Effective Techniques for Indonesian Text Retrieval

A thesis submitted for the degree of
Doctor of Philosophy

Jelita Asian B.Comp. Sc.(Hons.),
School of Computer Science and Information Technology,
Science, Engineering, and Technology Portfolio,
RMIT University,
Melbourne, Victoria, Australia.

30th March, 2007

Declaration

I certify that except where due acknowledgment has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Jelita Asian
School of Computer Science and Information Technology
RMIT University
30th March, 2007

Acknowledgments

First and foremost, I thank Justin Zobel, Saied Tahaghoghi, and Falk Scholer for their patience and general academic and moral support during my candidature. Without their guidance, the thesis would not exist. I thank Hugh Williams for supervising me for two years before leaving RMIT.

I thank Halil Ali for crawling most of the document collections, and Bobby Nazief for providing the S_NA source code and the dictionary used in this thesis; Vinsensius Berlian Vega for his S_V source code; Riky Irawan for the Kompas newswire documents; and Gunarso for the Kamus Besar Bahasa Indonesia (KBBI) dictionary. I also thank Wahyu Wibowo for his help in answering queries and Eric Dharmazi, Agnes Julianto, Iman Suyoto, Hendra Yasuwito, Debby Andriani, Sinliana, Malian, Susanna Gunawan and Hanyu for their help in creating our human stemming ground truth. I extend my gratitude to Beti Dimitrievska, Chin Scott, and Cecily Walker for their assistance to research students at RMIT.

I also thank my parents and friends for their moral support. I thank many students of the RMIT Search Engine Group: Steven Garcia, Pauline Chou, Ranjan Sinha, Michael Cameron, Bodo Billerbeck, William Webber, Nick Lester, Sarvnaz Karimi, Dayang Iskandar, Ying Zhao, Milad Shokouhi, Nikolas Astikis, Iman Suyoto, Jonathan Yu, Yaniv Bernstein, Abdusalam Nwesri, Jovan Pechevski, Vaughan Shanks, Abhijit Chattaraj, Yanghong Xiang, Pengfei Han, Yohannes Tsegay, and Rosette Kidwani. They have provided valuable assistance during my research and have made my candidature experience interesting.

This research was conducted with the support of an International Postgraduate Research Scholarship (IPRS) scholarship. Hardware used for experiments was provided with the support of the Australian Research Council and RMIT University VRII grant.

Credits

Portions of the material in this thesis have previously appeared in the following publications:

- Part of Chapter 3 appears in Asian et al. [2005b] and Adriani et al. [2007] (To appear)
- Part of Chapter 4 appears in Asian et al. [2004] and Adriani et al. [2007] (To appear)
- Part of Chapter 5 appears in Asian et al. [2005a]

Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

We use `the typewriter fonts` for user queries; *italics* for new terms or to emphasize some points; "double quotes" for Indonesian affixes or words or sentences; and ⟨angle brackets⟩ for the English translation.

All translations from Indonesian to English are done by the author and have been taken either from the author's own understanding of the languages or from the "Kamus Indonesia-Inggris" dictionary by Echols and Shadily [1998], unless otherwise stated.

When there is any comparison for performance shown in tables, the best results are indicated in bold font.

Contents

List of Figures

List of Tables

Abstract

The Web is a vast repository of data, and information on almost any subject can be found with the aid of search engines. Although the Web is international, the majority of research on finding of information has a focus on languages such as English, Chinese, and French. In this thesis, we investigate information retrieval techniques for Indonesian. Although Indonesia is the fourth most populous country in the world, little attention has been given to search of Indonesian documents.

Stemming is the process of reducing morphological variants of a word to a common stem form. Previous research has shown that stemming is language-dependent. Although several stemming algorithms have been proposed for Indonesian, there is no consensus on which gives better performance. We empirically explore these stemming algorithms, showing that even the best algorithm still has scope for at least an improvement of five percentage points. We propose novel extensions to this algorithm and develop a new Indonesian stemmer, and show that these can improve stemming correctness by up to three percentage points; our approach makes less than one error in thirty-eight words.

We propose a range of techniques to enhance the performance of Indonesian information retrieval. These techniques include: stopping; sub-word tokenisation; identification of proper nouns and not stemming proper nouns; and modifications to existing similarity functions. Our experiments show that many of these techniques can increase retrieval performance, with the highest increase achieved when grams of size five are used to tokenise words. We also present an effective method for identifying the language of a document; this allows various information retrieval techniques to be applied selectively depending on the language of target documents.

We also address the problem of automatic creation of parallel corpora — collections of documents that are the direct translations of each other — which are essential for cross-lingual information retrieval and other natural language processing tasks, including machine

translation. Well-curated parallel corpora are rare, and for many languages, such as Indonesian, do not exist at all. We describe algorithms that we have developed to automatically identify parallel documents for Indonesian and English. Unlike most current approaches, which consider only the context and structure of the documents, our approach is based on the document content itself. Even methods that make use the content of the documents make certain assumptions about the documents, for example that the sentences in the parallel documents are segmented at the same place, or that parallel documents share cognates. Our algorithms do not make any prior assumptions about the documents, and are based on the Needleman-Wunsch algorithm for global alignment of protein sequences. Our approach works well in identifying Indonesian-English parallel documents, especially when no translation is performed. It can increase the separation value, a measure to discriminate good matches of parallel documents from bad matches, by approximately ten percentage points. We also investigate the applicability of our identification algorithms for other languages that use the Latin alphabet. Our experiments show that, with minor modifications, our alignment methods are effective for English-French, English-German, and French-German alignment of parallel documents especially when the documents are not translated. Our technique can increase the separation value for the European corpus by up to twenty-eight percentage points compared to a baseline. Together, these results provide a substantial advance in understanding techniques that can be applied for effective Indonesian text retrieval.

Chapter 1

Introduction

The Web is a vast repository of data, and information on almost any subject can be found with the aid of search engines such as Google[1] and Yahoo.[2] In order to be able to return relevant answers to web users, provision of search involves a range of interrelated processes including collection of data through crawling, and parsing and processing the data to find items that are deemed likely to be relevant to queries by users. The study of these processes is known as Information Retrieval (IR).

In 1996, there were an estimated 40 million English-speaking Internet users but only 10 million non-English-speaking users. In 2005, of an estimated 1.12 billion Internet users, 820 million were not native English speakers.[3] Despite this growth, around two-thirds of accessible pages are in English.[4] Furthermore, the primary focus of IR research has been on monolingual English information retrieval, and cross-lingual information retrieval (CLIR) between English and other languages. Since 2003, a notable exception has been the Cross-Language Evaluation Forum (CLEF), when multilingual, bilingual and monolingual search tasks started to focus on European languages other than English [Braschler and Peters, 2004]. IR research on other languages has typically been driven by political or financial imperatives, such as for Chinese, Arabic, Japanese, and some European languages. This phenomenon is underlined by the increasing numbers of fora, workshops, corpora, and testbeds for these languages, among them CLEF and the Japanese National Institute of Informatics Test Collection for IR Systems (NTCIR).

[1] `http://www.google.com`

[2] `http://www.yahoo.com`

[3] `http://global-reach.biz/globstats/evol.html` accessed on 7th March 2007.

[4] `http://global-reach.biz/globstats/refs.php3`

Year	Subscribers	Users
1996	31 000	110 000
1997	75 000	384 000
1998	134 000	512 000
1999	256 000	1 000 000
2000	400 000	1 900 000
2001	581 000	4 200 000
2002	667 002	4 500 000
2003	865 706	8 080 534
2004	1 300 000	12 000 000

Table 1.1: The growth of Indonesian Internet subscribers and users [Hill and Sen, 2005]. The year 2004 numbers are estimates.

Indonesian has not been extensively investigated by the IR research community.[5] Indonesia is the fourth-most populous country in the world [Woodier, 2006, page 46] with over 240 million people[6] and its language is among the top ten most spoken when combined with the closely related Malay language [Quinn, 2001, page vii]. Indonesian is spoken in Indonesia, while Malay is spoken in Malaysia, Singapore, and Brunei Darussalam. Of the languages with a Latin character set, Indonesian is the fourth most widely spoken, after English, Spanish, and Portuguese.[7] The rapid growth of Internet users and subscribers in Indonesia is shown in Table 1.1. These factors, along with the fact that the author is a native Indonesian speaker, leads us to explore monolingual and cross-lingual retrieval for Indonesian text.

There are many aspects of monolingual text retrieval to be investigated. We investigate the parsing, indexing, and relevance assessment stages as we conjecture that these may need to be customised for Indonesian. These broad stages can be further divided into specific tasks such as stemming, stopping, tokenisation or dividing words into n-grams, measuring similarity functions, and parsing of proper nouns. Specifically, we consider four principal research questions that we now describe.

[5] There is a cross-lingual track for Indonesian and English in Cross-Language Evaluation Forum (CLEF) [Peters, 2006], however there is no track for Indonesian monolingual retrieval, and the Indonesian cross-lingual track does not have its own corpus.

[6] `https://www.cia.gov/cia/publications/factbook/print/id.html` accessed on 2nd March 2007.

[7] `http://www.linguasphere.org/language.html`

Which Indonesian stemming algorithm is the best?

Stemming — a technique that attempts to find the common root of words by applying morphological rules — has significant potential for impact on the ability to accurately identify relevant documents. For example, stemming could be used to reduce the words "retrieval", "retrieving", "retrieved", and "retrieves" to a common root, say "retriev". Stemming also has applications in machine translation [Bakar and Rahman, 2003], document summarisation [Orăsan et al., 2004], and text classification [Gaustad and Bouma, 2002]. Due to differences between languages, stemming is language dependent — techniques that work well in one language are unlikely to work in another. Stemming of Indonesian words is relatively challenging because Indonesian has a range of affix forms, including prefixes, infixes, suffixes, repeated forms, and combinations of these.

Although there is some prior work on stemming for Indonesian [Arifin and Setiono, 2002; Nazief and Adriani, 1996; Vega, 2001], there is no common testbed or ground truth to test this, and no consensus about which techniques perform best. We therefore build a testbed for Indonesian stemming and use it to compare the performance of different existing algorithms. We investigate improvements to the best of these algorithms by adding new stemming rules and modifying the rule order. We call this improved algorithm the CS stemmer, and show that it provides better performance than any of the existing algorithms for Indonesian, reducing the error rate from one in twenty-one words to one in thirty-eight words. We have also experimented with different dictionaries and concluded that the best dictionary is one that contains only root words.

Which information retrieval techniques increase retrieval effectiveness for Indonesian?

We are interested not only in finding the most accurate stemmer, but also in discovering whether stemming can help to increase retrieval effectiveness. However, there is again no publicly available testbed for Indonesian text retrieval. We build our own testbed for Indonesian text retrieval from newswire dispatches downloaded from an Indonesian media website.[8] We skim the corpus text to gauge topic availability, create the topics, and assess the relevance of each document to each query. We develop this testbed following the standardised Text REtrieval Conference (TREC) format [Voorhees and Harman, 1997].

[8] `http://www.kompas.com`

We hypothesise that, if existing IR techniques work well for English, they may work well for Indonesian as well, since the two languages share a similar character set. We conduct empirical studies to investigate which techniques work well, and whether any adjustment is required for Indonesian. Our results show that using only the title of the queries produces the highest mean average precision, which in fact reflects a typical web search situation. We also discover that the parameter settings for the Okapi BM25 similarity measure that work best for English do not necessarily work well for Indonesian. Stemming and stopping, which are also language-dependent, can increase retrieval effectiveness. General IR techniques applicable to both Indonesian and English include tokenisation. Tokenisation of words into n-grams is more effective than stemming in increasing retrieval effectiveness, which is also the case for English.

Most queries include proper nouns [Thompson and Dozier, 1997]. Proper nouns are considered to be root words, and should not be stemmed. We hypothesise that stemming proper nouns leads to decrease in precision, and we develop several methods to identify proper nouns in Indonesian. We show that, by identifying and not stemming proper nouns, retrieval effectiveness can be improved.

Before applying IR techniques specific to Indonesian, we must ascertain whether a document is in that language. We investigate techniques for language identification, and show that we can accurately identify whether a document is in Indonesian by counting the number of words in the document that is in Indonesian or in another language.

The key contribution of our analysis is that IR techniques that are known to work well for English can also work well for Indonesian, although they may need to be customised.

How can a parallel corpus for Indonesian and English be built automatically based on the content?

Another underdeveloped area of Indonesian IR research is cross-lingual retrieval between Indonesian and another language. To develop effective cross-lingual techniques, we must have a training corpus of parallel documents in Indonesian and the second language. The only prior work on Indonesian CLIR is that of Adriani and Wahyu [2005]. The retrieval task explored by these researchers does not in fact use a dual-language corpus. They only have an English corpus and English queries. They translate the English queries into Indonesian, and then translate the queries back into English to retrieve the English documents.

To allow more thorough research in CLIR between Indonesian and another language, we need document collections and queries in Indonesian and another language. One way of doing this is by manual translation of queries and documents. This method is not desirable, as it is time-consuming and costly. We need a method that can automate the construction of a bilingual document collection.

One possible method to automatically build a parallel document collection — parallel documents are documents that are translations of each other — is to use automatic machine translation. This method works well for short queries or queries within a specific domain, but can lead to ambiguous translation for longer documents [Fluhr, 1995]. Another method of building a bilingual corpus is to identify a set of documents that are translations of each other.

Existing methods of identifying parallel documents rely on the names, locations, or external structures of the documents [Chen et al., 2004; Chen and Nie, 2000; Kraaij et al., 2003; Nie et al., 1999; Resnik and Smith, 2003; Yang and Li, 2004] that may not be present in all documents. Even methods that use the content of documents make certain assumptions (for example, the assumption that parallel documents share cognates [Chen and Nie, 2000], or the assumption that sentences are segmented at the same place [Pike and Melamed, 2004]). We require a method that can automatically find parallel documents without making any assumptions about the documents. For this, we propose a technique based on the basic principle of the Needleman and Wunsch [1970] global alignment method, and treat words in documents as sequences to be aligned.

We empirically show that our alignment methods work well in separating Indonesian and English parallel documents from non-parallel documents, especially when the parallel documents are not translated. Our alignment methods also benefit from stopping with and without translation; our results show a significant improvement of around ten percentage points over a symmetric cosine baseline.

Are global alignment methods applicable for other languages that share the Latin alphabet?

To verify whether our methods work for other languages sharing the same character set, we conduct experiments on different language pairs: English-French, English-German, and French-German. These languages may present more challenges, as French and German words have accents while English words — with the exception of foreign words — do not. We show

that our alignment methods can separate parallel documents in these languages better than a symmetric cosine baseline when the documents are not translated, giving an improvement of around twenty percentage points with stopping, although the optimal parameter settings differ from those used for the Indonesian-English collection.

1.1 Thesis structure

The remainder of this thesis is organised as follows.

In Chapter 2, we summarise the history and features of the Indonesian language, and describe Indonesian morphology. We also provide an overview of information retrieval (IR) and cross-lingual information retrieval (CLIR), and outline the need for parallel corpora.

Stemming for Indonesian is discussed in Chapter 3. Using manual stemming as the baseline, we compare existing Indonesian stemming algorithms to determine which can stem the most accurately. We then describe the CS stemming algorithm, which improves upon the highly effective stemming algorithm of Nazief and Adriani [1996].

In Chapter 4, we introduce our Indonesian testbed for ad hoc information retrieval. Using this testbed, we explore the effectiveness of applying different text retrieval techniques, including stemming, stopping, tokenisation, and changing similarity measure parameter settings.

In Chapter 5, we describe algorithms and experiments for automatic identification of Indonesian and English parallel documents. Our identification algorithms are based on the Needleman and Wunsch [1970] global alignment method, whereby we align windows of words from the two documents to be evaluated.

In Chapter 6, we apply these alignment techniques to identify parallel documents for other language pairs: English-French, English-German, and French-German.

Finally, in Chapter 7, we conclude the thesis and outline directions for possible future research work.

Chapter 2

Background

Information Retrieval (IR) is the field of study about how to efficiently store and retrieve data [Witten et al., 1999, pages 6–8], and how to provide answers that satisfy a user's information needs [Grossman and Frieder, 2004, page 1]. A user may enter keywords such as `Melbourne weather report` or `universal declaration of human rights` to describe their information need. A good IR system ranks most relevant documents about Melbourne weather reports or the United Nations declaration of human rights at the top. Documents matching some of the query words, such as documents describing Melbourne or documents discussing other topics about humans, should not be deemed relevant. Despite the diversity of data types, locations, and languages a good IR system is supposed to manage the data properly.

Since this thesis is concerned with Indonesian information retrieval, the Indonesian language and its similarities and differences with English are introduced in Section 2.1. As Indonesian morphology is unique, it is described briefly in Section 2.2. Section 2.3 describes information retrieval in general, while cross-lingual information retrieval, where user queries and documents are in different languages is covered in Section 2.4.

2.1 Bahasa Indonesia

Bahasa Indonesia is the official language of Indonesia. Bahasa literally means "language" in Indonesian. In this thesis, we refer to "Bahasa Indonesia" as *Indonesian*, or *BI*.

Quinn [2001, page vii] 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. Below are the chronological orders of development of BI from Malay [Quinn, 2001, pages vii–xii; Robson, 2004, pages 5–7].

From the start of the recorded history, Malay has been the lingua franca of people living on both Malay Peninsula and the island of Sumatra. In the 17th century, when the Dutch started to colonise most islands in modern-day Indonesia (at that time called the Netherlands East Indies) they used Malay as the administrative language. Unlike other European colonisers, the Dutch did not promote their language to its colony, therefore only a few highly educated locals knew Dutch. Malay was favored over other local dialects such as Javanese, as it was simpler and widely spoken.

In the early years, the language used for Indonesian nationalist movement was not Malay. This changed from 1928 when the Congress of Young People made the Young People's Vow to adopt Malay as the national language of Indonesia. The congress also formally referred to Malay as *Bahasa Indonesia* to encourage patriotism.

The victory of the complete independence movement in 1959 strengthened *Bahasa Indonesia* position as the national language. In 1988, the *Pusat Bahasa* ⟨the Centre for Language Development⟩, an organisation promoting Indonesian, published a standard for Indonesian grammar called *Tata Bahasa Baku Bahasa Indonesia* ⟨A Standard Grammar of Indonesian⟩ and a standard dictionary named *Kamus Besar Bahasa Indonesia* ⟨A Comprehensive Dictionary of Indonesian⟩.

2.1.1 Character sets

Unlike most Asian languages such as Chinese, Japanese, Korean, and Vietnamese, which use special character sets, Indonesian uses the Latin alphabet [Robson, 2004, pages 10–12; Wilujeng, 2002, page 5] as used in many languages such as English, Italian, Spanish, Tagalog, and Swahili.

According to Quinn [2001, page xii], during the spread of Islam between the 14th and the 15th century, Javanese and Arabic scripts were used to write Malay. From the second half of the 19th century, due to the influence of European missionaries,[1] Latin script came into widespread use. By the early 20th century, all Malay words were written in Latin script.

There are some variations of Indonesian spellings, which can still be seen in proper nouns, as a consequence of two language reformation initiatives as in 1947 and in 1972 [Quinn, 2001, page 726]. Initially, Malay followed the Dutch spelling system. In 1947, a new spelling method called *Soewandi* was invented. This system changed "oe" to "u". As the

[1] `http://www.alhewar.com/habeeb\_salloum\_arabic\_language.htm`

result, "Soekarno" and "Soeharto", the names of former Indonesian presidents, are sometimes spelled as "Sukarno" and "Suharto" respectively. In 1972, there was a major spelling reformation called *Ejaan Yang Disempurnakan* ⟨the Updated and Improved Spelling⟩ that united spellings of *Indonesian* and *Malaysian*. The reform changed "tj" to "c", "j" to "y", "dj" to "j", and "nj" to "ny". "Jogjakarta" or "Yogyakarta", and "Djakarta' or "Jakarta", are examples of this variation. This reform also implemented separation of prepositions "di" ⟨at, in⟩ and "ke" ⟨to⟩ from the nouns following the preposition, and replacement of the number 2 behind a word to indicate a repeated word with a hyphen (-) followed by the same word again.[2] Prior to 1972, the sentence "Tadi pagi Djoko melihat anak2 Susan menjanji disekolah" ⟨This morning Joko saw Susan's children sing at school⟩ is correct. After 1972, it is written as "Tadi pagi Jack melihat anak-anak Susan menyanyi di sekolah".

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.

Some letters in Indonesian, such as "q", "v", "x", and "z" occur in loan words, which are words borrowed from other languages [Robson, 2004, page 12], but are not otherwise used in Indonesian. Wilujeng [2002, page 6] adds that the letters "q" and "x" are also used for proper nouns and scientific names.

2.1.2 Capitalisation

Since Indonesian uses the Roman alphabet, it has both uppercase and lowercase letters. The capitalisation rules for Indonesian are quite similar to those of English. These rules are important when we discuss how to find proper nouns; they appear in Appendix A. We discuss proper nouns in more detail in Chapter 4.

2.1.3 Vocabulary

In addition to Malay, BI has loan words from Sanskrit, Arabic, Dutch, English, Portuguese, and local dialects [Quinn, 2001, page vii; Wilujeng, 2002, page 28; Woods et al., 1995, page 5]. These foreign words could be assimilated into Indonesian with their original spellings intact; transliterated; or italicised [Dwipayana, 2001, pages 155–158].

Examples of words that have been assimilated into Indonesian are:

- "kungfu" ⟨kung fu⟩, "lihai" ⟨proficient⟩, and "hoki" ⟨fortune⟩ from Chinese;

[2] This repeated word is unique for Indonesian and is discussed in Section 2.1.5.

- "rekening" ⟨account⟩, "tante" ⟨aunt⟩, and "wortel" ⟨carrot⟩ from Dutch;
- "sastra" ⟨literature⟩, "karma" ⟨karma⟩, and "bahasa" ⟨language⟩ from Sanskrit;
- "halal" ⟨halal⟩, "kitab" ⟨book⟩, and "maaf" ⟨sorry⟩ from Arabic.
- "aku" ⟨I⟩, "cilik" ⟨small⟩, and "pangan" ⟨food⟩ from the local dialect Javanese; "saya" ⟨I⟩ and "nyeri" ⟨pain⟩ from Sundanese; "anda" ⟨you⟩ from Nias; and "lamban" ⟨slow⟩ from Minangkabau.

Examples of transliterated words are "teknologi" ⟨technology⟩, "kompas" ⟨compass⟩, and "narkotika" ⟨narcotics⟩.

Examples of italicised words are "*allegretto*", "*a la carte*", "*status quo*", "*cum laude*", "*curriculum vitae*", and "*esprit de corps*". They are usually foreign words and phrases that are widely used but not assimilated.

Foreign languages have also influenced Indonesian prefixes and suffixes. Prefixes are mostly influenced by Indo-Europe languages [Dwipayana, 2001, pages 179–180; Widyamartaya, 2003, pages 79–81] while suffixes can also be influenced by either Dutch [Wilujeng, 2002, pages 35–37] or other Indo-Europe languages [Widyamartaya, 2003, page 81]. These prefixes and suffixes can either be retained unchanged or transliterated into Indonesian. Examples of prefixes that have been adapted into Indonesian are "mono-" ⟨mono-⟩, "ekstra-" ⟨extra-⟩, "hiper" ⟨hyper⟩, "sin-" ⟨syn-⟩, "ultra-" ⟨ultra-⟩. Examples of suffixes are "-si" ⟨-sion and -tion⟩, "-asme" ⟨-asm⟩, "-bel" ⟨-ble⟩, "-ikel" ⟨-icle⟩, and "-or" ⟨-or⟩.

Dwipayana [2001, page 174] recommends that transliterated names should be written according to the ISO standards, common English spelling, or Chinese pinyin. Different languages have different ISO standards for transliteration, for example, Arabic uses ISO/R 233, Greek uses ISO/R 315, and Russian uses ISO/R 9 [Moeliono and Dardjowidjojo, 1988, page 441]. "John Howard", "Jacques Chirac", "Wolfgang Amadeus Mozart", and "Slobodan Milosevic" are examples of words of which the spellings are retained. "Sokrates" ⟨Socrates⟩, "Yesus" ⟨Jesus⟩, and "Hu Jingtao" are examples of names whose spellings have been adapted.

Although the original spellings of most place names are retained in Indonesian, some names are also transliterated. Examples include "Jerman" ⟨Germany, German⟩, "Kroatia" ⟨Croatia⟩, "Moskow" ⟨Moscow⟩, and "Skotlandia" ⟨Scotland⟩.

2.1.4 Metric and Numbering Systems

Dwipayana [2001, pages 170–172] explains that Indonesian uses the *Système International d'Unités* or the International System of Units (usually abbreviated as SI) for measurement, where "meter" ⟨metre⟩ is used for length, "kilogram" for weight, "detik" ⟨second⟩ for time, and "ampere" for electric current. Prefixes for metrics also use SI, for example, "tera-" for 10^{12}, "giga-" for 10^{9}, "mega-" for 10^{6}, and "kilo-" for 10^{3}.

BI uses Arabic and Roman numbering systems [Wilujeng, 2002, page 24]. For large numbers, Indonesian follows the American English convention [Moeliono and Dardjowidjojo, 1988, page 194]. Indonesian uses "biliun" ⟨billion⟩ for 10^{9}, "triliun" (trillion) for 10^{12}, "kuadriliun" ⟨quadrillion⟩ for 10^{15}, "kuantiliun" ⟨quintillion⟩ for 10^{18}, and "sekstiliun" ⟨sextillion⟩ for 10^{21}.

In most Indonesian text, full stops are used after every 3 digits from the right-hand end [Moeliono and Dardjowidjojo, 1988, page 195] and a comma (",") is used to indicate the decimal point [Moeliono and Dardjowidjojo, 1988, page 199]. For example, in Indonesian, π is usually written as 3,14159 as opposed to the English way of 3.14159. One million is usually written as 1.000.000 in Indonesian. For stock market reports, Indonesian usually follows the English convention.

2.1.5 Grammar

Indonesian has a rich grammar. In this section, we focus on aspects that could have an impact on information retrieval. Some other interesting aspects can be found in Appendix B.

Plural

In Indonesian, plurality can be shown by repeating words [Dwipayana, 2001, page 23; White, 1990, page 36; Woods et al., 1995, pages 21–22]. For example, the plural for "buku" ⟨book⟩ is "buku-buku" ⟨books⟩, for "rumah" ⟨house⟩ is "rumah-rumah" ⟨houses⟩, and for "botol" ⟨bottle⟩ is "botol-botol" ⟨bottles⟩.

If there are words at the front of the object to indicate plurality, repeated words need not be used [Widyamartaya, 2003, pages 45–46; Woods et al., 1995, page 22]. For example, words such as "sedikit" ⟨a few⟩, "seribu" ⟨a thousand⟩, "empat" ⟨four⟩, "banyak" ⟨a lot⟩, "sejumlah" ⟨a number of⟩ occur before an object such as "buku", there is no need to repeat the word "buku". There is no need for an agreement between a subject and a verb in term of plurality in Indonesian [Widyamartaya, 2003, page 45]. In the sentences "Seorang murid datang ke kantor saya" ⟨A student comes to my office⟩ and "Banyak murid datang ke kantor

saya" ⟨A lot of students come to my office⟩, the verb "datang" ⟨come⟩ is not affected by the number of students.

Articles

Indonesian has no articles [Widyamartaya, 2003, pages 46–48]. Instead, words such as "sebuah" ⟨one, usually for fruits⟩, "satu" ⟨one⟩, "seekor" ⟨one, usually for animals⟩, are used to indicate cardinality.

Sometime the word "yang" is used as "the" [Woods et al., 1995, page 13]. For example, the noun phrase "rumah yang tua" ⟨the old house⟩ derives from "rumah" ⟨house⟩ and "old" ⟨tua⟩, "mobil yang biru" ⟨the blue car⟩ derives from "mobil" ⟨car⟩ and "biru" ⟨blue⟩.

Word Order

In Indonesian, adjectives appear after the nouns they describe, in contrast to English [White, 1990, page 37; Woods et al., 1995, page 13]. For example, "teh hijau" ⟨green tea⟩ consists of "teh" ⟨tea⟩ and "hijau" ⟨green⟩, "pohon besar" ⟨big tree⟩ consists of "pohon" ⟨tree⟩ and "besar" ⟨big⟩, and "kolam renang" ⟨swimming pool⟩ consists of "kolam" ⟨pool⟩ and "renang" ⟨to swim⟩.

Some Indonesian possessive pronouns also appear after the nouns [Woods et al., 1995, pages 19–20]. "Buku saya" ⟨my book⟩, "buku Anda" ⟨your book⟩, and "buku mereka" ⟨their book⟩ are examples of this. Some possessive pronouns appear as suffixes, and are discussed in Section 2.2.

Adding "yang" ⟨the, that⟩ between a noun and an adjective can change the meaning of the noun phrase in an unpredictable manner [White, 1990, page 37]. For example, "orang tua" ⟨parents⟩ is different from "orang yang tua" ⟨old people⟩, and "kamar kecil" ⟨toilet⟩ is different from "kamar yang kecil" ⟨a small room⟩. Meanwhile, the noun phrase "merah muda" ⟨pink⟩ could not be written as "merah yang muda" ("merah" means "red" and "muda" means "young").

Repeated Words

Words may be repeated either in similar or in slightly altered format with a hyphen (-) used to separate the two occurrences. There are several formats of repeated words.

One format discussed in Section 2.1.5 is a repeated word that is repeated fully to indicate plurality. For example, the words "pohon-pohon" ⟨trees⟩, "rumah-rumah" ⟨houses⟩,

and “buku-buku” ⟨books⟩ stem from “pohon” ⟨tree⟩, “rumah” ⟨house⟩, and “buku” ⟨book⟩ respectively.

Repeated words could have one or more of their characters modified [Dwipayana, 2001, pages 17–18; Wilujeng, 2002, page 96], for example, “warna-warni” ⟨colourful⟩ is derived from “warna” ⟨colour⟩, “sayur-mayur” ⟨different kind of vegetables⟩ is derived from “sayur” ⟨vegetables⟩, and “compang-camping” ⟨in tatters, in rags⟩ is derived from “camping” ⟨in tatters, in rags⟩.

There are also repeated words with prefixes or suffixes where the prefix or suffix is not repeated [Dwipayana, 2001, pages 15–16; Wilujeng, 2002, page 96]. “Kehijau-hijauan” ⟨greenish⟩ is derived from “hijau” ⟨green⟩ with the prefix “ke-” and the suffix “-an”; “menari-nari” ⟨dance around⟩ is derived from “tari” ⟨dance⟩ with the prefix “me-”;[3] “dua-duanya” ⟨two of them⟩ is derived from “dua” ⟨two⟩ with the suffix “-an”.

Wilujeng [2002, page 96] adds that there are artificial repeated words, which means they look like repeated forms but in fact the meaning is completely different from the meaning of the original single word. Examples of these words are “pura-pura” ⟨pretend⟩ and “pura” ⟨Hindu temple⟩; “mata-mata” ⟨spy⟩ and “mata” ⟨eye⟩; and “laba-laba” ⟨spider⟩ and “laba” ⟨profit⟩.

There is a variant of repeated words where the second word is different from the first and where only the first syllable is repeated without any hyphen [Dwipayana, 2001, page 18]. The former type of repeated words are used in a literary sense to emphasise the first word, for example, “gelap-gulita” ⟨pitch black⟩ is derived from “gelap” ⟨dark⟩ and “gulita” ⟨dark, completely without light⟩, and “sunyi-senyap” ⟨dead silent⟩ from “sunyi” ⟨lonely, quiet, deserted⟩ and “senyap” ⟨silent, quiet⟩. Examples of the latter type are “lelaki” ⟨male⟩ originating from “laki” ⟨male⟩, “tetirah” ⟨to go somewhere for a cure⟩ from “tirah” ⟨to go somewhere for a cure⟩, “sesama” ⟨fellow, peer⟩ from “sama”[4] ⟨same, equal⟩, and “jejaka” ⟨young man, bachelor⟩ from “jaka” ⟨bachelor⟩.

Dwipayana [2001, pages 20–22] notes that besides showing plurality, repeated words can also indicate reciprocal action (“tarik-menarik” ⟨push and pull⟩ from “tarik” ⟨pull⟩), repeated actions (“menarik-narik” ⟨pull repeatedly⟩ from “tarik” ⟨pull⟩), and intensified adjectives (“besar-besar” ⟨very big⟩ from “besar” ⟨big⟩).

[3] The reason for changing “tari” to “nari” after addition of the prefix “me-” is discussed in Section 2.2.

[4] There is also a repeated word “sama-sama” that means together.

Compound Words

Like English, Indonesian has compound words — words that consist of two words or more and have a new meaning that is different from each of its components [Dwipayana, 2001, page 26; Wilujeng, 2002, page 97]. It is not possible to insert a new word between these compound words or to reverse the order of the words [Moeliono and Dardjowidjojo, 1988, page 122; Wilujeng, 2002, pages 97–98]. Words such as "panjang tangan" (⟨like to steal⟩ consisting of "panjang" ⟨long⟩ and "tangan" ⟨hand⟩), "darah daging" (⟨blood relation⟩ consists of "darah" ⟨blood⟩ and "daging" ⟨flesh⟩), "ikut serta" (⟨participate⟩ consists of "ikut" ⟨to follow⟩ and "serta" ⟨along with, as well as⟩), "anak mas" (⟨favourite child⟩ consists of "anak" ⟨child⟩ and "mas" ⟨gold⟩) are compound words. If a new word is inserted between these words, there may be no meaning or the meaning becomes the literal meaning of the elements ("darah dan daging" ⟨blood and flesh⟩). Furthermore, "panjang tangan" cannot be reversed to "tangan panjang" and "anak mas" to "mas anak".

In rare instances, a compound word can consist of a word or words that already have affixes [Moeliono and Dardjowidjojo, 1988, pages 124–126]. Examples of this are "hilang ingatan" (⟨crazy⟩ consists of "hilang" ⟨lose⟩ and "ingatan" ⟨memory⟩ that stems from "ingat" ⟨remember⟩); "haus kekuasaan" (⟨hungry for power, ambitious⟩ consists of "haus" ⟨thirsty⟩ and "kekuasaan" ⟨power, normally politically and socially⟩ that stems from "kuasa" ⟨authority, power⟩; and "akte kelahiran" (⟨birth certificate⟩ consists of "akte" ⟨official document⟩ and "kelahiran" ⟨birth⟩ that stems from "lahir" ⟨be born⟩).

According to *Tata Bahasa Baku Bahasa Indonesia* ⟨A Standard Grammar of Indonesian⟩ created by Pusat Bahasa [Moeliono and Dardjowidjojo, 1988, page 126], a compound word is written as one word if and only if it has both a prefix and a suffix. Otherwise it needs to be written separately. "Keikutsertaan" ⟨participation⟩ from '*ke*-ikut serta-*an*" and "menganakmaskan" ⟨to play favourites⟩ from "*me*-anak mas-*kan*"[5] are written as one word, whereas "berikut serta" ⟨to participate⟩ from "*ber*-ikut serta" and "anak masmu" ⟨your favourite child⟩ from "anak mas-*mu*" are written separately. In practice, some compound words are written as one word and, as they are often used as one word, they are accepted as the "right" format. For example "tanggung jawab" (⟨responsible, responsibility⟩ consisting of "tanggung" ⟨guarantee⟩ and "jawab" ⟨answer⟩) is often written as "tanggungjawab" and "beri tahu" (⟨inform⟩ consists of "beri" ⟨give⟩ and "tahu" ⟨know⟩) is often written as "beritahu".

[5]Transformation from "me-" to "meng-" is discussed in Section 2.2.3

Usually, when repeating these compound words, all components have to be repeated [Wilujeng, 2002, page 97], for example, “akte kelahiran-akte kelahiran” ⟨birth certificates⟩, “tanggung jawab-tanggung jawab” ⟨responsibilities⟩. There are some exceptional cases where only the first component is repeated if certain criteria are met [Moeliono and Dardjowidjojo, 1988, page 127]. The first criterion is that the second component of the repeated word explains the first component. The second criterion is that the first component has to be a verb that can be done repeatedly. Compounds such as “hilang ingatan” and “haus kekuasaan” meet the first criterion but not the second one — the verbs “hilang” ⟨lose⟩ or “haus” ⟨thirsty⟩ cannot be performed repeatedly. Examples of compounds that satisfy both criteria are “pindah-pindah tangan” (⟨change hands repeatedly⟩ originating from “pindah tangan” ⟨change hands⟩ that consists of “pindah” ⟨move⟩ and “tangan” ⟨hand⟩), “goyang-goyang kaki” (⟨repeated actions to relax while others solve his/her problems⟩ from “goyang kaki” ⟨relax while others solve his/her problems⟩ that consists of “goyang” ⟨shake⟩ and “kaki” ⟨foot⟩).

These different combinations of compounds words with the repeated forms and with the addition of prefixes or suffixes are discussed in Section 2.2.

2.2 Indonesian Morphology

In this section, we discuss Indonesian morphology which directly affects stemming (and hence may indirectly affect retrieval performance).

Indonesian is an agglutinative language that allows new words to be formed by adding prefixes and suffixes to a word [Quinn, 2001, page vii]. New words can also be formed by repeating the word as described in Section 2.1.5, and by inserting infixes into a word. For example, “kekemilauan” ⟨shininess⟩ is derived from “kilau” ⟨to shine⟩ with the addition of a prefix “ke-”, an infix “-em-”, and a suffix “-an”

Stemming is a method to reduce words to their root forms to get the stem [Paice, 1994]. For example, the words “retrieval”, “retrieves”, and “retrieving” can all be reduced to the root form “retriev”. Paice [1994] states that words are usually stemmed because forms that are syntactically different are assumed to have the same meaning. People entering `retrieval` as a query might also be interested in documents containing `retrieve` and `retrieving`.

Stemming requires good understanding of the language in question [Popovič and Willett, 1992]. English has prefixes such as “hyper-” as in “hypertension” and “hyperactive”; “anti-” as in “antisocial”; and “ultra-” as in “ultraviolet”. These prefixes create new meanings that

are different from the original meaning, therefore they are not considered in IR stemming. In English IR, stemming usually removes only suffixes. There are well-known stemming algorithms by Lovins [1968], Porter [1980], and Frakes [1992].

As Indonesian is an agglutinative language, stemming is relatively challenging. There are variations of affixes including prefixes, suffixes, infixes, and confixes. Indonesian also has repeated words, combinations of affixes, and combinations of affixes with repeated words. Moreover, Indonesian also has compound words that are written together when attached to a prefix and a suffix, as discussed in Section 2.1.5. Examples of Indonesian words and their stems are "pemerintah" ⟨the government, a government⟩, "pemerintahan" ⟨government, government administration⟩, "diperintah" ⟨be ruled, be ordered⟩, and "perintahnya" ⟨his/her order⟩, which can all be stemmed to "perintah" ⟨command, order⟩; and "buku-buku" ⟨books⟩, "bukumu" ⟨your book⟩, and "pembukuan" ⟨accounting⟩, which can all be stemmed to "buku" ⟨book⟩. Adding affixes can change the meaning of Indonesian words greatly. These kinds of affixes are called derivative affixes [Moeliono and Dardjowidjojo, 1988, pages 80–81].

Some Indonesian words require affixes to have meaning [Sahanaya and Tan, 2001, page xii]; the examples given are the root word "kemis" which requires the prefix "me-" ("mengemis" ⟨to beg⟩) or the prefix "pe-" ("pengemis" ⟨beggar⟩); on its own, "kemis" has no meaning.

In the next sections, we discuss different Indonesian affixes and how they are added to the root words.[6]

2.2.1 Suffixes

Suffixes are discussed first as, unlike prefixes and infixes, they do not change the form of the root word. According to the *Tata Bahasa Baku Bahasa Indonesia* (*TBBBI*) ⟨A Standard Grammar of Indonesian⟩ by Moeliono and Dardjowidjojo [1988, pages 85, 92–93], there are only three suffixes in Indonesia, namely "-i", "-kan", and "-an". There are particles and possessive suffixes attached at the end of a root word that are not considered as suffixes grammatically, but they can be considered as suffixes in the context of information retrieval. In the next sections, we discuss particles, possessive suffixes, and derivative suffixes.

[6]The meanings of affixes indicated in this thesis are by no means exhaustive, but are used to illustrate how affixes affect root words.

Particles

According to the *TBBBI* [Moeliono and Dardjowidjojo, 1988, pages 247–249], the particles "-lah", "-kah", and "-tah" do not modify the root words they are attached to. For example, the words "duduklah" ⟨please sit down⟩ and "diakah?" ⟨is it you?⟩, which stem from "duduk" ⟨sit⟩ and "dia" ⟨you⟩ respectively, do not change after being added the particles. The particle "-tah" is used for rhetorical questions and is now obsolete.

Moeliono and Dardjowidjojo [1988, page 248] add that the particle "-pun" can only be used in a declarative sentence. This particle emphasises the noun or the noun phrase it follows, and it should be written separately except when used as conjunction, as in "walaupun" ⟨although⟩ and "apapun" ⟨no matter what⟩. Example sentences are "*Susi pun* setuju" ⟨*Susi* agrees⟩ and "*Para aktivis pun* akhirnya dibubarkan polisi" ⟨*The activists* are finally dispersed by the police⟩. We have observed that in practice, the particle "-pun" is often attached to the word it follows, and this is accepted as the "right" format.

Possessive suffixes

There are three possessive suffixes in Indonesian, namely "-ku", "-mu", and "-nya", indicating possession by first, second, and third person respectively [Moeliono and Dardjowidjojo, 1988, pages 172–178]. Examples are "bukuku" ⟨my book⟩, "bukumu" ⟨your book⟩, and "bukunya" ⟨his/her book⟩. The suffix "-nya" can be used for the possessive of the third person plural ⟨theirs⟩ as well [White, 1990, page 11].

Besides indicating third person possessive, suffix "-nya" can turn an adjective into a noun [Wilujeng, 2002, page 80; White, 1990, page 106]. For example: "dalamnya" ⟨depth⟩ stems from "dalam" ⟨deep⟩ and "tingginya" ⟨height⟩ from "tinggi" ⟨high⟩. The suffix "-nya" also refers to a particular object depending on the context [Wilujeng, 2002, page 80]. Example sentences are "Itu bukunya yang saya beli" ⟨It is *that book* that I bought⟩ and "Apa judulnya?" ⟨What is *the title* [of something that the speaker and the audience are aware of]?⟩.

Derivative Suffixes

As with particles and possessive suffixes, the derivative suffixes "-i", "-kan", and "-an" do not modify the root words they are attached to [Moeliono and Dardjowidjojo, 1988, pages 92–93]. However, they change the meaning of the root word.

The suffix "-i" cannot be added to words ended with the letter "-i" [Moeliono and Dardjowidjojo, 1988, page 93]. Therefore, it is not acceptable to add the suffix "-i" to words like "lari" ⟨to run⟩, "mandi" ⟨to shower⟩, and "erosi" ⟨erosion⟩. Wilujeng [2002, pages 77-78] states that the suffix "-i" usually creates a verb from the root word and is usually combined with the prefix "me-" to create an active transitive verb or the prefix "di-" to create a passive transitive verb. Examples are "memanasi" ⟨to heat up⟩ stemming from "panas" ⟨hot⟩, and "ditandai" ⟨to be marked⟩ stemming from "tanda" ⟨mark⟩.

The suffix "-an" usually creates a noun [Wilujeng, 2002, page 79]. For example, the nouns "makanan" ⟨food⟩ and "bacaan" ⟨reading material⟩ are derived from the root words "makan" ⟨to eat⟩ and "baca" ⟨to read⟩ respectively.

Lastly, Moeliono and Dardjowidjojo [1988, pages 108–111] add that the suffix "-kan" creates transitive verbs. This suffix is often combined with the prefix "me-" and "di-" to create an active or a passive verb respectively. For example, the word "memasakkan" ⟨to cook for⟩ stems from "masak" ⟨to cook⟩ and the word "dibersihkan" ⟨to be cleaned⟩ stems from "bersih" ⟨clean⟩.

Other Derivative Suffixes

There are other suffixes in Indonesian adopted from foreign languages and not considered as native Indonesian suffixes. Examples include "-wan" ⟨male form⟩, "-wati" ⟨female form⟩, "-is" ⟨-ist⟩, and "-isme" ⟨-ism⟩ [Wilujeng, 2002, pages 80–82]. These suffixes usually do not change the root word. For example, the words "olahragawan" ⟨male athlete⟩ and "olahragawati" ⟨female athlete⟩ stem from "olahraga" ⟨exercise⟩; the word "sosialis" ⟨socialist⟩ stems from "sosial" ⟨social⟩; and the word "patriotisme" ⟨patriotism⟩ stems from "patriot" ⟨patriot⟩. There are some rare cases when the root words change after being attached to these suffixes. For example, adding the suffix "-wan" to the word "sejarah" ⟨history⟩ results in the word "sejarawan" ⟨historian⟩.

