 2, 99–105.

WILUJENG, A. 2002. *Inti Sari Kata Bahasa Indonesia Lengkap*. Serba Jaya, Surabaya, Indonesia.

WOODS, P., RINI, K. S., AND MEINHOLD, M. 1995. *Indonesian Phrasebook* 3rd Ed. Lonely Planet Publications, Hawthorn, Australia.

XU, J. AND CROFT, W. B. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inform. Syst. 16*, 1, 61–81.

ZOBEL, J. 1998. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*. W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, Eds. ACM, 307–314.

ZOBEL, J. AND DART, P. 1996. Phonetic string matching: Lessons from information retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*. H. P. Frei, D. Harman, P. Schauble, and R. Wilkinson, Eds. ACM, 166–173.

Received September 2006; revised March 2007; accepted October 2007.