2.2.2 Infixes

An infix is an affix inserted within a word [Moeliono and Dardjowidjojo, 1988, page 163]. Indonesian has three affixes: "-el-", "-em-", and "-er-". These infixes are not common in modern Indonesian usage.

These infixes are inserted after the first letter of a root word [Wilujeng, 2002, pages 75–76]. For example:

"tunjuk" ⟨to point⟩ + "-el-" → "telunjuk" ⟨pointer⟩; "kilau" ⟨shine⟩ + "-em-" → "kemilau" ⟨shiny⟩; and "gigi" ⟨tooth⟩ + "-er-" → "gerigi" ⟨serration⟩.

2.2.3 Prefixes

Indonesian prefixes create derivative words from the root words [Moeliono and Dardjowidjojo, 1988, pages 78–81]. These prefixes are complex because some prefixes can vary depending on the first letter of the root word, and the first letter of the root word may also be removed or changed depending on the prefix it is attached to. This removal or modification of the first letter of the root word is called recoding.

Indonesian prefixes are "se-", "ke-", "di-", "ter-", "ber-", "per-", "pe-", and "me-" [Wilujeng, 2002, pages 52–62]. "Ku-" and "kau-" are also considered as prefixes [Moeliono and Dardjowidjojo, 1988, pages 94–97] although they are less formal and not frequently used.

Prefixes "se-", "ke-", "di-", "ter-", "ber-", and "per-"

The prefix "se-", "ke-", and "di-" do not vary according to the root word and neither do they change the root word they are attached to [Wilujeng, 2002, pages 52–56]. Examples include "se-" + "cangkir" ⟨cup⟩ → "secangkir" ⟨one cup⟩; "se-" + "cerdik" ⟨smart⟩ → "secerdik" ⟨as smart as⟩; "ke-" + "dua" ⟨two⟩ → "kedua" ⟨the second⟩; and "di-" + "makan" ⟨eat⟩ → "dimakan" ⟨to be eaten⟩.

The prefixes "ber-", "per-", and "ter-" neither alter nor remove the first letter of the root word. However, these prefixes transform to "be-", "pe-", and "te-" when they are attached to a root word starting with a letter "r-", or a root word with the first syllable ending with "-er" [Moeliono and Dardjowidjojo, 1988, pages 90–92; Wilujeng, 2002, pages 56–59]. For example: "ter-" + "racun" ⟨poison⟩ → "teracun" ⟨to be poisoned⟩; "ter-" + "tidur" ⟨to sleep⟩ → "tertidur" ⟨fall asleep⟩; "ber-" + "anak" ⟨child⟩ → "beranak" ⟨to have a child or children⟩; "ber-" + "racun" ⟨poison⟩ → "beracun" ⟨poisonous⟩; "ber-" + "ternak" ⟨livestock⟩ → "beternak" ⟨to breed for a living⟩; "per-" + "keras"⟨hard, strong⟩ → "perkeras" ⟨to harden⟩; "per-" + "runcing"⟨sharp⟩ → "peruncing" ⟨to sharpen⟩; and "per-" + "ternak"⟨livestock⟩ → "peternak" ⟨breeder⟩. When the prefix "ter-" is added to a root word of with a first syllable ending with "-er", the prefix may become "te-", and the decision of which version of the prefix to use is arbitrary [Sneddon, 1996, page 9]. Sometimes both "ter-" and "te-" versions are accepted. For example: "ter-" + "percaya" ⟨belief⟩ → "terpercaya" or "tepercaya" [both variants mean "reliable, believable"], but "ter-" + "percik" ⟨spot, stain⟩ → "tepercik" ⟨be

Rule	First Letter of Root Word	Rule to Apply
1	{a|e|i|o|u}...	peng-...
2	{g|h}...	peng-...
3	{k}...	peng-[k]...
4	{c|d|j}	pen-...
5	{t}...	pen-[t]...
6	{b|f|v}...	pem-...
7	{p}...	pem-[p]...
8	{s}...	peny-[s]...
9	{l|m|n|r|w|y}...	pe-...
10	{z}...	pe-...|pen-...

Table 2.1: The prefix "pe-" with its variants and effects on the root words. This table is adapted from the rules specified by Sneddon [1996, pages 9–14] and Wilujeng [2002, pages 59–62]. A lowercase letter following a hyphen and inside a pair of square brackets is a recoding character. For the last rule, the prefix "pe-" can remain the same or become "pen-" depending on the root words.

splashed, be splattered⟩. Only a very experienced speaker can identify which variant to use. There are exceptional cases to these rules [Moeliono and Dardjowidjojo, 1988, pages 90–92; Wilujeng, 2002, pages 56–59]. When the words "ajar" ⟨to teach⟩ and "unjur"⟨extended or stretched [for legs]⟩, are attached to the prefix "ber-", the derivative words are "belajar" ⟨to study⟩ and "belunjur"⟨to sit with legs stretched out⟩. When the prefix "per-" is added to the root word "ajar" ⟨to teach⟩, the derivative word is "pelajar" ⟨student⟩.

Prefixes "pe-" and "me-"

The morphology of the prefix "pe-" is more complex than the previous prefixes. The prefix "pe-" changes according to the root word it attaches to, and it may alter the root word [Sneddon, 1996, pages 9–14; Wilujeng, 2002, pages 59–62]. The set of rules about the variants of the prefix "pe-" and the effects on a root word are shown in Table 2.1.[7] In the first rule, the prefix "pe-" becomes "peng-" when it is attached to any root words that start with a vowel,

[7]The prefixes "pe-" and "me-" are written using different variants including "peng-" and "peN-" for "pe-" and "meng-" and "meN-" in Indonesian grammar books. For clarity and compactness, we write them as "pe-" and "me-" respectively.

Rule	First Letter of Root Word	Rule to apply
1	{a|e|i|o|u}...	meng-...
2	{g|h|x}...	meng-...
3	{k}...	meng-[k]...
4	{c|d|j|z}	men-...
5	{t}...	men-[t]...
6	{b|f|v}...	mem-...
7	{p}...	mem-[p]...
8	{s}...	meny-[s]...
9	{l|m|n|r|w|y}...	me-...

Table 2.2: The prefix "me-" with its variants and effects on the root words. This table is adapted from the rules specified by Sneddon [1996, pages 9–14], Moeliono and Dardjowidjojo [1988, pages 87–90], and Wilujeng [2002, pages 52–55]. A lowercase letter following a hyphen and inside a pair of square brackets is a recoding character.

for example, "pe-" + "ambil" ⟨to take⟩ → "pengambil" ⟨taker⟩, and "pe-" + "isi" ⟨to fill⟩ → "pengisi" ⟨filler⟩. In the fourth rule, whenever the prefix "pe-" is attached to a root word starting with the letters "c", "d", or "j", it becomes "pen-". For example: "pe-" + "curi" ⟨to steal⟩ → "pencuri" ⟨thief⟩; "pe-" + "dayung" ⟨to row⟩ → "pendayung" ⟨rower⟩; and "pe-" + "jahit" ⟨to sew⟩ → "penjahit" ⟨tailor⟩. According to the fifth rule, the prefix "pe-" also becomes "pen-" when it is added to a word starting with "t-", but with the difference that the letter "t-" is removed or recoded. For example: "pe-" + "tari" ⟨to dance⟩ → "penari" ⟨dancer⟩; and "pe-" + "terima" ⟨to receive⟩ → "penerima" ⟨receiver⟩. For the last rule in the table, "pe-" could remain the same or become "pen-" when it is added to a root word starting with the letter "z", and both forms are accepted. Both "penziarah" and "peziarah", which mean "a visitor to a sacred place or grave", stemming from "ziarah" ⟨to make a devotional visit to a sacred place⟩, are valid.

The morphology of the prefix "me-" is also complex. This prefix varies based on the root word it is attached to, and the root word may need to be recoded as well [Sneddon, 1996, pages 9–14; Moeliono and Dardjowidjojo, 1988, pages 87–90; Wilujeng, 2002, pages 52–55]. The set of rules about the variants of the prefix "me-" and the effects on a root word are shown in Table 2.2. The rules in this table are similar to the rules in Table 2.1, and can be interpreted in similar fashion. Based on the first rule, whenever the prefix "me-" is added to

a word starting with any vowels, the prefix becomes "meng-". For example: "me-" + "ambil" ⟨to take⟩ → "mengambil" ⟨to take⟩ and "me-" + "injak" ⟨to tread on⟩ → "menginjak" ⟨to tread on⟩. Based on the third rule, whenever the prefix "me-" is added to a word starting with the letter "k", the prefix becomes "meng-" while the letter "k-" is altered. "Mengecil" ⟨to become small⟩ and "mengantuk" ⟨to be sleepy⟩, stemming from "kecil" ⟨small⟩ and "kantuk" ⟨sleepiness⟩ respectively, are examples of the recoded root words after being attached to the prefix "me-".

When the prefixes "pe-" and "me-" are added to a root word that consists of only one syllable, we can either follow the rules in Tables 2.1 and 2.2, or instead change the prefixes to "penge-" and "menge-" [Sneddon, 1996, page 13]. The following examples are adapted from Sneddon [1996, page 13]: "pe-" + "bom" ⟨bomb⟩ → "pembom" | "pengebom" ⟨bomber⟩; "me-"+ "bom" ⟨bomb⟩ → "membom" | "mengebom" ⟨to bomb⟩; "pe-" + "tik" ⟨to type⟩ → "pentik" | "pengetik" ⟨typist⟩; and "me-" + "tik" ⟨to type⟩ → "mentik" | "mengetik" ⟨to type⟩. Moeliono and Dardjowidjojo [1988, page 89] consider only the versions with the prefix "menge-" as the "valid" version. Sneddon [1996, pages 13–14] adds that the word "tahu" ⟨to know⟩ is treated as a one-syllable word and only the prefix variants of "penge-", as in "pengetahuan" ⟨knowledge⟩ (with an addition of the suffix "-an"), and "menge-", as in "mengetahui" ⟨to know⟩ (with an addition of the suffix "-i"), can be used.

When the root words are still considered as loan words, recoding is optional [Moeliono and Dardjowidjojo, 1988, pages 89–90; Sneddon, 1996, pages 11–12]. The following examples are adapted by J.A. from Moeliono and Dardjowidjojo [1988, page 90] and Sneddon [1996, page 12]. For example: "me-" + "protes" ⟨to protest⟩ → "memrotes"|"memprotes" ⟨to protest⟩ and "me-" + "kritik" ⟨to criticise⟩ → "mengritik" | "mengkritik" ⟨to criticise⟩. Both recoded and non-recoded forms are accepted. When the loan words have been accepted as Indonesian words, they need to be recoded. Sneddon [1996, page 12] mentions that there are some exceptional cases where the recoded and non-recoded forms have different meanings. The following examples are taken from Sneddon [1996, page 12]. Both the words "mengkaji" and "mengaji" stem from the word "kaji" ⟨to examine, religious knowledge or teaching⟩ but these two versions have different meanings — the first word means "to examine perfunctorily" while the latter means "to recite Koranic verses".

Prefixes "ku-" and "kau-"

The prefixes "ku-" and "kau-" do not vary and do not change the root words they are attached to [Moeliono and Dardjowidjojo, 1988, pages 94–96]. For example: "ku-" + "baca" ⟨to read⟩ → "kubaca" ⟨I read⟩ and "kau-" + "bawa" ⟨to bring⟩ → "kaubawa" ⟨you bring⟩. These prefixes are not considered to be formal prefixes, and their usage is less frequent compared to other prefixes.

2.2.4 Confixes

Moeliono and Dardjowidjojo [1988, pages 81–82] state that a confix is a combination of a prefix and a suffix that is considered as an affix of its own. Both the prefix and the suffix have to be added together to create a meaningful derived word — removing only the prefix or the suffix does not leave a meaningful derived word. The prefix "ber-" and the suffix "-an" form a confix "berkejaran" ⟨to chase each other⟩ that stems from the word "kejar" ⟨to chase⟩. Adding only the prefix "ber-", as in "berkejar", or only the suffix "-an", as in "kejaran", has no meaning. In contrast, the prefix "ber-" and the suffix "-an" in the word "bertumbuhan" ⟨to have plants⟩, that stems from the word "tumbuh" ⟨to grow⟩, do not form a confix because adding only the suffix "-an" to the word "tumbuh" creates the word "tumbuhan" ⟨plants⟩ that has a meaning on its own. Therefore, the fact that a word has a prefix or a suffix does not necessarily mean that it has a confix. Combinations of prefixes and suffixes that are not confixes are discussed in Section 2.2.5.

There is no official complete list of Indonesian confixes. From Moeliono and Dardjowidjojo [1988, pages 80–85], we conclude that the most common Indonesian confixes are "ber-an" and "ke-an". These pairs of prefixes and suffixes can form either confixes or combinations depending on the root word they are appended to.

Adding a confix to a word usually adheres to the rules of adding the current prefix and the suffix. From the previous example, adding the confix "ber-an" to a root word adheres to the rules of adding the prefix "ber-" and the rules of adding the suffix "-an".

2.2.5 Combinations

It is possible to form a new word by adding more than one prefix, more than one suffix, and an infix together into a root word or a repeated word. For example: "ke-" + "ber-" + "untung" ⟨lucky⟩ + "-an" + "-mu" → "keberuntunganmu" ⟨your luck⟩; "ke-" + infix "-em-" + "kilau" ⟨shiny⟩ + "-an" + "-nya" → "kekemilauannya" ⟨its shininess⟩; and "se-"

Prefixes	Suffix
“me-”, “per-”, “ber-”, “ter-”, and “di-”	“-kan”
“me-”, “per-”, “ter-”, and “di-”	“-i”
“ber-” and “ke-”	“-an”

Table 2.3: The list of common prefixes and suffixes combinations in Indonesian as adapted from Moeliono and Dardjowidjojo [1988, page 85]. This list is by no means exhaustive; the prefixes “pe-” and “se-” are not listed.

Prefix	Disallowed suffixes
“ber-”	“-i”
“di-”	“-an”
“ke-”	“-i” and “-kan”
“me-”	“-an”
“ter-”	“-an”
“per-”	“-an”

Table 2.4: The list of prefixes and suffixes combinations in Indonesian that are not supposed to appear together as adapted from Moeliono and Dardjowidjojo [1988, page 85]. The prefix “ke-” cannot appear together with the suffix “-i” except for the word “tahu” ⟨to know⟩ of which the derived word is “ketahui” ⟨to know⟩. This list is by no means exhaustive as the prefixes “pe-” and “se-” are not listed.

+ “merah” ⟨red⟩ + “-nya” → “semerah-merahnya” ⟨as red as possible⟩. It is also possible to add prefixes and suffixes to a compound word. For example: “mem-” + “per-” + “tanggung jawab” ⟨responsibility⟩ + “-kan” → “mempertanggungjawabkan” ⟨to account for⟩ and “ke-” + “ikut serta” ⟨to participate⟩ + “-an” + “-mu” → “keikutsertaanmu” ⟨your participation⟩. Adding these combination affixes still adheres to the rules of adding their component affixes.

The combinations of prefixes and suffixes that occur very often are shown in Table 2.3. Some prefixes and suffixes never appear together; these are listed in Table 2.4.

Moeliono and Dardjowidjojo [1988, page 86] add that there are prefixes that occur very often together and the order of occurrence is fixed. These prefixes are: “me-” + “per-” → “memper -”;[8] “me-” + “ber-” → “member-”; “di-” + “per- → “diper-”; “di-” + “ber- →

[8]This is a special case where the prefix “per-” is not recoded.

"diber-"; "ter-" + "per- → "terper-"; and "ter-" + "ber- → "terber-". This list is by no means exhaustive as there are other commonly used prefixes, such as "se-" + "per-" → "seper-", that are sometimes omitted from references or formal grammar; for example, this is not listed in Moeliono and Dardjowidjojo [1988] but is listed in Sneddon [1996, page 56]. Triple prefixes are possible but they do not occur as frequently as double prefixes, for example, "ke-" + "se-" + "per-" → "keseper-". The rules that disallow certain prefixes-suffixes combinations still apply to these double or triple prefixes, but, only the first prefix is examined. For example, as the prefixes "me-" and "di-" cannot appear together with the suffix "-an", consequently the prefixes "memper-" and "diper-" cannot appear together with the suffix "-an".

Summary

We have mentioned characteristics of Indonesian that are different from English and may affect information retrieval performance. There are other differences such as sentence structures, prepositions, and auxiliaries that are semantic and beyond the scope of this thesis. We have also discussed aspects of Indonesian morphology that affects stemming rules that are discussed in detail in Chapter 3. In the following sections, we discuss information retrieval (IR) and cross-lingual information retrieval (CLIR).

2.3 Information Retrieval

Information retrieval (IR) is different to database retrieval. A database retrieval system simply retrieves all documents or objects that satisfy certain criteria, whereas an information retrieval system needs to assess the information needs of its users and rank the answer documents based on likely relevance [Baeza-Yates and Ribeiro-Neto, 1999, pages 1–2].

Zobel and Moffat [2006] contrast a database retrieval system and an information retrieval (IR) system. A database retrieval system can accept complex queries and return all answers matching the logical conditions of the queries. A database retrieval system assigns a unique key for each of its record to allow searching using that key. Queries for an IR system are in the forms of lists of terms and phrases. An IR system usually returns certain number of answer documents ranked in descending order according to their similarity values — the value indicating how close a document is to a query. There is no concept of key for an IR system, instead it keeps statistics of terms.

In this work, we refer to text IR when we say IR. A practical example of a complex text IR system is a search engine on the Web.

This section is structured as follows. In Section 2.3.1, we describe typical tasks that can be performed with an IR system. The subsequent three sections describe three different steps involved in a search engine, namely parsing, indexing, and querying. The last section describes experimental methods for an IR system.

2.3.1 Search Tasks

Broder [2002] categorises web search tasks into three categories: informational, navigational, and transactional. Example of informational queries are `Melbourne weather report` and `universal declaration of human rights`, mentioned at the beginning of this chapter, where a user enters as a query and expects the system to return any documents that are relevant to the queries. Craswell et al. [2001] define this kind of search task as *topic finding* or *subject search*. Voorhees [2004] name this kind of task as the *ad hoc* search task which forms the basis of most TREC[9] retrieval tasks and is the typical search task performed by users of most web search engines. A navigational query is used when a user already has a particular Web page in mind and names the actual site in the query. Examples of navigational query are `RMIT university` and `Australian Taxation Office`. This query type is also known as *known item* search and Craswell et al. [2001] label them as *homepage finding* or *site-finding* tasks. For subject-search queries, a list of relevant documents are expected to be ranked at the top of the IR system's list of answers; whereas for site-finding queries, the users expect the particular page to be ranked at the top. A query is considered as transactional when the users expect further interaction after their initial query looking for a site. Examples of this kind of query are those entered when a user is doing online shopping or downloading files. Similar to the ad hoc search task, there is no one right answer for this kind of query; there is only an appropriate or inappropriate answer relative to the user.

This thesis covers only the traditional IR search task — the subject search task that is also referred to as the ad hoc task.

2.3.2 Parsing

Parsing in the IR sense means choosing the subset of terms in the documents that should be indexed [Meadow et al., 2000, pages 143–145]. A document in this thesis is defined as a unit of text whose relevance to a query can be judged, such as a Web page, a recipe, or a government bill.

[9] TREC is discussed in Section 2.3.5.

Parsing is done to unify the format of the document and the query. A parser usually removes elements that do not contribute much information for retrieval such as marked-up tags of HTML documents and punctuation marks. However, some punctuations, such as a dot (".") as the end of a sentence or a hyphen ("-") for plurals or repeated words in Indonesian, may be useful. Removal of these punctuation marks is optional depending on the implementation of the search engines. The removal is not limited to punctuation marks and markup-tags, some words can also be removed since not all words are created equal. Words can be categorised into different parts of speech, such as nouns, verbs, prepositions, conjunctions, and adverbs. As a result, some words carry more information than the others and not all words need to be indexed [Meadow et al., 2000, page 143].

Defining terms to be indexed

The choice of terms to index largely depends on the language of the documents [Ogawa and Matsuda, 1997]. The terms chosen can simply be at word level for most IR systems [Zhai, 1997]; this is relatively straightforward for languages that use the Roman alphabet, such as English, French, and Indonesian. For these languages, spaces and punctuation marks can be used to tokenise words. However, for languages such as Chinese, Japanese, and Korean (CJK), where the words boundaries are not clear, *morphemes* [Wu and Tseng, 1993] and *n-grams* [Lee and Ahn, 1996] are usually used as terms to index.

More complex methods of defining terms exist. These include using word senses instead of words as terms [Krovetz and Croft, 1992] and expanding the words or terms to be indexed using a morphological or a syntactical parser or even combination of the two parsers [Jacquemin et al., 1997]. These methods could increase the accuracy of the answer documents, however, they require a large vocabulary set and a good semantical knowledge of the language.

Since the focus of our work is Indonesian, we choose to index simply at word level. We ignore all punctuation other than the hyphen, which is needed to signify plurals.

As a consequence of this decision, we refer interchangeably to the terms we index as "term" or "word".

Case folding

Not all documents and queries use capitalisation consistently. For example, a user may enter `RMIT University`, `Rmit University`, or `rmit university` as queries for the same

information need. To address this problem, all letters in the documents are converted to lower case or upper case before indexing [Witten et al., 1999, page 146]. Similarly, all letters in the query are represented internally in the same case used for the documents. Thus, our example queries all become `rmit university` or `RMIT UNIVERSITY` for lower case and upper case respectively.

Witten et al. [1999, page 146] identify situations where the original case should be retained. For example, with the query `Standard and Poor`, a user is probably looking for the home page of the financial company rather than documents with text such as "the standard of living has risen, but the poor have become poorer". Since our search focuses on topic finding rather than home page finding, we convert all characters to lower case.

Stopping

As discussed earlier, not all words carry the same amount of information. Stopping is the act of removing words that do not contribute much to the content of the documents [Baeza-Yates and Ribeiro-Neto, 1999, pages 167–168]. The articles "the", "a", and "an" and the conjunctions "in", "at", and "to" are examples of such highly frequent words — referred to as *stopwords*.

Removing stopwords can save index size and processing time and also reduce the noise level. Stopping can be performed on documents prior to indexing and to the queries during querying.

However, using only frequency as the guidance to create a stopword list can backfire. Buckley et al. [1992] illustrates this issue with the query `The head and president of an American computer system company based in Washington said she expected to make a million systems by the end of the year`. All of the words in this query occur in more than 10% of documents in the TREC collection and stopping them leaves no words to search for. Therefore, the stopword list needs to be created more carefully, perhaps by including only prepositions and conjunctions. This will not always work either; the `to be or not to be` of Hamlet is an obvious example of a query that would not work.

It is not clear whether stopping can increase retrieval accuracy, as the results vary between different document collections [Meadow et al., 2000, pages 232–233]. We investigate the effects of stopping on Indonesian IR in Chapter 4.

Stemming

Stemming is the act of reducing a word into its stem or semantic root [Meadow et al., 2000, page 221]. For example, the words "sits" and "sitting" are stemmed to become "sit". Stemming is a basic text processing tool often used for efficient and effective text retrieval [Frakes, 1992], machine translation [Bakar and Rahman, 2003], document summarisation [Orăsan et al., 2004], and text classification [Gaustad and Bouma, 2002]. We have discussed Indonesian morphology in detail in Section 2.2. Stemming can be applied on both the terms of documents prior to indexing and the queries during querying. We describe our approach and experiments on Indonesian stemming in Chapter 3.

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. In other words, stemming is expected to increase recall, but might decrease precision.[10]

Despite the possibility that stemming could decrease precision, the actual results are language dependent [Pirkola et al., 2001]. It is unclear whether stemming improves retrieval in general as the results vary depending on the language [Pirkola et al., 2001], the queries and the collection [Harman, 1991]. For languages such as English [Hull, 1996], French [Savoy, 1999], Slovene [Popovič and Willett, 1992], and Arabic [Larkey et al., 2002], stemming increases precision values. Popovič and Willett claim that languages that are morphologically complex such as Slovene are more likely to benefit from stemming. For the same reason, we suspect that Indonesian might benefit as well. We compare retrieval performance with and without stemming for Indonesian in Chapter 4.

Identifying Proper Nouns

There is a positive correlation between the number of proper nouns in a query and retrieval performance [Mandl and Womser-Hacker, 2005]. This is understandable: if a user uses proper nouns in a query, they are looking for specific information such as `Haley's comet` or `the White House`. When stemming, proper nouns ought not to be stemmed. In Chapter 4, we compare the effects of not stemming, stemming all words except proper nouns, and stemming every word, on retrieval performance.

[10] Recall and precision are measures used to determine effectiveness in IR and are discussed in Section 2.3.5.

Document ID	Document Text
1	I bought a new mat.
2	A cat sat on the mat.
3	The cat is white and the mat is blue

Table 2.5: An example document collection.

In this thesis we use the term proper nouns to refer things that represent a concept. Proper nouns may be names of persons, organisations, locations, time expressions, brands, books, and movies [Wakao et al., 1996].

Tokenisation

Tokenisation is the process of breaking up a string of characters into tokens as basics units before further processing [Webster and Kit, 1992]. The tokens can be in different formats including words, idioms, morphemes, and n-grams.

Using n-grams as indexing terms instead of words can be beneficial in retrieval performance not only for languages where the word boundaries are indistinguishable, but also for languages with the Roman alphabet such as English, French, Dutch, German, and Italian [Mayfield and McNamee, 2003]. For example, the 4-grams of "information" are "info", "nfor", "form", "orma", "rmat", "mati", and "ation".

Tokenisation can be considered to be a form of stemming that is language independent; therefore it may have the benefits reaped from stemming without detailed semantical knowledge of the documents to be indexed [Mayfield and McNamee, 2003]. For example, the words "computer", "computing", and "compute" can all be stemmed to the same 6-gram of "comput" without prior knowledge of the English morphology.

We experiment with indexing using n-grams in Chapter 4.

2.3.3 Indexing

In IR, indexing all keywords or terms and the location of these terms is desirable as the document collection can be approximately rebuilt by merely using the indices [Witten et al., 1999, pages 105–109]. The drawback of this method is the amount of space taken for the indices. This can be alleviated by index compression techniques. Stopping and stemming could also help to save space, but at the expense of information loss. We discuss the most

Term	Doc Frequency; (Doc ID, Location)	(Doc ID, Term Frequency)
a	< 2; (1;3),(2;1)>	< (1,1), (2,1) >
and	< 1; (3;5)>	< (3,1) >
blue	< 1; (3;9) >	< (3,1)>
bought	< 1;(1;2)>	< (1,1)>
cat	< 2; (2;2),(3;2)>	< (2,1), (3,1) >
i	< 1; (1;1) >	< (1,1) >
is	< 1; (3;3,8)>	< (3,2) >
mat	< 3; (1;5), (2;6), (3;7) >	< (1,1), (2,1), (3,1) >
new	< 1; (1;4) >	< (1,1) >
on	< 1; (2;4) >	< (2,1)>
sat	< 1; (2;3) >	< (2,1) >
the	< 2; (2;5), (3;1,6) >	< (2,1), (3,2)>
white	< 1; (3;4) >	< (3,1) >

Table 2.6: Examples of an inverted list for the document collection in Table 2.5 with the terms ordered alphabetically. The second column is more complete than the third column as it includes the location of the terms in the collection.

common type of indexing in IR — inverted file indexing — and briefly describe less popular methods — signature files and bitmap indexing.

The inverted file, also called the posting list, is the most common form of indexing. Its concept is similar to the concept of the index list at the back of a book [Zobel and Moffat, 2006]. There are different versions of posting lists. Table 2.5 shows an example document collection, and Table 2.6 shows the corresponding inverted index. The first column of Table 2.6 shows all terms appearing in the document collection, while the second column shows the complete information of the documents in the form of an inverted list [Witten et al., 1999, page 113]. The inverted list consists of document frequencies f_t — the counts of how many documents contain each term — and a list of document identifiers and term offset pairs. For example, as can be seen from Table 2.6, the word "the" appears in two documents in the collection, it appears in document 2 at position 5 (five words from the beginning) and in document 3 at positions 1 and 6.

A different variant of the posting list does not include the term locations, but instead stores only pairs of document identifiers and the term frequencies in those documents [Zobel and Moffat, 2006]. This version is shown in the third column of Table 2.6. This is the most common form of posting list. These term frequencies are denoted with the symbol $f_{d,t}$ in this thesis. It can be seen from the postings that the word "the" appears once in document 2 and twice in document 3. Although the document frequencies f_t are not specified explicitly, they can be derived by calculating the numbers of pairs containing the document identifiers and the term frequencies.

Signature files or bitmaps indexing is less popular than using inverted files. The basic principle of signature file indexing is allocating a block to each document and hashing each term in the document, and setting several bits in the document chunk [Faloutsos, 1985]. Signature file is not efficient for large amount of text and not effective in ranking documents based on similarity [Zobel et al., 1998]. Bitmap indexing uses a chunk size as big as the number of distinct terms in the collection [Fraenkel et al., 1986]. The presence of each distinct term in the query sets one bit in the chunk. Bitmap indexing is also not efficient for large amount of text and cannot be used to rank document similarity.

2.3.4 Query Evaluation

Once the documents are parsed and indexed, an IR system needs to be able to process user queries and retrieve documents that are most relevant to the queries.

The two most common approaches to query evaluation are Boolean and ranked query evaluation. Both types of query evaluation are discussed in the next sections.

Boolean query evaluation

A Boolean query uses the Boolean operators *AND*, *OR*, and *NOT* [Witten et al., 1999, pages 153-154]. For example, if a user wants all words from the query

`Melbourne weather report`

to appear, they need to specify the query as

`Melbourne` *`AND`* `weather` *`AND`* `report`

whereas if the user is satisfied with only one of the terms appearing in the documents retrieved, they could specify the query as

`Melbourne` *`OR`* `weather` *`OR`* `report`.

If the user is not interested in the documents explaining about accommodation in Melbourne,

they could specify the query

`Melbourne` *AND* `weather` *AND* `report` *AND NOT* `accommodation`.

The terms can be nested to combine the operators *AND*, *OR*, and *NOT*. For example, the query

`(Melbourne` *OR* `Ballarat)` *AND* `(weather` *OR* `traffic)` *AND* `report`
AND NOT `(accommodation` *OR* `entertainment)`

indicates that the user is interested in documents that give reports about the weather or traffic condition in either Melbourne or Ballarat and do not contain the word "accommodation" or "entertainment".

It can be seen from the examples above that forming more complex Boolean queries can be cumbersome. Chowdhury [2004, pages 173–174] lists limitations of Boolean queries. First, the quality of documents obtained depends on how well the user forms their query. Second, as the documents obtained are not ranked based on relevance, the answer set might be too broad, hindering the user, or too narrow, excluding the required information.

A *phrase query* is like a Boolean query except that the operator *AND* is used implicitly between the query words, and the order of the keywords in the documents should follow the order of the keywords in the query [Bahle et al., 2002]. When a user enters the phrase `Melbourne weather report` as a query, a IR system that implements phrase querying should return documents containing the words contiguously and in this order.

Ranked Query evaluation

Ranked querying is a more natural form of querying [Spink et al., 2001], where the users enter the queries in natural language form, and the answer documents are returned ranked by decreasing estimated relevance to the query. Spink et al. [2001] report that users rarely use Boolean queries, and most of the Boolean queries they do form are not correct.

When a query `the mat` is entered into our document collection, the ranks of the documents returned are shown in Figure 2.1. Here, we use simple a similarity measure to illustrate the ranking system, the more the number of terms in a document matches the number of terms in the query, the higher the ranking of that particular document.

The similarity of document is not dependent solely on the presence of the keywords but also by the weighted frequencies of the terms in the query and in the documents [Zhai, 1997]. Major determinants to assign term weights are the scarcity of a term and the length of

DocID	*Rank*	*Content*
3	1	The cat is white and the mat is blue
2	2	A cat sat on the mat.
1	3	I bought a new mat.

Figure 2.1: Documents returned for the query "the mat" ranked by similarity, the ranking is based on the number of document terms that match query terms.

the document [Zobel and Moffat, 2006]. The scarcity of a term is determined by the *term frequency* and the *inverse document frequency*. The *term frequency* (TF) is the frequency of a term in a document and is denoted as $f_{d,t}$ [Zobel and Moffat, 2006] as explained in Section 2.3.3. The basic premise of using $f_{d,t}$ is that a document that contains more query terms t is deemed to be more likely to be relevant than the document with fewer occurrences of the query term. The term frequencies for the word "the" for the sample documents in Figure 2.1 are 2, 1, and 0 for documents with identifiers 3, 2, and 1 respectively. Meanwhile, *inverse document frequency* (IDF) can be denoted as $\frac{1}{f_t}$. where f_t is the number of documents containing the query term t [Zobel and Moffat, 2006]. The principle of using the inverse is the importance of a term diminishes in proportion to the frequency of the documents containing that term. Words such as "blue" and "sat" appear only once in the document collection, therefore are more important that the word "mat" that appears three times. As longer documents contain more terms, they thus have higher chance to be deemed relevant, so the length of the document needs to be taken into account [Singhal et al., 1996]. For example, the third document in the sample collection is the longest, therefore it has the most word matches of "the". We do not normalise the document length in this example as the difference in length is negligible as it is only a few words length.

We discuss two main approaches to ranked query retrieval, namely the *vector space model* and *probabilistic retrieval* in the next sections, and follow with a brief look at other techniques, including *language modeling* and *latent semantic indexing*.

Vector Space Model

The vector space model was first introduced by Salton and Lesk [1968] for the SMART retrieval system. The basic premise of the vector space model is that distinct terms in the query and in the documents occupy N-dimensional vectors, where N is the number of

distinct terms that are in both the query and the document. The more similarities between the query and the document in the vector space, the closer the angle between them will be. As the result, the cosine measure is usually used to evaluate how well the two vectors are aligned.

The cosine measure formulas below are discussed in Lee et al. [1997]. The mathematical definition of the cosine between two vectors is:

$$\cos(q, d) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| \cdot |\vec{d}|} \tag{2.1}$$

where q is a query; d is a document; $\vec{q}$ is a query vector; $\vec{d}$ is a document vector; $|\vec{q}|$ is the vector length of the query q; and $|\vec{d}|$ is the vector length of the document d. This means that the cosine measure is determined by the dot products between the query vector and the document vector, and normalised by the lengths of the document and query vectors [Lee et al., 1997]. The query and document vectors are determined by the weight of each term t in the query $w_{q,t}$ and document $w_{d,t}$ respectively. Meanwhile, the vector length of the query and the document depends merely on the terms that are present either in the query or in the document. Therefore, the formula becomes:

$$\cos(q, d) = \frac{\sum_t w_{q,t} \cdot w_{d,t}}{\sqrt{\sum_{t \in q} w_{q,t}^2 \times \sum_{t \in d} w_{d,t}^2}} \tag{2.2}$$

Only the terms that exist in both the query and the document $t \in q \cap d$ contribute to the dot products [Zobel and Moffat, 2006]. From this assumption, a new formula arises:

$$\cos(q, d) = \frac{\sum_{t \in q \cap d} w_{q,t} \cdot w_{d,t}}{\sqrt{\sum_{t \in q} w_{q,t}^2 \times \sum_{t \in d} w_{d,t}^2}} \tag{2.3}$$

As this model assigns relevance based on weighted frequencies, rarer terms are given higher weights (the term scarcity rule) [Zobel and Moffat, 2006]. The weight of a term in the query q is usually specified as:

$$w_{q,t} = \ln\left(1 + \frac{N}{f_t}\right) \tag{2.4}$$

where N is the number of documents in the collection and f_t is the frequency of documents containing the term t. Equation 2.4 is often referred as the *inverse document frequency* rule; the more documents that contain a term t, the smaller the weight of that term [Zobel and Moffat, 2006]. The natural logarithm (ln) function is used to curb the quick progression of the weights.

The weights of the terms in the documents are usually formalised as:

$$w_{d,t} = 1 + \ln f_{d,t} \tag{2.5}$$

where $f_{d,t}$ is the number of occurrences of a term t in a document d. Equation 2.5 is often called the *term frequency* rule; a document with more occurrences of a term t is deemed to be more important than a document that has fewer occurrences.

Substituting Equation 2.4 and Equation 2.5 into Equation 2.3, we obtain:

$$\cos(q,d) = \frac{\sum_{t \in q \cap d}(\ln(1 + \frac{N}{f_t}) \times (1 + \ln f_{d,t}))}{\sqrt{\sum_{t \in q}(\ln(1 + \frac{N}{f_t}))^2 \times \sum_{t \in d}(1 + \ln f_{d,t})^2}} \tag{2.6}$$

Since the query length is a constant for any given query, the ranking is unchanged if we drop the normalisation by query vector length, giving:

$$\cos(q,d) = \frac{\sum_{t \in q \cap d}(\ln(1 + \frac{N}{f_t}) \times (1 + \ln f_{d,t}))}{\sqrt{\sum_{t \in d}(1 + \ln f_{d,t})^2}} \tag{2.7}$$

Equation 2.6 is often denoted as the symmetric cosine measure, where the lengths of both the query and the document vectors contribute towards the similarity estimation. Equation 2.7 ignores the query vector length, and we refer to it in this thesis as the cosine measure formula. The higher the similarity score, the more likely to be relevant a document is estimated to be to the query, and so documents are presented to the user ranked by decreasing similarity score.

Probabilistic Model

The probabilistic model of IR relies on the notion that each document has a certain probability of being relevant to a query. The documents that are most likely to be relevant and to be useful to the users are ranked by decreasing order of probability. This principle referred to as the "probability ordering principle" [Robertson and Sparck Jones, 1976] or the "probability ranking principle" [Robertson, 1977].

Several major methods regarding probabilistic models for IR can be found in Crestani et al. [1998]. In this thesis, we only discuss the most popular probabilistic model, the Okapi model [Robertson and Walker, 1999], especially the Okapi BM25 ranking function [Robertson et al., 1994]. The Okapi formula we use is explained in detail in Sparck Jones et al. [2000] and Robertson et al. [1994]. As we only experiment with different parameters of the model, and

do not attempt to extend the model as part of our research, we do not present the derivation of the model from first principles, but instead focus on explaining its key characteristics.

The similarity score between a query q and a document d in OKAPI can be formalised as:

$$\text{score}(d,q) = \sum_{t \in V} \log \frac{(r_t + 0.5)(N - f_t - R + r_t + 0.5)}{(f_t - r_t + 0.5)(R - r_t + 0.5)} \tag{2.8}$$

where t is a term, V is the vocabulary set of distinct terms, r_t is number of relevant documents in which the term t occurs, N is the number of documents in the collection, f_t is number of documents containing the term t, and R is the number of relevant documents.

This formula is derived from the Bayesian theorem about the probability of the presence or the absence of a term in the relevant and non relevant documents. The constant 0.5 is added into each r_t to normalise the equation in case there is no relevant document that contains the term t. Robertson and Sparck Jones [1976] cite statistical justification for this choice of value.

In an ad hoc retrieval task, where the relevance of documents is unknown, the values of R and r_t can be set to 0; simplifying the Equation 2.8 into:

$$\text{score}(d,q) = \sum_{t \in V} \log \frac{N - f_t + 0.5}{f_t + 0.5} \tag{2.9}$$

Equation 2.8 is the basic probability model used in TREC-1 [Robertson et al., 1992]. This equation is based on the unrealistic assumption that all documents have the same length. If document length varies, the equation is biased towards longer documents, as they are more likely to contain the term t. As the result, some document length normalisation needs to be incorporated into the equation. Furthermore, the equation takes into account only the document frequencies (f_t) without considering the frequency of a term t in the document d ($f_{d,t}$) nor the frequency of a term in the query q ($f_{q,t}$). The widely successful combination of parameters is the Best Match 25 function, often shortened as *BM25*. This *BM25* was first introduced in TREC-3 [Robertson et al., 1994]. The formula of *BM25* is:

$$\text{BM25}(d,q) = \text{score}(d,q) \times \sum_{t \in q} \frac{(k_1 + 1) f_{d,t}}{k_1 \times (((1-b) + b) \times \frac{|d|}{ADL}) + f_{d,t}} \times \frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}} \tag{2.10}$$

where $|d|$ is the document length (this can be expressed for example as the number of characters, or the number of words before or after stopping); ADL is the average length of the documents in the collection in the same measurement unit as the document length and; k_1, k_3, and b are tuning constants, which we explain in the next paragraph. As our research

focuses on ad hoc retrieval, for which the relevance of the documents is unknown, we use Equation 2.9 rather than Equation 2.8 to calculate $score(d, q)$.

The first part of the multiplication in Equation 2.10 is the original OKAPI formula that takes into account only the document frequency and the number of documents in the collection. The second part takes into account the frequency of a term in a document $f_{d,t}$ and the normalisation of document length. The k_1 constant is a positive number used to determine how strongly $f_{d,t}$ affects the whole weight in the equation. If the value of k_1 is very small or 0, the contribution of $f_{d,t}$ is effectively limited to whether the term t is present in the document ($k_1 = 0$ means that this second part of the multiplication becomes 1) without taking into account how many times the term t present in that document. Conversely, larger k_1 value indicates that the weight increases more quickly with the $f_{d,t}$. A simple way to normalise document length is to divide the length of that document with the average document length in the collection. The tuning constant b, which is between 0 and 1 inclusive, is used to determine how much document length normalisation is required. If the value of b is 0, there is no document length normalisation; whereas the value b of 1 indicates that normalisation is in full effect. The third part of the equation takes into account the frequency of a query term, which is useful for long queries where a term can be repeated in the query. The function of k_3 tuning constant is similar to the function of k_1 in terms of determining how much $f_{q,t}$ affects the whole equation. Small k_3 values limit the effect of $f_{q,t}$ whereas with larger values indicates the weight increase is linear to $f_{q,t}$.

Robertson and Walker [1999] state that the optimum values of b, k_1, and k_3 depend on the queries and the document collection. In general, the default values for b and k_1 are set to 0.75 and 1.2 respectively. These numbers are considered to be the best across various TREC collections. The value of k_3 for long queries is normally set to between 7 and 1000 inclusive. Since we deal with Indonesian queries and document collections, we suspect that the ideal tuning constants might be different, and we try to find the best b and k_1 tuning constants for our collection in Chapter 4. We do not try to find the optimum k_3 value since it is only beneficial when the queries are long, which is not the case for our Indonesian queries.

Other Retrieval Models

Other retrieval models have been proposed in the IR literature. However, since we use the cosine and BM25 similarity scores in our experiments, we only briefly describe these other approaches.

Language Models. Statistical language modelling (SLM) aims to capture regularity in natural language, and are used in various natural language application tasks including IR, machine translation, and spelling correction [Rosenfeld, 2000]. SLM is used to estimate the probability distribution of words, or sentences, or other linguistic units. The most common language modelling techniques are n-gram models [Chen and Goodman, 1998] where the probability of a term t_i occurring in a string s is dependent on the product of the probability of the preceding $n-1$ terms. For an n-gram of size 2 (or bigram), the probability of a term t_i depends only on the previous term t_{i-1}. The probability distribution for the string s is then:

$$P(s) = \prod_{i-1}^{l} P(t_i|t_i - 1)$$

where l is the length of s.

Smoothing is used in language modelling to produce more accurate probabilities by making the distribution more uniform and normalising zero probabilities [Chen and Goodman, 1998]. Language modelling requires a large number of parameters due to large vocabulary and ambiguity of language, therefore it needs large amount of training data [Rosenfeld, 2000]. In IR environment, language models are used to rank documents by estimating the probability that a query is generated from a document model [Nallapati et al., 2003]:

$$P(Q|D) = \prod_{t \in Q} P(t|D)$$

where Q is the query, D is a document, and t is a term.

In this thesis, we focus on vector space and probabilistic information retrieval models, such as cosine measure and Okapi BM25.

Latent Semantic Indexing. This technique is based on the proximity of concepts between terms in the query and in the documents [Syu et al., 1996]. The promixity is measured by mapping the terms in queries and documents in latent semantic space, a vector space the dimensions of which have been reduced based on term frequencies of queries and documents [Hofmann, 1999]. The premise of LSI is that documents with co-occuring terms should share similar latent space. Tang et al. [2004] state that when the corpus is large and contains diverse materials, the performance of LSI is not as good as other similarity measures such as Okapi. Moreover, LSI cannot be measured properly in term of memory usage and computation time. We choose not to use LSI for our Indonesian collection.

Rank	Score	% out of top score	(Relevance) from ground truth
1	125	100.00%	R
2	122	97.60%	N
3	121	96.80%	N
4	110	88.00%	R
5	103	82.40%	R
6	100	80.00%	N
7	98	78.40%	R
8	92	73.60%	N
9	75	60.00%	N
10	73	58.40%	R

Table 2.7: An example of relevance judgements of the top 10 documents retrieved by an IR system in response to a query and their similarity scores. We assume that there are 20 relevant documents to the query in the collection as a whole. The first column contains the rank, the second column contains the similarity scores between the query and that document, the third column contains the absolute percentage from the top score, and the fourth column indicates the relevance to the query (R is relevant and N is not relevant). The similarity scores used here are examples only.

2.3.5 Experimental Methods

We have described major IR models that are used to identify potentially relevant documents. In this section, we describe experimental methods commonly used to evaluate the performance of alternative techniques.

Standard measures and testbeds are needed to evaluate the performance of different approaches. First, we discuss the main effectiveness and efficiency measures used in IR. We later describe IR testbeds and their standardised format. We also discuss statistical significance tests.

Evaluating Retrieval Effectiveness

The most common measures of retrieval performance are recall and precision [Witten et al., 1999, pages 188–189]. Others include the mean reciprocal rank (MRR) and separation values.

We describe these measures here.

Recall. To measure how successful a system is at retrieving relevant documents from the collection, we use recall. This is the fraction of all relevant documents retrieved from the collection.

$$\text{Recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Number of relevant documents}} \tag{2.11}$$

The higher the recall, the better the system.

An answer set that a hypothetical IR system has returned in response to a query is shown in Table 2.7. Suppose that there are 20 relevant documents in the whole collection and the system manages to retrieve 5 of these in its 10 returned candidate answers, the recall is 5 out of 20 or 25% at the cut-off value of 10. If we only see the top 5 answers — cutoff value of 5 — the recall value is 3 out of 20 (the number of relevant documents in the collection is constant) or 15%. Therefore, the recall value is dependent on the cutoff value.

Precision. The answers returned by an IR system should have a high proportion of relevant documents; this can be measured using precision, which is the proportion of relevant documents in the documents retrieved in the answer set.

$$\text{Precision} = \frac{\text{Number of relevant documents retrieved}}{\text{Number of retrieved documents}} \tag{2.12}$$

As with recall, the precision value is specified at a cutoff value. From Table 2.7, the precision value for the cutoff value of 10 (*precision@10*) is 5 out of 10, or 50%. The *Precision@5* value for the same table is 3 out of 5, or 60%. Precision is typically reported for cutoff values such as 5, 10, 20, or 100. According to a study conducted by Spink et al. [2001], more than a quarter of users only look at the first 10 answers returned by a search engine, indicating the importance of *Precision@10*.

Another common measure of precision is the *R-precision* where R is the cutoff value that reflects the actual number of relevant documents in the collection. In our case, since there are 20 relevant documents in the whole collection, the R value is 20 so the *R-Precision* is the value for *Precision@20*, which is 25%.

A further precision measure that is commonly used is *mean average precision* (MAP). The *mean average precision* is obtained by taking the precision value whenever a relevant document is found. As shown in Table 2.7, the first relevant document is at position 1, therefore the average precision for that document is $\frac{1}{1}$. Similarly, the second relevant document is located at position 4, therefore the average precision is $\frac{2}{4}$. If the relevant document is not in the answer set, the precision value for that document is zero. Based on the answer set in

Table 2.7, the *mean average precision* is:

$$\text{MAP} = \frac{(\frac{1}{1} + \frac{2}{4} + \frac{3}{5} + \frac{4}{7} + \frac{5}{10} + (15 \times 0))}{20} = \frac{\frac{280+140+168+160+140}{280}}{20} = \frac{888}{5600} = 15.86\%$$

Mean average precision is more sensitive to the rank of relevant items than the previous measures. When measuring the precision only at a certain cutoff value, the order of the ranks of relevant documents does not matter. For example, the value of *Precision@5* for the answer set shown in Table 2.7 is still 60% even if the top two documents are not relevant but are followed by three relevant documents. On the other hand, the value of *mean average precision* is affected by rankings. The more relevant documents ranked at the top, the higher the *mean average precision*. Mean average precision also has a recall component, since it is also affected by the number of relevant documents. Mean average precision is a widely-used metric.

In this thesis, we use the averages of *precision@10*, *R-precision*, and *mean average precision* across a set of queries to evaluate retrieval effectiveness.

Mean Reciprocal Rank. The reciprocal rank is the inverse of the rank of the first relevant document. The values of reciprocal rank are averaged over all queries to obtain mean reciprocal rank (MRR). When there is no relevant document in the answer set for that particular query, the MRR value is 0. From the result in Table 2.7, which is for a single query, the reciprocal rank is $\frac{1}{1} = 1$. Similar to the *mean average precision*, the value of *mean reciprocal rank* (MRR) is sensitive to the rank position of relevant documents.

The MRR value is usually used for tasks where very few — usually one — answers or relevant documents are expected, as in question answering [Voorhees, 1999] and homepage finding [Ogilvie and Callan, 2003]. We use MRR to measure the effectiveness of finding parallel documents in Chapter 5.

Separation Value. The separation (SEP) value is used to judge the effectiveness of identifying parallel documents. A document is parallel to another document if they are direct translations of each other [Sadat et al., 2002]. The SEP value is used to measure how well a system can discriminate good matches from bad matches [Hoad and Zobel, 2003]. We use it to measure the difference between similarity scores of parallel and non-parallel documents. We use the terms *relevant* to represent parallel answers (good matches) and *not relevant* to represent non-parallel answers (bad matches) to simplify our description.

The SEP value is defined as the difference between the score of lowest true match (LTM) and the highest false match (HFM) in an answer set, assuming that all relevant documents are already in the answer set [Hoad and Zobel, 2003]. Scores are normalised such that score of the top ranked document is 100. The LTM is the normalised score for the *relevant* answer

ranked bottommost, assuming there are only five relevant documents in Table 2.7, the LTM is the normalised score of the document at rank 10 — 58.40% . The HFM is the normalised score for the topmost *not relevant* answer — the score of document at rank 2 of 97.60%. The SEP value is the difference between the LTM and the HFM, which in this case is −39.2%. This negative SEP value is undesirable, as it indicates that the system cannot distinguish between the *relevant* and *not-relevant* answers, and even ranks some false answers above true answers. The desired result is to see the *relevant* answers are all ranked at the top of a result list above all the *not relevant* answers with a big gap between the LTM and the HFM. The higher the SEP value, the more confidence we can have in the system's ability to distinguish the *relevant* answers from the rest.

Evaluating Retrieval Efficiency

The main components of efficiency in terms of IR are storage space and speed. An efficient IR system is expected to use as little space as possible to store the document collection and to process documents and queries as fast as possible. The processing involves parsing and indexing documents, evaluating queries, and returning answers. Our focus is on effectiveness of the methods, and we do not consider efficiency in this work.

Testbeds, TREC, and *trec_eval* Formats

By using standard testbeds, researchers from different institutions can compare the performance of their systems. The Text REtrieval Conference (TREC)[11] has provided these testbeds for different IR tasks since 1992 [Harman, 1992]. Some of the search tasks have been mentioned in Section 2.3.1. TREC has also provided some other tasks such as question answering and topic distillation [Voorhees, 2004]. For the question-answering task, systems return a set of answers to a particular question rather than a large volume of documents that users are unlikely to peruse. In topic distillation task, systems return a list of links to homepages that are the key pages to a particular topic and that provide a better overview of a particular topic.

There are also other organisations providing testbeds such as the Cross-Language Evaluation Forum (CLEF),[12] which deals mainly with cross-lingual retrieval and also monolingual retrieval for a range of languages, principally European languages.

[11] `http://trec.nist.gov/`

[12] `http://www.clef-campaign.org/`

```
<DOC>
<DOCNO>FT911-5</DOCNO>
<PROFILE>_AN-BEOA7AAGFT</PROFILE>
<DATE>910514 </DATE>
<HEADLINE>
FT 14 MAY 91 / World News in Brief:  Newspaper pays up
</HEADLINE>
<TEXT>
A Malaysian English-language newspaper agreed to pay former Singapore prime
minister Lee Kuan Yew Dollars 100,000 over allegations of corruption.
</TEXT>
<PUB>The Financial Times</PUB> <PAGE>
International Page 1
</PAGE>
</DOC>
```

Figure 2.2: An example TREC document taken from Financial Times collection from TREC [Voorhees and Harman, 1997].

An IR testbed consists of three components. We describe first the component of a testbed and later the standardised TREC format.

An IR document collection. For research purposes, a static document collection is used.

Figure 2.2 shows an example of document taken from the TREC Financial Times collection. This collection was introduced from TREC 6 [Voorhees and Harman, 1997]. The documents are tagged using the SGML format. The essential tags are the <DOC> tags to indicate the start of the document, </DOC> tags to indicate ends of documents, and the <DOCNO> and the </DOCNO> tags to delimit document identifiers. The texts between the <TEXT> and </TEXT> are the contents of the documents and have to be present.[13] Other tags are optional depending on system requirements.

The ad hoc tasks for TREC have been well established with well-known collections including newswire data from sources such as Wall Street Journal and Associated Press

[13]The <TEXT> and </TEXT> tags themselves are optional.

```
<top>
<num> Number:  405
<title> cosmic events
<desc> Description:
What unexpected or unexplained cosmic events or
celestial phenomena, such as radiation and
supernova outbursts or new comets, have been detected?
<narr> Narrative:
New theories or new interpretations concerning
known celestial objects made as a result of new
technology are not relevant.
</top>
```

Figure 2.3: An example TREC query number 405 for the ad hoc task from TREC 8 [Voorhees and Harman, 1999].

[Voorhees and Harman, 1999] and collection of data crawled from the Web, such as `WT10g` [Voorhees and Harman, 2000], are also widely used.

Queries. The TREC queries are also formatted with SGML mark-up, following the formats as shown in Figure 2.3. The tags for the TREC queries are <`title`>, <`desc`>, and <`narr`> that describe three components of a TREC query — title, description, and narrative. Users may choose one or more of these three components as a query. The most common component to be used for querying is the <title> section that reflects what typical users might enter as their query. The <`num`> tags are used to indicate query identifiers, while the <`top`> and <`/top`> tags are used as query delimiters.

Relevance judgements. Relevance judgements are required to indicate whether each document in the collection is relevant for that query. This allows us to apply the evaluation measures described in Section 2.3.5. Where large collections make it impossible to judge every document, a pooling method is used [Voorhees and Harman, 1999]. The pool is created by collecting the top n results for each query, where n is usually 100, from each system participating in a TREC track. This pool of documents is then passed to human assessors for relevance judgement. This judgement is binary, 1 for relevant

and 0 for not. Documents not in the pool and hence not judged are considered as not relevant. To prevent biased assessment, these pooled documents are sorted by using the document identifiers. In this way, the human assessors cannot know in advance which documents are ranked at the top by a certain system and how many systems consider a particular document as relevant to a query.

Sanderson and Zobel [2005] state that an IR system can benefit more from shallower pools such using the top 10 results of each query than from deeper pools such as the standard of using the top 100 results. Judging 50 topics using a pool of depth 10 and judging 10 topics using a pool of depth 100 take the same amount of effort. Since the density of relevant documents ranked at the top is higher than the density of relevant documents ranked at the bottom, given the same amount of time, more relevant documents can be assessed using a shallower pool than using a deeper pool [Sanderson and Zobel, 2005]. However, using a shallow pool may disadvantage new systems that pick up relevant documents that have not been assessed, hence deem not relevant. The stability and error rates of a system using a shallow pool requires further study.

Since there is no publicly available corpus used for for Indonesian text retrieval, we build our own collection conforming to the TREC format in Chapter 4.

Statistical Significance Tests

When the performance of system A is higher than the performance of system B, it does not necessarily mean that system A is in fact better than system B [Zobel, 1998]. Statistical significance tests are required to see whether the performance of the two systems is indeed different, and how confident we can be about any difference. Statistical significance testing is used to show that system A is indeed better that system B, and the differences are not due to chance, by estimation of errors averaged over all queries [Hull, 1993]. Although there have been attempts to calculate error rates empirically to compare the performance of different systems, as done by Voorhees and Buckley [2002] from TREC 3 to TREC 2001, this calculation is not valid for future runs.

A paired t-test is used to measure the magnitude of difference between two methods and compare it with standard variance of difference [Hull, 1993]. If the difference is larger than the standard variance then a system is considered as better than another system. A paired t-test assumes the data to be normally distributed, which might not be the case for IR data. We choose the Wilcoxon signed ranked test because it is empirically more reliable [Zobel, 1998]

and does not need to assume that the data is normally distributed, since it is a non-parametric test. The Wilcoxon signed ranked test is also more powerful than the sign test because it considers both the magnitude and direction of difference between the paired data [Daniel, 1990, pages 38–42; Hull, 1993].

Because of the binary nature of the stemming data, we choose the McNemar one-tailed test [Sheskin, 1997, pages 315–327] to compare the accuracy of various stemming algorithms against the baseline. The data is binary because the result of stemming can only be the same as (correct) or different from (incorrect) the baseline stems.

We use † to indicate a particular result is statistically significant when $p < 0.05$ (95% confidence level) compared to the baseline.

In this section, we have discussed different aspects of information retrieval (IR) and the experiments involved. In the next section, we discuss a more specific aspect of IR — cross-lingual information retrieval (CLIR).

2.4 Cross-Lingual Information Retrieval

As the Web has matured, the proportion of non-English documents has increased. In 1996, there were an estimated 40 million English-speaking Internet users but only 10 million non-English-speaking users. In 2005, of an estimated 1.12 billion Internet users, 820 million were not native English speakers. Despite this growth, around two-thirds of accessible pages are in English.[14]

These language barriers can be reduced through cross-lingual information retrieval (CLIR), where users can enter a query in one language, and receive answer documents in other languages, for possible later translation. Many users who are able to read a language may not be sufficiently fluent to use it to express a query, and even fluent users would rather pose a single query to a multilingual collection than multiple queries to disjoint collections. Oard and Dorr [1996] add that it is impractical to form queries in each language given the number of languages available on the Web, and note that documents could contain words or phrases from other languages such as technical terms, quotations, and citations of publications.

Cross-lingual information retrieval (CLIR) is a subset of multilingual information retrieval (MLIR) [Hull and Grefenstette, 1996]. Multilingual information retrieval covers broad topics including IR for non-English languages, IR for parallel corpora where the query can only be in one language, IR for monolingual or multilingual corpora where the queries can be in any

[14] `http://global-reach.biz/globstats/refs.php3`

languages, and IR for multilingual documents where a document contains more than one language. As long as the IR system allows users to enter a query in a language different from the language of the document collection, it can be categorised as a CLIR system.

CLIR as a research area has attracted significant interest. From 1996, the Association for Computing Machinery Special Interest Group on Information Retrieval (ACM SIGIR) started to be involved with CLIR with papers such as that of Hull and Grefenstette [1996] and Sheridan and Ballerini [1996]. In the same year, TREC also started to host the CLIR track for English and other languages including German, French, Spanish, and Dutch [Voorhees and Harman, 1997]. In 1999, the National Institute of Informatics (NII) started to conduct NTCIR (NII Test Collection for IR Systems),[15] a TREC-like workshop for Japanese CLIR. The NTCIR workshop was later expanded to other languages such as Chinese and Korean. In 2000, the Cross-Language Evaluation Forum (CLEF) started to provide IR and CLIR testbeds for European languages. Later, CLEF expanded to include CLIR tasks for non-European languages such as Amharic, Hindi, Telugu, and Indonesian. Although CLEF does not provide Indonesian queries or an Indonesian corpus. Adriani and Wahyu [2005] translated the original CLEF English queries into Indonesian, and translated these Indonesian queries back to English. The precision of retrieving English documents using original English queries were compared against retrieval of the documents using the doubly translated queries.

We cover different aspects of CLIR briefly in the following section as we do not investigate CLIR in general, but focus on techniques to identify parallel corpora in Chapter 5 and Chapter 6. Detailed descriptions of CLIR can be obtained elsewhere [Hull and Grefenstette, 1996; Oard and Dorr, 1996; Pirkola et al., 2001].

2.4.1 Similarities and differences with monolingual IR

With the exception of IR techniques that are language-dependent, such as stemming or defining word boundaries for indexing, most of the concepts explained in Section 2.3 for information retrieval are also applicable for cross-lingual information retrieval (CLIR). However, applicability does not mean that there is no need for modification. For example, stopping may help increase precision, but the stopwords are likely to be different. For example, the OKAPI BM25 similarity measure might be useful, but the parameters such as k_1, k_3, and b might differ.

[15] `http://research.nii.ac.jp/ntcir`

Recall and precision values for monolingual information retrieval are usually used as the benchmark for CLIR performance [Hull and Grefenstette, 1996]. The maximum performance for a CLIR system is expected to be close to that of a monolingual system.

The major difference between a monolingual IR and a CLIR system is in the testbeds used. In a monolingual IR system, the queries and the documents are in the same language, whereas in a CLIR system they are in different languages. The process of making relevance judgements is therefore different. One possibility is to judge documents in language A using queries in another language B. For example, we could judge documents in English using queries in Indonesian. Since this method requires the assessor to understand both languages equally well, which in practice is not very common, most IR researchers, including the NTCIR [Chen et al., 1999; Sato et al., 1999], opt to use a parallel corpus and to translate either the queries or the documents.

In a parallel corpus, for each document in one language there is a manual translation of the document in the other language. Documents are parallel if one is a translation of the other [Sadat et al., 2002]. This parallel corpus can be used as basic building block for bi-directional testbeds; relevance judgements in one language can be used for the other, and queries in either language,[16] can be used for experiments in monolingual or cross-lingual retrieval.

The alternative to use a parallel corpus is to translate either the documents or the queries; each approach has its own advantages and shortcomings [Hull and Grefenstette, 1996]. Translating documents provides more context, so that it is easier to remove translation ambiguity, which often occurs for short queries consisting only one or two words. However, with large numbers of documents, translation of documents takes a lot of storage space and processing time. It is more efficient to translate the queries; this step can easily be added to an existing IR system. The shortcoming of translating queries is translation diasambiguity, an inherent problem with any automatic translation. Most research concerns query translation as it is more practical.

2.4.2 Translation techniques

We discuss various translation techniques that can be applied to either documents or queries. For ease of discussion, we focus on translating queries instead of documents, as this is more common in the research literature.

[16] Query translation is required since the queries are originally only in one language.

Manual translation generally produces the best results and is used to measure the performance of machine translation [Papineni et al., 2001]. Humans can understand the contexts of a word to be translated, and are more likely than machines to identify the correct translation. However, manual translation is not practical as it is time consuming and costly.

A more common and practical approach is to use automated translation. These automated translations include machine translation, translation using bilingual thesauri, and translation using information derived from parallel corpora [Hull and Grefenstette, 1996].

Machine translation is the simplest form of translation. Here, a system accepts words in one language and produces the translations in another. Systran,[17] Toggletext,[18] and the Google translation tool[19] are examples of machine translation systems. These tools are not always reliable [Fluhr, 1995]. For example, Alfred Lord Tennyson's quote "`A lie which is half a truth is ever the blackest of lies`" when translated into French using the Google translation tool becomes "`Un mensonge qui est moitié d'une vérité est jamais le plus noir des mensonges`". When the French phrase is translated back into English, the quote becomes
"`A lie which is half of a truth is never the blackiest lies`"
at the first time and
"`A lie which is half of a truth is never blackest of the lies`"
at the second and subsequent translations; there translations have the opposite intended meaning.[20] The problem of ambiguities in machine translation is unavoidable as we are dealing with human languages [Slocum, 1984]. Even the best human translator may not fully understand the content of a particular document, for example when it contains advanced technical or professional jargon. Moreover, a human translator uses syntactic and lexical understanding that is hard to incorporate into a machine. Machine translation works best only on short queries [Oard and Dorr, 1996] and on specific domains [Fluhr, 1995].

Another translation method is by using bilingual thesauri or bilingual dictionaries [Hull and Grefenstette, 1996]. Oard and Dorr [1996] describe a bilingual dictionary as an ontology that defines a word in one language by another word or words in another language, that is, it is a word replacement translation. A vector translation dictionary is effectively a lookup table; a word in one language is replaced by words in another language [Hull and

[17] `www.systransoft.com/index.html`

[18] `www.toggletext.com/kataku_trial.php`. Toggletext specialises in translating from Indonesian to English and vice versa.

[19] `www.google.com/language_tools`

[20] All these translations were done on 5th September 2006.

Grefenstette, 1996]. Such a dictionary is also referred to as *bilingual transfer dictionary* or *bilingual thesaurus*. Since in bilingual thesauri the relationships between words are easily understandable by humans and they are often domain specific, users can formulate better queries. However, these very features also have drawbacks. Translation using thesauri is limited by the broad nature of most thesauri, with few technical terms. An additional limitation is that the thesauri may produce different meanings of a word without the translation probability, therefore it is not known which is the most likely translation. Furthermore, converting a dictionary meant to be used by humans into a bilingual thesaurus is not trivial, since a general dictionary is more verbose [Hull and Grefenstette, 1996].

Translation dictionaries can also be built from parallel corpora [Hull and Grefenstette, 1996; Resnik and Smith, 2003]. With the growth in the numbers of pages in minor languages on the Web, for which manual construction of a machine translation system or a bilingual thesaurus is a prohibitive cost, translation dictionaries built from parallel corpora are invaluable. However, building such dictionaries requires huge training sets and statistical models to determine translation probabilities [Hull and Grefenstette, 1996]. Major work in this area was done by Brown et al. [1991; 1993], although their work focused on machine translation rather than CLIR. Hull and Grefenstette [1996] add that these translation probabilities generated may be too specific to certain domains, as there are not many parallel corpora on general topics.

Unlike machine translation, translation using parallel corpora can give several meanings of a word together with the probability of each translation, which is beneficial for query translation [Nie and Chen, 2002]. For example, the word “lucu” in Indonesian could mean either “funny” or “cute”. An automatic machine translation tool would normally provide only one of the meanings, which might not be appropriate based on the context. Translation using parallel corpora can use the co-occurrence of words or phrases in parallel documents, allowing either the most appropriate meaning to be used, or both meanings to be used to increase recall. The more often two words or phrases occur together between parallel documents, the more likely they are translation of each other. Statistical translation using parallel corpora can guide such selection. Query translation using parallel corpora is similar to traditional IR search using query expansion [Kraaij et al., 2003]. Query translation using parallel corpora includes all possible translations that are semantically closely related to the query word.

Existing parallel corpora tend to be limited to legal collections such as EU legislation and Hansard records of Canadian parliamentary proceedings, or religious texts such as the

Qur'an and the Bible. For that reason, we want to build a system that can identify parallel corpora automatically based merely on the content, rather than the structure or location of documents. This automatic identification allows us to build parallel corpora in different domains and therefore broaden the contexts of statistical translation. Such techniques are discussed in Chapter 5 and Chapter 6.

There are other translation problems inherent in dictionary-based translation, as outlined by Pirkola et al. [2001]. These problems include the existence of proper nouns, words with different spellings, compound words, phrases, and terms that are domain-specific. We do not discuss these problems further in this thesis, as we focus on identification of parallel corpora.

2.5 Summary

In this chapter, we have presented background information about the history, characteristics, and morphology of the Indonesian language. We have also outlined general techniques involved in information retrieval and cross-lingual information retrieval.

In Section 2.1 we gave a brief history of the Indonesian language, together with its similarities and differences with English. Indonesian uses the Roman alphabet and capitalises letters at the beginning of sentences, names, and letters in acronyms. Like English, Indonesian does not have word genders. However, Indonesian does not have any tenses or articles. As opposed to English, in which adjectives are before a noun, in Indonesian nouns appear before their adjectives. The Indonesian numbering system differs from English in terms of usage of a full stop or a comma in separating numbers or decimals. Indonesian uses repeated words to indicate plurals. There are some other features that are unique but not necessarily comparable to English, for example, the use of negation and superlative forms.

In Section 2.2, we provided a brief description of Indonesian morphology. Indonesian has complex affixes that include prefixes, suffixes, infixes, confixes, repeated forms, and the combinations of all these affixes. Suffixes do not change the forms of root words they are attached to, but infixes and some prefixes do. Some prefixes may change depending on the first letters or syllable of the root word, and they may also alter the first letter of the root word. Certain pairs of prefixes and suffixes can form a confix if and only if the root word has to be appended with both a prefix and a suffix to have meaning, and where adding either one would not produce a meaningful word. Otherwise, those pairs form combinations. Combination could consist of multiple prefixes, an infix, and multiple suffixes. Repeated

words have other functions besides indicating plurals, and affixes can also be added to the repeated words.

In Section 2.3, we described different techniques involved in implementing and evaluating information retrieval systems. The primary steps in an IR system can be considered to be as parsing, indexing, and query evaluation. Parsing in IR means choosing which terms in the documents are to be indexed. The terms to be indexed can be case-folded, stopped, and stemmed. Stemming can be done using language knowledge or tokenisation. Indexing allows faster querying time; the most common indexing approach is to use an inverted file.

Query evaluation methods can be divided into Boolean and ranked evaluation. The most common ranked evaluation methods are the vector space and the probabilistic models. We introduced recall and precision for use in measuring the effectiveness of an IR system, and the mean reciprocal rank and separation values for use in measuring effectiveness of identifying parallel documents. We also described the TREC standard testbeds for IR. We explained the Wilcoxon signed ranked test for statistical analysis of IR systems, and the McNemar test for statistical analysis of stemming algorithms.

In Section 2.4, we considered aspects of cross-lingual information retrieval. The difference between a monolingual IR and a CLIR system is in the language of the query and the documents. In a monolingual system, they are in the same languages, whereas in CLIR they are in different languages. Most techniques discussed for IR in general are applicable to CLIR, although some techniques such as stemming need to be adapted to the different languages. A testbed consisting of queries and documents in different languages can be formed by judging the documents despite the language difference; by using parallel corpora as the basic building block for bi-directional testbeds; or by translating either the queries or the documents. Query translation may have more ambiguities than document translation but is preferable as it is more efficient. The translation can be done manually or automated using machine translation, bilingual thesauri, or parallel corpora. We focus on automatic parallel corpora identification as parallel corpora are very useful for both building CLIR testbeds and performing translations.

Chapter 3

Stemming Indonesian

Stemming is a core natural language processing technique for efficient and effective information retrieval [Frakes, 1992], and one that is widely accepted by users. It is used to transform word variants to their common root by applying — in most cases — morphological rules. For example, in text search, it should permit a user searching using the query term "stemming" to find documents that contain the terms "stemmer" and "stems" because all share the common root word "stem". 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 of terms 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.

The actual effect is language dependent [Pirkola et al., 2001]. For languages such as English [Hull, 1996], French [Savoy, 1999], Slovene [Popovič and Willett, 1992], and Arabic [Larkey et al., 2002], stemming increases precision. Popovič and Willett claim that languages that are morphologically complex such as Slovene is more likely to benefit from stemming. For the same reason, we suspect that Indonesian might benefit as well.

For the English language, stemming is well-understood, with techniques such as those of Lovins [1968] and Porter [1980] in widespread use. However, 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], Malay [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. Indeed, for Indonesian the schemes are nei-

ther easily accessible nor well-explored. There are no comparative studies that consider the relative effectiveness of alternative stemming approaches for this language.

As discussed in Section 2.2, Indonesian affixes are complex — they include prefixes, suffixes, infixes (insertions), confixes, repeated forms and combinations of these affixes. These affixes must be removed to transform a word to its root word, and the application and order of the rules used to perform this process requires careful consideration. Consider a simple example: the word "minuman" ⟨a drink⟩ has the root "minum" ⟨"to drink"⟩ and the suffix "-an". However, many examples do not share the simple suffix approach used by English:

- "kemilau" ⟨shiny⟩ is derived from the root "kilau" ⟨to shine⟩ through the process of inserting the infix "em" between the "k-" and "-ilau" of "kilau".
- "menyimpan" ⟨to store⟩ is derived from the root word "simpan"⟨to store⟩ with the prefix "me-"
- "buku-buku" (books) is the plural of "buku" ("book")

We cater only for native Indonesian affixes, we do not consider foreign affixes such as "pro-" ⟨pro-⟩ and "anti-" ⟨anti-⟩ that form words with meanings of their own that can be completely different from the original.

Several techniques have been proposed for stemming Indonesian. We evaluate these techniques through a user study, where we compare the performance of the scheme to the results of manual stemming by four native speakers. Our results show that an existing technique, proposed by Nazief and Adriani [1996] in an unpublished technical report, correctly stems around 93% of all word occurrences (or 92% of unique words). After classifying the failure cases, and adding our own rules to address these limitations, we show that this can be improved to a level of 95% for both unique and all word occurrences. We hypothesise that adding a more complete dictionary of root words would improve these results even further. We conclude that our modified Nazief and Adriani stemmer, the CS stemmer, should be used in practice for stemming Indonesian.

The remainder of this chapter is structured as follows. We report on problems faced by stemming algorithms in general, and in Indonesian specifically, in Section 3.1. Different approaches for stemming Indonesian words are presented in Section 3.2. The experimental framework used to test stemming approaches is explained in Section 3.3. Results and discussion of the existing techniques are presented in Section 3.4. Section 3.5 shows the extension to the Nazief and Adriani stemmer to address some of the problems. A summary is presented in Section 3.6.

3.1 Stemming Issues

Stemming algorithms may be hampered by several issues generic to all natural language processing (NLP) tasks, and some that are specific to the language. One of the most common problems for all NLP tasks is word-sense ambiguity [Krovetz, 1993]. This problem also occurs for homonyms, words that have the same spelling but different meanings; examples include "bank", "can", and "mean". If the words "banking", "banker", and "bankable" are conflated with the word "bank" with the meaning of "the land along a lake or a river", instead of "a financial establishment", the retrieved documents may not reflect the user's intended meaning. Non-homonym words, such as "get" and "fix", may also have different meanings depending on the context.

Another common problem for stemming is dependency on a comprehensive dictionary. Many stemming algorithms depend on a dictionary to check whether the root word has been found. If the root word has been found, the stemming process stops; otherwise the word is overstemmed, with words of different meanings being grouped to the same stem. Suppose that the word "selatan" ⟨south⟩, which is a root word, is not in the dictionary; a dictionary-based stemmer would probably wrongly stem this word to "selat"⟨strait⟩. In addition, some dictionaries may contain non-root words that in turn cause understemming, where words derived from the same root word are not stemmed to the correct root word. Overstemming and understemming can be problems in any language. Examples of overstemming and understemming in English include the words "provenance" and "proverbial", which can be stemmed erroneously to the word "prove", and the word "beautifully" stemmed to "beautiful" instead of "beauty".

Overstemming and understemming can also be caused by the stemming algorithm itself: whether it is a heavy or light stemmer [Paice, 1994; 1996]. A heavy stemmer is a stemmer that removes as many affixes as possible, tending towards overstemming, while a light stemmer is a stemmer that tries to remove as few affixes as possible, tending towards understemming.

Indonesian has stemming problems that are specific to the language. One of the problems is having different types of affixes, another is having some prefixes that change according to the first letters of the root words as explained in Section 2.2.3. For example, the prefix "me-" becomes "mem-" when attached to a root word starting with the letter "b-" as in "*mem*-buat" ⟨to make⟩, but it becomes "meny-" when attached to a root word starting with the letter "s-" as in "*meny*-[s]impan"[1] ⟨to store⟩. Furthermore, since there can be more than one

[1]The letter "s" is removed when the root word is attached to the prefix "meny-".

affix attached to a word, the order of removal is important, otherwise the resultant root word may not be what is expected. For example, the word "*di*-beri-*kan*" ⟨to be given⟩ is derived from the word "beri" ⟨to give⟩. If we remove the suffix "-kan" first before the prefix "di-", we get the correct stem. However, if the stemming algorithm attempts to remove prefixes before suffixes, the resultant root word becomes "ikan" ⟨fish⟩ (after removing valid prefixes "di-" and "ber-") which is a valid word but not the correct root word.

3.2 Stemming Algorithms

In this section, we describe the five schemes we have evaluated for Indonesian stemming. In particular, we detail the approach of Nazief and Adriani [1996], which performs the best in our evaluation of all approaches in Section 3.4. We propose extensions to this approach in Section 3.5.

With the exception of the S_V algorithm of Section 3.2.3, all the algorithms described here use the University of Indonesia dictionary described by Nazief and Adriani [1996]; we refer to this dictionary as DICT-UI. We use this dictionary since it has quite a reasonable number of root words, a total of 29 337.[2]

3.2.1 Nazief and Adriani's Algorithm

The stemming scheme of Nazief and Adriani [1996] is described in an unpublished technical report from the University of Indonesia. In this section, we describe the steps of the algorithm, and illustrate each with examples. We refer to this approach as S_NA.

The algorithm is based on comprehensive morphological rules that group together and encapsulate allowed and disallowed affixes, including prefixes, suffixes, and *confixes* (combination of prefixes and suffixes), which are also known as circumfixes.[3] As explained in section 2.2, affixes can be *inflectional* or *derivational* [Payne, 1997]. This classification of affixes leads to the rules:

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

[2]This number is reasonable as it is comparable to the size of dictionary used by other languages. For example, the Malaysian stemming algorithm by Ahmad et al. [1996] uses a dictionary of 22 293 root words and the Spanish derivative stemming algorithm by Figuerola et al. [2002] uses 15 000 root words.

[3]Not all prefix and suffix combinations form a confix [Moeliono and Dardjowidjojo, 1988, pages 81–82], but we choose to treat them as such during stemming.

where DP is *derivational prefix*; DS is *derivational suffix*; PP is *possessive pronoun*; and P is *particle* (both PP and P are inflectional affixes).

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.

2. In practice, affixes are never repeated, so a stemmer should remove only one of a set of repeating affixes.

3. We can use confix restriction during stemming to rule out invalid affix combinations. The list of invalid affix combinations are listed in Table 2.4 in Section 2.2.4.

4. If characters are being restored after prefix removal, we perform recoding if necessary. We explain this in Step 5 of the next section.

Detailed approach

We now describe S_NA 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 in Section 2.2, 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", or "-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 is present in the dictionary, so stemming stops.

   According to our affix model, this leaves 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 in the next step.

4. Remove any derivational prefixes {"be-", "di-", "ke-", "me-", "pe-", "se-", and "te-"}:[4]

   (a) Stop processing if:

   - the identified prefix forms an invalid affix pair with a suffix that was removed in Step 3; the invalid pairs are listed in Table 2.4;
   - 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-"} must be further disambiguated using the rules described in Table 3.1 because these have different variants. The prefix "me-" could become "mem-", "men-", "meny-", or "meng-" depending on the letters at the beginning of the root word.[5]

   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.

   For the word "mempertinggi" ⟨to heighten⟩, we remove the prefix "mem-" to obtain the word "pertinggi" ⟨to heighten⟩.

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

[4] In Section 2.2.3, the prefixes "pe-" and "per-" are considered different prefixes and the prefixes "be-" and "te-" are listed as "ber-" and "ter-"; here we follow the description by S_NA.

[5] Based on Table 3.1, the number of letter to be considered is up to five.

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_2...	be-C_1erC_2 ... where C_1!={'r'| 'l'}
6	terV...	ter-V... | te-rV...
7	terCerV...	ter-CerV... where C!='r'
8	terCP...	ter-CP... where C!='r' and P!='er'
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{r|l}...	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}...	meng-{g|h|q}...
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...
23	perCAP...	per-CAP... where C!='r' and P!='er'
24	perCAerV...	per-CAerV... where C!='r'
25	pem{b|f|v}...	pem-{b|f|v}...
26	pem{rV|V}...	pe-m{rV|V}... | pe-p{rV|V}...
27	pen{c|d|j|z}...	pen-{c|d|j|z}...
28	penV...	pe-nV... | pe-tV...
29	peng{g|h|q}...	peng-{g|h|q}...
30	pengV...	peng-V... | peng-kV...
31	penyV...	peny-sV...
32	pelV...	pe-lV... Exception: for "pelajar", return ajar
33	peCerV...	per-erV... where C!={r|w|y|l|m|n}
34	peCP...	pe-CP... where C!={r|w|y|l|m|n} and P!='er'

Table 3.1: 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 'P' indicates a short fragment of a word 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 Section 2.2.3.

(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. After the recursive prefix removal, the word "pertinggi" becomes the correct stem "tinggi" ⟨high⟩ that is found in the dictionary after removal of the prefix "per-".

If the word is not found after recursive prefix removal and the three conditions in 4a are not violated yet, proceed to the next step. 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-", we obtain "angkap", which is not a valid root word. Further recursive prefix removal does not succeed since there is no other valid prefix to be removed.

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 3.1, which 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 a lowercase letter following the hyphen and outside the braces. Not all prefixes have a recoding character.

   From the example "menangkap" above, there are two possible recoding characters based on Rule 15, "n" (as in "men-*n*V...") and "t" (as in "men-*t*V..."). This is somewhat exceptional; in most cases there is only one recoding character. 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 the confixes are not explicitly removed in the above steps, they are indirectly removed by the removal of prefixes and suffixes. There may be some exception cases. For example, the confix "pe-an" in the word "pengusutan" could mean either "entanglement", which is derived from "kusut" ⟨tangled⟩ or "examination, investigation", which is derived from "usut" ⟨examine⟩. Without using context, neither an automatic stemmer or humans can tell which is the correct stem.

The following section describes a feature unique to the S_NA stemmer.

Prefix disambiguation

When we encounter a complex prefix, we determine the prefix limits according to the rules shown in Table 3.1. Consider the word "menangkap". Looking at the rules for the prefix letters "me-", we exclude Rule 10, 11, 12, and 13 since the third letter of our word is "n" instead of "l", or "r", or "w", or "y", or "m", and also exclude Rule 14 since the fourth letter "a" is not "c", "d", "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" 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*-[k]aku" with the root "kaku" ⟨stiff⟩.[6] Both are valid root words and we can only determine the correct root word from the context. The same ambiguity can also occur for a word that can be a stem or an affixed word. The word "mereka" can be a stem, which means "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, the context surrounding the words is required. This is beyond the scope of this thesis, which focuses on stemming on a word by word basis.

3.2.2 Arifin and Setiono's Algorithm

Arifin and Setiono [2002] propose a less complex scheme than that of Nazief and Adriani, but one that follows a similar approach of using a dictionary, progressively removing affixes, and dealing with recoding. We refer to this approach as S_AS. Their approach attempts to remove up to two prefixes first and then remove 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.

The particular advantage of this approach is that, if the word cannot be found after the removal of all affixes, the algorithm then tries to restore all combinations of removed affixes. For example, the word "kesendirianmu" ⟨your solitude⟩ has the prefix "ke-" and the suffixes

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

"-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 process would have stepped through "kesendirian" ⟨solitude⟩ to "sendirian" ⟨being alone⟩ (both are valid words but not the root word).

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 pre-recoding 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 "-an" is 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 prefix as "peny-", and so removes the letters "ny" to obtain "endiri".[7] Adding the recoding letter "s-" results in "sendiri"; this appears in the root word dictionary, so the operation ends. The recoding rules used by all stemming algorithms in this chapter follow the standard recoding rules specified by Moeliono and Dardjowidjojo [1988] for *Tata Bahasa Baku Bahasa Indonesia* ⟨A Standard Grammar of Indonesian⟩. These rules are listed in Table 3.1.

This restoration process helps avoid overstemming when some parts of the words can be mistaken as prefixes or suffixes. This is illustrated by the first letters "se" and the last letter "i" that were mistaken as a prefix and a suffix in the previous example.

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" seem to appear twice. Arifin and Setiono remove these in succession to obtain the valid word "per" ⟨spring⟩ instead of the correct root word "peran" ⟨to play the role of⟩.

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

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 combination pair "mem-" and "-kan".

The S_NA algorithm does not share these problems.

3.2.3 Vega's Algorithm

The approach of Vega [2001] is distinctly different because it does not use a dictionary; instead, it uses rule sets to determine whether affixes can be removed from a word. The rules are accessed in order. For each word to be stemmed, rules are applied that attempt to segment the word into smaller components. When one rule fails, the algorithm proceeds to the next. We refer to this approach as S_V.

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*, a combination of prefixes and suffixes. 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 combination "me-...-i".

There are four variants of this algorithm: standard, extended, iterative standard, and iterative extended. Standard deals with standard affix removal of prefixes such as "ber-", "di-", and "ke-", the suffixes "-i", "-an", and "-nya", and the infixes "-el-", "-em-", "-er-". In contrast, extended — unlike all other approaches described in this paper — deals with non-standard affixes used in informal spoken Indonesian. The iterative versions recursively stem words. In our results, we report results with only the first scheme, which we refer to as S_V-1; the other variants are ineffective, performing between 10%– 25% worse than S_V-1.[8]

A major shortcoming of the S_V 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.

3.2.4 Ahmad, Yusoff, and Sembok's Algorithm

The approach of Ahmad et al. [1996] has two distinct differences to the others: first, it was developed for the closely-related Malay language, rather than Indonesian; and, second, it does not progressively apply rules (we explain this next). We could have adapted the scheme to Indonesian: the sets of affixes are different between Indonesian and Malay, some rules are not applied in Indonesian, and some rules applicable to Indonesian are not used in Malay. However, it unclear how much improvement is possible with additional work. Therefore, we use the original algorithm first to see the baseline performance and to check whether the effort of adapting the rules is justified.

The algorithm uses a root word dictionary and a list of valid affix combinations in the form of templates. Ahmad et al. [1996] say 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 use the University of Indonesia dictionary, DICT-UI described earlier. 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, and, where possible, affixes are removed. If all matching templates are exhausted without a successful dictionary lookup, the original word is returned

[8] All percentage differences given in this chapter are absolute percentage points.

unstemmed. The advantage of not progressively applying rules is that overstemming is minimised. In addition, as in other successful approaches, the scheme supports recoding.

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 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. [1996] 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 additional 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" ⟨infrared⟩ 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: S_AYS-B_1, S_AYS-B_2, S_AYS-B_3, S_AYS-C_1, S_AYS-C_2, and S_AYS-C_3. In the S_AYS-B_1 and S_AYS-C_1 variants, the additional rules are appended to the previous rules as shown in Ahmad et al. [1996]. In the S_AYS-B_2 and S_AYS-C_2 variants the rules are ordered alphabetically without considering the affix types. In the S_AYS-B_3 and S_AYS-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; the other schemes either perform the same or at most 1% worse, in the case of AHMADA. In this paper, we describe results for the ordering (S_AYS-b_2) that we have found to perform the best.

We suspect that the better performance of S_AYS-b_2 is due to its catering for general affixes before considering more specific affixes such as those from the Qur'an and excluding modern Malaysian affixes as they are not similar to Indonesian.

Because the scheme is not progressive, its accuracy depends closely on the rule ordering as illustrated by the word "berasal" earlier, where "berasal" can be stemmed to either "basal" or the correct stem "asal" depending on whether we apply the infix or the prefix rule first.

3.2.5 Idris

Idris [2001] extends the scheme of Ahmad et al. [1996] to progressive stemming and recoding. The algorithm alternates between removal of prefixes and of 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 DICT-UI 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 checks recoding first. For our example, after removing the prefix "mem-", we obtain "asuk", which is not in the dictionary. From recoding rules shown in Table 3.1, 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 but incorrect 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 — which we call S_I-2 — performs slightly better, around 0.3%, and we only report 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 S_I 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 Malay, this algorithm uses a set of prefixes and suffixes that are slightly different from those used in Indonesian, which in turn contributes to overstemming.

3.3 Experimental Framework

To investigate the performance of stemming schemes, we have carried out a user experiment. In this, we compared the results of stemming with each of the algorithms to manual stemming by native Indonesian speakers. This section explains the collection we used and the experimental design.

3.3.1 Collection

We formed a collection of words to be stemmed for training by extracting every fifth word from a collection of 9 898 news stories from the online edition of the Kompas[9] newspaper between January and June 2002. We define a word as a sequence of characters enclosed by whitespaces, 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 the 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 non-unique words were not root words, and so we decided to omit words with fewer than six characters from our training collection. Note that by design, S_NA does not stem words shorter than three characters; this is an orthogonal issue to the collection creation process.

We obtained 1 807 unique words forming a final collection of 3 986 non-unique words, reflecting a good approximation of their frequency of use. We chose to extract non-unique words to reflect the real-world stemming problem encountered in text search, document summarisation, and translation. The frequency of word occurrence in normal usage is highly skew [Williams and Zobel, 2005]; there are a small number words that are very common, and a large number of words that are used infrequently. In English, for example, "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 as mentioned in Section 2.1.5) is the most common word. It is important that an automatic stemmer processes common words correctly, even if this means that it fails on some rarer terms.

We use the training collection in two ways. First, we investigate the error rate of stemming algorithms relative to manual stemming for the non-unique word collection. This permits

[9] `http://www.kompas.com`

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 of stemming for 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.

The error rate we use is different from the method for counting error rate relative to truncation (ERRT), overstemming index (OI), and understemming index (UI) proposed by Paice [1994]. We do not use the Paice method because it requires considerable semantic knowledge of the languages and all the words in the collection are already known. In his method, all words in the collection are categorised into different concept groups prior to stemming — for example the words "converser" and "conversazione" are grouped into one group and the words "convert", "converter", "convertible", and "conversion" into another group. After stemming, the algorithm then checks whether there is any conflation error which means that a resultant stem is in the wrong group.

3.3.2 Baselines

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.[10] The words were listed in their order of occurrence, that is, repeated words were distributed across the collection and words were not grouped by prefix. Table 3.2 shows the level of agreement between the assessors: as expected, there is some disagreement as to the root words between the speakers; agreement ranges from around 93% (for speakers A and C) to less then 89% (for C and D). For example, the word "bagian" (part) is left unstemmed in some cases and stemmed to "bagi" (divide) in others.

Having established that native speakers disagree and also make errors, we decided to use the majority decision as the correct answer. Table 3.3 shows the number of cases where three and four speakers agree. All four speakers 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 speakers 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 then we keep the original word in our collection and note its answer as the majority decision.

[10] Three of the assessors are undergraduate students and the fourth is a PhD candidate.

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

Table 3.2: Results of manual stemming by four Indonesian native speakers, denoted as A to D, on the training set. The values shown are the number of cases out of 3 986 where participants agree, with the percentage indicated in parentheses.

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

Table 3.3: Consensus and majority agreement for manual stemming by four Indonesian native speakers, denoted as A to D, on the training set. The values shown are the number of cases out of 3 986 where participants agree.

We refer to this training collection as C_TR_MAJORITY; it contains 3 799 words. Words that do not have a majority stemming decision are omitted from the training collection.

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" ⟨to be⟩ was stemmed by three assessors to "ada" ⟨to exist⟩ 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. This increase of 2 words is due to cases such as "adalah" remaining unstemmed by 3 out of 4 speakers in some cases and being stemmed by 3 out of 4 to "ada" in other cases.

Stemmer	C_TR_MAJORITY		C_TR_UNIQUE		C_TR_SUBJECTIVE	
	Correct (%)	Errors (words)	Correct (%)	Errors (words)	Correct (%)	Errors (words)
S_NA	**92.8**	**272**	**92.1**	**139**	**95.0**	**119**
S_AYS-b_2	88.8	424	88.3	205	91.4	344
S_I-2	87.9	458	88.8	197	89.8	405
S_AS	87.7	466	88.0	211	90.0	397
S_V-1	66.3	1 280	69.4	536	67.7	1 286

Table 3.4: Automatic stemming performance on the training set: C_TR_MAJORITY, C_TR_UNIQUE *and* C_TR_SUBJECTIVE.

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 this training set as C_TR_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 non-unique words, and we refer this training set as C_TR_SUBJECTIVE. For example, consider a case where the word "spiritual" (spiritual) is stemmed by two speakers to "spiritual", by a third to "spirit" (spirit), and the fourth to "ritual" (ritual). In this case, if an automatic approach stems to any of the manual three stems, we deem it has correctly stemmed the word. This baseline is just used for comparison and the trends of performance across all algorithms are similar to those of C_TR_MAJORITY.

3.4 Results and Discussion

The results of automatic stemming for the training set C_TR_MAJORITY, C_TR_UNIQUE, and C_TR_SUBJECTIVE are shown in Table 3.4. The S_NA scheme produces the best results, correctly stemming 93% of word occurrences in C_TR_MAJORITY, 92% of C_TR_UNIQUE, and 95.0% of C_TR_SUBJECTIVE. For C_TR_MAJORITY, this is some 4% better than the best-performing other scheme (S_AYS-b_2), making less than two-thirds of the errors, the difference is statistically significant ($p < 0.001$, one-tailed McNemar test). There is a drop of around 4% each for C_TR_UNIQUE and C_TR_SUBJECTIVE, this difference is also statistically significant ($p < 0.001$). The remaining dictionary schemes — S_AYS-b_2, S_I-2, and S_AS —

are comparable and achieve 87%–91% on three collections. We observe that the only non-dictionary scheme, S_V-1, is less effective than even the S_I-2 and S_AYS-b_2 schemes designed for Malaysian stemming. It makes almost five times as many errors on C_TR_MAJORITY as S_NA, illustrating the importance of validating decisions using an external word source.

Interestingly, the S_I-2 approach offers no improvement to the S_AYS scheme on which it is based. On C_TR_UNIQUE, S_I-2 is 0.5% (eight words) better than S_AYS-b_2. However, on C_TR_MAJORITY, S_I-2 is 0.9% (34 words) worse than S_AYS-b_2. This illustrates an important characteristic of our experiments: stemming algorithms should be considered in the context of word occurrences and not unique words. While S_I-2 makes less errors on rare words, it makes more errors on common words, and is less effective overall for stemming document collections.

As expected, performance on C_TR_SUBJECTIVE is slightly better than for C_TR_MAJORITY or C_TR_UNIQUE, since an automated approach need only agree with a single assessor. S_AS produces slightly better results than S_I-2 for C_TR_SUBJECTIVE, but the difference is very small (0.2%).

The S_NA stemmer has addressed some of the stemming issues mentioned in Section 3.1. It tries to avoid overstemming by two methods. The first method is by checking whether which prefix has been removed so it does not remove the same prefix repeatedly. The S_NA algorithm will not stem the word "kekerasan" ⟨violence⟩ erroneously to "ras" ⟨race⟩ as it checks that the prefix "ke-" has been used so it will not remove another prefix "ke-"; instead it will produce the correct stem of "keras" ⟨hard⟩. The second method is by checking whether a certain prefix is allowed with certain suffix. Therefore, the word "senilai" ⟨to be priced at⟩ is not overstemmed to "nila" ⟨indigo⟩ but is correctly stemmed to "nilai" ⟨price, value⟩ as the algorithm checks that the prefix "se-" and the suffix "-i" cannot appear together. The S_NA algorithm has also addressed the complexity of Indonesian affixes, along with its changing affixes according to the root words and removing the first letter of the root word, by well-crafted prefix rules that closely follow Indonesian morphology, as listed in Table 3.1.

The performance of the S_NA scheme is indeed impressive and, for this reason, we focus on it in the remainder of this paper. Under the strict majority model — where only one answer is allowed — the scheme incorrectly stems less than 1 in 13 words of longer than 5 characters; in practice, when short words are included, this is an error rate of less than 1 in 21 word occurrences. However, there is still scope for improvement: even under a model where all 3 986 word occurrences are included and any answer provided by a native speaker

is deemed correct, the algorithm achieves only 95%. Considering both cases, therefore, there is scope for an at least 5% improvement in performance by eliminating failure cases and seeking to make better decisions in non-majority decision cases. We consider and propose improvements in the next section.

3.5 CS Stemmer

In this section, we discuss the reasons why the S_NA scheme works well, and what aspects of it can be improved. We present a detailed analysis of the failure cases, and propose solutions to these problems. We then present the results that incorporate these improvements, and describe our modified S_NA approach that is called confix-stripping or CS stemmer.

3.5.1 Analysis of S_NA

The performance of the S_NA approach is perhaps unsurprising: it is by far the richest approach, being based closely on the detailed morphological rules of the Indonesian language. In addition, it supports dictionary lookups and progressive stemming, allowing it to evaluate each step to test if a root word has been found and to recover from errors by restoring affixes to attempt different combinations. However, despite these features, the algorithm can still be improved.

In Table 3.5, we have classified the failures made by the S_NA scheme on the training set C_TR_MAJORITY.[11]

The two most significant faults are dictionary related: around 34% of errors are the result of non-root words being in the dictionary, causing stemming to end prematurely; and around 11% are the result of root words not being in the dictionary, causing the algorithm to backtrack unnecessarily. Hyphenated words, usually indicating plurals, contribute around 16% of the errors. Of the remaining errors, around 49 errors or 18% are related to rules and rule precedence. The remaining errors are foreign words, misspellings, acronyms, and proper nouns.

In summary, three opportunities exist to improve stemming with NAZIEF. First, a more complete and accurate root word dictionary may reduce errors. Second, features can be added to support stemming of hyphenated words. Last, new rules and adjustments to rule precedence may reduce overstemming and understemming, as well as support affixes not

[11] We classify human errors and misspellings as two separate issues. Misspellings are created when the words are written, while human errors occur when an assessor stems a word wrongly.

	Examples			
Fault Class	Original	Error	Correct	Total Cases
Non-root words in dictionary	sebagai	sebagai	bagai	92
Hyphenated words	buku-buku	buku-buku	buku	43
Incomplete dictionary	bagian	bagi	bagian	31
Misspellings	penambahanan	penambahanan	tambah	21
Incomplete affix rules	siapapun	siapapun	siapa	20
Overstemming	berbadan	bad	badan	19
Peoples' names	Abdullah	Abdul	Abdullah	13
Names	minimi	minim	minimi	9
Compound words	pemberitahuan	pemberitahuan	beritahu	7
Recoding ambiguity (dictionary related)	berupa	upa	rupa	7
Acronyms	pemilu	milu	pemilu	4
Recoding ambiguity (rule related)	peperangan	erang	perang	3
Understemming	mengecek	ecek	cek	1
Foreign words	mengakomodir	mengakomodir	akomodir	1
Human error	penebangan	terbang	tebang	1
Total				272

Table 3.5: Classified failure cases of the S_NA *stemmer on the training set* C_TR_MAJORITY. *The total shows the total occurrences, not the number of unique cases.*

currently catered for in the algorithm. We discuss the improvements we propose in the next section.

3.5.2 Improvements

To address the limitations of the S_NA scheme, we propose the following improvements:

1. Using a more complete dictionary — we have experimented with two other dictionaries. The discussion of the dictionary and the results are in Section 3.5.4.
2. Adding rules to deal with plurals — when plurals, such as "buku-buku" ⟨books⟩ are encountered, we propose stemming these to "buku" ⟨book⟩. However, care must be taken

New Rule	Construct	Return
35	terC_1erC_2...	ter-C_1erC_2... where C_1!='r'
36	peC_1erC_2...	pe-C_1erC_2... where C_1!={r|w|y|l|m|n}
Modified Rule	**Construct**	**Return**
12	mempe...	mem-pe...
16	meng{g|h|q|k}...	meng-{g|h|q|k}...

Table 3.6: Additional and modified template formulas for derivation prefix rules in Table 3.1

with other hyphenated words such as "bolak-balik" ⟨to and fro⟩, "berbalas-balasan" ⟨mutual action or interaction⟩ and "seolah-olah" ⟨as though⟩. For these later examples, we propose stemming the words preceding and following the hyphen separately and then, if the words have the same root word, to return the singular form. For example, in the case of "berbalas-balasan", both "berbalas" and "balasan" stem to "balas" ⟨response or answer⟩, and this is returned. In contrast, the words "bolak" and "balik" do not have the same stem, and so "bolak-balik" is returned as the stem; in this case, this is the correct action, and the approach works for many hyphenated non-plurals.

3. Adding prefixes and suffixes, and additional rules:

   (a) Adding the particle (inflection suffix) "-pun" to the list of suffixes to be stemmed. This is used in words such as "siapapun" (where the root word is "siapa" ⟨who⟩). As mentioned in Section 2.2.1, the particle "-pun" is not supposed to be attached to the root word except for conjunction; however, from observation, people often attach this particle with the root word. Therefore, we choose to deal with this common mistake.

   (b) For the prefix type "te-", we have added a new condition (Rule 35) in Table 3.6. Previously, words such as "terpercaya" ⟨the most trusted⟩ are not stemmed. By the addition of this new rule, it is correctly stemmed as "percaya" ⟨believe⟩.

   (c) For the prefix type "pe-", we have added Rule 36 in Table 3.6 so that words such as "pekerja" ⟨worker⟩ and "peserta" ⟨member⟩ are stemmed correctly to "kerja" ⟨to work⟩ and "serta" ⟨along with, as well as⟩, rather than not stemmed as the prefix is not recognised by S_NA.

   (d) For the prefix type "me-", we have modified Rule 12 so that words such as "mem-

pengaruhi" ⟨to influence⟩ can be stemmed correctly "pengaruh" ⟨influence⟩ instead of not successfully stemmed.

(e) We have modified Rule 16 for the prefix type "me-", so that the word "mengkritik" ⟨to criticise⟩ can be stemmed correctly to "kritik" ⟨critics⟩.

4. Adjusting 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 "ber-" and the root word "tingkah" ⟨behaviour⟩. However, the algorithm will remove the suffix "-kah" to obtain the word "berting", and then remove the prefix "ber-" to obtain the valid word "ting" ⟨lamp⟩. This particular problem arises only in limited cases with specific prefixes and particles. The list of rules that have been adjusted are:

   (a) If a word is prefixed with "be-" and suffixed with the inflection suffix "-lah", try to remove prefix before the suffix. This addresses problems with words such as "bermasalah" ⟨having a problem⟩ and "bersekolah" ⟨be at school⟩ to be stemmed correctly to "masalah" ⟨problem⟩ and "sekolah" ⟨school⟩ instead of the erroneous "seko" ⟨spy⟩ and "masa" ⟨time, period⟩.

   (b) If a word is prefixed with "be-" and suffixed with the derivation suffix "-an", try to remove prefix before the suffix. This solves problems with, for example, "bertahan" ⟨to hold out⟩ is stemmed correctly to "tahan" ⟨to last, to hold out⟩ instead of "tah" which is a shortened form of "entah" ⟨who knows⟩.

   (c) If a word is prefixed with "me-" and suffixed with the derivation suffix "-i", try to remove the prefix before the suffix. This solves problems with, for example, "mencapai" ⟨to reach⟩ stemmed correctly to "capai" ⟨to reach⟩ instead of "capa" ⟨game of heads or tails⟩.

   (d) If a word is prefixed with "di-" and suffixed with the derivation suffix "-i", try to remove the prefix before the suffix. This solves problems with, for example, "dimulai" ⟨to be started⟩ stemmed to "mulai" ⟨to start⟩ rather than "mula" ⟨beginning⟩.

   (e) If a word is prefixed with "pe-" and suffixed with the derivation suffix "-i", try to remove prefix before the suffix. This solves problems with, for example, "petani"

	B	C	D
A	3 822 (95%)	3 835 (96%)	3 776 (94%)
B		3 811 (95%)	3 755 (94%)
C			3 771 (94%)

Table 3.7: Results of manual stemming by four Indonesian native speakers, denoted as A to D, on the test set. The values shown are the number of cases out of 4 012 where participants agree, with the percentage indicated in parentheses.

	ABCD	ABC	ABD	ACD	BCD	Any three
Number	3 624	3 735	3 676	3 690	3 674	3 903
(%)	90.3	93.1	91.6	92.0	91.6	97.3

Table 3.8: Consensus and majority agreement for manual stemming by four Indonesian native speakers, denoted as A to D, on the test set. The values shown are the number of cases out of 4 012 where participants agree.

⟨farmer⟩ stemmed to "tani" ⟨farm, farmer⟩ rather than "petan" ⟨a game of hide and seek⟩.

(f) If a word is prefixed with "ter-" and suffixed with the derivation suffix "-i", try to remove prefix before the suffix. This solves problems with, for example, "terabai" ⟨ignored⟩ stemmed to "abai" ⟨neglectful⟩ instead of "aba" ⟨father⟩.

We present results with these improvements in Section 3.5.4. These rules address some of the stemming ambiguity issues discussed in Section 3.1.

3.5.3 Baselines

We have modified the S_NA stemmer to address some of the issues listed in Table 3.5. To allow more objective assessment, we created a new set of collections to test the improved S_NA stemmer. We extracted every seventh word, without repeating the same word used in C_TR_SUBJECTIVE, from the collection of 9 898 news stories referred in Section 3.3.1. We used the same definition of word as in the training collection and chose words longer than five characters. We obtained 4 012 words and 1 688 of them are unique.

We asked four Indonesian native speakers to stem the 4 012 words manually.[12] Only one of the assessors (assessor A) is common to the training and test collections. Similar to the training collection, speakers sometime disagree or make mistakes. Tables 3.7 and 3.8 show the levels of agreement between users. The highest level of agreement between two users is 96% while the lowest is 94%. All four speakers agree on 90.3% of the words. These levels of agreement in general are higher that those of the training collection. We use the set where at least three users agree as our first test baseline; this set consists of 3 903 words and we name this test collection set as C_TE_MAJORITY.

There are 1 653 unique words in C_TE_MAJORITY. However, since users do not stem consistently, these 1 653 words map to 1 657 stems based on the majority decision. We name this test collection consisting of 1 653 words as C_TE_UNIQUE.

The original 4 012 words, referred as C_TE_SUBJECTIVE, where a stem is considered correct if it matches at least one user's stem, is used as a third baseline.

We use these three test collections and the three training collections as the baselines to judge whether we have improved S_NA in the next section.

3.5.4 Results and Discussion

Tables 3.9 and 3.10 show the results of our improvements to the S_NA stemmer with the training and test collections. The symbol † in this chapter is used to indicate a statistically significant difference compared to the original S_NA result ($p < 0.05$). Using a different, well-curated dictionary does not guarantee an improvement: the second and third rows of Table 3.9 and the third row of Table 3.10 show the result when the 29 337 word dictionary used in developing the original S_NA approach is replaced with the 27 828 word Kamus Besar Bahasa Indonesia (KBBI) dictionary and with an online dictionary[13] of unknown size. For the training collection, the stemming accuracy using KBBI dictionary drops by 4.0% on C_TR_MAJORITY, 3.9% on C_TR_SUBJECTIVE, and 5.2% on C_TR_UNIQUE. Using the same dictionary, the accuracy drops even more for the test collections, it drops by 5.8% on C_TE_MAJORITY, 5.4% on C_TE_SUBJECTIVE, and 6.5% on C_TE_UNIQUE. We hypothesise that the drop is not merely caused by the size of the dictionary but due to three other factors: first, dictionaries often contain non-root words and, therefore, can cause stemming to stop before the root word is found; second, the dictionary is only part of the process and its improvement addresses only some of the failure cases; and, last, inclusion of new, rare words

[12] Three of the accessors have bachelor degrees and the fourth one is a PhD candidate.

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

Stemmer	C_TR_MAJORITY		C_TR_UNIQUE		C_TR_SUBJECTIVE	
	Correct (%)	Errors (words)	Correct (%)	Errors (words)	Correct (%)	Errors (words)
Original	92.8	272	92.1	139	95.2	191
(A_1) Alternative KBBI dictionary	88.8 †	426	86.9 †	229	91.3 †	348
(A_2) Alternative Online dictionary	93.8 †	236	92.5	131	94.4	225
(B) Adding repeated word rule	93.9 †	232	94.0 †	105	96.2 †	153
(C) Changing to rule precedence	93.3 †	255	92.7 †	128	95.6 †	174
(D) Adding additional affixes	93.3 †	253	92.8 †	127	95.6 †	174
(E) Combining (B) + (C) + (D)	94.8 †	196	**95.3** †	**82**	**97.0** †	**119**
(F) Combining (A_2) + (B) + (C) + (D)	**95.4** †	**173**	95.2 †	85	95.8 †	166

Table 3.9: Improvements to the NAZIEF *stemmer on the training set, measured with* C_TR_MAJORITY, C_TR_UNIQUE, *and* C_TR_SUBJECTIVE. *The symbol † is used to indicate a statistically significant difference compared to the original* S_NA *result ($p < 0.05$).*

Stemmer	C_TE_MAJORITY		C_TE_UNIQUE		C_TE_SUBJECTIVE	
	Correct (%)	Errors (words)	Correct (%)	Errors (words)	Correct (%)	Errors (words)
Original	93.4	257	91.9	134	95.1	198
(A_1) Alternative KBBI dictionary	87.6 †	484	85.4 †	242	89.7 †	412
(B) Adding repeated word rule	94.5	216	94.0 †	99	96.0	161
(C) Changing to rule precedence	93.7	244	92.4	126	95.4	185
(D) Adding additional affixes	93.5	253	92.2	130	95.2	194
(E) Combining (B) + (C) + (D)	**94.9** †	**199**	**94.7** †	**87**	**96.4** †	**144**

Table 3.10: Improvements to the NAZIEF *stemmer on the test set, measured with* C_TE_MAJORITY, C_TE_UNIQUE, *and* C_TE_SUBJECTIVE. *The symbol † is used to indicate a statistically significant difference compared to the original* S_NA *result ($p < 0.05$).*

can cause matches with incorrectly or overstemmed common words, leading to decreases in performance for some cases while still improving others. The results shown by the online dictionary are better than the results of using the original dictionary for the training collection C_TR_MAJORITY and C_TR_UNIQUE but not for C_TR_SUBJECTIVE, although the only difference that is significant is for C_TR_MAJORITY ($p < 0.001$), while the increase for C_TR_UNIQUE and decrease for C_TR_SUBJECTIVE are not significant ($p = 0.243$ and $p = 0.092$, respectively). To test our other improvements, we used the original dictionary, which we hypothesise has partly satisfied the issue of having a comprehensive dictionary as mentioned in Section 3.1. Due to the latency issues, we test the effects of the online dictionary only on the combined improvement methods. We do not use the online dictionary for the test set as the dictionary site is no longer available.[14]

The fourth (B), fifth (C), and sixth (D) rows of Table 3.9 and the third (B), fourth (C), and fifth (D) rows of Table 3.10 show the effect of including the algorithmic improvements we discussed in the previous section. The results show the accuracy gains of including only the improvement into the original version, while the second last row of Table 3.9 and the last row of Table 3.10 show the additive effect of including all three. Dealing with repeated words improves the result with C_TR_MAJORITY by 1.1%, the result with C_TR_UNIQUE by 1.9%, and the result with C_TR_SUBJECTIVE by 1.0%. A similar trend is observed for the test collection; with C_TE_MAJORITY the accuracy improves by 1.1%, with C_TE_UNIQUE by 2.1%, and with C_TE_MAJORITY by 0.9%. Adjustments to the rule precedence improve the results by 0.5%, 0.6%, and 0.4% on the three training collections, and by slightly smaller margins for the test collections, 0.3%, 0.5%, and 0.3%. Adding additional affixes improves results by 0.5% on C_TR_MAJORITY, 0.7% on C_TR_UNIQUE, and 0.4% on C_TR_SUBJECTIVE for the training set, by 0.1% on C_TE_MAJORITY, 0.3% on C_TE_UNIQUE, and 0.1% on C_TE_SUBJECTIVE for test set. The results of the training and test collections show that the combined effect of the three improvements lowers the error rate to 1 in 19 words of 5 or more characters, or an average of only 1 error every 38 words in the original Kompas collection. Overall, the CS stemmer is highly effective for stemming Indonesian words. All the differences produced by different techniques (without changing the dictionary) or by using the KBBI dictionary (without changing the techniques) for the training set C_TR_MAJORITY, C_TR_UNIQUE, and C_TR_SUBJECTIVE are statistically significant ($p = 0.003$ for changing rule precedence (D) on C_TR_UNIQUE and $p < 0.001$ for the rest). Table 3.10 shows that only the improvement

[14]The last time we accessed it successfully was in May 2005.

produced by combining three different techniques (E) on the three test collections ($p = 0.008$ for C_TE_MAJORITY, $p = 0.002$ for C_TE_UNIQUE, and $p = 0.004$ for C_TE_SUBJECTIVE) and adding the repeated word rule (B) on C_TE_UNIQUE produces results that are statistically significantly better ($p = 0.026$) than the original S_NA. Using the KBBI dictionary produces results that are statistically significantly worse than using the original dictionary for the three collections in the test set ($p < 0.001$ for all).

The last row of Table 3.9 shows the effect of combining the online dictionary with additive effects of three algorithmic improvements on the training set. The results show that using the online dictionary rather than the original dictionary with the combination of all three algorithmic improvements increases the stemming accuracy of C_TR_MAJORITY by only 0.6%, and does not improve performance for the other two collections. When compared to the original S_NA, all the differences shown in the last row are statistically significant ($p = 0.019$ for C_TR_MAJORITY and $p < 0.001$ for the rest). Given the highly skew nature of text distribution, this result is good, as it stems more non-unique words correctly. However, since the online dictionary is no longer available, we choose to use the original dictionary for subsequent experiments.

As discussed in Section 2.2.1, there is a particle "-tah" in Indonesian. This particle is excluded by S_NA. This particle is also 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]. In experiments including the particle "-tah" into the list of additional affixes, we found that not catering for this prefix actually improves effectiveness from 94.7% (with 201 errors) to 94.8% (with 196 errors) for C_TR_MAJORITY, and from 95.2% (with 85 errors) to 95.3% (with 82 errors) for C_TR_UNIQUE. The most likely reason it does not help is that the particle "-tah" is rarely used in modern Indonesian. Incorporating the particle 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⟩. Therefore, all subsequent experiments we report here, including experiments with the test set, exclude this prefix.

The CS stemmer has addressed some of the issues presented in Section 3.1 that have not been solved by the S_NA stemmer. It has addressed the issues of understemming by adding rules to deal with hyphenated words and by introducing new prefix rules and modifying some of the existing prefix rules as shown in Table 3.6. It reduces some overstemming and

	Example			Cases	
Fault Class	Original	Error	Correct	Training	Test
Non-root words in dict.	sebagai	sebagai	bagai	93	111
Incomplete dictionary	bagian	bagi	bagian	31	31
Misspellings	penambahanan	penambahanan	tambah	20	11
Peoples' names	Abdullah	Abdul	Abdullah	13	0
Recoding ambiguity	berupa	upa	rupa	9	10
Names	minimi	minim	minimi	9	4
Compound words	pemberitahuan	pemberitahuan	beritahu	7	4
Acronyms	pemilu	milu	pemilu	4	3
Understemming	mengecek	ecek	cek	5	6
Hyphenated words	masing-masing	masing-masing	masing	3	3
Foreign words	mengakomodir	mengakomodir	akomodir	1	3
Human error	penebangan	terbang	tebang	1	3
Overstemming	melangkah	lang	langkah	0	10
Total				196	199

Table 3.11: Classified failure cases of the CS *stemmer on* C_TR_MAJORITY *and* C_TE_MAJORITY*. The total shows the total occurrences, not the number of unique cases.*

rule-related ambiguity problems by adjusting rule precedence — removing the prefixes first before the suffixes.

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 the derivation suffix "-i". For example, the phrase "di sisi" ⟨at the side⟩ — with the correct stem "sisi" ⟨side⟩ — is sometimes misspelt 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.

The categories of errors created by the CS stemmer on C_TR_MAJORITY and C_TE_MAJORITY are shown in Table 3.11. We compare this table with Table 3.5 to see which errors are solved

by the CS stemmer, and whether any new errors are introduced. The CS stemmer has reduced the number of errors caused by hyphenated words to three for both collections. The errors can be categorised into two classes:

- Incomplete dictionary. For example, the word "alasan-alasan" ⟨reasons⟩ should be stemmed to "alasan" ⟨reason⟩, but is wrongly stemmed to "alas" ⟨base⟩ since "alasan" is not in the dictionary,
- Recoding. For example, the word "menimbang-nimbang" ⟨to consider⟩ should be stemmed to "timbang" ⟨to consider, to weigh⟩. However, since the recoded word "nimbang" is not in the dictionary, the word is unsuccessfully stemmed.

The CS stemmer has also removed all the errors caused by incomplete affix rules in the S_NA stemmer for C_TR_MAJORITY by additional rules mentioned in part 3 of Section 3.5.2. There are still some errors related to incomplete affixes for the test collection C_TE_MAJORITY which are caused by:

- An informal affix. For example, the word 'finalis" ⟨finalist⟩ should be stemmed to "final" ⟨final⟩. However the suffix "-is" is not listed in the formal suffix list so it is left unstemmed.
- Variation of recoding. For example, the word "mengritik" ⟨to criticise⟩ can be written as "mengkritik"; CS can stem the latter but not the former.

We categorise these errors as understemming. Adjusting the rule precedence removes all overstemming errors for the training collection C_TR_MAJORITY, on which the adjustment is based. However, it introduces three new cases of understemming errors, all three caused by the word "mengalami" explained earlier. Some overstemming errors caused by rule precedence still occur on the test collection C_TE_MAJORITY since the list of all combinations of rule precedence is not exhaustive. Since there is always a trade-off —when a rule precendence is followed, then some other words will be stemmed wrongly—and a fairly large number of words are required to get a more exhaustive rule precedence list, we choose to continue using the CS scheme for subsequent experiments.

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 stem the word "mengecek" ⟨to check⟩ properly 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 need to 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 statistical data to be collected.

Some prefixes are mutually exclusive, for example the prefix "me-" can never appear with the prefix "di-" and the prefix "ke-" can never appear with the prefix "di-". The word "mendidik" ⟨to educate⟩ is derived from the prefix "me-" and the stem "didik" ⟨to educate⟩. However, except for S_AYS which uses template rules, none of the Indonesian stemming algorithms that use a dictionary restrict the combination and order of derivational prefix removal. This could lead to overstemming. If the word "didik" is not in the dictionary, the fragments "di-" at the beginning of the word is considered as a prefix, and the final resulting stem is "dik" ⟨a younger sibling⟩. This problem is rare as all algorithms check whether a word is in the dictionary after each removal. It occurs only when the dictionary is not complete. This is a problem for any language processing tasks that rely on a dictionary, and is not unique for Indonesian.

While we deal with generic stemming, many words can 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 regardless of the context. We plan to investigate stemming further by considering the context surrounding a word. We also plan to investigate the effect of different types of dictionaries, general or domain-based, including the CICC [1994] dictionary, on the stemming algorithms.

Bacchin et al. [2005] propose a language-independent suffix stemmer that reinforces the relationship between stems and derivations using probabilistic models. While this probability model can be applied to Indonesian, it is unlikely to be effective without substantial language-specific modification due to the existence in Indonesian of prefixes, infixes, and confixes.

3.6 Summary

In this chapter, we have investigated Indonesian stemming and presented an experimental evaluation of stemmers for this language. Our results show that a successful stemmer is complex, and requires the careful combination of several features: support for complex morphological rules, progressive stemming of words, dictionary checks after each step, trial-and-error combinations of affixes, and recoding support after prefix removal.

Our evaluation of stemmers began with a user study. Using four native speakers and a newswire collection, we evaluated five automatic stemmers. Our results show that the S_NA stemmer is the most effective scheme, making less than one error in twenty-one words on newswire text. With detailed analysis of failure cases and modifications, we have improved this to less than one error in thirty-eight words. We conclude that the modified S_NA stemmer, which we call CS stemmer, is a highly effective tool. It addresses some of the stemming issues listed in Section 3.1 such as overstemming, understemming, and ambiguity. We have also discovered that the Indonesian dictionary we use, DICT-UI, is sufficiently comprehensive to deal with stemming issues that are dictionary-related.

We test this CS stemmer and other well-known text retrieval techniques on an Indonesian testbed in Chapter 4.

Chapter 4

Techniques for Indonesian Text Retrieval

An information retrieval (IR) system succeeds when a candidate answer that it provides to the user does in fact satisfy their information needs; a better system provides a higher proportion of relevant documents as part of the set of documents returned in response to a query. The measures of *recall* — the fraction of relevant items that are retrieved — and *precision* — the fraction of retrieved items that are relevant — were introduced in Section 2.3.5, and provide a quantitative indicator of the effectiveness of a system. To allow an objective comparison of alternative retrieval approaches, we require a collection of documents and example user queries for which the relevant documents are known; these documents with the queries and relevance judgements form a testbed. There is no publicly available testbed for Indonesian text retrieval, and the Indonesian document collections that do exist [Adriani, 2002; Fahmi, 2004; Tala, 2003; Vega, 2001] either do not have query topics and relevance judgements, or are not available publicly. We have constructed an Indonesian text retrieval testbed which we explain in Section 4.1.

Using this testbed, we explore different known text retrieval techniques: using only title, description, narrative, or any combination of these as a query in Section 4.2; varying the parameters for the cosine and Okapi BM25 similarity measures in Section 4.3; stopping in Section 4.4; and stemming in Section 4.5. Tokenisation or converting words into n-grams, which can be considered as a language independent form of stemming, is discussed in Section 4.6. In Section 4.7, we explore the combination of stemming and n-gram as a form of spelling correction. We hypothesise that stemming all words except proper nouns

will increase precision, and so investigate methods to identify proper nouns and experiment with combining these methods with stemming and correction of misspelling using n-grams in Section 4.8. In Sections 4.9 and 4.10, we explore some miscellaneous methods that are not standard text retrieval techniques, such as identifying the language of a document and identifying compound words. We summarise and conclude our findings in Section 4.11.

4.1 Building an Indonesian Text Retrieval Testbed

A testbed for evaluating ad hoc retrieval consists of three parts: a document collection, a list of query topics, and a set of *relevance judgements* [Voorhees and Harman, 1999]. The Text REtrieval Conference (TREC) series provides researchers with appropriate testbeds for evaluating Information Retrieval (IR) techniques for several retrieval paradigms [Harman, 1992]. The Linguistic Data Consortium (LDC) [Liberman and Cieri, 1998] also provides data collections, some of which are used by the IR community. However, they do not provide any testbed for Indonesian, so we need to build our own.

4.1.1 Building a Document Collection

A document collection for evaluating ad hoc retrieval must be static. Voorhees [2004] adds that it is better if the topics of document are diverse, to represent different natures of queries of typical users. We use a collection of newswire articles from the popular online Indonesian newspaper Kompas between January and June 2002; this is the collection we used to extract stemming data as explained in Section 3.3.1.

To allow rigorous evaluation of IR techniques for Indonesian, we separated this into two collections: one test collection consists of 3 000 articles, which contains 38 601 distinct words and is around 750 KB; and one training collection of size 16 MB, containing 6 898 articles and 68 199 distinct words, which can be used for experiments described in Section 4.4, Section 4.8, Section 4.9, and Section 4.10. We label these test and training collections C_INDO-TEST-SET and C_INDO-TRAINING-SET.

Our test collection is relatively small. For example, two collections widely used for English IR research are the Wall Street Journal collection (1987–1989) of size 276 MB with 98 732 documents, and the Associated Press newswire collection (1989) of size 254 MB with 84 678 documents [Voorhees and Harman, 1999]. Nevertheless, it is still a useful resource that can be extended with collaborative input from other researchers. The small size of our collection also allows detailed ground truth to be prepared; with TREC document collections, not every

```
<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>
```

Figure 4.1: An example Kompas newswire document from our test collection, marked up in the TREC format.

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 favour newer systems that combine and improve the retrieval techniques of old systems. Pooling may also discount actual relevant documents that have not been seen by the reviewers during the relevance judgement process, and hence lead to lower reported recall.

Following the TREC approach, we kept the data as close to the original as possible, and did not correct any faults such as spelling mistakes or incomplete sentences [Voorhees and Harman, 1999]. The documents are stored in a single file, marked up using standard TREC tags. 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. Using the TREC format allows straightforward use in the Zettair[1] search engine and other IR research tools. An example document is shown in Figure 4.1. Unless specified otherwise, we use Zettair for all experiments in this chapter.

[1] http://www.seg.rmit.edu.au/zettair. Zettair was previously named Lucy. For consistency, we use the name Zettair throughout this thesis. The version we used to index our corpus was Lucy version 0.5.4.

```
<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>
```

Figure 4.2: An example topic (left) and its English translation (right).

4.1.2 Building Queries

The next step in building a testbed is to define a set of queries or *topics* that represent user information needs. There are different formats of TREC topics from different years of the workshops [Voorhees and Harman, 1997], with recent examples containing fewer fields. We followed the final ad hoc track format from TREC-8 [Voorhees and Harman, 2000].

The ad hoc topics from TREC-8 have three major fields: title, description, and narrative. The title (encapsulated in a <title> element) is a short string that summarises the information need. The description (<desc>) is a longer, one-sentence description of the topic, and the narrative (<narr>) gives more detailed explanation that aims to completely describe the documents that are relevant to the query. The topics also have the additional <top> and </top> tags to delineate each query in a file, and a <num> element to denote the query identifier. For an IR system, a query can be made of title, description, narrative, or any combination of them.

The Kompas newswire is different in topicality and time span to the newswire collections used at TREC, and so we defined our own topics. We began by reading all the 3 000 documents in C_INDO-TEST-SET to see what topics were available. Since we have limited

resources in this research, we use only twenty 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 in our collection; for example, the query `What are the symptoms and causes of asthma?` (Topic 10). An example of an Indonesian topic and its English translation is shown in Figure 4.2. The full list of Indonesian queries with their English translation are shown in Appendix C and D.

Voorhees [2000] observes that queries having fewer than five relevant documents could lead to unstable mean average precision (MAP) — a measure that depends greatly on the ranking of relevant documents. For a query that has only one relevant document, the MAP relies heavily on how this one answer document is ranked — system A may rank the document first and system B may rank it second – resulting in MAP of 1.0 and 0.5 respectively. If a testbed contains many such queries, the resulting mean average precisions are not good indicators that system A is better than system B. Voorhees [2000] conjectures that having enough queries can offset this problem. For any IR experiments using queries, reliability of experiments depend on the number of queries. Since the number of our queries is limited to 20, it is harder to get statistically significant results and the results obtained are not necessarily reproducible with other collection sets [Sanderson and Zobel, 2005]. Having more queries such as 100 may lead to more statistically significant results. Since our intention is to explore the effects of different parameters towards effectiveness, these preliminary results can aid further research in Indonesian IR. Using larger number of queries is not within the scope of this thesis.

4.1.3 Making Relevance Judgements

The final step in constructing a testbed is to make *relevance judgements*, that is, to define which documents are relevant to the information needs expressed by each query. The relevance judgements are then used as the benchmarks to decide whether the documents deemed to be relevant by retrieval systems are indeed relevant according to a human assessor.

In TREC, candidates for relevance assessment are collected via *pooling* [Voorhees and Harman, 1997]. The drawback of pooling is that documents that are not collected are con-

sidered not relevant. Since we limited our collection to 3 000 documents (C_INDO-TEST-SET) and 20 query topics we were able to make an exhaustive tabulation of $20 \times 3\,000 = 60\,000$ relevance assessments.

Each document is marked as either relevant or not relevant to the topic (information need). We format our relevance judgment in a way suited to the *trec_eval* tool used in the TREC ad hoc task [Voorhees, 2003]; *trec_eval* is a program written by Chris Buckley and used for evaluation in the ad hoc retrieval task, where relevant documents are returned in ranked list format. This program returns various results of a run including recall, precision at 10, R-precision, and mean average precision (MAP). In this chapter, unless specified otherwise, all results in text and tables are rounded to two decimal figures while the results in graphs are rounded to three decimal figures.

Voorhees [2000] reports that recall and precision is affected not only by the system performance, but also by different characteristics of the testbed. These include whether the relevance judgements are done by the query author, and whether the judgements are done by a single judge — both conditions are met by our testbed. She argues that, although different conditions affect the actual figures of recall and precision, the relative performance between different systems remain the same. All our experiments here use the same testbed. Due to limited resources, only the author performed the relevance judgements.

4.2 Text Retrieval: Using Different Query Structures

As explained in Section 4.1.2, TREC topics have three fields: the title (<title>), the description (<desc>), and the narrative (<narr>). For the TREC ad hoc retrieval task, each field or a combination of them can be used as a query. Used individually, these fields help to show the effects of query length on recall and precision [Voorhees and Harman, 2000]. The title section represents a short query; the description is slightly longer and used to describe the topic in one sentence; and the narrative is the longest and used to define more precisely which documents are relevant. We experimented with these fields, as well as different combinations of the fields, to determine which one is the most effective grouping to be used for our subsequent experiments. We use the Okapi BM25 measure for experiments in this section (Section 4.2) as it is the default similarity measure of the Zettair search engine.

	T	D	N	TD	TN	DN	TDN
MAP	**0.46**	0.40 †	0.30 †	0.40 †	0.43	0.41	0.42
Precision at 10	**0.38**	0.31 †	0.33	0.33	0.37	0.36	0.35
R-precision	0.42	0.40	0.34	0.40	**0.44**	0.43	0.43
Recall	0.73	0.68	0.66 †	0.69 †	**0.76**	0.74	0.74

Table 4.1: The precision and recall values for the top 100 documents returned using different combinations of topic fields as queries. The letters "T", "D", and "N" are used to indicate title, description and narrative fields. Multiple letters indicate a combination of the individual fields. The symbol † is used to indicate a statistically significant difference compared to using only the title as the query ($p < 0.05$).

4.2.1 Results and Discussion

The results of this experiment are shown in Table 4.1.[2] These indicate that the combination of title and narrative gives the highest recall (0.76), highest R-precision (0.44), and the second-highest precision@10 (0.37) and the second-highest MAP (0.43). The highest MAP of 0.46 and precision@10 of 0.38 are achieved using the scheme that uses only the title as the query. In terms of R-precision, the second-highest values are achieved by the combination of the description and narrative scheme and by the combination of all three schemes as queries.

As using only the title for querying reflects what a typical user might enter during a web search [Voorhees and Harman, 2000], we choose this as our baseline for statistical significance test. The symbol † is used to indicate a statistically significant difference compared to using the title only as the query ($p < 0.05$). The results show that although the R-precision produced by the combination of title and narrative is higher than that produced by using only the title, the difference is not statistically significant ($p > 0.05$, one-sided Wilcoxon signed rank test). The only significantly worse results than the baseline are the MAP values when using only the description ($p = 0.002$), only the narrative ($p = 0.025$), and the combination of title and description ($p = 0.001$); and the precision@10 value for using only the description ($p = 0.028$); and the recall values for using only the narrative ($p = 0.029$) and the combination of title and description ($p = 0.047$).

[2] All the precision and recall values described in this chapter are obtained when we retrieve the first 100 documents, unless specified otherwise.

Using only the title itself without any combination produces better precision and recall values than only description or only narrative. This is perhaps because the titles are more likely to contain keywords or phrases that describe the user's information need [Brown and Chong, 1997]. In contrast, the description and narrative are more verbose, and, in the case of the narrative, may contain descriptions of documents that are not relevant. For example, the TREC-8 query number 405 for the ad hoc task is `cosmic events` [Voorhees and Harman, 1999]. The narrative of the query is

`New theories or new interpretations concerning known celestial objects made as a result of new technology are not relevant.`

Feeding only this narrative part of the query into an IR system may not get the desired results as the documents returned are likely to contain

`celestial objects made as a result of new technology`

since they are used in the keywords. Nevertheless, combining the title with the narrative — with or without the description — improves recall and R-precision values; this may be due to more keywords being used as part of the query. Adding more keywords is similar to expanding a query [Harman, 1988].

In a study by Jansen and Spink [2006], $20\% - 29\%$ of queries received by the Excite and AltaVista (US-based) search engines, and $25\% - 35\%$ of queries received by the Fireball, BWIE, and AlltheWeb.com (European-based) web search engines consist of only one term. In another study by Jansen et al. [2000], average query length is 2.21; queries between 1 to 3 words in length make up 80% of total queries, while queries with more than 6 words represent less than 4% of all queries. The length of our Indonesian titles of query topics is between 2 and 8 words, with the mean of 4.3 and median of 4. Since our average query length is longer than typical query length and we want to emulate typical user web search behaviour, we choose to use only titles as queries in all subsequent experiments.

4.3 Text Retrieval: Varying Ranking Parameters

As explained in Section 2.3.4, there are two well-known similarity measures: cosine and BM25 measures. They have a few parameters that can be adjusted to optimise ranking for a collection [Chowdhury et al., 2002]. These parameters are pivot p for the cosine measure, and tuning constants for document length b, document term frequency k_1, and query term frequency k_3 for the BM25 measure. The role of each of these parameters and the effects of these measures on retrieval effectiveness are explained below.

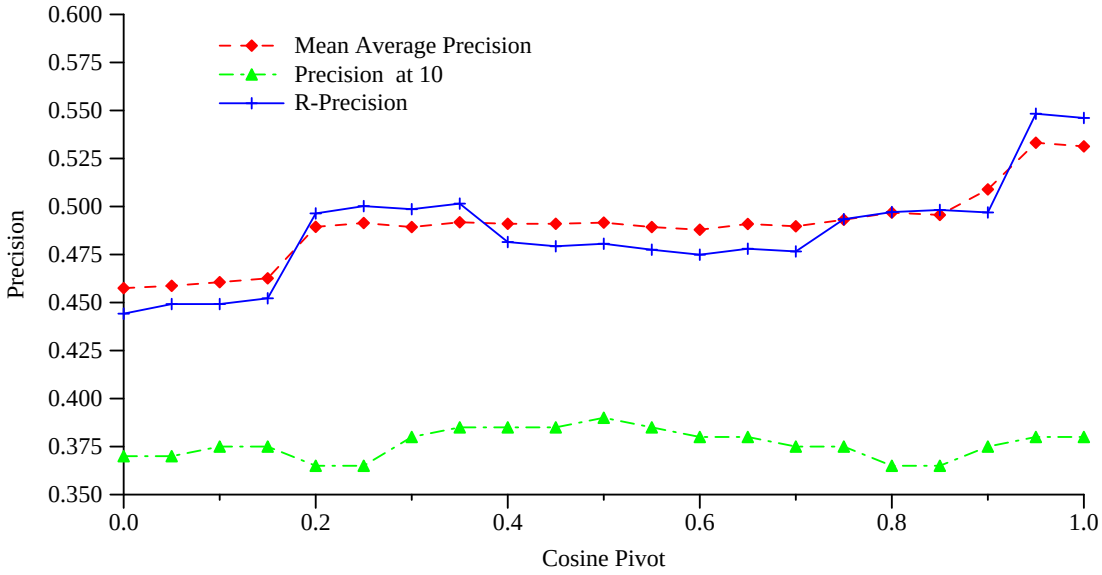

Figure 4.3: Effectiveness for varying values of the cosine pivot values. The valid range of the pivot value is between 0 and 1 inclusive. A pivot value of 0 means that there is no document length normalisation, while a pivot value of 1 means that document length normalisation is in full effect. We use 0.05 as the interval.

4.3.1 Cosine Measure

We described the cosine measure in Chapter 2. Normalising Equation 2.7 for document length [Singhal et al., 1996], we get:

$$\text{pivoted cos}\,(q,d) = \frac{\frac{\sum_{t\in q\cap d}(\ln(1+\frac{N}{f_t})\times(1+\ln(f_{d,t})))}{\sqrt{\sum_{t\in d}(1+\ln(f_{d,t}))^2}}}{(1.0-p)+p\times(\frac{W_d}{aW_d})} \tag{4.1}$$

where p is the pivot, W_d is the weight of the document, and aW_d is the average weight of all documents in the collection. The weight of a document, W_d, is defined as:

$$W_d = \sqrt{\sum_{t\in d} {w_{d,t}}^2} \tag{4.2}$$

where $w_{d,t}$ is the weight of a term t in a document d as defined in Equation 2.5. In this case, a pivot of 0 means that there is no document length normalisation, whereas a pivot of 1 means that the document length normalisation is in full effect.

As can be seen from Figure 4.3, the highest MAP value of 0.53 and the highest R-precision value of 0.55 are achieved by using a pivot value of 0.95. These values are much higher than

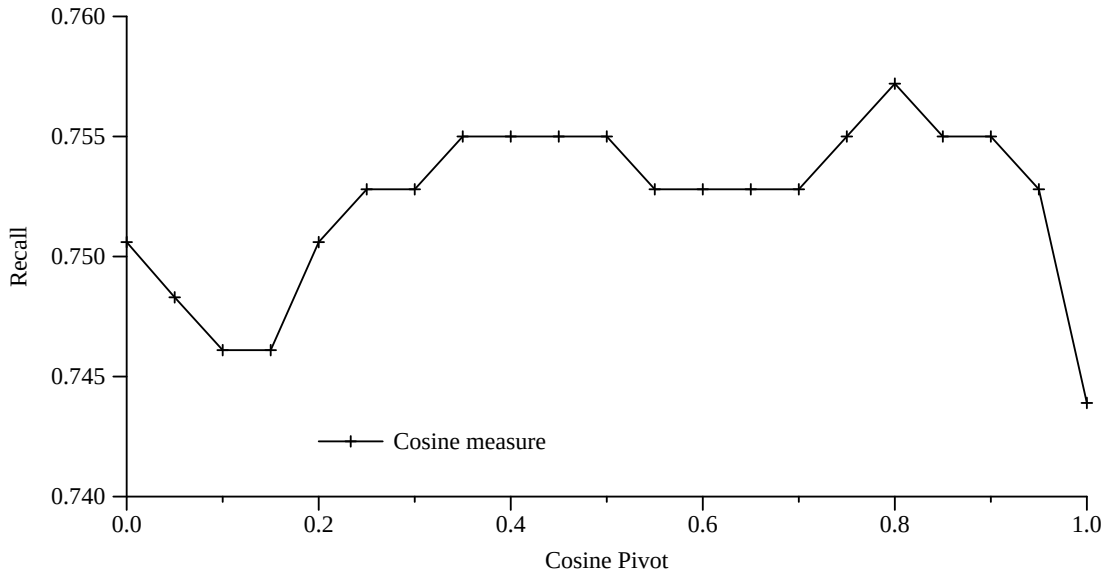

Figure 4.4: Recall for varying values of the cosine pivot (out of 453 relevant documents). The valid range of the pivot value is between 0 and 1 inclusive. A pivot value of 0 means that there is no document length normalisation, while a pivot value of 1 means that document length normalisation is in full effect. We use 0.05 as the interval.

the MAP value of 0.46 and R-precision value of 0.44 when no normalisation is used. Despite the big differences, these highest results are not statistically significantly better than the results produced by no normalisation. This phenomenon illustrates our statement in Section 2.3.5 that, unless verified by statistical significance testing, a much higher precision value does not always indicate a better system. We observe the highest precision@10 of 0.39 is obtained when the pivot value is 0.5, while using no normalisation produces precision@10 of 0.37. This difference is not significant ($p = 0.201$).

The highest recall value of 0.76 (343 relevant retrieved out of 453 relevant documents)[3] is obtained when the pivot value is 0.8 as shown in Figure 4.4. This is not statistically significantly different from the recall value of 0.75 (340 relevant retrieved) when no normalisation takes place ($p = 0.156$).

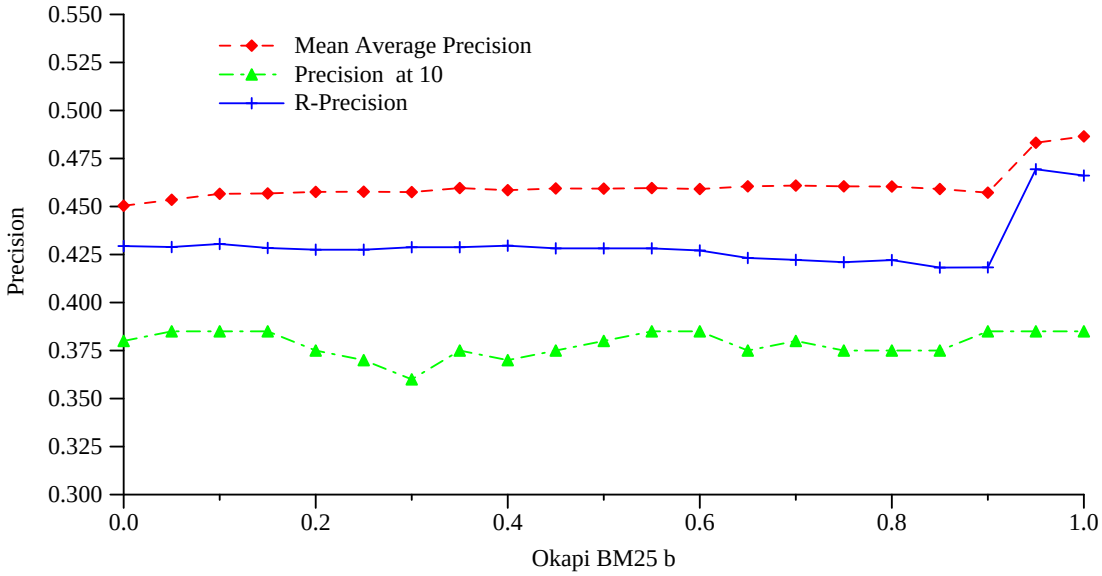

Figure 4.5: Effectiveness for varying b values of the Okapi BM25 measure. The valid range of b is between 0 and 1. If the value of b is 0, there is no document length normalisation; whereas the value b of 1 indicates that normalisation is in full effect. We use 0.05 as the interval. The value of k_1 is 1.2 and the value of k_3 is 0.

4.3.2 The Okapi BM25 Measure

As shown in Equation 2.10, there are three adjustable parameters for the Okapi BM25 measure: b, k_1, and k_3. The value of b, which is between 0 and 1 inclusive, determines how much document length normalisation is applied. If the value of b is 0, there is no document length normalisation, whereas a value of 1 indicates that normalisation is in full effect. The value of k_1, a positive number, indicates how strongly $f_{d,t}$ affects the whole weight in the Equation 2.10. If k_1 is very small or 0, the contribution of $f_{d,t}$ is effectively limited to whether the term t is present in the document without taking into account how many times t is present. Conversely, a larger k_1 value indicates that the weight increases more quickly with $f_{d,t}$. The value of k_3 indicates how much frequency of a term in a query, $f_{q,t}$, affects the equation. Robertson and Walker [1999] state that the optimum values of b and k_1 for TREC-8 ad hoc collections are 0.75 and 1.2, while the value of k_3 is set between 7 and 1 000 for long queries. Since our queries consist of only a few words each, we set the value of k_3 to 0 for all experiments, which means that the effect of term frequencies toward the calculation

[3]Since the number of relevant documents is always the same for our corpus, we later omit the "out of 453 relevant documents" and only write "x relevant retrieved".

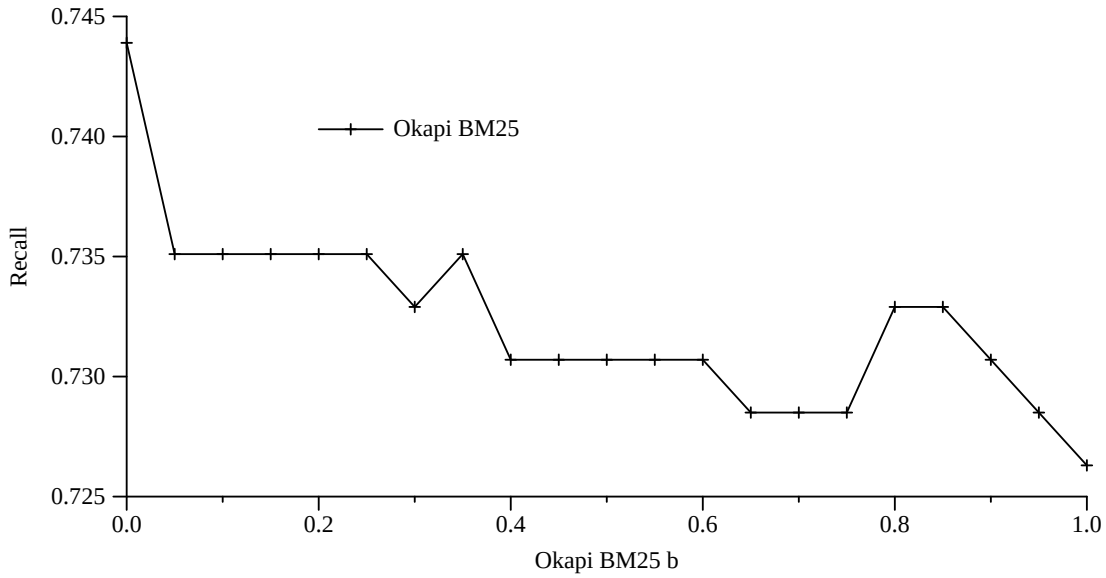

Figure 4.6: Recall for varying values of b values for the Okapi BM25 measure (out of 453 relevant documents). The valid range of b is between 0 and 1. If the value of b is 0, there is no document length normalisation; whereas the value b of 1 indicates that normalisation is in full effect. We use 0.05 as the interval. The value of k_1 is 1.2 and the value of k_3 is 0.

is limited to whether the terms are present in the query. We vary the values of b and k_1 to see whether most appropriate values for TREC-8 ad hoc collection are also applicable to our Indonesian collection.

The results of varying b are shown in Figure 4.5. The highest MAP value of 0.49 and the second-highest value of 0.48 are achieved when the values of b are 1.0 and 0.95; these are much higher than the MAP value of 0.46 produced using the default b of 0.75. Other b values produced MAP values ranging from 0.45 and 0.46. None of these MAP values are significantly different from the MAP value for the default b of 0.75 ($p > 0.05$). The highest R-precision of 0.47 is achieved when b is 0.95; however, this result is not statistically significant ($p > 0.05$). The highest precision@10 value of 0.39 is achieved by various b values, including the b of 0.95 that produces the highest R-precision. None of the increase in precision@10 is statistically significant ($p > 0.05$).

From Figure 4.6, it seems that there is a trend towards lower recall as b increases. The highest recall of 0.74 (337 relevant retrieved) is obtained when no normalisation takes place, and the default setting of b produces a recall value of 0.73 (330 relevant retrieved). This difference is not statistically significant ($p > 0.05$).

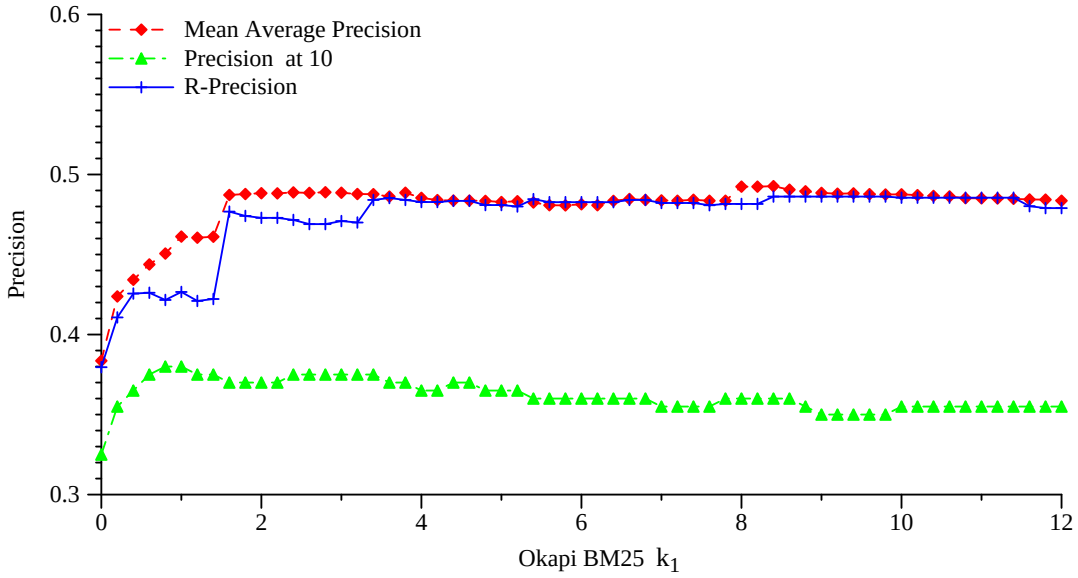

Figure 4.7: Effectiveness for varying k_1 values of the Okapi BM25 measure. If the value of k_1 is 0, we only see whether a term is present in a document; whereas greater values k_1 indicate that the weight of the terms increases with the number of time a term t appears in a document d. We use 0.2 as the interval. The value of b is 0.75 and the value of k_3 is 0.

There is no limit for the value of k_1 as long as it is a positive number. Zero or smaller values of k_1 indicate that contribution of the terms in a document, $f_{d,t}$, is limited to the presence of the terms, whereas larger values indicate that the weight of the terms increases with the number of times a term t appears in a document d.

As shown in Figure 4.7, the MAP and R-precision values increase with k_1. Using four decimal places, the highest MAP is produced for $k_1 =$ 8.4 (0.4927). This MAP is not significantly different from the MAP of 0.4605 produced by the default k_1 of 1.2 ($p > 0.05$). The R-precision produced by the default pivot is 0.42; whereas the highest R-precision of 0.49 is produced by a few k_1 values ranging from 8.4 to 11.4 inclusive with 0.2 as the interval. Despite the big difference, none of the differences are statistically significant compared to the R-precision produced by the default k_1 of 1.2 ($p > 0.05$). Conversely, smaller k_1 values produce higher precision@10 results. The default precision@10 of 0.38 produced when k_1 is 1.2 is already high, and only the k_1 values 0.8 and 1.0 surpass it by 0.005 percentage points. The differences in precision@10 produced by these two k_1 values are not statistically significant ($p > 0.05$).

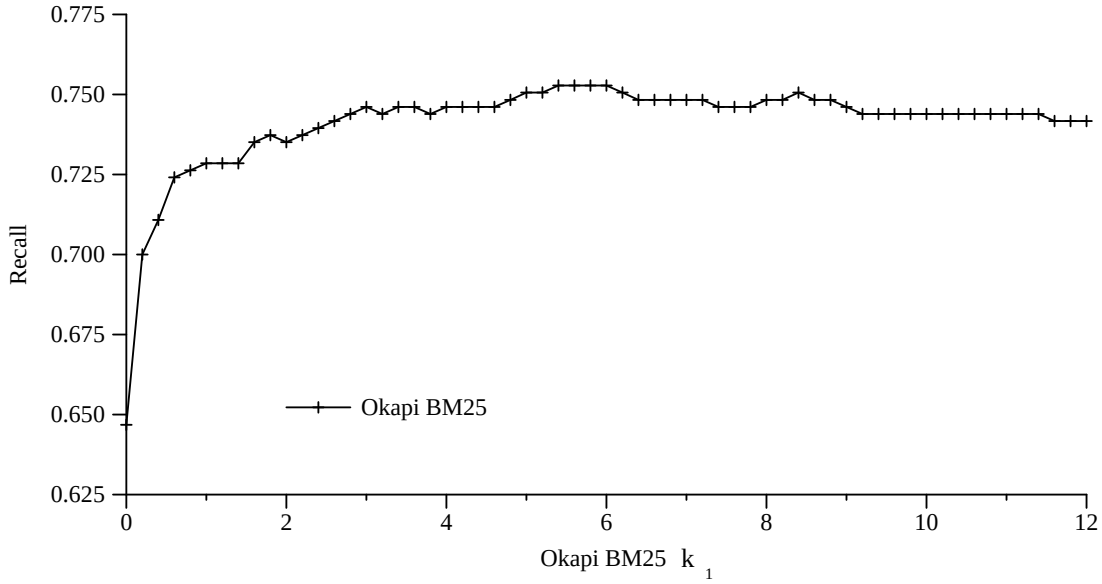

Figure 4.8: Recall for varying k_1 values of the Okapi BM25 measure (out of 453 relevant documents). If the value of k_1 is 0, we only see whether a term is present in a document; whereas greater values k_1 indicate that the weight of the terms increases with the number of time a term t appears in a document d. We use 0.2 as the interval. The value of b is 0.75 and the value of k_3 is 0.

The recall value produced by the default k_1 is 0.73 (330 relevant retrieved). From Figure 4.8, it can be seen that higher k_1 values tend to produce higher recall. The highest recall of 0.75 is produced by a range of k_1, which are 3.0, 3.4, 3.6, from 4.0 to 9.0 inclusive with 0.2 as the interval. None of the increase is statistically significant ($p > 0.05$).

The b value that produces the highest MAP is 0.95, while the k_1 value that produces the highest MAP is 8.4. The default setting and the best-performing parameter settings for both the Okapi BM25 and cosine measures are shown in Table 4.2. The cosine with pivot of 0.95 setting gives the best MAP and R-precision values, while the ($b = 0.75$ and $k_1 = 8.4$), ($b = 0.95$ and $k_1 = 8.4$), and ($b = 0.95$ and $k_1 = 1.2$) settings also give higher MAP and R-precision compared to the default Okapi BM25 settings. However, none of these differences are statistically significant ($p > 0.05$). The difference of R-precision values between the best Okapi setting and the best cosine measure setting is statistically significant ($p = 0.022$).

The symbol † in Table 4.2 is used to indicate a statistically significant difference in recall as compared to the default Zettair setting (Okapi BM25 $b = 0.75$ and $k_1 = 1.2$). Except

Measures	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
Cosine *pivot* = 0.00	0.46	0.44	0.37	**0.75** †	340
Cosine *pivot* = 0.95	**0.53**	**0.55**	0.38	**0.75**	341
Okapi BM25 $b = 0.75, k_1 = 1.2$	0.46	0.42	0.38	0.73	330
Okapi BM25 $b = 0.95, k_1 = 1.2$	0.48	0.47	**0.39**	0.73	330
Okapi BM25 $b = 0.75, k_1 = 8.4$	0.49	0.49	0.36	**0.75**	340
Okapi BM25 $b = 0.95, k_1 = 8.4$	0.49	0.48	0.36	0.74	333

Table 4.2: Precision and recall for different parameter values for the cosine and Okapi BM25 measures. "MAP" is mean average precision, "R-Prec" is R-precision, "Prec@10" is precision@10, and "Rel Retrieved" is relevant retrieved. All recall values are derived from "Rel Retrieved" divided by 453, which is the number of relevant documents. We assume that the default pivot for cosine measure is 0.0 (no normalisation), whereas the default for Okapi BM25 measure is the best setting for TREC-8 of $b = 0.75, k_1 = 1.2$. All other settings are based on parameters that give the highest mean average precision. The symbol † is used to indicate a statistically significant difference compared to the default Zettair setting.

for the default cosine measure setting with pivot 0 ($p = 0.040$), the recall values are not statistically significantly different from the recall value produced using the default Okapi BM25 setting. The difference in recall between the default cosine measure setting (pivot is 0) and the best cosine measure setting (pivot is 0.95) is not significant ($p = 0.233$).

4.3.3 Discussion

Although the MAP and R-precision values produced by the best cosine measure and some Okapi BM25 settings with altered b and k_1 are higher than the default, the differences are not statistically significant. The same applies for the difference in precision@10 values produced by both measures.

We conclude that the optimal settings of Okapi BM25 depends on the language and the corpus. We explore the value of b and k_1 separately, instead of finding all possible combinations of b and k_1. This means that we first find which b produces the highest MAP using the default value of k_1 (1.2), we then find which k_1 produces the highest MAP using the default value of b (0.75). For our collection, the optimal setting of b is 0.95, and the optimal setting for k_1 is 8.4, whereas for TREC-8 the optimal values for b and k_1 are 0.75 and 1.2

respectively. We note that the maximum k_1 value is unbounded; we did not experiment with k_1 values beyond 12, as precision stabilised around this value.

Moreover, these values may not be directly applicable to other Indonesian collections. More corpora are required to determine the optimum settings. Since the differences between Okapi BM25 and cosine measure are not statistically significant for MAP, and the default for Zettair is the Okapi BM25 measure, we use the Okapi BM25 measure with its setting ($b = 0.75$, k_1 =1.2, and k_3 =0) for the rest of the text retrieval experiments in this chapter.

4.4 Text Retrieval: Stopping

As discussed in Section 2.3.2, stopping removes words that do not carry topic-specific information, with the aim of reducing the noise in retrieval. Stopping may increase precision but is likely to reduce recall, as there are fewer keywords to be used in the query. The stopping technique is similar for all languages, but the word lists are unique to each language.

Stopword lists can be made by using words that appear very frequently, or by choosing words that do not contribute much information and serve only as grammatical markers. We have experimented with both types of stopword list.

4.4.1 Experiments

Our stopword lists are of two types: frequency-based and semantic-based. The frequency-based stopword list contains the n most frequent words in C_INDO-TRAINING-SET. We vary the values of n to see which value leads to the highest mean average precision (MAP); we limit the upper bound of n to the level where some queries return no answer document because the queries are empty. The 50 most frequent words are shown in Figure 4.9. The list is ordered from left to right, top to bottom, with the most frequent word appearing at the first column of the first row, the second most frequent at the second column of the first row, and so on, until the fiftieth most frequent word at the last column of the last row.

In our C_INDO-TRAINING-SET, there are 68 199 unique words including numbers and dates. After removing numbers and strings that have no letters attached to them such as “01:30”,“0.46%”, “01-06-02”, or “09.00-12.30”, there are 56 755 words left. These include a mixture of letters and numbers such as “ol-01”, “10.000km”, “grup-8” ⟨the eighth group⟩, “ke-300” ⟨the three hundredth⟩, and “oktober-31” ⟨31st October⟩. We experimented with

the top 5, top 10, and top 25 words, and then in steps of 25 to 375; beyond 375, we find that important query terms are also stopped.

The semantic-based stopword list takes into account the semantical functions of a word in a sentence. There are some semantic-based stopwords readily available [Tala, 2003; Vega, 2001]. The Tala stopword list consists of 758 unique words, the Vega stopword list 1 consists of 169 words and the Vega stopword list 2 consists of 556 words. We refer to these as TALA-STOP, VEGA-STOP1, and VEGA-STOP2, respectively. We show 50 words from each list in Figure 4.10, 4.11, and 4.12. The complete semantic-based stopword list can be found in Appendices F, G, and H.

4.4.2 Results and Discussion

The results of stopping using the n most frequent words are shown in Figure 4.13. Precision values monotonically decrease with increasing n. Stopping using the 10 or more most frequent words is statistically significantly less effective than with no stopping ($p < 0.05$). Some queries do not return any documents when n is 400 or more, because some queries are dropped altogether as all of the keywords are removed; hence we report results only up to n=375.

As shown in Figure 4.14, using the 100 or more most frequent words as stopwords leads to a drop in recall. Without stopping, recall is 0.73, but after stopping using the 100 most frequent words it drops to 0.68. Starting from $n = 100$, the recall values keep dropping. Stopping using the 100 or more most frequent words produce recall values that are significantly worse than no stopping ($p < 0.05$).

A comparison of no stopping and stopping using semantic-based stopwords is shown in Table 4.3. The symbol † indicates a statistically significant difference compared to no stopping. Using customised stopword lists increases MAP (at the fourth decimal place), except for VEGA-STOP1, which leads to decreased precision. The highest increase occurs with VEGA-STOP2, with MAP increasing from 0.4605 to 0.4632. In fact, using VEGA-STOP2 as stopwords produces the highest precision@10, R-precision, and recall values. Except for the recall value using VEGA-STOP2 as stopwords ($p = 0.038$), the rest of the differences are not significant ($p > 0.05$).

We conclude that constructing a stopword list on word frequencies alone is not enough. Frequency-based stopwords remove valuable keywords from queries such as the proper nouns "Jakarta", "Indonesia", and "presiden" ⟨president⟩, as shown in Figure 4.9. In general, semantic-based stopwords lists are better at increasing precision and recall. While the im-

yang	dan	di	itu	dengan
untuk	tidak	dari	dalam	akan
pada	ini	jakarta	tersebut	juga
ke	karena	presiden	katanya	ada
kata	kepada	mengatakan	indonesia	mereka
media	oleh	telah	mpr	sudah
as	saat	sebagai	bisa	saya
para	menjadi	melakukan	pemerintah	dpr
namun	ant	negara	bahwa	ketua
menurut	harus	masih	orang	terhadap

Figure 4.9: Top 50 most frequent words in C_INDO-TRAINING-SET *used as stopwords.*

ada	adalah	adanya	adapun	agak
agaknya	agar	akan	akankah	akhir
akhiri	akhirnya	aku	akulah	amat
amatlah	anda	andalah	antar	antara
antaranya	apa	apaan	apabila	apakah
apalagi	apatah	artinya	asal	asalkan
atas	atau	ataukah	ataupun	awal
awalnya	bagai	bagaikan	bagaimana	bagaimanakah
bagaimanapun	bagi	bagian	bahkan	bahwa
bahwasanya	baik	bakal	bakalan	balik

Figure 4.10: Sample of TALA-STOP *stopwords. We show the first 50 words appearing on the alphabetically sorted list.*

a	adalah	agar	akan	aku
anda	andaikata	antara	apa	apakah
apalagi	asal	atas	atau	b
bagaimana	bagaimanakah	bagi	bahkan	bahwa
begitu	begitulah	berkat	biji	bolehkan
bongkah	buah	buat	bungkus	butir
c	d	dalam	dan	dapatkah
dari	daripada	demi	demikian	dengan
di	dia	dimana	dimanakah	e
ekor	f	g	guna	h

Figure 4.11: Sample of VEGA-STOP1 *stopwords. We show the first 50 words appearing on the alphabetically sorted list.*

a	acuh	ada	adalah	adil
agak	agar	akal	akan	akhir
akhir-akhir	akibat	akibatnya	aku	amat
ambil	anda	antara	antri	anu
apa	apakah	apalagi	apapun	asumsinya
atas	atau	ayo	ayolah	b
bagaimana	bagaimanakah	bagaimanapun	bagian	bagus
bahwa	baik	bakal	banyak	baru
bawah	beberapa	beda	bekas	belakang
belakangan	benar	berbagai	berbeda	bergaul

Figure 4.12: Sample of VEGA-STOP2 *stopwords. We show the first 50 words appearing on the alphabetically sorted list.*

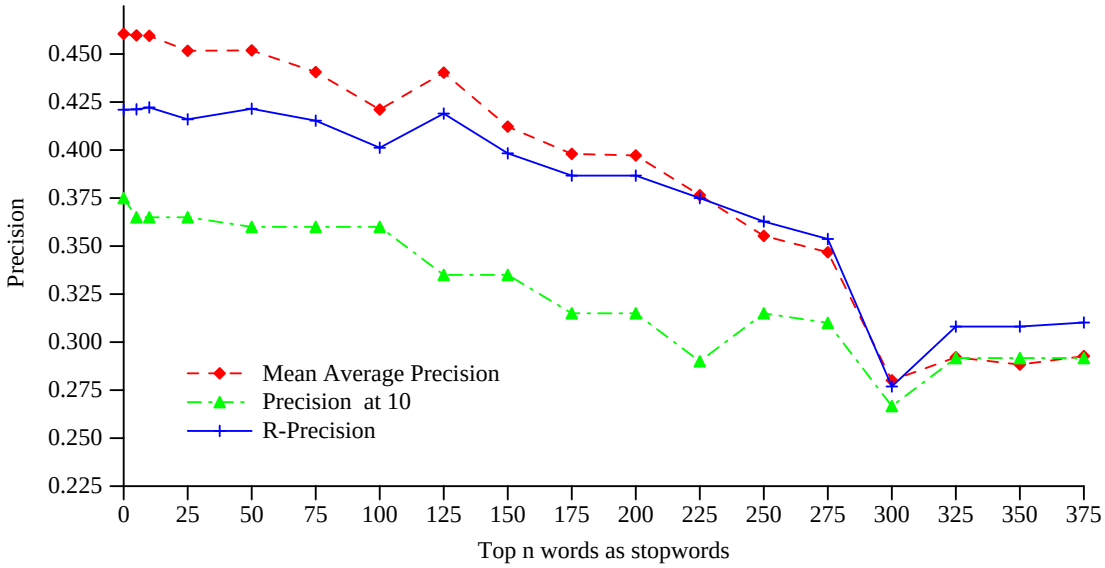

Figure 4.13: Effectiveness for varying n (the number of most frequent words used in the stopword list); n = 0 corresponds to no stopping.

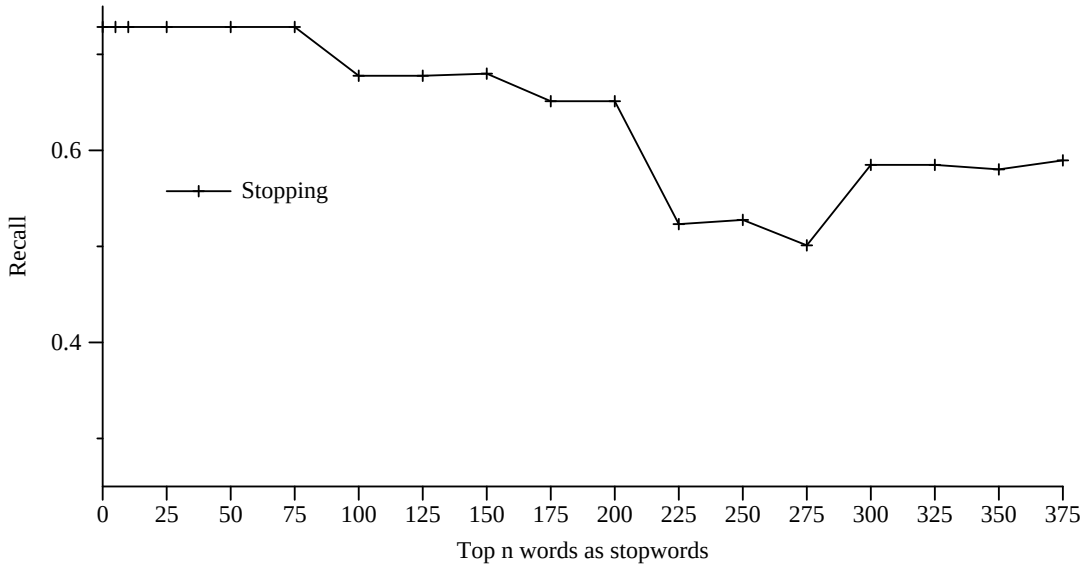

Figure 4.14: Recall (out of 453 relevant documents) for varying n (the number of most frequent words used in the stopword list); n = 0 corresponds to no stopping.

Stopword List	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
No Stopping	**0.46**	0.42	0.38	0.73	330
TALA-STOP	0.45	0.42	0.39	0.74	334
VEGA-STOP1	**0.46**	0.42	0.38	0.73	330
VEGA-STOP2	**0.46**	**0.43**	**0.40**	**0.75** †	**341**

Table 4.3: Precision and recall for original documents and documents stopped using different stopword lists. The symbol † indicates a statistically significant difference compared to no stopping.

provement is not significant for our collection, this approach appears promising. We explore the combination of stopping and stemming in the following section.

4.5 Text Retrieval: Stemming

In Chapter 3, we described stemming approaches for Indonesian, and reported on our confix-stripping (CS) approach. In this section, we experiment with various stemming algorithms to assess their impact on recall and precision. We also test the effect of combining stopping with the best stemming algorithm.

4.5.1 Experiments

We experiment with all six stemming algorithms we discussed in Chapter 3, namely S_NA, S_AYS-b_2 , S_I-2, S_AS, S_V-1, and CS.[4] We use the DICT-UI dictionary created by Nazief and Adriani [1996], as discussed in the previous chapter, for all stemming algorithms that require a dictionary. To test the impact of using a different dictionary, we use the DICT-KBBI dictionary, also discussed previously, for the stemming algorithm that produces highest MAP with DICT-UI. We were unable to use the online DICT-KEBI dictionary because the service was not available at the time of writing.

To test the effect of both stopping and stemming, we first stop the queries and documents using the stopword list that produces the highest MAP, VEGA-STOP2, then stem the stopped documents using the stemming algorithm that also produces the highest MAP.

[4]Some algorithms have a few variants; we chose the variant that produced the highest stemming accuracy against manual stemming.

Stemming Algorithm	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
No Stemming	0.46	0.42	**0.38**	0.73	330
S_NA	0.48	**0.45**	0.36	**0.78**	**354**
S_AYS-b_2	0.48	**0.45**	0.36	**0.78**	352
S_AS	0.48	**0.45**	0.35 †	**0.78**	353
S_V-1	**0.49**	**0.45**	0.36	**0.78**	352
S_I-2	0.48	**0.45**	0.36	**0.78**	**354**
CS	0.48	**0.45**	0.36	**0.78**	**354**
CS using DICT-KBBI	0.48	0.44	0.36	**0.78**	353

Table 4.4: Precision measures and recall for no stemming and stemming using different algorithms. The symbol † is used to indicate a statistically significant difference compared to no stemming.

4.5.2 Results and Discussion

A comparison of using different stemming algorithms with using the unstemmed documents is shown in Table 4.4. Stemming consistently increases MAP for all algorithms; surprisingly, the S_V-1 stemming algorithm that produced the poorest stemming results as presented in Chapter 3, leads to the the highest MAP of 0.49. Nevertheless, none of the increases in MAP or R-precision are significant. Stemming using the most accurate stemmer, the CS stemmer, also increases MAP. However, stemming appears to hurt precision@10, the decreases are not significant ($p > 0.05$), except for the precision@10 produced by S_AS ($p = 0.038$).

We use the DICT-KBBI dictionary with the CS stemmer to assess the impact of using a different dictionary on retrieval effectivenesss. The result is shown on the last line of Table 4.4. Except for R-precision, the results of using DICT-UI are similar to the results of using DICT-KBBI. A possible reason for this is that DICT-UI contains more root words than DICT-KBBI. For example, the word "tantangan" ⟨challenge⟩ should be stemmed to "tantang" ⟨to challenge⟩; this is done correctly when DICT-UI is used, but not when DICT-KBBI is used since DICT-KBBI contains the word "tantangan" in addition to the stem "tantang". None of the difference in precision values between the CS stemmer using DICT-UI and DICT-KBBI is significant ($p > 0.05$).

We expected stemming to increase recall, and that is indeed the case, as the stem can

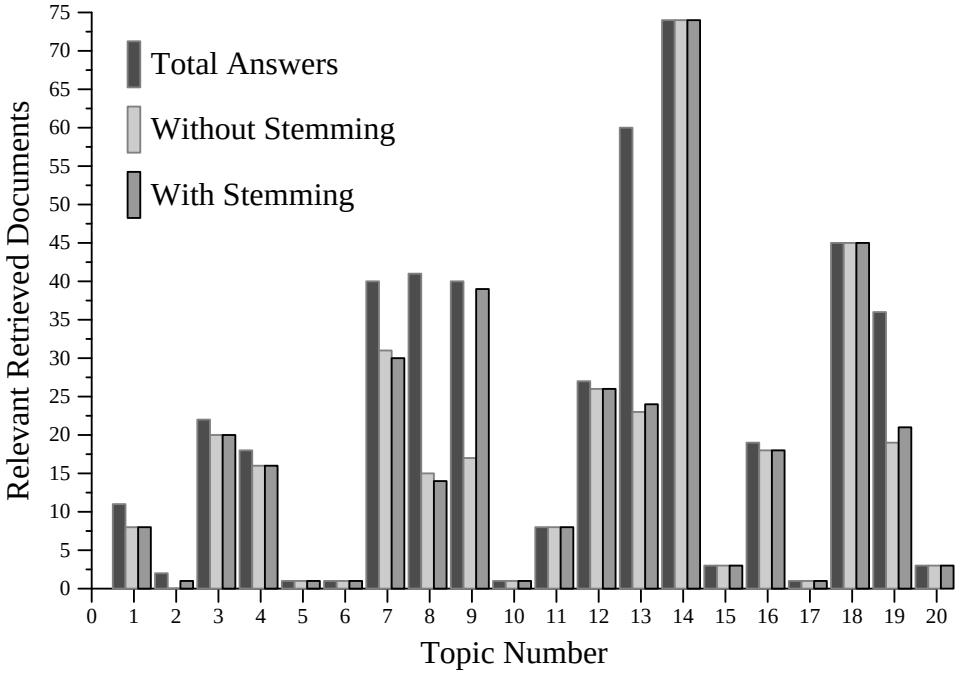

Figure 4.15: Topic-by-topic performance with and without stemming. For each topic, the left column shows the number of relevant documents, the middle column represents the number retrieved without stemming, and the right column shows the number retrieved with stemming. Only the query title text is used for querying.

match more words in different documents. None of the increase in recall is significant ($p > 0.05$).

Hull [1996] states that the absolute increase in MAP due to stemming is ranging from 1% to 3%. When our Indonesian corpus is stemmed using the CS stemmer, the absolute improvement is slightly over 4%, but this is not statistically significant. The slight difference between the languages is perhaps because Indonesian words have many more variants — with prefixes, infixes, suffixes, and confixes — than English does.

Hull [1996] elaborates further that, although stemming does not in general greatly improve effectiveness, the effect on the performance on individual queries varies greatly. We show the effect of stemming on recall for each query in Figure 4.15. For each topic, three bars are shown: on the left, the total number of relevant documents; in the middle, the number of relevant documents found without stemming; and, on the right, the number of relevant documents found with stemming using the CS stemmer. The results show that — with the exception of Topic 2, Topic 9, and Topic 19 — there is little difference between recall with

and without stemming. The overall recall value without stemming is 0.73; with stemming, this increases to 0.78.

We suspect that the lack of improvement 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 boss of Manchester United⟩ does not retrieve one document that discusses `the manager of MU`. 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.

Stemming hurts precision@10, although, except for S_AS, the difference is not significant. This usually happens when nouns derived from verbs are stemmed to become verbs again. The words "laporan" ⟨report⟩ in "laporan piala dunia" ⟨world cup report⟩ and "kenaikan" ⟨increase⟩ in "akibat kenaikan harga bbm" ⟨effects of the increase of petrol price⟩ are stemmed to "lapor" ⟨to report⟩ and "naik" ⟨to climb⟩ respectively. For these cases, no stemming can lead to the exact matches of the phrases, hence higher precision@10 values, while stemming can lead to spurious matches.

Stemming appears to aid MAP and R-precision, while reduces precision@10, although results are generally not significant. As the CS stemmer was shown to be the most accurate stemmer, we choose this algorithm for subsequent experiments.

In Section 4.4, we have described that using the VEGA-STOP2 stopword list produces the highest precision values. We used the CS stemmer to stem our corpus after stopping using the VEGA-STOP2 stopword list to see the effect of both stopping and stemming. The results are shown in Table 4.5. MAP and R-precision are both improved by stopping and stemming, with increases of more than 5%. Stopping in general increases precision@10, while stemming decreases it, the combination improves precision@10, although stemming counters the effect of stopping. Combination of stopping and stemming also increases recall. Despite the large differences compared to the original collection, none of these increases is significant ($p > 0.05$). The increase in precision@10 produced by combining stopping and stemming is statistically significant compared to the very low precision@10 produced by stemming ($p = 0.029$).

Scheme	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
Original	0.46	0.42	0.38	0.73	330
Stopped	0.46	0.43	**0.40**	0.75	341
Stemmed	0.48	0.45	0.36	0.78	354
Stopped and stemmed	**0.51**	**0.48**	**0.40**	**0.79**	**356**

Table 4.5: Precision and recall for no stemming, stopping, stemming, and combination of stopping and stemming. The stopping algorithm used is VEGA-STOP2 *and the stemming algorithm is the* CS *stemmer; each produces the highest MAP across its variants.*

4.6 Text Retrieval: Tokenisation

In Section 2.3.2, we discussed tokenisation techniques, among them n-grams. Use of n-grams is a form of stemming that is language independent [Mayfield and McNamee, 2003]. We experiment with different tokenisation schemes, with the expectation that similar to stemming, they will lead to improved recall and precision.

The content of n-grams is affected by various factors, including gram or token sizes, and whether to add some special characters such as space (␣) between words to join them into a single token. For example, the word "information" can be tokenised into the following 4-grams: "info", "nfor", "form", "orma", "rmat", "mati", "atio", and "tion".

There is a more complex version of tokenisation that spans word boundaries; this can help in phrase identification [McNamee and Mayfield, 2004b], and is done by adding spaces (␣) between words. Under this scheme, when the phrase "train station" is tokenised into 7-grams, the results are "␣train␣", "train␣s", "rain␣st", "ain␣sta", "in␣stat", "n␣stati", "␣statio", "station", and "tation␣". Some of the grams such as "in␣stat" and "n␣stati" are less common than others such as "␣train␣" and "station", therefore the tokens "in␣stat" and "n␣stati" are good indicators that the phrase we are seeking for is "train station". McNamee and Mayfield [2004a] choose to not span sentences. Unlike spanning word boundaries that can help in identifying phrases, spanning sentence boundaries is not likely to help increasing recall or precision. When the two sentences "I bought a new mat. A cat sat on the mat." are tokenised into 4-grams, the resulting tokens for the first sentence are "i␣bo", "␣bou", "ough", "ught", "ght␣", ht␣a", "t␣a␣", "␣a␣n", "a␣ne", "␣new", "ew␣m", "w␣ma", "␣mat".

The tokens for the second sentence start with "a␣ca". Should the tokenisation span across sentences, there will be additional tokens connecting the two sentences: "at␣a", "t␣a␣ ', and "␣a␣c".

4.6.1 Experiments

We experiment with variants of tokenisation both spanning and not spanning word boundaries. When not spanning word boundaries, we can add spaces either before and after each word, or leave the words unchanged. McNamee et al. [2000] suggest that special tokens be used for numbers. Positive integer numbers that contain four or fewer digits remain unchanged, whereas positive integers that are more than four digits are written as `INTNUMBER`, and all positive floating point numbers are written as `FPNUMBER`. We use the Indonesian convention for thousands, which are separated by dots, and decimal numbers, which are separated by a comma as defined in Section 2.1.4. For example, the integer "75.000", which is equivalent to "75,000" in English, is tokenised as `INTNUMBER`, whereas the decimal "0,435", which is equivalent to "0.435" in English, is tokenised as `FPNUMBER`.

We use three definitions of sentence delimiter. One variant uses the traditional delimiters of sentences: dot (.), question mark (?), and exclamation mark (!). Another variant adds colons (:) as the end of sentences and the last variant adds both colons (:) and commas (,) as sentence delimiters.

Since a "." can act either as a sentence delimiter or a separator for thousands, we look at the first character preceding the ".". If it is a character, then we consider it as the end of a sentence and if it is a number and there is also another number follows the "." then we consider it as a number instead of a sentence delimiter.

We experiment with all these variants for both spanning and no spanning versions, using tokens of sizes between 2 and 7 inclusive, to see the impacts of different factors on precision and recall. We show only the variants that achieved higher MAP in general for both versions. For the no spanning word boundary version, the highest MAP is achieved by the variant that adds a space(␣) before and after each word; for the spanning word boundary version, the highest MAP is obtained by the variant that uses the traditional delimiters of sentences.

4.6.2 Results and Discussion

Table 4.6 shows the precision and recall of using different token sizes for tokenisation that does not span the word boundary. The 4-gram scheme produces the highest MAP, R-precision, and

Scheme	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
Original	0.46	0.42	0.38	0.73	330
Words	0.48	0.45	0.36	0.78	354
2-gram	0.20 †	0.24 †	0.17 †	0.54 †	246
3-gram	0.48	0.48	0.37	0.78	354
4-gram	**0.53**	**0.52** †	0.38	**0.80** †	**361**
5-gram	0.51	0.45	**0.42**	0.79	358
6-gram	0.48	0.46	0.37	0.72	328
7-gram	0.34 †	0.31 †	0.34	0.62	279

Table 4.6: Precision and recall for use of n-grams that does not span word boundaries compared against untokenised version and version that are tokenised at word level (stemmed). Extra spaces are added before and after words that are divided into n-grams. The token sizes are from 2 to 7 inclusive. The symbol † is used to indicate a statistically significant difference compared to original (no stemming).

recall, while the 5-gram scheme produces the second-highest MAP and increases precision@10 by 4 percentage points compared to no stemming. The 3-gram scheme also produces the second-highest R-precision. Only the increase in R-precision and recall produced by 4-gram tokenisation is significant ($p = 0.013$ and $p = 0.047$). All the precision and recall values produced when using 2-grams are significantly worse than using no tokenisation ($p < 0.001$), and also worse than using stemming ($p < 0.001$). MAP and R-precision produced by 7-grams are also significantly worse than using no tokenisation ($p = 0.021$ and $p = 0.027$), and worse than stemming ($p = 0.001$ and $p = 0.013$). The remaining token sizes produce higher precision values than the stemmed version, although the significant differences are only in the R-precision produced by tokens of size 4 ($p = 0.042$) and in the precision@10 ($p = 0.009$) produced by tokens of size 5.

Table 4.7 shows the precision and recall for the tokenisation scheme that spans the word boundaries. Although the highest MAP value is achieved when tokenisation across word boundaries, the best results were observed using 5-grams instead of 4-grams. The highest R-precision values are achieved by tokenisation using 4- and 5-grams; and the highest precision@10 is achieved by 5-gram tokenisation. Most differences are not statistically significant; however, all precision and recall values produced by 2-gram tokens are significantly worse

Scheme	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
Original	0.46	0.42	0.38	0.73	330
Words	0.48	0.45	0.36	0.78	354
2-gram	0.20 †	0.24 †	0.17 †	0.54 †	246
3-gram	0.49	0.49	0.37	0.78	354
4-gram	0.53	**0.52** †	0.37	**0.79**	**357**
5-gram	**0.55**	**0.52** †	**0.40**	0.78	355
6-gram	0.51	0.51 †	0.37	**0.79**	356
7-gram	0.49	0.44 †	0.36	**0.79**	**357**

Table 4.7: Precision and recall for tokenisation that spans word boundaries using traditional sentence delimiters: dot (.), question mark (?), and punctuation mark (!) compared against untokenised version and version that are tokenised at word level (stemmed). The token sizes are from 2 to 7 inclusive. The symbol † is used to indicate a statistically significant difference compared to original (no stemming).

than no tokenisation ($p < 0.001$), and the R-precision produced by 4-, 5-, 6-, and 7-gram tokens are significantly better than no tokenisation ($p = 0.008$, $p = 0.003$, $p = 0.012$, and $p = 0.034$). The highest recall is also achieved when using 4-, 6-, and 7-grams, although it is not statistically significant ($p > 0.05$). Apart from all precision and recall values for 2-grams (all $p < 0.001$) and R-precision for 7-grams ($p = 0.212$), the remaining recall and precision values by other token sizes are higher than the precision and recall values of the CS stemmer; only increase in the R-precision for token of sizes 3, 4, and 5 is significant ($p = 0.025$, $p = 0.027$, and $p = 0.307$).

The good results obtained by tokens of size 4 and 5 for both versions are similar to the maximum MAP achieved on many European languages [McNamee and Mayfield, 2004b] — 4-grams for English, French, German, Italian, Spanish, and Swedish and 5-grams for Dutch and Finnish. Besides gram lengths of 2 for both versions and 7 when not spanning word boundaries, the rest of gram lengths outperform the CS stemmer, which takes into account word morphology. A similar phenomenon is observed by McNamee and Mayfield, who suspect that one of the factors that determine the best gram size is the mean word length. As analysed in Section 3.3, the mean word length for our Indonesian testbed is 6.75 and most words of less than 6 characters are root words. We conjecture that using gram sizes between 3 and 5 may

in fact be similar to stemming. The higher-than-normal stemming precision values produced by tokenisation are likely to be due to its sensitivity towards phrases, which can be achieved by the tokenisation that spans the word boundary. Since the average word length is 6.75, the most likely candidate token sizes producing the best higher precision are 6 or 7. However, that is not the case, possibly because mean word length is only one of the factors, other factors include morphological complexity [McNamee and Mayfield, 2004b]. We conclude that it is better to span word boundaries, and it is better to use 5-grams as this produces the highest MAP values.

4.7 Text Retrieval: Dictionary augmentation using n-grams

Table 3.11 shows that misspellings account for a little over 10% of stemming errors. To correct misspellings, we propose a stemming scheme that uses n-grams; this is a common mechanism for evaluating string similarity [Bakar et al., 2000; Ng et al., 2000]. A string of length N can be decomposed into $N - n + 1$ substrings or n-grams, where n is the length of the gram. 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 misspelt 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 those of the correctly spelt word.

The basic principle of tokenisation is similar to the one explained in the previous section, but we use it differently here, namely to augment the word list used for stemming. We stem words using the best stemming algorithm, the CS stemmer; when words are not stemmed successfully and do not appear in the dictionary, they could be misspelt. We use tokenisation to find possible matches in the dictionary for that word. The first match returned by the tokenisation method is assumed to be the correct stem.

To determine which dictionary word is the closest, a variety of measures have been described in the literature; these include *Q-grams* [Ukkonen, 1992] and *tapering* [Zobel and Dart, 1996]. The *Q-gram* method is used to measure the distance between two strings s and t [Ukkonen, 1992], which 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, G_t is the number of n-grams in string t, and $|G_s \cap G_t|$ is the number of identical n-grams in the two strings. For the strings "perintah"

⟨order⟩ and "perimtah" (misspelt word) and 2-grams, we compute the distance to be $7+7-(2\times|5|)=4$.

The *tapering* method is an edit distance method that gives a higher penalty when deletion and replacement occurs at the beginning rather than at the end of a string [Zobel and Dart, 1996]. An edit distance method counts the minimum number of insertion, deletion, and replacement operations required to transform one string into another string. For example, the minimum edit distance between "perimtah" and "perintah" is one, by replacing "m" to "n". When the tapering method is used, the penalty from transforming "serintah" (misspelt word) to "perintah" is higher than transforming "perimtah" to "perintah", hence the distance for the former will be higher than the latter. Although both instances of misspelt words have only one character difference, the location of the character replacement differs — one occurring at the first character while the other occurring at the fifth character, any replacement or deletion occurring earlier has higher penalty than replacement or deletion occurring later.

We have explored different approaches to finding the closest words, and experiment with different n-gram sizes to see which combination produces the highest stemming accuracy. The closest match returned by each approach is considered to be the best answer; the answer that is deemed best by the algorithm is not necessarily the right answer. In Section 4.5.2, we reported that the CS stemmer produces increased recall and precision over unstemmed queries. We conjecture that the most accurate stemming scheme will produce the highest MAP, therefore we test the stemming accuracy of each method by first comparing the stemmed results against the manually stemmed results described in Chapter 3 before reporting on the effects of each method on recall and precision. We later report the effects of combining stemming and tokenisation towards precision and recall.

Results and Discussion

To discover the measure, such as Q-gram and tapering, and n-gram size that can find the closest dictionary word, and hence result in the best stemming accuracy, we conducted a preliminary study using the test collection C_TE_MAJORITY and C_TE_UNIQUE. We discovered from the study that the Q-gram approach, using an n-gram length of between 5 and 7 characters produces the highest stemming accuracy.

However, incorporating n-grams does not always result in improved stemming precision. Table 4.8 shows that 5-, 6-, and 7-grams produce results that are similar to the CS stemmer without any extension. The symbol † in this section is used to indicate a statistically sig-

Stemmer	C_TE_MAJORITY		C_TE_UNIQUE	
	Correct (%)	Errors (words)	Correct (%)	Errors (words)
CS	**94.9**	**199**	**94.7**	**87**
CS, 3-grams	76.3 †	925	75.2 †	411
CS, 4-grams	79.5 †	802	79.2 †	344
CS, 5-grams	**94.9**	**199**	**94.7**	**87**
CS, 6-grams	**94.9**	**199**	**94.7**	**87**
CS, 7-grams	**94.9**	**199**	**94.7**	**87**

Table 4.8: Comparison of stemming performance for original CS*, and* CS *with dictionary extension using n-grams and the Q-gram distance measure, for the test collection* C_TE_MAJORITY *and* C_TE_UNIQUE*. The symbol † is used to indicate a statistically significant difference compared to the* CS *stemmer without any extension.*

nificant difference compared to the CS stemmer baseline without any extension ($p < 0.05$). A possible reason that dictionary augmentation using larger n-grams is more accurate than using smaller n-grams is that shorter grams tend to match more words in the dictionary, including incorrect matches, than longer grams; our algorithm picks the closest match, which is not always correct. The incorrect matches usually occur for words that are not in dictionary, these include proper nouns, compound words, foreign words, and words with different spellings. Based on the results we described in Chapter 3, humans usually stem compound words, correct and stem misspelt words, and do not stem proper nouns and foreign words. When larger gram sizes are used, most resulting grams do not match dictionary words, so these words remain unchanged, whereas when smaller gram sizes are used, there are likely to be more matches, so trying to simply find the closest match may not be correct. For example, the proper nouns "indonesia" and "errikson" are stemmed to "amnesia" ⟨amnesia⟩ and "periksa" ⟨to check⟩ when the algorithm tries to find the closest match for 4-grams. However, the algorithm will leave these words unchanged using larger gram sizes, because there are no equivalent matches in the dictionary. Compound words such as "kerjasama" ⟨to cooperate⟩ and foreign words such as "champion" are also stemmed to "kerja" ⟨to work⟩ and "lampion" ⟨lantern⟩ respectively using smaller grams. In a few cases, it is better to use shorter n-grams. For example, with 4-grams we correctly stem "mengritik" ⟨to criticise⟩

Measure	Unstemmed	CS	CS 3-grams	CS 4-grams	CS 5-grams	CS 6-grams	CS 7-grams
MAP	0.46	0.48	0.50	**0.51**	0.48	0.48	0.48
Precision at 10	**0.38**	0.36	0.34	0.34	0.36	0.36	0.36
R-precision	0.42	0.45	0.48	**0.50**	0.45	0.45	0.45
Recall	0.73	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**	**0.78**
Rel Retrieved	330	**354**	**354**	353	**354**	**354**	**354**

Table 4.9: Retrieval performance for unstemmed, stemmed using CS*, and stemmed using* CS *with n-gram extension using different gram lengths and the Q-gram distance measure.*

to "kritik" ⟨to criticise⟩, whereas we fail to find a stem when using 5-grams or longer. On balance, proper nouns occur more often than long words such as "mengritik", and so the benefit of using longer n-grams outweighs the benefit of using shorter ones. While we aimed to use n-grams to correct misspellings, we discovered that using smaller n-grams, in this case grams of size 4 or smaller, manage to correctly stem 4 of the 11 misspellings that appeared in C_TE_MAJORITY and C_TE_UNIQUE. For example, the word "kahawatirkan", which is supposed to be spelled as "khawatirkan" ⟨to worry about⟩ and the word "ditaklukan", which is supposed to be spelled as "ditaklukkan" ⟨to be conquered⟩, are correctly stemmed to "khawatir" ⟨to worry⟩ and "takluk" ⟨to conquer⟩ after n-grams are incorporated.

For both C_TE_MAJORITY and C_TE_UNIQUE, there is no difference between stemming using the CS stemmer with and without spelling correction, and using 5-, 6-, and 7-grams ($p > 0.05$, one-tailed McNemar test). Although combining stemming with 3-grams and 4-grams can correct some misspellings, the accuracy is significantly worse than not using n-grams, for both collections ($p < 0.001$).

The best stemming effectiveness is not necessarily accompanied by the best retrieval effectiveness. From Table 4.9, we see that the best MAP and R-precision values are produced when using 4-grams. However, using stemming — with and without adding n-grams — hurts precision@10. The precision values produced by the combination of CS with n-grams are not significant compared to both stemming using CS without n-grams and also not stemming at all ($p > 0.05$, one-sided Wilcoxon signed ranked test). Only the precision@10 values for 3- and 4-grams produce significantly worse results than not stemming ($p = 0.022$ and $p = 0.019$). Similarly, combining the CS stemmer with n-grams does not help recall, with no

Stemmer	C_TE_MAJORITY		C_TE_UNIQUE	
	Correct (%)		Correct (%)	
	DICT-UI	DICT-KBBI	DICT-UI	DICT-KBBI
CS	**94.9**	**88.8**	**94.7**	**87.9**
CS, 3-grams	76.3 †	71.5 †	75.2 †	69.2 †
CS, 4-grams	79.5 †	74.6 †	79.2 †	73.1 †
CS, 5-grams	**94.9**	**88.8**	**94.7**	**87.9**
CS, 6-grams	**94.9**	**88.8**	**94.7**	**87.9**
CS, 7-grams	**94.9**	**88.8**	**94.7**	**87.9**

Table 4.10: Comparison of stemming performance (precision) for CS *and* CS *with dictionary extension using n-grams and the Q-gram distance measure for the test set* C_TE_MAJORITY *and* C_TE_UNIQUE *using the* DICT-UI *and* DICT-KBBI *dictionaries. The symbol † is used to indicate a statistically significant difference compared to the* CS *stemmer without any dictionary augmentation.*

recall value significantly different from not stemming and from stemming using CS without n-grams. Using n-grams can correct some misspellings, and also leads to more matches, which in turn increases MAP. Overall, we consider that 4-grams offer the best trade-off.

4.7.1 Extensions

The effectiveness of stemming depends on dictionary quality. Table 3.11 shows that around 22% (43 of 199) to 26% (51 of 196) of stemming errors can be traced to an incomplete dictionary or to misspellings. In this section, we explore the effect of using the DICT-KBBI dictionary instead of the DICT-UI dictionary for dictionary augmentation of the CS stemmer. The online dictionary DICT-KEBI is no longer available.

Results and Discussion

The results shown in Table 4.10 are consistent with those for the CS stemmer without any n-gram extensions: using DICT-UI tends to produce more accurate results than using the DICT-KBBI dictionary. This phenomenon exhibits a similar trend to combining DICT-UI with n-grams as shown in Table 4.8. The symbol † in this table is used to indicate a statistically significant difference compared to the CS stemmer without any augmentation using each own

Approach	Dictionary					
	dict-ui	dict-kbbi				
	MAP	MAP	R-Prec	Prec@10	Recall	Rel Ret
cs	0.48	0.48	0.44	**0.36**	**0.78**	**353**
cs, 3-grams	0.50	0.49 †	0.47	0.35	**0.78**	352
cs, 4-grams	**0.51**	**0.51**	**0.49**	0.35	**0.78**	**353**
cs, 5-grams	0.48	0.48	0.44	**0.36**	**0.78**	**353**
cs, 6-grams	0.48	0.48	0.44	**0.36**	**0.78**	**353**
cs, 7-grams	0.48	0.48	0.44	**0.36**	**0.78**	**353**

Table 4.11: Comparison of IR performance for the cs *stemmer, and* cs *with n-gram extensions using the Q-gram distance measure with different dictionaries —* dict-ui *and* dict-kbbi*. The symbol † is used to indicate a statistically significant difference compared to the* cs *stemmer without any dictionary augmentation.*

dictionary ($p < 0.05$), comparing results of using dict-ui with dict-ui and using dict-kbbi with dict-kbbi. There is no difference in stemming accuracy for the cs stemmer using dict-kbbi with and without n-grams ($p > 0.05$, one-tailed McNemar test) when n-grams of size 5 or larger are used. Meanwhile, the stemming accuracy of using 3-grams and 4-grams is significantly worse than when not using n-grams for both dictionaries ($p < 0.001$).

Table 4.11 shows that using dict-kbbi is similar to using dict-ui in terms of recall and precision. The symbol † is used to indicate a statistically significant difference compared to the cs stemmer without any augmentation using each own dictionary ($p < 0.05$), comparing results of using dict-ui with dict-ui and using dict-kbbi with dict-kbbi. The highest MAP and R-precision values are achieved by 4-grams using dict-kbbi; these results are similar to those obtained when using dict-ui, but using 4-grams does not produce statistically significantly better results than stemming without any dictionary augmentation ($p > 0.05$, one-sided Wilcoxon signed ranked test). Using larger grams of size 5, 6, and 7 does not increase MAP values, and the differences are also not significant. Only the slight increase of MAP by using 3-grams is statistically significantly better than not using any dictionary augmentation ($p = 0.031$). The R-precision values for 5-, 6-, and 7-grams also remain unchanged while the R-precision values for 3-grams and 4-grams have increased by around 3 and 5 percentage points respectively. None of the increases in precision values are significant

($p > 0.05$). The precision@10 values for 5-, 6-, and 7-grams are similar to the value without using any dictionary augmentation, whereas 3-grams and 4-grams hurt precision@10. The precision@10 values produced by the 3-grams and 4-grams are not significantly worse than using only the CS stemmer with DICT-KBBI ($p > 0.05$), but they are significantly worse than not stemming at all ($p = 0.046$ for both). All recall values are not statistically significantly different compared to non-stemming and the CS stemmer ($p > 0.05$).

The reason shorter n-grams of size 3 and 4 produces higher retrieval precision is similar to that reason given in Section 4.7 for DICT-UI — shorter n-grams correct some of the misspelling errors and increase the number of matches during document retrieval.

4.8 Text Retrieval: Identifying and Not Stemming Proper Nouns

In the Concise Oxford English Dictionary [Soanes et al., 2004], a proper noun is defined as "a name for an individual person, place, and organisation, having an initial capital letter." A proper noun refers to a specific instance of a common noun [Mann, 2002]; examples of common nouns include "chair", "book", "person", "air", and "imagination", while examples of proper nouns include "George W. Bush", "President of the USA", "RMIT University", "TREC", and "Jakarta". Although the above definition states that proper nouns are initialised with capital letters, this is not always the case, in practice especially in the text IR environment where users may enter queries containing proper nouns in any capitalisation. Even in non-IR environments, not all proper nouns are capitalised — "the president of the USA" is an example of a proper noun that is not all capitalised, and the Indonesian acronym "narkoba" ⟨drugs⟩ is an example of a proper noun that is frequently written with different capitalisation, namely Narkoba and NARKOBA.

Thompson and Dozier [1997] state that proper nouns make up between 39% and 68% of news database queries. Proper nouns are considered to be root words, and should not be stemmed. We hypothesise that stemming proper nouns leads to decreased precision: Table 3.11 shows that up to 13% of stemming errors are caused by improperly stemming proper nouns.

We continue with an exploration of different approaches to find proper nouns in Indonesian text, and explore the impact of stemming on retrieval performance when proper nouns are excluded.

4.8.1 Proper Noun Identification and Experiments

For this experiment, we use C_INDO-TRAINING-SET explained in Section 4.1.1. We use grams of size 4 and 5 since they seem to be most promising: tokenisation with grams of size 4 in Table 4.6 and tokenisation with grams of size 5 in Table 4.7 lead to the highest MAP values, while stemming with dictionary augmentation using 5-grams produces the highest stemming accuracy as shown in Table 4.8. Stemming with dictionary augmentation using 4-grams produces the highest MAP as shown in Table 4.9.[5]

We approach proper noun identification from four aspects similar to those described by Curran and Clark [2003]. First, we identify words that are likely to be acronyms, and should therefore not be stemmed. Acronyms are typically written in uppercase (all-caps, 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, for example, in a sentence "Pasien penyalahguna narkotika dan obat-obat berbahaya (narkoba) dari kalangan keluarga miskin berhak mendapat pelayanan pengobatan gratis di rumah sakit (RS)." ⟨Drug users from poor families are entitled to free drug treatments in hospitals.⟩, the words "narkoba"[6] ⟨drugs⟩ is the acronym for "narkotika dan obat-obatan terlarang" and "RS" ⟨hospital⟩ is the acronym for "Rumah Sakit". 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 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.

When conducting the preliminary study, we conjectured 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 encompasses words appearing in titles of documents or in the names of organisations or committees, such as "keterlibatan" ⟨involvement⟩, which can be stemmed to "libat" ⟨involve⟩. Not stemming such words decreases the MAP of ad hoc retrieval.

[5]The stemming accuracy and retrieval effectiveness displayed by using grams of size 6 and 7 is the same as that displayed by using grams of size 5, so we choose to use only grams of size 5.

[6]The acronym Narkoba appears with initial letter in uppercase (IU) in only 22.7% of instances, in all lowercase (AL) in 75.0% of instances, and in all uppercase (AU) in 2.3% of instances.

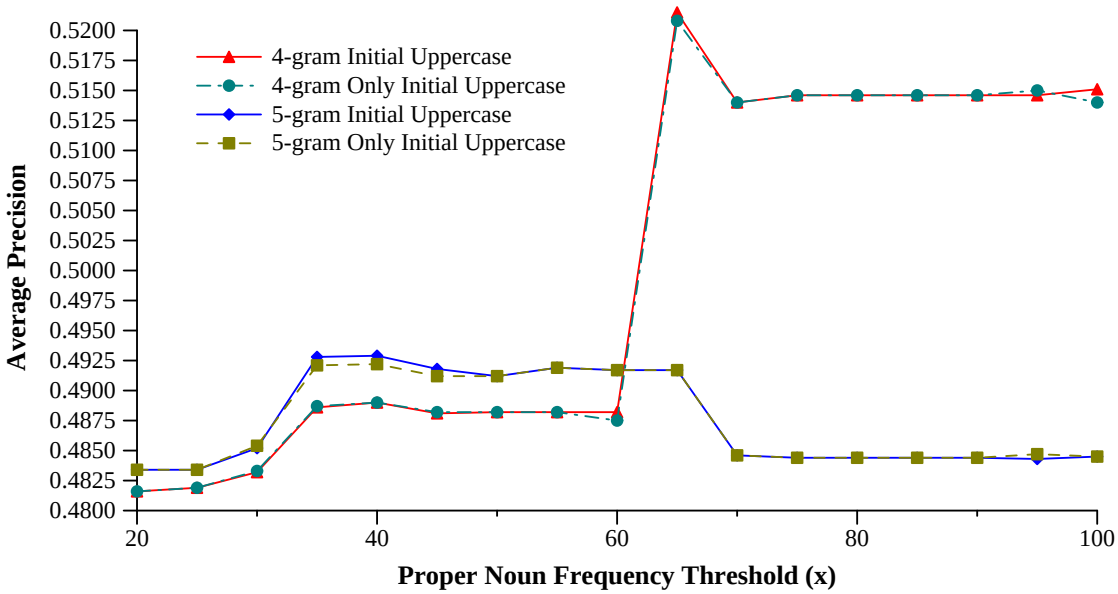

Figure 4.16: MAP for the CS *stemmer using n-grams and proper noun identification. Here, proper nouns are words that appear mid-sentence at least x times with the initial letter in uppercase (*IU*) or only initial letter in uppercase (*OIU*). The 4-gram variants overlap, as do the 5-gram variants.*

This result, and the fact that proper nouns do not always appear with consistent capitalisation, lead us to apply a required ratio between the capitalisation 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 experimentation with different threshold frequencies of words appearing in a particular way in C_INDO-TRAINING-SET. Figure 4.16 shows the retrieval effectiveness for varying thresholds. The MAP values for 4-gram IU and 4-gram OIU are quite similar, as are the MAP values for 5-gram IU and 5-gram OIU; this leads to two overlapping lines in the figure. When the threshold is exceedingly 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 MAP. In Figure 4.16, MAP peaks for a threshold of 65 for 4-grams IU and OIU and at 40 for 5-grams 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 initial letter in uppercase, OIU, (75% for 4-grams and 63% for 5-grams). Results on experiments with different combinations of IU and OIU for 4-grams

Scheme (Threshold)	Gram Size	
	4-gram	5-gram
IU (40)	0.4890	0.4929
IU (65)	0.5215	0.4917
OIU (40)	0.4890	0.4922
OIU (65)	0.5208	0.4917
IU (40) OIU (40)	0.4889	0.4922
IU (40) OIU (65)	0.4889	0.4922
IU (65) OIU (40)	0.4889	0.4922
IU (65) OIU (65)	0.5208	0.4917

Table 4.12: MAP produced by IU, OIU, *and combinations of* IU *and* OIU *with different thresholds. All these thresholds give the highest MAP for* IU *and* OIU *for either 4-grams or 5-grams.*

and 5-grams that produce the highest MAP are shown in Table 4.12.[7] These indicate that a threshold of 65 — equal to 62% of proper nouns in IU and 75% in OIU — is appropriate. We use this ratio, along with n-grams of size 4 for further experiments as BEST-IU and BEST-OIU in Section 4.8.2.

We consider any words in 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.[8] These documents comprise content from the Associated Press (AP), the San Jose Mercury (SJM), the Wall Street Journal (WSJ), the Financial Times (FT), and the Los Angeles Times (LATimes). We consider all words in these documents as English words as long as they do not start with numbers. We remove any words in C_INDO-TRAINING-SET that start and end with numbers, such as "10.35" and "rp23,575" ⟨23,575⟩, and, if the remaining words in C_INDO-TRAINING-SET also occur in our English word list, they are considered as English words. Using this definition, 894 717 words (15 271 unique) or around 43% of the words in C_INDO-TRAINING-SET are EW. In this way, we obtain English words and proper nouns that are not supposed to be stemmed, for example "account", "switzerland", and "indonesia". However, we also obtain some affixed Indonesian words such as "pelaksanaan" ⟨implementation⟩, de-

[7] We usually use two decimal places rounding for precision and recall results but since we want to determine the best n-gram size and IU and OIU combination, we use four decimal places to differentiate the results.

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

rived from "laksana"⟨resembling⟩, and "pendidikan" ⟨education⟩, derived from "didik"⟨to educate⟩.

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 to 4 letters from C_INDO-TRAINING-SET, and manually selecting valid titles that are typically 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 immediately following the second title to be a proper noun.

4.8.2 Results and Discussion

Table 4.13 shows the best MAP that these methods achieve. As explained earlier, the BEST-IU and BEST-OIU results were obtained using a threshold of 65. For the techniques shown in the last two rows, the word lists are combined by merging proper nouns from one list with those of another, and removing duplicates. The symbol † is used to indicate a statistically significant difference compared to CS stemmer without any extension ($p < 0.05$).

Except for EW, applying each individual technique increases MAP, although only the increase created by WAT is significantly better than the CS stemmer ($p = 0.049$, one-sided Wilcoxon signed ranked test). None of these increases are significantly better than not stemming despite the 6 percentage-point increase for BEST-IU and BEST-OIU ($p > 0.05$). Compared to no stemming, applying AU, PIU, and WAT hurts precision@10 significantly ($p = 0.038$, $p = 0.046$, and $p = 0.019$ respectively). The R-precision values produced by BEST-IU and BEST-OIU are also significantly worse than no stemming ($p = 0.011$ for both).

Each individual technique can be combined (after duplicates are removed). In Table 4.13, we show only those combinations that gives highest MAP, R-precision, and precision@10 plus the combination of all techniques. The effect of combining different techniques is additive; if one technique increases MAP, combining it with another technique that also increases MAP will increase MAP further. The same can be said for a technique that reduces MAP: combining it with other techniques usually lowers the final MAP. The highest MAP (0.53), the highest R-precision (0.51), and the highest precision@10 (0.37) are achieved by the combination

Scheme	MAP	R-Prec	Prec@10	Recall	Rel Retrieved
Unstemmed	0.46	0.42	0.38	0.73	330
CS	0.48	0.45	0.36	**0.78**	**354**
AU	0.51	0.50	0.35	0.78	352
PIU	0.51	0.50	0.35	**0.78**	**354**
BEST-IU	0.52	**0.51**	**0.37**	**0.78**	353
BEST-OIU	0.52	**0.51**	**0.37**	**0.78**	353
EW	0.47	0.43	0.36	**0.78**	**354**
WAT	0.51 †	0.50	0.34	**0.78**	353
AU+PIU+BEST-IU+BEST-OIU+WAT	**0.53**	**0.51**	**0.37**	**0.78**	353
ALL	0.48	0.45	**0.37**	**0.78**	353

Table 4.13: Precision and recall for queries using proper noun identification for not stemming, stemming with CS*, and stemming with* CS *extended with* 4*-gram dictionary augmentation when not stemming proper nouns. The symbol † is used to indicate a statistically significant difference compared to* CS *stemmer without any extension. Multiple acronyms indicate technique combinations, where proper noun lists of component techniques are merged. Key:* AU*=All Uppercases,* PIU*=Parentheses Initial Uppercase,* BEST-IU*=Best Initial Uppercase,* BEST-OIU*=Best Only Initial Uppercase,* EW*=English Words,* WAT*=Words After Titles,* ALL*=All Combinations.*

of AU+PIU+BEST-IU, AU+PIU+BEST-OIU, AU+PIU+BEST-IU+BEST-OIU, and AU+PIU+BEST-IU+BEST-OIU+WAT. We choose to use AU+PIU+BEST-IU+BEST-OIU+WAT in further experiments as it contains the most complete proper noun list. These increases are not statistically significantly better than the CS stemmer. The MAP ($p = 0.035$) and R-precision ($p = 0.011$) produced by these combinations are significantly better than no stemming. Combining all techniques increases only precision@10.

Adding a proper noun identification component does not significantly affect recall value. Using n-grams and proper noun identification may increase precision, but not necessarily recall. Adding proper noun identification causes some words not to be stemmed; this in turn leads to fewer matches between queries and documents. None of the recall values produced by these individual and combination techniques is statistically significantly different from the recall produced by no-stemming or the unmodified CS stemmer ($p > 0.05$).

We then combine stopping using the stopword list that produces the highest MAP (VEGA-STOP2), stemming using the CS stemmer with the best proper noun identification scheme (AU+PIU+BEST-IU+BEST-OIU+WAT), and dictionary augmentation using 4-grams. This produces MAP, R-precision, precision@10, and recall of 0.55, 0.54, 0.40, and 0.79 (359 relevant retrieved out of 453 relevant documents), respectively. All these values are higher than the precision and recall produced when not stemming and when using the unmodified CS stemmer. Of all these increases, the significant ones are the increase in R-precision compared to not stemming ($p = 0.006$) and to the unmodified CS stemmer ($p = 0.014$), the increase in precision@10 compared to the unmodified CS stemmer ($p = 0.021$), and the increase in recall compared to not stemming ($p = 0.038$).

The queries for both not stemming and the CS stemmer with some proper noun identification techniques are quite similar: many keywords are incorrectly considered to be proper nouns and are therefore not stemmed. Examples of such words include "hubungan" ⟨relationship⟩, "laporan" ⟨report⟩, "pertandingan" ⟨contest⟩, and "kenaikan" ⟨increase⟩ that should be stemmed to "hubung" ⟨be connected⟩, "lapor" ⟨to report⟩, "tanding" ⟨match⟩, and "naik" ⟨to ascend⟩, respectively. The increase in MAP is more likely to be caused by a small number of queries that perform very well, leading to an increase in the overall mean average precision. As Sanderson and Zobel [2005] point out, we need at least fifty queries for significant results and a stable system.

Despite the lack of statistical significance when compared to the CS stemmer, combining our proper noun identification scheme with this stemmer and dictionary augmentation using 4-grams still increases precision, more so when we also include stopping. We have shown that MAP can be significantly increased by stemming while excludes proper nouns; satisfactory results can be obtained using a proper noun list comprising words that are all in uppercase (AU), words inside brackets that are capitalised (PIU), a portion of words that are capitalised and are not located at the beginning of a sentence (IU), a portion of words that have only the first letter capitalised and are not located at the beginning of a sentence (OIU), and words that appear immediately after titles. For conclusive confirmation of this hypothesis, further experiments are required with different and more comprehensive testbeds of Indonesian text. Since we do not presently have access to such collections, we must leave this to future work.

4.9 Text Retrieval: Language Identification

With the increasing number of languages used for Web documents, it is essential to be able to identify the language of a document, so as to be able to return answer documents in particular languages requested by users [Lins and Gonçalves, 2004]. In multilingual environments, such as pan-European companies, language identification helps in the automatic classification of, and response to, electronic messages [Zhdanova, 2002]. Systems for natural language processing tasks such as question answering and automatic translation, also require awareness of the language of a document [Padró and Padró, 2004]. Automatic language identification is useful for our Indonesian text retrieval tasks: by knowing whether a document is in Indonesian, we can decide whether we can apply our CS stemmer or stop documents using an Indonesian stopword list. Language identification can also be used when crawling the Web to build a corpus of only Indonesian documents [Vega, 2003].

Padró and Padró [2004] summarise different approaches to language identification. Some approaches use special features of the language such as unique letter sequences (for example, "vnd" for French and "eux" for German) [Dunning, 1994], diacritics (for example, "é", "à", and "ö"), and special characters (for example, "ß" for German). Other methods use statistics, including a set of most common words [Dunning, 1994], low-order n-gram models [Churcher et al., 1994], text categorisation based on n-grams [Cavnar and Trenkle, 1994], and visible Markov models [Padró and Padró, 2004], to determine the language of a document

Indonesian does not have any distinguishing characters; moreover, it is common to have some foreign words in a text document, especially in web documents. We hypothesise that statistical methods are more likely to successfully discriminate whether a document is in Indonesian. In preliminary work, Vega and Bressan [2001] have used the weighted 3-gram approach to decide whether a document is in Indonesian; they conclude that, although the results are encouraging, further language modelling is required to produce a robust system.

The simplest statistical method uses word statistics, and works well for a sufficiently large volume of training data [Dunning, 1994]. We first collect word statistics from a training set of different languages. We use Indonesian, Malay, and English as our training languages. Malay was chosen because it has the same origin as Indonesian, as mentioned in Section 2.1, while English was chosen because Indonesian documents often contain English words. To determine the language of a document in the test set, we count how many words of the test document, not including numbers, occur in each of the Indonesian, Malay, and English word lists; we assume that the document language is the one with the most matches. If there is a

Language	Type	Total	Source/URL
Indonesian	Training	4 950	http://www.kompas.com
	Test 1	2 475	http://www.kompas.com
	Test 2	2 473	http://www.kompas.com
	Test 3	88	www.bbc.co.uk/indonesian
	Test 4	5 435	http://www.antara.co.id/
Malay	Training	3 990	http://www.bharian.com.my/
	Test 1	1 994	http://www.bharian.com.my/
	Test 2	1 994	http://www.bharian.com.my/
	Test 3	45	http://www.ukm.my/ukmportal/
English	Training	37 261	Wall Street Journal (1990-1992)
	Test 1	18 630	Wall Street Journal (1990-1992)
	Test 2	18 629	Wall Street Journal (1990-1992)
	Test 3	208	http://yallara.cs.rmit.edu.au/~imsuyoto/
	Test 4	13 273	http://news.bbc.co.uk/2/hi/

Table 4.14: Training and test data sets for language identification. The column "Total" refers to number of documents in each data type, which can be either training or test. The source for the English corpus are the Wall Street Journal articles from TREC-8 [Voorhees and Harman, 1999]; while the sources for the Indonesian training, Test 1, and Test 2 collections are our 9 898-documents first described in Section 3.3.1.

tie, we report a match to both languages. If the words do not occur in any of the languages in the training set word lists, we mark the language of document as "unidentified".

The lists of training and test collections are shown in Table 4.14. To investigate the robustness of our method, we use documents from different domains for the training and test sets; for example, we use the Wall Street Journal articles as our English training, and the Wall Street Journal articles, English documents obtained from the homepage of an Indonesian student, and BBC articles as our test sets. The student's homepage was chosen to ensure that our method works well for documents that are not news articles and may contain some informal words.

To measure how well our identification algorithm works, we calculate the identification precision, which we define as:

Actual Language	Type		Identified As					
	Test Set	Total	Indonesian		Malay		English	
			%	Total	%	Total	%	Total
Indonesian	1	2 475	**99.96**	2 474	0.04	1	0.00	0
	2	2 473	**100.00**	2 473	0.00	0	0.00	0
	3	88	**100.00**	88	6.00	3	0.00	0
	4	5 435	**99.98**	5 434	0.06	3	0.00	0
Malay	1	1 994	0.00	0	**99.75**	1 989	0.30	6
	2	1 994	0.00	0	**100.00**	1 994	0.00	0
	3	45	0.00	0	**100.00**	45	0.00	0
English	1	18 630	0.00	0	0.00	0	**100.00**	18 630
	2	18 629	0.00	0	0.00	0	**100.00**	18 629
	3	158	0.00	0	0.00	0	**100.00**	158
	4	13 273	0.00	0	0.00	0	**100.00**	13 273

Table 4.15: The actual language of documents and the language identified by our algorithm. Some documents are identified with two languages, causing the sum of total percentages of some test sets to be more than 100%, and the total number of identified documents to sometimes be more that the actual number of documents in the test set. Precision for a particular test collection is set in bold.

$$\text{Precision} = \frac{\text{Number of documents identified as of language L}}{\text{Number of documents of language L}} \tag{4.3}$$

4.9.1 Results and Discussion

Table 4.15 shows the precision values of each of the test collections in bold, together with the language incorrectly associated with the documents. Our approach is most effective on English (precision consistently 100%), then Indonesian (precision varying from 99.98% to 100.00%), and then Malay (precision varying from 99.75% to 100.00%). This is similar to the results reported for identifying other languages such as English and Spanish (precision varying from 92.00% to 99.90%) [Dunning, 1994]; and English, Indonesian, Malay, German, and Tagalog (precision varying from 89% to 100% using a tiny set of documents: seventeen Indonesian and four Malay documents, with only one document for each of the other lan-

guages [Vega and Bressan, 2001].[9] When the "identified as" percentages for each test set are summed up, the total may sometime be more than 100%, since our algorithm allows a document to be identified with more than one language. For example, two documents in Indonesian Test Set 4 were identified as both Malay and Indonesian because the two documents have an equal number of words identified as Malay and Indonesian. Although the total number of documents in Test Set 4 is 5 435, the total sum of number of documents identified as Indonesian, Malay, or English is 5 437 (equivalent to 100.03%).

In analysis, we have determined that the primary reason that some documents are not identified correctly is that they have a mixture of languages. Many documents in our Malay test collections have both Malay and English sentences, with the number of English sentences sometimes much larger than the number of Malay sentences, making it hard to define the actual language of the document. Dictionary size also plays an important part in determining identification precision. The number of unique words in our English, Malay, and Indonesian training collections are 293 537, 36 847, and 53 170 respectively. Our Indonesian word list is relatively small and does not contain words such as "pemanggil" ⟨caller⟩ and "pemotong" ⟨cutter⟩ that are valid words for both Indonesian and Malay, while our Malay word list contains these words. As a result, documents with such words are more likely to be identified as Malay. Our Indonesian word list also does not contain proper nouns, such as "Milwaukee", "Cheung", "Potter", and "Rowling", that exist in the English or Malay word lists.

Furthermore, some Malay documents use some English words such as "music" and "computer" without any transliteration, whereas the same words are transliterated to "musik" and "komputer" in Indonesian — the words "music" and "computer" do not exist in our Indonesian dictionary. This makes documents with proper nouns such as "Universal Music" and "Apple Computer" more likely to be considered as Malay or English rather than Indonesian.

To increase identification precision, we can increase the dictionary size and the range of domains covered by the training collection.

4.10 Text Retrieval: Compound Word Splitting and Identification

Compound words are formed by two or more words. As discussed in Section 2.1.5, Indonesian compound words usually have a new meaning that is different from each component; these

[9] We do not report the results of Vega and Bressan with higher precision because the number of documents is very small.

are called opaque or non-compositional compound words [Hedlund, 2001]. Indonesian compound words are usually not written together, for example, "darah daging" ⟨blood relation⟩, consisting of "darah" ⟨blood⟩ and "daging" ⟨flesh⟩, "ikut serta" ⟨participate⟩, consisting of "ikut" ⟨to follow⟩ and "serta" ⟨along with, as well as⟩, unless they have a prefix and a suffix, for example "keikutsertaan" ⟨participation⟩ from "*ke-ikut serta-an*". Exceptions to this rule include "bulutangkis" ⟨badminton⟩, composed of "bulu" ⟨feather⟩ and "tangkis" ⟨repel⟩, and "beritahu" ⟨to inform⟩, composed of "*beri*" ⟨to give⟩ and "*tahu* ⟨to know⟩"; in such cases, the combined words are widely accepted as the "right" format.

The process of splitting compound words into their components is called compound splitting — also known as decompounding [Hollink et al., 2004]. In an IR environment, compound words in documents should be split before being indexed and compound words in queries should be split before evaluation with the expectation of increasing MAP. Since decompounding can cause topic drift, where the components of the compound word have different meanings from the intended meaning of the compound words, Hollink et al. [2004] suggest adding the original compound words to the query. For example, the compound word "headstand" has a different meaning from its component words "head" and "stand", therefore all three words "head", "stand", and "headstand" need to be included in the query.

Although decompounding is expected to increase MAP in both monolingual and cross-lingual IR, Hedlund et al. [2002] state that it may only help in increasing precision of some queries. Airio [2006] reports that decompounding may increase MAP, especially when the queries are phrases and the documents to be retrieved are compound words. However, the increase may not be significant, and in some cases decompounding decreases precision. One of the examples given by Airio [2006] is the English phrase "maternity leave" and the German compound word "Mutterschaft<u>s</u>urlaub"; the latter consists of Mutterschaft ⟨mother⟩ and "Urlaub" ⟨leave⟩ with an "s" in the middle that acts as connecting character (underlined in this example). Without decompounding, the documents retrieved only contain the word "Mutterschaftsurlaub" whereas with decompounding documents containing the words "Mutterschaft" and "Urlaub" are also retrieved. Hollink et al. [2004] indicate that the increase in MAP is more marked when stemming follows decompounding. Airio [2006] and Hollink et al. [2004] agree that the differences in performance depend on the language.

Decompounding may not increase MAP for an Indonesian corpus because most Indonesian compound words are written separately, except when both a prefix and a suffix are added to the compound word as explained earlier.

Hollink et al. [2004] state that even though some decompounding algorithms require extensive knowledge of the language, their own algorithm uses only a word list. Our decompounding or compound word identification algorithm is similar to the algorithm listed by Hollink et al., but extended to split and identify compound words. Since we do not know which words in the collection are compound words, this algorithm serves primarily as a compound word identifier.

We try to find compound words from C_INDO-TRAINING-SET consisting of 6 898 documents by iteratively breaking a string in the collection into two substrings (Indonesian compound words usually consist of two words) and checking whether the two substrings are in the DICT-UI dictionary. If they are, we assume the mixed string to be a compound word. With this algorithm, different combinations of compound words are possible as long as both components are in the dictionary. The algorithm of Hollink et al. [2004] caters for European languages that often use a linking element such as an "s". Since Indonesian does not use any linking element to form a compound word, we do not need to cater for this. However, some prefix and suffix combinations lead our algorithm to incorrectly identify a word as a compound word. These combinations are the prefix "ke-", or "se-", or "men-" with either the suffix "-i" or "-an" and the prefix "pe-' with the suffix "-an". Our algorithm does not consider words with these affix combinations to be compound words.

4.10.1 Results and Discussion

This algorithm classifies 15 563 words (991 unique words), or around 0.75%, out of 2 085 203 words from C_INDO-TRAINING-SET are compound words. We have manually analysed these 991 unique words and categorised the answers as "correct", "incorrect", and "unknown" when the annotator was not sure whether the word is a compound word.[10] We find that only 2 042 words (130 unique) are recognised correctly as compounds. Some 12 806 words (711 unique) are incorrectly identified as compound words, while the remaining 715 words are categorised as unknown.

Words are incorrectly identified as compound words for reasons similar to those that can cause stemming to fail. The first type of words that are mistakenly identified as compound words are proper nouns. For example, while the proper nouns "abdul" and "gani" are in the dictionary, the proper noun "abdulgani" is not, therefore "abdulgani" is mistakenly identified as a compound word. Some words with affixes are also incorrectly considered as

[10]Not all compound words are listed in a published dictionary.

compound words. Words such as "keperempat" ⟨the one fourth⟩, deriving from "ke-"+"per-"+"empat" ⟨four⟩, and "agamawan" ⟨religionist⟩, deriving from "agama" ⟨religion⟩+"wan", are incorrectly assumed to derive from "keper" ⟨twilled cloth⟩ + "empat" and "agam" ⟨virile⟩ + "awan" ⟨cloud⟩, respectively. Some misspelt words or words with missing spaces in between are also incorrectly identified as compound words. For example, the misspelt word "kasalahan" is mistakenly assumed to be derived from "kasa" ⟨gauze⟩ and "lahan" ⟨terrain⟩, while in fact it is a misspelling of the word "kesalahan" ⟨mistake⟩. Words such as "anggota" ⟨member⟩ and "penting" ⟨important⟩ are wrongly written without any space in between to become "anggotapenting", and our algorithm wrongly identifies it as a compound word. The last type of words that are mistakenly identified are foreign words such as "laundering", which is assumed to derive from "laun" ⟨to delay⟩[11] and "dering" ⟨tinkling sound⟩.

Of all these errors, the one that we can possibly fix is the type of error where affixed words are mistakenly considered as compound words. For example, the words "perantara" ⟨mediator⟩ and "pendahulu" ⟨predecessor⟩ are assumed to be composed of "per" ⟨per⟩ + "antara" ⟨between⟩ and "pen" ⟨pen⟩ + "dahulu" ⟨previous⟩ while they are actually words that have been prefixed with the variants of the prefix "pe-". We can try to remove these affixes first, but doing so may dismiss some valid compound words. For example, if we try to remove the prefix "pe-" and its variants, valid compound words such as "peranserta" ⟨to participate⟩, consisting of "peran" ⟨character⟩ and "serta" ⟨along with⟩ will not be identified.[12]

Given that Indonesian compound words are not usually written together unless they are prefixed and suffixed first,[13] and that the number of compound words correctly identified by our algorithm are less than 1% of words in the whole collection, we hypothesise that compound words identification and splitting does not merit further investigation.

4.11 Summary

In this chapter, we have described the first publicly available testbed for Indonesian text retrieval. It includes 3 000 documents from newswire dispatches, 20 topics, and exhaustive

[11] "Laun" is usually added after "lambat" to have any meaning (the meaning is "very slowly").

[12] The compound words "peran serta" and "ikut serta" both mean "to participate".

[13] Since Indonesian compound words are prefixed and suffixed, they need to be stemmed first before the compound word can be identified. However, stemming may not work if the compound word is not in the dictionary. For example, the stem of the compound word "keikutsertaan" is "ikut serta". In order to identify the compound word, the prefix "ke-" and the suffix "-an" need to be removed, since our stemming algorithm requires the stem to be in the dictionary; it will not be able to stem "ikut serta" since this compound word is not in the DICT-UI dictionary.

relevance judgements. The testbed is stored in the TREC format, and can be used in TREC-like ad hoc evaluations with standard TREC retrieval and evaluation tools.

Using this testbed, we have explored several well-known text retrieval techniques and applied them to Indonesian text. We have discovered that using only the title of the TREC-like queries produces the highest MAP. Since this reflects the length of typical web queries, we choose to use only the title for the queries in subsequent experiments. We have experimented with different normalisation pivot values for the cosine measure, and b and k_1 values for the Okapi BM25 measure. For the cosine measure, a pivot value of 0.95 produces the highest MAP and R-precision, although the results are not significantly better than not using any pivot. For our Indonesian text collection, the optimum Okapi BM25 setting for b is 0.95, and for k_1 is 8.4; these are different from the optimum settings for the English TREC-8 collection (b=0.75 and k_1=1.2). The difference in MAP between the optimum setting for Indonesian and the optimum English setting is not statistically significant. Since the differences in MAP values between cosine and Okapi BM25 measures are not significant, we proceed to report results of our text retrieval experiments using the optimum Okapi BM25 setting for English.

We have also experimented with stopping using frequency-based and semantic-based stopword lists. Stopping using a frequency-based stopword list, which contains the most frequent words in the training collection, decreases MAP, while stopping using semantic-based stopwords — words that do not contribute much information and serve as grammatical markers — generally increases MAP. The highest increase is produced by using the VEGA-STOP2 list although the increase is not statistically significant compared to the unstopped version.

Stemming — even using the S_V-1 algorithm that performs poorly in the stemming experiments of Chapter 3 — increases MAP, even though the increases are not statistically significant compared to no stemming. Using the DICT-UI dictionary rather than DICT-KBBI not only produces higher stemming accuracy but also leads to higher R-precision, both for stemming with and without dictionary augmentation. Consequently, we use DICT-UI for our text retrieval experiments. Combining stopping and stemming produces even higher MAP compared to unstopped and unstemmed queries and documents, even though the increase is not statistically significant. Stopping, stemming, and combining both stopping and stemming, generally increases recall.

Tokenisation, a form of language-independent stemming, has varieties that span or do not span word boundaries. Grams of size 4 produce the highest MAP when not spanning word boundaries, while grams of size 5 produce the highest MAP when spanning word boundaries.

These MAP values are higher than not stemming or stemming using the CS stemmer, although the differences are not significant. We recommend that word boundaries be spanned, as this produces the highest MAP.

We have also tried to find alternative spellings or correct misspellings by using n-grams for words that cannot be stemmed, and have explored different methods to find the closest distance between misspelt words with words in the dictionary. We have discovered that the Q-gram method performs best; this dictionary augmentation using grams of size 3 and 4 can handle some of the misspellings and increase MAP values, while the stemming accuracy and MAP produced by grams of size 5, 6, and 7 are similar to the stemming accuracy and MAP produced by CS without any dictionary augmentation. Dictionary augmentation with grams of size 4 produces the highest MAP.

Stemming all words, except proper nouns, with dictionary augmentation can also increase MAP, but not necessarily recall. We have described schemes for identifying proper nouns in Indonesian text. We observe that the best result is achieved by stemming all words, except for proper nouns identified using PIU+AU+BEST-IU+BEST-OIU+WAT with additional dictionary augmentation using 4-grams. The MAP and R-precision values are statistically significantly better than no stemming, but not significantly better than CS without dictionary augmentation. The threshold we recommend for BEST-IU and BEST-OIU using 4-grams is 65, which is equivalent to 62% of all words in IU list and 75% of all words in OIU list.

We have also experimented with language and compound word identification using simple methods. Our language identification method using word frequency accurately identifies English documents, but is slightly less accurate at identifying Indonesian and Malay documents. This is largely due to the disparity in the training set sizes. Our compound word identification algorithm does not perform well, although the errors, such as proper nouns and foreign words being mistakenly identified as compound words, are inherent to natural language processing tasks that rely on a dictionary. Since compound words make up less than 1% of our training set, we consider further investigation to be unproductive.

Our Indonesian collection is relatively small, and larger corpora and further experiments are required for more significant results. However, the techniques outlined in this work represent a considerable advancement in the literature on Indonesian text retrieval.

Another promising avenue for future research is query expansion, which can increase retrieval effectiveness [Abdelali et al., 2007]. We need to find the appropriate terms to include and how many terms to include for the expansion to produce optimum results [Billerbeck and Zobel, 2004]. We can also add a query-biased summary feature, which shows sentences

or fragments of sentences to indicate how the query terms appear in a document as employed by some popular search engines such as Google, to allow users to judge quickly whether a document is relevant without needing to retrieve the whole document [Tombros and Sanderson, 1998]. It would need to be investigated whether query-biased summaries need to be customised for Indonesian. We can also optimise the various effective techniques considered to make them more efficient.

In the following chapter, we investigate automatic identification of English and Indonesian parallel documents using the contents of the documents.

Chapter 5

Identification of Indonesian-English Parallel Documents

Having discussed how a range of existing IR techniques can be incorporated into an Indonesian text retrieval system in Chapter 4, we now move on to a different topic — automatic identification of parallel documents. A document is deemed parallel to another document if it is a direct translation [Sadat et al., 2002]. If the content of the two documents is essentially similar, but not a direct translation, then the texts are called comparable [Braschler et al., 1999; Demner-Fushman and Oard, 2003]. As highlighted in Section 2.4, parallel corpora are useful resources for CLIR research as they are the basic building blocks for bi-directional testbeds and translation dictionaries [Nie and Chen, 2002; Resnik and Smith, 2003].

Parallel corpora are also useful for natural language processing (NLP) tasks. They can be used to build lexical resources such as bilingual dictionaries or ontologies for general or domain-specific texts. Examples include the work of Widdows et al. [2002], who propose the use of vector space models and cosine similarity measures to correlate words between parallel documents. Their results show that words that are highly correlated are likely to be translations or synonyms of each other. Gale and Church [1993] demonstrate that parallel corpora can be used to align sentences, which can then be used to build bilingual concordances and probabilistic dictionaries for machine translation. Kupiec [1993] uses parallel corpora to build bilingual collocations, to disambiguate word-sense, and to map nouns and noun phrases.

In this chapter, we propose methods for the automatic construction of bilingual corpora from web data. Some web sites, such as newspaper archives and government resources,

contain versions of the same document in multiple languages. These versions may be identical, or may be redrafted to suit the target audience; and articles in some languages tend to be more verbose than in others. Since the naming or organisation of the website cannot be relied on as a mechanism for identifying which documents correspond to each other, it is necessary to use the document content for matching.

Our method is designed for languages that share a character set. In this chapter, we report on results using Indonesian and English. In Chapter 6, we investigate the effectiveness of our techniques for English, French, and German documents.

As a baseline, we use simple information retrieval techniques to locate matches. We either leave the corpus in the original language, or substitute the words in a corpus in one language with the words in another language. As we show, this baseline can be effective, with the matching document typically ranked in the top five answer positions when the words in the documents are substituted, and in the top ten positions when the documents are not substituted. However, our alignment method is as precise as the baseline in identifying parallel documents. It is even better than the baseline in indicating the separation between the parallel and non-parallel documents when no word substitution is involved. Stopping also helps in increasing separation, especially for collections where words have been substituted. Stemming helps only slightly.

The remainder of this chapter is structured as follows. We report on related work that has considered the identification of parallel documents in Section 5.1. Our approach for aligning windows of words is presented in Section 5.2. The experimental framework used to test our approaches — including collections, queries, and evaluation metrics — is explained in Section 5.3. We present results and discussion in Section 5.4. Finally, we summarise our results in Section 5.5.

5.1 Background

Previous work on finding parallel documents has focused on two main features: the external structure of files, such as filenames or URLs; or the internal structure of the files, such as file structure elements and sentence alignment. In this section, we explore previous approaches to identifying relevant documents, and discuss possible associated problems.

5.1.1 External Structures

The simplest method for finding parallel documents is to use their filenames or URLs [Chen et al., 2004; Chen and Nie, 2000; Kraaij et al., 2003; Nie et al., 1999]. This method assumes that documents in different languages usually have the same filename (such as `intro.htm` for both English and Indonesian) or use labels to indicate their languages (`intro-english.htm` and `intro-indo.htm`). The documents could also be located in corresponding directories such as `http://students.idp.com/indonesian/aboutaustralia/default.asp` and `http://students.idp.com/english/aboutaustralia/default.asp`. When the URLs or filenames are organised consistently and systematically in this way, they can be relied on to identify parallel documents.

A variant of this method involves analysing parent pages and sibling pages on the Web [Kraaij et al., 2003; Resnik and Smith, 2003]. A parent page contains links to documents with similar content in different languages, whereas a sibling page includes both content and links to different documents in different languages. An example of a parent page is an Australian education institution web site `http://students.idp.com` that has links to parallel documents in English, Japanese, Indonesian, Chinese, and Spanish. A shortcoming of this method is that filenames or URLs often do not follow a consistent naming convention [Chen et al., 2004]. Consider the following URLs containing parallel documents in Indonesian and in English, `http://www.antara.co.id/seenws/?id=36431` and `http://www.antara.co.id/en/seenws/?id=14960`. From the URLs alone, it is not obvious that these are parallel documents and other information is required to identify parallel documents.

Another simple method is to analyse anchor text of documents based on the assumption that relevant documents in different languages may refer to each other [Kraaij et al., 2003]. For example, an English document may have the anchor text "Indonesian version". By following this link, the user can view the parallel page in Indonesian. This page may in turn contain the anchor text "versi Inggris" ⟨English version⟩ that takes the user back to the English page. This method analyses queries containing requests for specific anchor text, such as "Indonesian version", and specific languages such as "English". Therefore, a query log is required to see which documents are returned by the search engine; these documents are assumed to be parallel. There are two disadvantages of this method. First, it is dependent on a query log that may not be easily accessible. Second, this method only works for documents that provide links to parallel documents.

<html> <head> <title>BBC NEWS | Europe | No traces of Milosevic poisoning</title> <body marginwidth="0" topmargin="0" leftmargin="0" marginheight="0"> Independent Dutch tests on the body of Slobodan Milosevic show no signs that he was poisoned, the international war crimes tribunal in The Hague has said.	<html> <head> <title>BBCIndonesia.com | Berita Dunia | Tak ada 'tanda' Milosevic diracuni</title> <body bgcolor="#FFFFFF"> Pengadilan kejahatan perang PBB di Den Haag mengatakan, hasil sementara tes toksologi terhadap jenazah mantan pemimpin Yugoslavia Slobodan Milosevic tidak menunjukkan tanda-tanda dia diracun.

Figure 5.1: The left and right texts are parallel HTML sources in English and Indonesian respectively. They are adapted following the style of Resnik and Smith [2003]. These are incomplete HTML sources from the BBC newswire.

5.1.2 Internal Structures

Resnik and Smith [2003] present a method of identifying parallel documents by aligning document mark-up tags. This method relies on the presence of mark-up tags such as in the tags in HTML documents. Examples of aligned mark-up tags are shown in Figure 5.1, which shows two aligned HTML source files. These sources are categorised into different tokens, for example start [`Start:token_type`], end [`End:token_type`] and data tokens together with the length [`Chunk:length`]. The length in the chunk is defined as the number of characters other than the white spaces. The differences between the tokens of two HTML documents are then compared, as shown in Table 5.1. If the difference falls below a certain threshold, the documents are considered as candidates for being parallel. The drawback of this method is that it is only applicable to marked-up documents, and even then such documents may not always share a similar internal structure [Chen and Nie, 2000].

Sentences can be aligned using terms that convey meaning [Chen et al., 2004], that is, using words other than stop words. Cognates — words that have similar roots — can also be

English	Indonesian
[Start:HTML]	[Start:HTML]
[Start:Title]	[Start:Title]
[Chunk:43]	[Chunk:59]
[End:Title]	[End:Title]
[Start:Body]	[Start:Body]
———	[Start:Div_StoryText]
[Chunk:129]	[Chunk:167]

Table 5.1: Aligned HTML tokens for documents in Figure 5.1. These tokens are adapted from Resnik and Smith [2003].

used for aligning sentences [Chen and Nie, 2000]. Example of cognates include the English word "father", the German word "Vater", and the Latin word "pater", all of which share the same root [Soanes et al., 2004]. This method is not applicable for Indonesian and English, as they do not share the same origin.

Yang and Li [2004] propose aligning sentences and titles using statistical methods and lexical information. Their approaches require extensive linguistic knowledge of the two languages to be aligned, so the number of potential language pairs is limited. In addition, a large amount of data is needed to build the statistical information used for the alignment process.

Pike and Melamed [2004] propose a text mapping algorithm for the identification of parallel documents. The approach requires that sentences in parallel documents have been properly aligned. Pike and Melamed categorise documents that are parallel but not aligned at sentence level as "comparable". The SIMR-cl method is not applicable for our corpus since sentence boundaries are not necessarily preserved for our Indonesian-English corpus, our collection size is relatively small, and there is no exact one-to-one mapping between English and Indonesian words. For example, the two Indonesian sentences:
"Kami optimis pertemuan Tim Teknis kedua negara pada 22 Maret akan ada penyelesaian yang baik," katanya. Ia juga mengatakan hendaknya masalah itu tidak perlu dibesar-besarkan. ⟨"We are optimistic that the technical team meeting of the two countries on 22 March will find a good resolution," he said. He also said that this matter should not be exaggerated.⟩
can be written as one sentence in English:

"We are optimistic that a meeting of the technical team of the two countries on March 22, 2005 will reach a comprehensive resettlement of the dispute," he said, calling on all sides not to exaggerate the problem.".
Furthermore, the SIMR [Melamed, 2000] method, on which SIMR-cl is based, assumes a one-to-one word mapping between the source and target language. The word "drug" in English can also map to the words "narkotika dan obat-obat terlarang" in Indonesian.

Landauer and Littman [1990] proposes the Latent Semantic Indexing (LSI) technique that relies on the premise that words with the same meaning or synonyms should share similar statistical frequency in order to build co-occurrence statistics of the words in the two languages to be aligned. This assumption may not hold for parallel Indonesian-English documents. An Indonesian document may use the name of a former Indonesian president "Megawati" repeatedly, while the parallel English document may change it to "she" after the first few instances of "Megawati".

5.2 Windowed Alignment

Previous techniques for identifying parallel documents require consistent file naming and structuring conventions, or access to query logs. More advanced schemes require prior knowledge of either semantic or statistical information, which can only be derived if sufficient data is available, and make assumptions about the documents, for example that the sentences in parallel documents need to be properly aligned, and that words in parallel documents have similar term frequency. In this chapter, we introduce a new approach that extends the Needleman and Wunsch [1970] algorithm for global sequence alignment, and does not require prior statistical or semantic knowledge of the documents. It also does not assume documents have to be structured in a certain way before alignments can be done.

Our algorithms align candidate documents using variants of the global alignment methods developed for applications such as protein sequence search. In most applications of global alignments, two strings are regarded as a match if they share many symbols in the same order. In this application, we focus on corpora in languages with a Latin alphabet, and treat words as symbols; even untranslated versions of parallel documents will share symbols such as proper nouns and loan words. As an example, consider the parallel Indonesian and English documents of Figure 5.2, drawn from the Antara newswire service. As can be seen, the documents have some words in common (such as "sea games", "Roberto Pagdanganan", "Chi", and "2003") as well as simple variations such as compounds ("Ha Noi"/"Hanoi").

Vietnam Ready To Help Philippines Organise 23rd Sea Games

Ha Noi (ANTARA News/VNA) - Viet Nam's National Committee for Physical Training and Sports (NCPTS) has agreed in principle to lend the Philippines electronic equipment free of charge to organise the 23rd SEA Games, said a NCPTS official.

Duong Nghiep Chi, director of the NCPTS's Institute of Physical Training and Sports Sciences, said that Chairman of the Philippines SEA Games Organising Committee Roberto Pagdanganan has expressed his thanks for Viet Nam's goodwill, and that the Philippines would send a delegation to Viet Nam for detailed discussions on the issue in the coming time.

The electronic equipment used in the 22nd Southeast Asian (SEA) Games in Ha Noi in December 2003, has proven to meet both regional and world standards, Chi said, adding that Viet Nam would also help the Philippines with software processing.

Vietnam Siap Bantu Filipina Selenggarakan Sea Games

Hanoi (ANTARA News/VNA) - Komite Nasional Untuk Kesegaran Jasmani dan Olahraga Vietnam (NCPTS) pada prinsipnya setuju untuk meminjamkan peralatan elektronik tanpa biaya kepada Filipina untuk menyelenggarakan SEA Games ke-23, kata seorang ofisial NCPTS.

Duong Nghiep Chi, Direktur Institut Kesegaran Jasmani dan Ilmu Pengetahuan Olahraga NCPTS mengatakan, ketua penyelenggara SEA Games, Roberto Pagdanganan, menyatakan terima kasihnya atas maksud baik Vietnam, dan Filipina akan mengirim perutusan ke Vietnam untuk mengadakan pembicaraan lebih lanjut menyangkut masalah itu pada waktu yang akan datang.

Peralatan elektronik, yang digunakan dalam pekan olahraga Asia Tenggara (SEA Games) di Hanoi pada Desember 2003 itu, memenuhi standar regional dan dunia, kata Chi, dan menambahkan bahwa Vietnam juga akan membantu Filipina dengan prosesing perangkat lunak.

Figure 5.2: The upper and lower passages are parallel documents in English and Indonesian respectively. These are incomplete documents from our Antara newswire collection.

Simple dictionary substitution can greatly increase the number of shared words.[1] However, the word order tends to be jumbled, and there are likely to be many local permutations in ordering due to grammatical differences in the languages. Consider the words "green tea", written as "teh hijau" in Indonesian; "teh" means "tea" and "hijau" means "green". Another example is the noun phrase "National Committee" written as "Komite Nasional" in Indonesian; these appear in Figure 5.2. At a higher level, structure such as paragraph order tends to be preserved, but with breaks added or removed.

We therefore propose a method for relaxing the ordering constraint in the alignment. Instead of aligning words, we align bags or *windows* of words. We take windows of words from both the query documents and the collection documents — each window of the query document is compared against each window of the collection documents; this is different from the LSI technique [Landauer and Littman, 1990] that treats the query as a unit and using bags of words only on the collection documents. The strength of match between two windows for our method is determined by how many words they have in common. The intuition is that, if two documents contain a sequence of windows in the same order with reasonable numbers of matching words, then the documents are likely to be translations of each other. We use a predetermined fixed window length to match words between two sequences, instead of matching word by word. These fixed-length windows can overlap. We accept any matches within two windows without a particular ordering or threshold. Since our main concern is global alignment for two documents, we continue with a description of the Needleman-Wunsch algorithm that is commonly used for global alignment between two sequences.

5.2.1 The Needleman-Wunsch algorithm

Needleman and Wunsch [1970] describe a global alignment method for finding the maximum similarity of two sequences. Each sequence is treated as an ordered list of symbols; the algorithm typically assesses the similarity between the sequences by rewarding matches between identical symbols and penalising mismatches between differing symbols. We note that this scheme is used for finding global alignment between two sequences, but is not appropriate for all types of sequences. For example, while it is suitable for DNA sequences, a different scoring scheme must be used for protein sequences [Henikoff and Henikoff, 1992]. Other ap-

[1] In substitution, one word in one language is substituted with synonyms or meanings in the other language without considering which meaning is more likely. This differs from translation, where the most appropriate meaning is used.

$$\begin{array}{cccccccc} G & A & T & - & C & G & C & T \\ | & & | & & | & & & | \\ G & - & T & A & C & - & - & T \end{array}$$

Figure 5.3: Two sample genomic sequences, GATCGCT and GTACT, aligned; the dashes indicate indicate insertions or deletions; the vertical bars indicate matches.

plications of this algorithm include genome alignment [Höhl et al., 2002] for bioinformatics, string similarity measurement [Bakar et al., 2000], and network intrusion detection [Takeda, 2005].

Consider the two strings "GATCGCT" and "GTACT". These sequences can be aligned as shown in Figure 5.3. Insertions and deletions are shown as horizontal dashes while matches are shown as vertical bars.

For two sequences of length N and M, this algorithm creates a matrix with dimensions $N \times M$ as shown by the left matrix in Figure 5.4; the symbols of one sequence correspond to rows and the symbols of the other correspond to columns. A "1" is placed in each cell for a match, and a "0" is placed for a mismatch (alignment between a symbol and a dash) between elements in that particular row and column. We call this the *start matrix* (SM). In practice the SM values are determined on the fly. Another matrix, the *traversal matrix* (TM), is shown by the matrix on the right in Figure 5.4. The TM has the dimensions of $N + 1$ and $M + 1$ and is used to find the global alignment between the two sequences; it is used to store and traverse the maximum scores achieved by aligning the two sequences so far with the final traversal score located at the bottom-right of the matrix. The additional row and column of the TM are used to store initialised penalty values for insertion before traversal; in this instance, they are initialised to 0. Non-zero initial penalty values indicate that any insertion at the beginning of a sequence are penalised. An example of insertion is the insertion of a dash between "G" and "T" in the lower sequence of Figure 5.3. This insertion in a sequence is penalised in the global alignment method. The value of the penalty can be fixed, or increased proportionally according to the gap length. We have used the gap penalty value of 0 for TM in the right matrix of Figure 5.4. We describe our global alignment algorithms and the penalty value in Section 5.2.2, in the context of our specialised application. Details of the basic algorithm can be found elsewhere [Needleman and Wunsch, 1970].

	G	T	A	C	T
G	1	0	0	0	0
A	0	0	1	0	0
T	0	1	0	0	1
C	0	0	0	1	0
G	1	0	0	0	0
C	0	0	0	1	0
T	0	1	0	0	1

(a) *Start Matrix*

	$*$	G	T	A	C	T
$*$	0	0	0	0	0	0
G	0	1	1	1	1	1
A	0	1	1	2	2	2
T	0	1	2	2	2	3
C	0	1	2	2	3	3
G	0	2	2	2	3	3
C	0	2	2	2	3	3
T	0	2	3	3	3	4

(b) *Traversal Matrix*

Figure 5.4: The start and traversal matrices of genomic sequences aligned in Figure 5.3 using the Needleman-Wunsch algorithm; the matrix on the left is the start matrix and the matrix on the right is the traversal matrix. The initialisation and gap penalty values for this traversal matrix are set to 0.

To find the maximum possible alignment between these two sequences, we need to traverse the matrix. Traversal of the TM involves identifying the maximum value derived from one of three cells: from the above, from the left, or from the diagonal above left of the current cell. While the traversal formula can vary, the essential principle is to give a higher score for traversing diagonally (reflecting a match), and to impose a penalty if the score is derived from the above or left (reflecting an insertion or deletion). The traversal starts from the first row of the matrix with direction from left to right and ends at the last row of the matrix. To identify the path of the best alignment sequence, the algorithm traces the maximum scoring path from the bottom right corner of the TM to the first cell at the top left.

5.2.2 Window-based Needleman-Wunsch

For alignment of parallel documents, instead of using one-by-one symbol comparison, as is done in all previous applications of alignment, we use *windows* of words. The degree of match between two windows is indicated by the number of words they have in common. In our method, we form windows of size k by starting at position 0, $k/2$, k, $3k/2$, and so on; that is, each window overlaps half of the window to either side. Punctuation symbols

a cat sat on a table drinking milk in a saucer i bought yesterday it drank it quickly

yesterday a white cat sat on a mat next to a saucer it has come here frequently lately i do not know who it belongs to

Figure 5.5: The upper and lower texts are sample documents used to illustrate our window-based alignment algorithms. The text has been case folded and punctuation marks have been removed.

are removed before forming the windows. Hyphens within words, as in "buku-buku" or "state-of-the-art" are kept since Indonesian plural words are often hyphenated as shown in Section 2.1.5. There are not many words with an apostrophe in Indonesian. Words such as "Jum'at" ⟨Friday⟩ and "Qur'an" ⟨Quran, Koran⟩ can be written as "Jumat" and "Quran", and we choose to write them together rather than separating them because they are part of a word.

In our experiments, we use window sizes that are multiples of 4, with a minimum size of 8. If the number of words in the last window is less than the chosen window size, it is left unaltered. Consider the example document shown at the top of Figure 5.5. If a window of size 8 is used to group words in this document, the windows would be:

W_a={a, cat, sat, on, a, table, drinking, milk}

W_b={a, table, drinking, milk, in, a, saucer, i}

W_c={in, a, saucer, i, bought, yesterday, it, drank}

W_d={bought, yesterday, it, drank, it, quickly}

Using windows of words has several advantages. First, even without substitution, documents in two languages — such as Indonesian and English — that share an alphabet, are likely to have some words in common, such as proper nouns, as shown in Figure 5.2. Even if the vast majority of words are mismatches, windows of a reasonable size may well have words in common, allowing the alignment to progress. Second, use of windows helps to manage the resulting noise when simple dictionary substitution is used. A dictionary may not necessarily contain all terms; there are out-of-vocabulary words [Hall and Dowling, 1980; Nwesri et al., 2006] such as "Roberto Pagdanganan" and "NCPTS" that cannot be discarded, as they may be proper nouns that are highly indicative of related content. Third, words often have multiple translations. The English word "scholar" can be translated as "anak sekolah"

Document 2

Document 1		w_1	w_2	w_3	w_4	w_5	w_6
	w_a	4	3	1	0	0	0
	w_b	1	2	2	1	1	0
	w_c	2	2	3	2	2	1
	w_d	1	0	1	1	1	1

Figure 5.6: The start matrix built by using windows of words from the documents in Figure 5.5. Rows are formed from windows of words of the top document, and columns are formed from windows of words of the bottom document. The numbers in the cells show the number of unique words in common between windows. A window size of eight is used in this example.

⟨school kid⟩; "mahasiswa yang dapat beasiswa" ⟨tertiary students with scholarship⟩; "sarjana" ⟨graduates⟩; "terpelajar" ⟨learned⟩; or "ahli" ⟨experts⟩. With a windowing approach, all translations can be included. Using windows of words also allows us to overcome restrictions on word order, which could affect noun phrases such as "teh hijau" ⟨green tea⟩ as illustrated earlier. While the basic premise of Needleman and Wunsch [1970] is to maintain the order of sequences, this is not suitable for natural language documents where word order is not rigid. A further advantage is that use of windows reduces the cost of the alignment, because the number of windows is much smaller than the number of words.

In the next sections, we introduce two algorithms based on the window-based Needleman-Wunsch approach. Algorithm 1 uses a constant penalty value for insertions, whereas Algorithm 2 uses linearly increasing penalty values for insertions.

Algorithm 1

To illustrate the alignment process, we now work through an example, using the two documents shown in Figure 5.5. First, we build the two-dimensional matrix Start Matrix (SM) where the rows are windows from the top document, and the columns are windows from the bottom document. We label windows of the upper document W_a, W_b, and so on; similarly,

Document 2

Document 1

	$w_{\#}$	w_1	w_2	w_3	w_4	w_5	w_6
$w_{\$}$	0	−1	−3	−5	−7	−9	−11
w_a	−1	4	3	2	1	0	−1
w_b	−3	3	6	5	4	3	2
w_c	−5	2	5	9	8	7	6
w_d	−7	1	4	8	10	9	8

Figure 5.7: A completed TM with OGP of −1 and EGP of −2 for Algorithm 1, based on the SM in Figure 5.6. The $W_{\$}$ row indicates the initialisation row and the $W_{\#}$ column indicates the initialisation column.

windows of the lower document are labeled W_1, W_2, and so on. To simplify the similarity calculation, we ignore duplicates within windows. The first window of the upper document is {a, cat, sat, on, table, drinking, milk} and the first window of the lower document, is {yesterday, a, white, cat, sat, on, mat}. W_a has four words in common with W_1, three words in common with W_2, one word in common with W_3, and no words in common with W_4, W_5, and W_6; this is shown as the values in the W_a row of Figure 5.6. We denote a cell in the SM in row W_i and column W_j as $SM[i, j]$; and similarly, we denote a cell in the TM in row W_i and column W_j as $TM[i, j]$.

We now describe the initialisation of penalty values. First, the TM is initialised with 0 at the $W_{\$}$ row and $W_{\#}$ column $TM[\$, \#]$ as shown in Figure 5.7; the \$ and # signs are used to indicate the row and column that contain the initialisation values. There are two types of initialisation values: the opening gap penalty and the extension gap penalty values. Both values are important in the Needleman-Wunsch algorithm. The opening gap penalty (OGP) is used when there is only one gap, such as between "T" and "C" in the genomic sequence "GAT−CGCT" in Figure 5.3. In Figure 5.7, the opening gap penalties are the initial penalty values in $TM[\$, 1]$ and $TM[a, \#]$, both are -1. Meanwhile, the extension gap penalty (EGP) is used as a penalty value when there is more than one consecutive gap, such as between "C" and "T" in the sequence "G−TAC−−T". The EGPs are dependent on the initial EGP e and the OGP g: the value of an EGP at k positions from the starting

point is $e_k = g + (k - 1 \times e)$. In this example, OGP $= -1$ and EGP $= -2$. We explain our choice of penalty values in Section 5.3.4. These EGP values are shown at row $W_\$$, starting with -3 and ending with -11, and at column $W_\#$, starting with -3 and ending with -7.

After initialisation of the TM with penalty values at the $W_\$$ row and the $W_\#$ column, the subsequent rows of the TM are computed horizontally starting from the W_a row in Figure 5.7. Each cell of the TM matrix, $TM[i, j]$, is computed as:

$$TM[i, j] = max(a, b, c)$$

where

$$\begin{aligned} a &= TM[i-1, j] + g \\ b &= TM[i, j-1] + g \\ c &= TM[i-1, j-1] + SM[i, j] \end{aligned}$$

This means that the value of each cell $TM[i, j]$ is the maximum of three possible values: from above (a), left (b) and above left (c) values of $TM[i, j]$. Only the value of c is influenced by the SM. Therefore, matching words in corresponding windows increase the traversal score. The opening gap penalty g is used in equations a and b to minimise results derived from matches occurring vertically or horizontally, reflecting a mismatch or an insertion. The value in the top left cell of the traversal matrix $TM(a, 1)$ is the maximum of:, $TM[\$, 1] + g = -2$, $TM[a, \#] + g = -2$, and $TM[\$, \#] + SM[a, 1] = 4$. In this case, the maximum value is 4. The remaining cells of the matrix are calculated in a similar fashion. Finally, the overall similarity value V is located at the bottom-right cell of the TM. In our running example, $V = 8$.

As we try to find the parallel equivalent of a document in the other language, there will be multiple documents in the other language; for a query document in one language and N documents in a second language, there will be N such values. We rank the similarity values V of each document in decreasing order assuming the one ranked at the top to be the parallel document.

We may find sets of documents that have identical similarity scores V, for example in Table 5.2, document `21` and `36` have the same scores of 34. We assign the same rank to all documents in such a set; we use the average rank of the documents in the set as the rank. As the result, both document `21` and `36` have the same ranking of 3.5.

Doc ID	Similarity Score	Rank
34	90	1
23	85	2
21	34	3.5
36	34	3.5
78	33	5
56	23	6

Table 5.2: Similarity scores for various documents after being ranked by decreasing order. It is possible for documents to have same similarity scores as shown in document 21 and 36.

The answer document with the top similarity value is considered to be parallel to the query document. Clearly, in some cases there may be no equivalent, or the equivalent may not be ranked at the top.

Since we are interested only in the similarity scores rather than the actual alignment between the two documents, we do not trace the maximum scoring path from the bottom-right hand corner of the TM for both Algorithm 1 and Algorithm 2.

Algorithm 2

This algorithm extends Algorithm 1 and uses the same SM as in Figure 5.6. It also uses the same initial opening gap penalty and extension gap penalty values as Algorithm 1. However, it uses different definitions of a and b in the traversal process. The a and b values in this algorithm are affected by the gap length — the longer the gap length, the bigger the penalty value. Instead of using an opening gap as the penalty value in the calculation, extension gap penalties are used:

$$\begin{aligned} a &= TM[i-1, j] + e_j \\ b &= TM[i, j-1] + e_i \\ c &= TM[i-1, j-1] + SM[i, j] \end{aligned}$$

where e_j is the extension penalty at column j, and e_i is the extension gap penalty at row i. Recall that an EGP at k positions from the starting point is formulated as $e_k = g + (k - 1 \times e)$. Using Algorithm 2, $TM(a, 1)$ is the maximum of: $TM[\$, 1] + e_1 = -2$, and $TM[a, \#] + e_a = -2$, and $TM[\$, \#] + SM[a, 1] = 4$. The final result of this traversal is shown in Figure 5.8.

Document 2

Document 1	$w_{\#}$	w_1	w_2	w_3	w_4	w_5	w_6
$w_{\$}$	0	−1	−3	−5	−7	−9	−11
w_a	−1	4	3	2	1	0	−1
w_b	−3	3	6	5	3	2	0
w_c	−5	2	5	9	7	5	3
w_d	−7	1	2	6	10	8	6

Figure 5.8: A completed TM with OGP of −1 and EGP of −2 for Algorithm 2 based on the SM of Figure 5.6. The $W_{\$}$ row indicates the initialisation row and the $W_{\#}$ column indicates the initialisation column.

The overall similarity score, V, is again located at the bottom-right corner of the matrix (in the example, $V = 6$). Similar to Algorithm 1, when there are documents with same similarity scores, the rankings are averaged. Note that when no extension gap penalty is applied ($e = 0$), Algorithm 2 is identical to Algorithm 1.

Algorithm 1 and Algorithm 2 do not produce a binary decision to indicate for each document whether it is parallel or not. Instead, they assign similarity scores to the documents, based on the difference between each document. The top ranked documents are assumed to be most likely to be parallel. We investigate the effectiveness of our methods in the next section.

5.3 Experimental Framework

To explore the effectiveness of our methods for identifying parallel documents, we performed experiments on several corpora collected from the Web. We conducted training experiments on two collections to identify effective alignment parameters, and then tested the effectiveness of these parameters on a third collection.

HSBC Indonesia Business Banking - Investasi
Investasikan dana anda sesuai dengan kebutuhan anda, dengan perlindangan modal penuh dan menjamin kembalinya dana anda secara maksimal.
-Surat obligasi (Debt Securities)
-Pinjaman usaha (Corporate Bonds)
-Pemodalan usaha (Recapitalization Bonds)

HSBC Indonesia Business Banking - Investments
Invest your business' surplus cash in a product that meets your investment objectives, whether you want full capital protection or to maximise your return.
-Debt Securities
-Corporate Bonds
-Recapitalization Bonds

Figure 5.9: The upper and lower passages are parallel documents in English and Indonesian respectively. These are incomplete documents from the HSBC Indonesia web site.

5.3.1 Collections

We constructed three collections of parallel documents covering a variety of topics.

Collection A. This collection consists of 1 007 English and Indonesian documents that are known to be parallel, based only on the URLs that adhere to a pattern. The institution names, the number of documents from each domain, and the seed URLs from which we started the crawl to build collection A are listed in Table 5.3. These documents include typical homepages of institutions; an example is shown in Figure 5.9. The average lengths of documents for the English and Indonesian collections are 265 and 274 words respectively. The queries used are 10 of the Indonesian documents from these collections, chosen at random.

Collection B. This collection is made up of 1 964 English documents and 5 615 Indonesian documents from the Antara[2] web site, with an average document length of 304 and 300 words respectively. A sample document can be seen in Figure 5.2.

[2] `http://www.antara.co.id/en` (English) and `http://www.antara.co.id` (Indonesian)

Institution	Type	No of documents	URL
IDP	Education	139	http://students.idp.com/
Allianz	Company	144	http://www.allianz.co.id/
BI		85	http://www.bi.go.id/web/
BII		318	http://www.bii.co.id/
HSBC		109	http://www.hsbc.co.id/
Telkom		194	http://www.telkom-indonesia.com/
Qantas		18	http://www.qantas.com.au/international/
Total		1 007	

Table 5.3: List of institution names that constitute collection A along with the number of documents and the seed URL for each domain.

To form the testbed, we manually identified the Indonesian document corresponding to each English document; there is no definable pattern for equivalence based on the URL structure or filename. Therefore, methods to identify parallel documents based on URL patterns, such as those discussed in Section 5.1.1, cannot be applied for this collection. For the queries, we chose 10 Indonesian documents that had parallel equivalents in English.

Collection C. This collection contains 13 274 newswire documents from the BBC Indonesian web site,[3] added as noise to collection B. The average document length for this English collection is 520 words. For this collection, we also manually identified the Indonesian document corresponding to each English document. We chose 20 Indonesian documents that had parallel equivalents in English; these 20 queries are different from the 10 queries used in Collection B.

In collection A, all the English documents have parallel equivalents in Indonesian. That is not the case for collection B and C — not all documents have a corresponding document in the other language.

[3] http://www.bbc.co.uk/indonesian

5.3.2 Word Substitution

It is plausible that parallel documents in different languages can be successfully aligned without any translation. For example, documents could contain shared proper nouns such as "Vietnam" and "Jakarta", or phrases such as "cum laude". To test this, we used both unsubstituted and substituted versions of our collections. Thus the unsubstituted collection consists of English and Indonesian documents, and the substituted collection consists of Indonesian documents and English documents in which words have been substituted by Indonesian words through simple dictionary lookups. We refer to these collections as C_SUBSTITUTED and C_UNSUBSTITUTED.

We use a bilingual dictionary that has multiple meanings for one word to cater for all possible contexts, rather than automatic machine translation tools that produce only one translation for one word. Our dictionary substitution method does not disambiguate different senses of a word. Pirkola [1998] disambiguates word senses by using two dictionaries: one general dictionary and one specialised dictionary. His method works well for disambiguating a query that consists of a few words, not a document, to retrieve the relevant documents. We do not use this disambiguation technique because we do not have a specialised dictionary available for Indonesian. Another method for word-sense disambiguation is to give different weights to query terms [Hiemstra, 2001]. The weights can be derived from statistical modelling of the collection, or assigned by users. Although Hiemstra [2001] provides different alternatives of disambiguation, his experiments indicate that structured queries, where all possible meanings of a word are included, perform significantly better than manual and automatic disambiguation, where only the best sense is included in CLIR tasks. We therefore do not consider word-sense disambiguation.

We choose the English-Indonesian dictionary[4] created by Hantarto Widjaja containing 14 562 entries. We choose this dictionary because it is free, is in a downloadable text form rather than in an online interactive dictionary form, and contains multiple meanings to a word. Table 5.4 shows the extract of English to Indonesian substitution dictionary.

The substitution process we used is straightforward except for number formatting. If a word is found in the dictionary, it is replaced by its possible meanings; no attempt is made to choose correct translations. If a word is not in the dictionary, it is left unchanged.

[4]This dictionary was obtained from `http://www.geocities.com/CapeCanaveral/1999/kamus.zip`. It was made to help Indonesian speakers to read English text, and does not include some common words.

English Word	Indonesian Substitution
iron	besi, setrika, rantai, belenggu
jam	selai
jade	batu permata hijau
lay up	menimbun
qualm	ragu-ragu, sangsi, sesal, mual
wrought iron	besi tempa

Table 5.4: Extract of English to Indonesian dictionary. During substitution, all meanings are used and the comma signs are removed.

For example, for English words are "no qualm to lay up wrought iron and jam", and a substitution dictionary consisting only of words in Table 5.4, the resulting substitution will be "no ragu-ragu sangsi sesal mual to menimbun besi tempa besi setrika rantai belenggu and selai". For construction of the Start Matrix (SM), all these substitutions are included in only one matrix, not separate matrices. Since the numbering systems between English and Indonesian are different, as discussed in Section 2.1.4, we convert the numbering system during translation. For example, the numbers 1.5 and 5,000 in English are respectively converted into 1,5 and 5.000 in Indonesian.

The average length of documents after substitution for Collection A, B, and C is 468, 507, and 867 words respectively.

5.3.3 Evaluation Measures

We used two measures to judge how accurately a system identifies parallel documents, namely the mean reciprocal rank (MRR) and separation (SEP) values that we introduced in Section 2.3.5. The MRR value is used to measure how well a system identifies parallel pairs of documents based on an ordered list of candidate answer documents. The maximum MRR is 1.0, indicating that the correct parallel document is consistently ranked at position 1. The SEP value, which is the difference between the score of the highest false match (HFM) and the lowest true match (LTM), is used to measure how well a system separates parallel documents from non-parallel documents [Hoad and Zobel, 2003]. A higher SEP value means that the algorithm is better at differentiating between parallel and non-parallel documents, while a negative value indicates that the parallel document is ranked below at least one non-parallel

document. Higher SEP values give higher confidence that the documents ranked at the top are the parallel documents, and that the rest of the documents can be ignored.

We focus on SEP, rather than MRR, as our main basis for comparison, because the former gives a more meaningful indication at the confidence with which the system has been able to identify parallel documents. MRR is very sensitive to the rankings, making comparison more difficult. When the true parallel document is ranked at the top, it has MRR of 1; when it is ranked second, it has MRR of 0.5; and when it is ranked third, it has MRR of 0.333. Furthermore, we only have twenty queries for our test set; and the MRR metric may be less stable in such situations. Meanwhile, SEP values are not sensitive to the rankings but to the actual similarity value — a measure of how similar two documents are. A good system is one that can correctly categorise whether a document is parallel. Using SEP can give a measure of confidence for whether a document is parallel.

We note that it is not appropriate to rely on SEP as an absolute numerical measure of performance, since order-preserving transformation of similarity functions could impact on the SEP score. However, for any pair of scoring functions, SEP does provide a useful measure for the confidence with which a system has been able to differentiate between parallel and non-parallel documents. To increase the stability of our SEP results further, we use the full symmetric cosine measure, as shown in Equation 2.6, for our baseline approach, rather than a simplified order-preserving cosine variants as are commonly used in information retrieval [Witten et al., 1999, pages 185-187]. Moreover, our queries were documents rather than words, therefore we wanted the similarity scores to be consistent for a pair of documents, regardless of the order they are evaluated in. In this case, the symmetric cosine measure is a more suitable choice.

For our algorithms, it is possible for the similarity scores to be negative. When this occurs, the similarity scores need to be normalised before calculation of LTM and HFM can proceed. Suppose the lowest similarity score is -5 as shown in Table 5.5. We add a 6 to each of the similarity score so all similarity values will be greater than 0. The normalised scores in the rightmost column of Table 5.5 are then used to compute the LTM and HFM values as explained in Section 2.3.5.

Doc ID	Similarity Score	Normalised Score
5	14	20
7	9	15
1	6	12
3	6	12
2	2	8
6	-3	3
4	-5	1

Table 5.5: Normalised similarity scores for calculating LTM and HFM.

5.3.4 Performance Baseline and Experimental Parameters

As a baseline, we used the Zettair[5] search engine to index our three collections on both unsubstituted English documents and on Indonesian documents of which words are substituted from English. A search engine is likely to find parallel documents because it gives different weights to words depending on their frequencies in a collection — the *term frequency* (TF) and *inverse document frequency* (IDF) rules as explained in Section 2.3.4. We expected certain words such as proper nouns to carry more weight, as they appear less often in the collections (that is, they have higher inverse document frequencies). As discussed in Section 2.1.3, proper nouns are mostly written in the same way between the two languages.

We tested our alignment algorithms on both the substituted and unsubstituted versions of collections A, B, and C. We varied the window size from 8 to 32, and where applicable, used opening gaps and extension gaps from 0 to 4 to identify the combination that produced the best results.

For comparison against approximated word-by-word alignment, we used alignment with the minimum possible window size of 2.[6] For this window size, we applied small values (0 and 1 for opening gap penalties, and only 0 for extension gap penalties), as there could be at most two matches in a window of size 2.

Note that corresponding windows in parallel documents do not tend to share many words, either before or after substitution: some words are not substituted, the substitution process introduces words that are irrelevant to content, such as functional and grammatical markers,

[5]We use Zettair version 0.6.1.

[6]Our algorithms require windows overlap, which is why window of size 2 is chosen instead of 1.

		C_UNSUBSTITUTED			C_SUBSTITUTED		
Scheme	Window	A	B	Mean	A	B	Mean
Cosine baseline	-	-9.74	40.22	15.24	3.04	37.85	20.45
No penalty	2	22.09	43.15	32.62	22.20	22.47	22.33
Windowed Alignment	8	35.03	**56.99**	**46.01**	-1.07	-1.87	-1.47
	12	33.73	56.13	44.93	9.48	16.22	12.85
	16	**35.43**	56.00	45.71	12.08	26.02	19.05
	20	33.01	56.48	44.75	18.70	31.48	25.09
	24	32.83	54.46	43.64	19.96	**33.33**	26.65
	28	31.57	52.49	42.03	**22.97**	32.11	**27.54**
	32	28.29	50.08	39.19	21.78	32.64	27.21

Table 5.6: SEP values for different window sizes using training sets (collections A and B). Values in bold denote the maximum SEP for either C_UNSUBSTITUTED *or* C_SUBSTITUTED*, while italics indicate SEP values that are lower than the cosine baseline. This scheme, with an opening gap penalty (OGP) of 1 and an extension gap penalty (EGP) of 0, produces the best results for collections A and B.*

and there can be substantial differences in word order. For this reason, the matching process has to be highly tolerant, and penalties cannot be large.

5.4 Results

We divided our collections into training and test sets. Performance of the two approaches, using different parameter values on Collections A and B, was used to identify good parameter settings for our algorithms. Since collections A and B both have ten queries each, we took the average of collections A and B to determine the best penalty scheme and window size. The larger collection C was then used to verify the effectiveness of our approach.

5.4.1 Training

Table 5.6 shows the SEP values for our training collections A and B. Cosine matching does not strongly differentiate between parallel and non-parallel documents. Moreover, the SEP value for collection B is much higher than the SEP value of collection A, which indicates that

		C_UNSUBSTITUTED			C_SUBSTITUTED		
Scheme	Window	A	B	Mean	A	B	Mean
Cosine baseline	-	0.469	0.914	0.692	0.451	**1.000**	0.726
No penalty	2	0.754	0.915	0.835	**0.813**	**1.000**	**0.907**
Windowed Alignment	8	**0.860**	**0.940**	0.900	0.484	0.573	0.529
	12	0.830	0.920	0.875	0.625	0.789	0.707
	16	0.844	0.918	0.881	0.667	0.911	0.789
	20	0.883	0.929	**0.906**	0.771	**1.000**	0.885
	24	0.845	0.914	0.880	0.722	**1.000**	0.861
	28	0.857	0.918	0.888	0.773	0.950	0.862
	32	0.793	0.909	0.851	0.777	**1.000**	0.889

Table 5.7: MRR values for different window sizes using training sets (collections A and B). Values in bold denote the maximum MRR for either C_UNSUBSTITUTED *or* C_SUBSTITUTED*, while italics indicate MRR values that are lower than the cosine baseline. We show this scheme, with an opening gap penalty (OGP) of 1 and an extension gap penalty (EGP) of 0, to reflects the common MRR values produced by the scheme which produces the best SEP for collections A and B.*

cosine matching separates the parallel documents from non-parallel documents better in collection B. Alignment with window size 2, which approximates word-by-word alignment, is more effective than the cosine baseline. We show only the scheme with no penalty for window size 2, as this produces almost all the highest SEP values. The alignment with window size 2, with an opening gap penalty (OGP) of 1 and an extension gap penalty (EGP) of 0, produces SEP values of 17.07 and -31.30 for unsubstituted and substituted collection A, and 18.45 and -26.77 for collection B. Alignment with window size 2 is computationally expensive since the TM matrices are large.

For alignment with larger window sizes, there are many penalty scheme variations; from a large number of experiments, we have selected typical values, showing a range of outcomes for the unsubstituted and substituted collections. Since the alignment scheme with OGP of 1 and EGP of 0 produces overall the best SEP for the training sets, we used this to find the optimal window sizes for both C_UNSUBSTITUTED and C_SUBSTITUTED. Table 5.6 shows that window size 8 works best on average for C_UNSUBSTITUTED, while window size 28 works best

on average for C_SUBSTITUTED. Therefore, we chose window sizes 8 and 28 for experiments on the test set for C_UNSUBSTITUTED and C_SUBSTITUTED, respectively.

Table 5.7 shows the MRR values of the cosine baseline and other schemes with similar parameter settings as in Table 5.6. Cosine-based matching is quite effective according to the mean reciprocal rank (MRR) measures for unsubstituted and substituted collection B, with MRR values of 0.914 and 1.000 respectively. This is not the case for collection A, where less than half of the parallel documents for both C_UNSUBSTITUTED and C_SUBSTITUTED are ranked at the top. Alignment with window size 2 in general performs better than the cosine baseline, with the highest MRR still produced by the no-penalty scheme. Although the windowed alignment scheme with OGP 1 and EGP 0 does not always achieve the highest MRR values, it is still as good as the cosine baseline except for smaller window sizes for the substituted collection B. This is to be expected, since the baseline MRR for the substituted collection B is exceptionally high, with the perfect value of 1.000.

Tables 5.8 and 5.9 show SEP values for different penalty settings for C_UNSUBSTITUTED and C_SUBSTITUTED respectively. For the optimum window size — 8 for C_UNSUBSTITUTED and 28 for C_SUBSTITUTED — the scheme with OGP of 1 and EGP of 0 does indeed give the highest SEP value. Using higher EGP values is not beneficial because the initial penalty values at the $W_{\$}$ row and $W_{\#}$ column of the traversal matrices in Figure 5.7 and 5.8 will be too large. Larger initial penalty values can lead to negative SEP results for each aligned document. Using OGP of 1 for all collections is good because it gives small penalty values when matches do not occur diagonally; using higher OGP values has similar effects to using higher EGP values. Note that when EGP is 0, Algorithm 1 and Algorithm 2 are identical. As OGP of 1 and EGP of 0 is the best combination of penalty values for either C_UNSUBSTITUTED or C_SUBSTITUTED, we used these settings for our experiments with the test collection C.

5.4.2 Windowed Alignment Results

The results from running 20 queries on our test collection (collection C) are shown in Tables 5.10 and 5.11. We use the symbol † to indicate a particular result is statistically significantly different from the cosine baseline. The SEP value of our alignment method, with optimised parameters for the training sets, is higher than the cosine baseline for C_UNSUBSTITUTED but is lower for C_SUBSTITUTED. The difference of the SEP values between the baseline and our alignment method is not significant ($p = 0.184$, one-sided Wilcoxon signed ranked test) for C_UNSUBSTITUTED and is significant ($p = 0.034$) for C_SUBSTITUTED. For alignment win-

Scheme	OGP	EGP	A	B	Mean
No Penalty	0	0	25.04	43.38	34.21
Algorithm 1	1	0	**35.03**	**56.99**	**46.01**
	1	1	15.95	7.86	11.90
	2	0	24.27	42.13	33.20
	2	1	15.30	5.87	10.58
	2	2	11.5 0	3.19	7.35
	3	0	18.18	23.34	20.76
	3	1	13.80	3.79	8.80
	3	2	10.79	2.05	6.42
	3	3	9.10	1.41	5.25
	4	0	13.56	14.33	13.94
	4	1	12.70	1.73	7.21
	4	2	9.82	0.86	5.34
	4	3	8.56	0.59	4.58
	4	4	7.66	0.46	4.06
Algorithm 2	1	0	**35.03**	**56.99**	**46.01**
	1	1	11.33	-0.44	5.44
	2	0	24.27	42.13	33.20
	2	1	11.13	-0.90	5.12
	2	2	8.41	-1.66	3.38
	3	0	18.18	23.34	20.76
	3	1	11.08	-1.10	4.99
	3	2	8.30	-1.91	3.20
	3	3	6.81	-2.09	2.36
	4	0	13.56	14.33	13.94
	4	1	11.05	-1.28	4.89
	4	2	8.29	-2.03	3.13
	4	3	6.72	-2.27	2.23
	4	4	5.79	-2.32	1.73

Table 5.8: SEP values for the training data set on C_UNSUBSTITUTED *collection A and B with window size 8, and the averages of collection A and B with window size 8. When EGP is 0, Algorithm 1 and Algorithm 2 have identical results.*

Scheme	OGP	EGP	A	B	Mean
No Penalty	0	0	22.74	27.33	25.03
Algorithm 1	1	0	**22.97**	**32.11**	**27.54**
	1	1	15.71	9.56	12.63
	2	0	12.97	17.58	15.28
	2	1	14.76	3.19	8.97
	2	2	11.66	1.82	6.74
	3	0	4.98	6.67	5.83
	3	1	13.32	-1.54	5.89
	3	2	10.26	-1.89	4.18
	3	3	8.42	-1.45	3.49
	4	0	-1.36	0.91	-0.23
	4	1	12.22	-4.23	4.00
	4	2	9.12	-4.70	2.21
	4	3	7.28	-3.99	1.65
	4	4	6.22	-3.22	1.50
Algorithm 2	1	0	**22.97**	**32.11**	**27.54**
	1	1	6.81	-3.88	1.46
	2	0	12.97	17.58	15.28
	2	1	5.84	-6.16	-0.16
	2	2	2.52	-6.24	-1.86
	3	0	4.98	6.67	5.83
	3	1	5.57	-6.48	-0.45
	3	2	1.90	-7.57	-2.84
	3	3	0.80	-7.25	-3.23
	4	0	-1.36	0.91	-0.23
	4	1	5.21	-6.81	-0.80
	4	2	1.83	-7.79	-2.98
	4	3	0.32	-8.16	-3.92
	4	4	-0.32	-7.79	-4.05

Table 5.9: SEP values for the training data set on C_SUBSTITUTED *collection A and B, and the averages of collection A and B with window size 28. When EGP is 0, Algorithm 1 and Algorithm 2 have identical results.*

Scheme	C_UNSUBSTITUTED	C_SUBSTITUTED
Cosine baseline	34.46	32.64
Window size 2, no penalty	31.28	−31.59 †
Windowed alignment	36.32	25.50 †

Table 5.10: SEP results for test collection C. Windowed alignment uses an OGP of 1, an EGP of 0, and a window size of 8 and 28 for the C_UNSUBSTITUTED *and* C_SUBSTITUTED*, respectively (these are the optimal parameters learned from the training collections). The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.*

dow size 2, the no-penalty scheme produces the highest SEP for the unsubstituted collection. However, for window size 2 alignment of the substituted collection, the scheme with OGP 1 and EGP 0 produces a slightly better SEP value of -31.55 than the no-penalty scheme with the SEP value of -31.59. Since the difference is small and both values are negative, we still show the no-penalty results for consistency. The SEP value for alignment with window size 2 for C_SUBSTITUTED is significantly worse than the cosine baseline ($p < 0.001$), while the difference for C_UNSUBSTITUTED is not significant ($p = 0.227$).

As with the training data, the best results are observed on C_UNSUBSTITUTED. A possible reason why results are better for C_UNSUBSTITUTED is because the alignment relies largely on proper nouns. Our simplistic substitution approach introduces noise into the matching process in many cases, reducing the performance on substituted documents. In contrast, as shown in Equation 2.6, the cosine baseline incorporates the inverse document frequency rule (IDF) as the default setting. Therefore, the noise that appears in more documents contributes less weight to the similarity score of the cosine measure leading to higher SEP values for the cosine measure than for our alignment without any IDF rule. We do not incorporate an IDF rule for our alignment method because the alignment method judges each document independently without prior knowledge of the whole collection.

Mean reciprocal rank results of our experiments are shown in Table 5.11. The scheme with OGP 1 and EGP 0 has lower MRR values than the baseline. However, the differences between our alignment method and the baseline are not statistically significant for either C_UNSUBSTITUTED or the C_SUBSTITUTED ($p = 0.251$ and $p = 0.291$, respectively). For C_UNSUBSTITUTED, using the penalty of OGP 1 and EGP 0 with window size 8 is as effective as using no penalty with window size 2. Although the two systems are not significantly

Scheme	C_UNSUBSTITUTED	C_SUBSTITUTED
Cosine baseline	0.917	0.950
Window size 2, no penalty	0.917	0.125 †
Windowed alignment	0.873	0.923

Table 5.11: MRR results for test collection C. Windowed alignment uses an OGP of 1, an EGP of 0, and a window size of 8 and 28 for the C_UNSUBSTITUTED *and* C_SUBSTITUTED*, respectively (these are the optimal parameters learned from the training collections). The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.*

different in term of MRR ($p = 0.356$), there are large differences in running time, since the alignment matrices for window size 2 are significantly larger. For C_SUBSTITUTED, using window size 2 produces SEP that is significantly worse than the cosine baseline ($p < 0.001$). The choice of approach in practical settings should therefore also depend on performance constraints.

5.4.3 Discussion

Our window-based alignment algorithms are effective at identifying parallel documents. The window size is an important parameter. Alignment with small window sizes works well for C_UNSUBSTITUTED, where the matching process relies on proper nouns and phrases that are common to both documents. Alignment with larger windows works better for C_SUBSTITUTED because larger windows can cater for cases where a word in one language is substituted by several words in another language.

Our approach is more effective at separating parallel documents from non-parallel documents for C_UNSUBSTITUTED than for C_SUBSTITUTED for this Indonesian-English collection. Through failure analysis, we have identified four classes of factors that may cause non-parallel documents to be ranked highly:

(a) Some parallel documents use synonyms, or different representations of the same name. For example, “Aceh” is sometime referred to as “Nanggroe Darussalam” or “NAD”. This reduces the number of matches, and results in true parallel documents being considered less similar. This issue is particularly important for unsubstituted documents, where matching relies heavily on the presence of proper nouns. This problem also arises

with differences in representation conventions. For example, in Indonesian documents time is represented using the 24-hour clock, while English documents often use the 12-hour clock.

(b) In some documents, proper nouns are misspelt or spelt differently. Therefore, although the documents are parallel, they are not ranked highly by the alignment process. For example, the name of the son of the first president of Indonesia is sometimes spelt as "Guruh Soekarnoputra" and at other times as "Guruh Sukarnoputra". This variation is a product of the spelling reforms discussed in Section 2.1.1.

(c) Phrases or proper nouns often occur in different positions within parallel documents. Two phrases might fall into two separate windows in the query document, because there are other words in between, whereas — perhaps due to language structures — in the parallel document the two phrases appear in the same window. This phenomenon can affect the number of matches. This problem is more noticeable for alignment with smaller window sizes. When the word matches between a query and a document occur at the first row or column of the SM, or occur consecutively within one row or column in the SM, the similarity scores tend to be high.

(d) In substituted collections, where words in the documents are substituted by words in the language of the queries, the substitution process itself may cause some documents to be ranked highly, even when they are not parallel. In particular, stopwords — words that have a grammatical function but no meaning of their own, discussed in Section 2.3.2 — can lead to large numbers of matches between documents, and therefore produce misleadingly high similarity scores. The alignment process relies on the number of meanings introduced by the substitution dictionary. An effective dictionary substitutes most words by the correct meaning, leading to more matches, while a less effective dictionary can introduce misleading words into a document because all meanings of a word or a phrase are included.[7] Although this problem is more likely to occur in longer documents, that is not the case for this collection. The average length of an answer document that is ranked first is 501 words; this length is below the average length of documents for the substituted Indonesian-English collection, which is 867 words.

[7]Our definition of an effective dictionary is a dictionary that introduces a few but accurate meanings of a word depending on the context. This definition is different from the conventional meaning of a good dictionary, which is a dictionary that contains all meanings of a word.

Solutions to the first two problems, such as finding synonyms or correcting misspellings, require deeper understanding of both languages. A plausible solution for the third problem is to alter the window size.

In contrast, removing stopword noise is straightforward, as stopword lists are available. Stemming may also help by conflating terms that refer to a particular topic. We therefore experimentally evaluated the effect of stopping and stemming on the effectiveness of our approaches.

Stopping and Stemming

As described in Section 2.3.2, stopping and stemming are two widely-used information retrieval techniques that can aid the term matching process [Witten et al., 1999, pages 146–150]. Stopping involves the removal of very common terms from the collection. Stopwords are typically words that convey little or no meaning of their own, but are required for largely grammatical reasons. For our parallel document identification experiments, stopping such words would lead to matches being more likely to be based on content-bearing terms. This would be of particular importance for substituted collections; without stopping, many matches may occur in a window simply because of the presence of stopwords, rather than similar topical content.

Stemming aims to merge variant forms of words by removing common affixes — suffixes for English; prefixes, suffixes, infixes and repeated forms for Indonesian. This technique could be of benefit for our alignment method, because slight variation in matching terms would be removed, thereby increasing the frequency of a correct alignment based on topical similarity.

To test the effect of stopping and stemming, several permutations of alignment between the queries and the target documents need to be considered for both languages. For Indonesian documents and queries, we could leave them untouched, stop them, stem them, or apply both stopping and stemming. We use the three semantic-based stopword lists: TALA-STOP, VEGA-STOP1, and VEGA-STOP2 first described in Section 4.4, and the CS stemmer first described in Chapter 3. Similarly, the unsubstituted English documents could be left untouched, stopped, stemmed, or both stopped and stemmed. We use two well-known stoplists compiled by Salton and Buckley[8] and the Porter stemmer [1980].[9] The English stopword lists can be seen in Appendix I and J.

[8] `http://www.lextek.com/manuals/onix/stopwords1.html` and `http://www.lextek.com/manuals/onix/stopwords2.html`

[9] `http://tartarus.org/~martin/PorterStemmer`.

	C_UNSUBSTITUTED		C_SUBSTITUTED	
	Cosine Baseline	Windowed Alignment	Cosine Baseline	Windowed Alignment
Unstopped, unstemmed	34.46	36.32	32.64	25.50 †
Stopped	35.59	**46.65** †	**34.96**	**45.65** †
Stemmed	33.63	39.85	20.49	23.11
Stopped and stemmed	**35.64**	45.73 †	32.80	40.00

Table 5.12: SEP results for the windowed alignment scheme on test collection C with stopping and stemming. Parameters are set to the optimal values learned from the training collection with no stopping and stemming. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

For C_SUBSTITUTED — where the words in the English documents are first substituted into Indonesian words — the collection contains both English and Indonesian words. As with C_UNSUBSTITUTED, these documents can be left untouched, or stopped using English or Indonesian stoplists, or stemmed using the CS or Porter stemmer, or any combination of these.

We tested our alignment techniques using various combinations of stopping and stemming. The parameter settings used in our algorithms are those that we obtained from the training collections (A and B) as described in Section 5.4.1, with OGP 1 and EGP 0 and window size 8 for the C_UNSUBSTITUTED and window size 28 for the C_SUBSTITUTED.

The results for our experiments are presented in Tables 5.12 and 5.13. Although there are three stopwords for Indonesian and two stopwords for English, we show only the results of the variants that produce the highest SEP values. For SEP, the introduction of stopping gives a relative increase in performance of 28% ($p = 0.017$) and 79% ($p < 0.001$) for C_UNSUBSTITUTED and C_SUBSTITUTED respectively. As anticipated, stopping is very important for the alignment of substituted documents, serving to remove terms that have no topical content and otherwise lead to spurious matches. Stopping for our alignment also produces higher SEP values than stopping for the cosine baseline. The most likely reason is that stopping removes terms that occur frequently, so the impact of using an IDF rule is minimised for the cosine baseline. Stemming leads to a small improvement for C_UNSUBSTITUTED ($p = 0.131$), but significantly harms performance when applied to

	C_UNSUBSTITUTED		C_SUBSTITUTED	
	Cosine Baseline	Windowed Alignment	Cosine Baseline	Windowed Alignment
Unstopped, unstemmed	**0.917**	0.873	0.950	0.923
Stopped	**0.917**	**0.935**	**1.000**	**1.000**
Stemmed	**0.917**	0.904	0.892	0.893
Stopped and stemmed	**0.917**	0.934	0.950	0.975

Table 5.13: MRR results for the windowed alignment scheme on test collection C with stopping and stemming. Parameters are set to the optimal values learned from the training collection with no stopping and stemming. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

C_SUBSTITUTED ($p = 0.047$). Applying a combination of stopping and stemming leads to an increase in performance for both collections, $p = 0.013$ and $p < 0.001$ for C_UNSUBSTITUTED and C_SUBSTITUTED respectively, but this is dominated by the application of stopping alone.

When compared to the corresponding cosine baseline (such as comparing stopped windowed alignment with stopped cosine baseline), application of stopping and the combination of stopping and stemming both produce a significant increase in SEP ($p = 0.014$ and $p = 0.010$, respectively) for C_UNSUBSTITUTED. For the substituted collection, of the stopping and stemming variants, only the application of stopping produces a SEP value that is significantly better than the cosine baseline ($p = 0.004$).

The effect on MRR of applying stopping and stemming is shown in Table 5.13. While the trends of our alignment methods are similar to those for SEP values — stopping improves performance, particularly for the substituted collection — the performance increases achieved for MRR are not statistically significant ($p > 0.05$). Stopping or stemming offer no improvement for the cosine baseline using the unsubstituted collection, and only stopping increases MRR for the cosine baseline on the substituted collection.

As our results show, stopping can help to reduce the noise caused by non-content words in the alignment process. Stopping increases both the SEP and MRR values for both C_UNSUBSTITUTED and C_SUBSTITUTED. As the negative impact of stopwords is greater for the substituted collection (there are more stopword matches in windows), the performance increase when stopwords are removed is more evident than for C_UNSUBSTITUTED.

	C_UNSUBSTITUTED			C_SUBSTITUTED		
	A	B	C	A	B	C
Cosine without IDF	0.88	32.40	25.75	7.03	21.83	13.35
Cosine with IDF	-9.74 †	40.22 †	34.46 †	3.04	37.85 †	32.64 †

Table 5.14: SEP results for cosine baseline with and without IDF. The symbol † is used to indicate a statistically significant difference compared to not using IDF.

	C_UNSUBSTITUTED			C_SUBSTITUTED		
	A	B	C	A	B	C
Cosine without IDF	0.572	0.912	0.935	0.565	1.000	0.917
Cosine with IDF	0.469	0.914	0.917	0.451 †	1.000	0.950

Table 5.15: MRR values for cosine baseline with and without IDF. The symbol † is used to indicate a statistically significant difference compared to not using IDF.

However, stopping also causes the ranks of some parallel documents to fall. This can occur when stopping removes words that are parts of a phrase, for example the word "and" in "the meteorological and geophysical agency".

As mentioned earlier, the term frequency (TF) and inverse document frequency (IDF) may also affect SEP and MRR values. We conjecture that IDF, rather than TF, may play an important part in our alignment process. The TF is accounted for as we count the occurrence of each term per window in a document for our alignment methods. As our alignment methods traverse each document separately, rather than the corpus as a whole, it does not make use of the IDF rule. To see whether the IDF rule has a significant effect on alignment, we experiment with using and not using the IDF rule and the impact that this has on the performance of the cosine baseline. Please note that the default setting of the cosine baseline is to use the IDF rule. The results are shown in Table 5.14 and 5.15.

C_SUBSTITUTED benefits more from incorporating the IDF rule than C_UNSUBSTITUTED does. This is expected, since the introduction of noise by the substitution process is reduced by the IDF rule. However, the effects of using IDF for the cosine baseline are collection dependent. Incorporating the IDF rule increases SEP values for collection B and C, with all increases being statistically significant($p = 0.008$ and $p = 0.003$ for unsubstituted and

substituted collection B; and $p = 0.015$ and $p < 0.001$ for unsubstituted and substituted collection C). Meanwhile, using the IDF rule decreases SEP values for the unsubstituted and substituted collection A, but only the decrease for C_UNSUBSTITUTED is statistically significant ($p = 0.038$).

The trend of changes in MRR values is similar to the trend in the SEP values. Except for the substituted collection A ($p = 0.022$), all the differences in MRR values introduced by incorporating and not incorporating the IDF rule are not statistically significant ($p > 0.05$).

Since the effects of incorporating the IDF rule varies between collections, we decide that this merits further investigation using other data sets, and believe that this would be an interesting area to explore for future work.

5.5 Summary

In this chapter, we have presented two alignment algorithms that are effective in identifying parallel documents, showing high precision and discrimination as measured by mean reciprocal rank and separation. These algorithms can be applied to languages that share a character set and do not make any assumption about the external structure or the internal content of parallel documents.

We have shown that our alignment method works well in separating parallel documents for C_UNSUBSTITUTED. The cosine baseline separates parallel documents better than our alignment for C_SUBSTITUTED because it incorporates the IDF rule, while our method treats each document separately without prior knowledge of the corpus. While a simple alignment method using window size 2 can also outperform the baseline in some cases, approaches with larger windows can take advantage of situations where long phrases need to be matched, and where a word in one language needs to be mapped to many words in another. For our window-based algorithms, using an open gap penalty of 1 and an extension gap penalty of 0 results in better overall SEP results.

For unsubstituted collections, where the alignment relies on the occurrence of proper nouns and phrases, small window sizes work best. Larger window sizes work well for substituted collections because larger windows can capture mappings of one word to many possible words.

We have also investigated the use of stemming and stopping for our approach. Stopping helps for both unsubstituted and substituted collections, as it removes noise in the alignment process for both collections, and also removes inflated matches caused by the presence of

stopwords for substituted collections. Stemming does not have a significant effect on the alignment process. However, stopping has a strong effect on our windowed alignment approach, leading to significant improvements in performance compared to the baseline for both substituted and unsubstituted collections.

We conclude that our alignment method works best for C_UNSUBSTITUTED when a substitution dictionary is not available.

In Chapter 6, we test whether our alignment methods are able to identify parallel documents for English, French, and German.

Chapter 6

Identification of European Parallel Documents

In the previous chapter, we discussed automatic identification of parallel documents for Indonesian and English, and hypothesised that our methods can also work effectively for other languages, as long as they use the same character set. In this chapter, we test our hypothesis for several other language pairs to see whether our algorithms are still applicable, and investigate suitable modifications.

For this, we chose English, French, and German since they use a similar character set. These languages have some words in common, often with slight variation; for example, the word "confection" in English is written as "confection" in French and as "Konfektion" in German [Bolton, 1988, pages 224–225]. Both English and German are classified as West Germanic languages, and share some grammatical rules and vocabulary [Baugh and Cable, 2002, page 11; Bolton, 1988, pages 227–229; Fennell, 2001, pages 33–34]. English has also adopted many loan words from French, which belongs to the Latin group of languages [Barber, 1993, pages 61–62; Baugh and Cable, 2002, pages 11–12].

In our experiments with these languages, we use the parameter settings that we found to work best for our unsubstituted and substituted Indonesian and English collections, to investigate whether these settings are transferable across languages. Since stopping and stemming were demonstrated to be helpful for increasing separation (SEP) values for Indonesian and English in the previous chapter, we will also study their effect on the European corpus.

The remainder of this chapter is structured as follows. In Section 6.1, we discuss the major differences between English and each of the language that we study, Indonesian, French, and

German — the latter pair use accented characters — and how this affects our baseline approach. We then describe our experimental framework in Section 6.2, followed by results and discussion of our findings in Section 6.3. Finally, we summarise our results in Section 6.4.

6.1 Accented Characters

In Section 2.1.1, we explained that the Indonesian alphabet does not include accented characters. Similarly, English words do not use diacritics, except for some loan words such as "déjà vu" and "naïve". However, French and German words often use diacritics. The previous two words "déjà vu" and "naïve" are examples of accented French words. Examples of accented German words are "Doppelgänger" and "Götterdämmerung". French and German diacritics or accented characters are included in the ISO 8859 character set [Lunde, 1999, page 75]. Indonesian and English use the American Standard Code for Information Interchange (ASCII) character set, whereas ISO 8859 is an extended version of ASCII [Lunde, 1999, pages 74–75]. We retain all accented characters for our experiments to see whether they affect performance.

6.2 Experimental Framework

To test our windowing algorithm, we require multilingual corpora. We present the methods of collecting documents in Section 6.2.1, and explain the difference between parsing these European documents and our Indonesian corpus in Section 6.2.2.

We use the same evaluation measures used for our windowing alignment in Chapter 5 — mean reciprocal rank (MRR) and separation (SEP) values — to measure the performance of our alignment methods. Similar to the previous chapter, we also use the symmetric cosine baseline provided by the Zettair[1] search engine for both unsubstituted and substituted collections.

6.2.1 Collection

We used the official European Union (EU)[2] web site — which has parallel documents in languages including English, French, German, Italian, Spanish, Finnish, and Dutch — as our domain for collecting documents. We used certain parts of the URL structure as preliminary indicators to differentiate whether the documents crawled were in English (the URL contains

[1] We use Zettair version 0.6.1.

[2] `http://europa.eu`

the word `en`), French (the URL contains the word fr), and German (the URL contains the word de).

From the crawled results, we filtered out documents that do not contain any rendered text (for example, documents containing only links, image files, forms, or scripts redirecting to other documents). We also removed duplicates, since the same documents may be represented with different URLs. For example, the URLs

`http://europa.eu.int/grants/topics/employers/employers_en.htm`

and

`http://europa.eu.int/grants/topics/workers/workers_en.htm`

refer to the same document. We also checked the contents of the documents manually, because classifying on the basis of URL alone does not guarantee that the contents will be in the language indicated by the URL — some documents that are categorised as French or German based on the URLs are in fact English, or contain more English words than French and German words.

We employ several methods to remove documents that are not in the correct language. The first method is by counting how many English words occur in the French and German documents;[3] if most of the words in a document are English words (we use a threshold of 75%) there is a high chance that the document is in English. We consider words appearing in the list of Wall Street Journal words between 1990 and 1992 from TREC-8 [Voorhees and Harman, 1999] used by our language identification method in Section 4.9 as English words. To avoid false positives, we then manually check whether documents that are indicated as English are indeed English documents. We determine that documents that contain at least a complete sentence such as "President Barroso visited Poland for two days"[4] in English to be an English document; documents containing a lot of English proper nouns, such as "Gloucestershire", "Wiltshire", and "North Somerset",[5] but not forming a sentence, are not considered as English documents.

The other method is by using words that occur very often in one language to identify the language of a document. For example, if the words "the" or "is" occur in French or German documents, the documents are likely to contain English words. We use the words such as

[3] It is more likely for French or German documents to contain English words in our collection rather than vice versa.

[4] `http://ec.europa.eu/commission_barroso/president/index_fr.htm` accessed on 17th November 2006.

[5] `http://europa.eu.int/rapid/pressReleasesAction.do?reference=STAT/05/13&format=HTML&aged=1&language=FR&guiLanguage=en` accessed on 17th November 2006.

French Word	English Substitution
abaisse	crust, reduces
imitiez	were imitating
imitions	were imitating
notassent	might note
notassiez	might note
notassions	might note

Table 6.1: Extract of French to English dictionary. During substitution, all meanings are used and the commas are removed.

"est" ⟨is⟩ and "et" ⟨and⟩ to determine whether a document is in French, and the words such as "ist" ⟨is⟩ and "das" ⟨that⟩ to determine whether a document is in German. We still need to determine manually whether a document that is supposedly in French or German but contains the word "the" or "is" is indeed an English document using the principle explained earlier.

After removing duplicates and documents in the wrong language, we end up with 6 721 documents each for our English, French, and German corpus. On average a document contains of 616, 713, and 575 words for each of the languages, respectively. The documents in each group are either parallel or comparable to documents in another language group. Documents in each group are from different genres including home pages, and articles containing news; discussion of work permits; and European Union history. Samples of parallel documents can be seen in Figure 6.1. We chose 50 documents at random to use as our queries.

As in the previous chapter, we use both unsubstituted and substituted collections for our alignment experiments. The unsubstituted collection is that without any word substitution, which means that the documents are still in the original language. As in the previous chapter, we label these unsubstituted collections C_UNSUBSTITUTED. In the substituted collection, documents that were originally in French and German have their content words substituted with English words. We label these substituted collections C_SUBSTITUTED. We use the French to English dictionary created by the American and French Research on the Treasury of the French Language (ARTFL).[6] This dictionary is available in downloadable text form

[6] `http://machaut.uchicago.edu/frengdict.sql`

EUROPA - Your Europe - Judicial procedures Judicial procedures USEFUL INFORMATION ON NATIONAL PROVISIONS "COMPLAINTS PROCEDURE" IN CASES BROUGHT UNDER SOCIAL LEGISLATION

USEFUL INFORMATION ON NATIONAL PROVISIONS

If you feel that an administrative authority's final decision is unconstitutional (e.g. constitutes a breach of the European Convention on Human Rights), you can lodge a complaint with the Constitutional Court (Verfassungsgericht). A complaint in respect of an administrative authority's final decision which you consider to be illegal for other reasons usually falls under the jurisdiction of the Administrative Court (Verwaltungsgericht). The final decision must point this out, as well as the fact that all internal administrative procedures have been exhausted.

EUROPA - L'Europe est à vous - Procédures judiciaires Procédures judiciaires PRECISIONS UTILES SUR LES DISPOSITIONS NATIONALES LA "PROCÉDURE DE RECOURS" DANS LES AFFAIRES DE DROIT SOCIAL

PRECISIONS UTILES SUR LES DISPOSITIONS NATIONALES

Contre une décision prise en dernier ressort par une autorité administrative et entachée d'inconstitutionnalité (violation de la Convention Européenne des Droits de l'Homme, par exemple), vous avez la possibilité d'introduire un recours auprès de la Cour constitutionnelle, et contre une décision prise en dernier ressort par une autorité administrative et que vous considérez comme contraire au droit pour d'autres raisons, vous pouvez, en règle générale, introduire un recours auprès du Tribunal administratif supérieur. Cette circonstance, ainsi que le fait que les voies de droit purement administratives sont épuisées, doivent être mentionnés dans la décision de la dernière instance.

EUROPA - Europa für Sie - Gerichtliche Verfahren Gerichtliche Verfahren WISSENSWERTES ÜBER DIE NATIONALE REGELUNG ZUM "BESCHWERDEVERFAHREN" IN SOZIALRECHTSSACHEN

WISSENSWERTES ÜBER DIE NATIONALE REGELUNG

Gegen einen letztinstanzlichen Bescheid einer Verwaltungsbehörde, der mit Verfassungswidrigkeit (z.B. einem Verstoß gegen die Europäische Menschenrechts-konvention) behaftet ist, können Sie Beschwerde beim Verfassungsgerichtshof, gegen einen letztinstanzlichen Bescheid einer Verwaltungsbehörde, den Sie aus anderen Gründen für rechtswidrig halten, im Regelfall Beschwerde an den Ver-waltungsgerichtshof erheben. Auf diesen Umstand sowie darauf, daß ein ver-waltungsinternes Rechtsmittel nicht mehr zur Verfügung steht, ist im letztinstanz-lichen Bescheid hinzuweisen.

Figure 6.1: The top, middle, and bottom passages are parallel documents in English, French, and German respectively. They are incomplete documents from the European Union web site.

German Word	English Substitution
aus dem Offensichtlichen eine Tugend machen	to make a virtue of the obvious
Bachstelze	wagtail, white wagtail
Feuerverzinkungsschicht	galvanized coating, hot-dip galvanized zinc coating
gesund	daffy, fit, hale, healthful, healthy, nonhazardous, nonvenomous, salubrious, salubriously, salutarily, salutary, sanely, sound, well, wholesome, wholesomely
sich einschleichen	to creep in, to sneak in, to slip in, to steal in
sich leicht aus der Ruhe bringen lassen	to be flappable

Table 6.2: Extract of German to English dictionary. During substitution, all meanings are used and the commas are removed.

rather than in online dictionary form, and is made by an authoritative organisation. This dictionary contains 60 099 distinct entries; an extract is shown in Table 6.1. For the German to English dictionary, we use the word list of Paul Hemetsberger.[7] We choose this dictionary as it is available in downloadable text form. We have 175 195 entries in this German to English dictionary of which an extract is shown in Table 6.2. We did not make French-to-German or German-to-French substitution because our knowledge of both languages is limited.

For the substituted collection, we substitute not only single words but also phrases. For example, if we encounter the phrase "vingt deux" ⟨twenty two⟩, the substitution results are "twenty" (for "vingt"), "twenty two" (for "vingt deux"), and "two" (for "deux"). In this case, we need to provide a look-ahead value which is the maximum number of words in a phrase. The look-ahead value is dependent on our dictionary entries; we use the value of 3 for French-to-English substitution, and 13 for German-to-English substitution. These numbers are chosen because they are the length of the longest phrase in the dictionary. The average number of words for French and German document collections after substitution are 912 and 2089 words.

[7] `http://www.dict.cc`

6.2.2 Parsing

The parsing for our European corpus is slightly different from parsing for the Indonesian corpus. For the Indonesian corpus, we retain hyphens, as they usually indicate plurals. For European collections, hyphens can be removed; hyphenated words such as "feed-back", "check-list", and "re-orientation" are written as "feedback", "checklist", and "reorientation". There are cases where hyphenated words should not be merged; examples include "Internet-based", "pollution-reduction", and "most-favoured-nation", which become "Internetbased", "pollutionreduction", and "mostfavourednation" respectively. It is difficult to automatically distinguish between these two classes of hyphenated words at initial pass unless some preprocessing such as comparing each hyphenated words against a dictionary is performed. Since hyphenated words are not very common — they make up less than 1% of each of the English, French, and German collections — we choose to consistently remove all hyphens.

French words often use apostrophes (') between words such as "l'Université" ⟨the University⟩ and "d'une" ⟨of one⟩. For these cases, we remove the apostrophe and separate the words so the previous examples are split into two words "l Université" and "d une". These words need to be separated because the dictionary only contains separate entries without the apostrophe; we can find "Université" and "une" in the dictionary but not "l'Université" or "d'une". The "l" and "d" act more like stopwords in French.

6.3 Results And Discussion

Table 6.3 shows that the separation (SEP) values for the symmetric cosine baseline and for our alignment algorithm with optimum parameters for identifying Indonesian and English parallel documents — window size 8, OGP of 1, EGP of 0 for C_UNSUBSTITUTED; window size 28, OGP of 1, EGP of 0 for C_SUBSTITUTED. With the exception of unsubstituted French-German and substituted English-German collections, all SEP values produced by our alignment algorithms are higher than the cosine baseline. The improvement is most for unsubstituted English-French and English-German collections. Negative SEP values, displayed by the unsubstituted English-German collection for the cosine baseline and unsubstituted French-German collection for both the cosine baseline and alignment method, indicate that the system cannot differentiate true parallel documents from the rest. The symbol † is used to indicate that the result is statistically significant ($p < 0.05$), either better or worse, compared to the symmetric cosine baseline. All SEP values are significantly

	C_UNSUBSTITUTED		C_SUBSTITUTED	
	Baseline	Alignment	Baseline	Alignment
English-French	20.48	46.51 †	37.22	40.73 †
English-German	−2.77	30.88 †	33.01	8.34 †
French-German	−31.08	−42.24 †	—	—

Table 6.3: SEP values for the cosine baseline and alignment algorithms using the optimum values on both unsubstituted and substituted for the Indonesian and English collection (Window size 8, OGP 1, EGP 0 for C_UNSUBSTITUTED*; Window size 28, OGP 1, EGP 0 for* C_SUBSTITUTED*). The substitution from French to German or German to French is not done. The symbol † is used to indicate that a statistically significant difference compared to the cosine baseline.*

different from the cosine baseline ($p = 0.004$ for unsubstituted French-German, $p = 0.036$ for substituted English-French, and $p < 0.001$ for the rest; one-sided Wilcoxon signed ranked test).

Table 6.4 shows the mean reciprocal rank (MRR) results for the cosine baseline and our windowed alignment using the parameter setting that we found to be optimal for the Indonesian-English collections. In terms of MRR, our alignment results are statistically significantly better than the cosine baseline for unsubstituted English-French and English-German collections ($p < 0.001$). Our alignment methods produce MRR values that are statistically significantly worse than the MRR values of the cosine baseline for unsubstituted French-German and substituted English-German collections ($p < 0.001$). The decrease in MRR for substituted English-French is not statistically significant ($p = 0.159$).

Alignment results for C_UNSUBSTITUTED in most cases are better than the baseline due to the cosine measure relying on unique words (*term frequency* and *inverse document frequency* rule). Since our European collection does not contain a lot of unique words, only query documents with unique words such as the URL `www.poland.gov.pl`, which is parsed to `wwwpolandgovpl`; proper nouns such as `Valentinas Junokas` and `Pierluigi Vigna`; and acronyms such as `SPECIALIUJU TYRIMU TARNYBA (STT)`, can be easily distinguished hence having higher SEP values than their non-parallel counterparts.

Using the parameter settings that are optimal for the Indonesian-English collection does not necessarily increase SEP and MRR values for our alignment methods; we hypothesise that

	C_UNSUBSTITUTED		C_SUBSTITUTED	
	Baseline	Alignment	Baseline	Alignment
English-French	0.838	0.954 †	1.000	0.987
English-German	0.545	0.847 †	0.987	0.691 †
French-German	0.332	0.207 †	—	—

Table 6.4: MRR values for the cosine baseline and alignment algorithms using the optimum values on both unsubstituted and substituted for the Indonesian and English collection (Window size 8, OGP 1, EGP 0 for C_UNSUBSTITUTED*; Window size 28, OGP 1, EGP 0 for* C_SUBSTITUTED*). The substitution from French to German or German to French is not done. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.*

the results can be improved further by using customised parameter settings. Before deciding which setting to use, we analyse what causes non-parallel documents to be considered as parallel documents.

There are several problems that challenge any parallel document identification method. Some of the problems have been discussed in Chapter 5. There are additional problems discussed below that may occur in any collection; we discuss these using examples from our English, French, and German collections.

(a) Some parallel documents across two languages may not always have a match — proper nouns may have been normalised or transliterated, and some words may have been written in different formats. For example, the month "May" is written as "Mai" in French and German, and "July" as "Juillet" in French and as "Juli" in German; the proper nouns "Korea" as "Corée" in French, and "Brazil" as "Brésil" in French and as "Brasilien" in German; the words "Thème" ⟨topic⟩ in French as "Thema" in German, "commission" as "kommision" in German; and "article" as "artikel" in German. This problem is more apparent for French and German documents as they do not have as many words in common as English and French. In our collection, English and German have fewer words in common than English and French, but they still have more words in common than French and German. Numbers and words can also be written differently even in the same language, for example, "15-25" can be written as "between 15 and 25", and "€317 million" can be written "317 millions euros".

(b) There are some artefacts that create false matches when they occur often. If a document contains links to other documents in different languages such as "de" (German), "fr" (French), and "en" (English) and links to different types of documents such as "html", "pdf", and "doc", it will match any documents containing such artefacts. These matches are usually small, perhaps only 1 or 2 words in one window occurring randomly as opposed to 4 words or more in one window occurring diagonally for the true parallel documents in the alignment matrix.

(c) Documents may contain phrases, proper nouns, or words in common with the language of the query. A German or French document may contain English phrases such as "delegation of the European commission" and "information centre of the European Union". Many German documents contain the word "in" (with the same meaning as the English word "in"); documents containing such words are sometimes ranked higher than the actual parallel documents, especially when the actual parallel documents contain few common proper nouns or phrases.

(d) The alignment process relies on the quality of the substitution dictionary. In Section 5.4.3, we have stated that substitution process introduce noise into the translated documents. The amount of noise introduced is directly related to the quality of the dictionary. An effective dictionary substitutes most words to the correct meaning, hence leading to more matches, while a less effective substitution dictionary introduces more words into a document because all meanings of a word or a phrase are included. Our French-to-English dictionary is effective for our alignment purpose as it introduces few but accurate meanings, whereas our German-to-English dictionary is not as good because it introduces more meanings, hence more noise, into the substituted documents.

The first problem is inherent in any natural language processing task, and we will not delve into it further as it requires deeper understanding of these languages. The number of artefacts, which is the second cause for true parallel documents being inaccurately identified, can be reduced by using larger penalty values. Stopping can help in removing some of the foreign words, for example the words "in", "also" ⟨thus⟩, and "an" ⟨on⟩ are in the German stopword list although they may not have the same meaning as the similarly-spelt English words. Using different window sizes and stopping may solve the last problem. Large window sizes can capture more proper nouns, numbers, and phrases in one window even when there are many incorrect substitutions. When an effective substitution dictionary is used,

	C_UNSUBSTITUTED				C_SUBSTITUTED			
	Parameter			SEP	Parameter			SEP
	Window Size	OGP	EGP		Window Size	OGP	EGP	
English-French	12	1	0	48.62 †	24	1	0	41.79 †
English-German	16	1	0	31.24 †	60	1	0	17.76 †
French-German	28	1	1	0.07 †	—	—	—	—

Table 6.5: The best SEP values for both the unsubstituted and substituted European collections are produced with parameters settings that are different from the optimum parameter settings for English and Indonesian. OGP is Opening Gap Penalty and EGP is Extension Gap Penalty. The substitution from French to German or German to French is not done. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

a smaller window size is generally more beneficial. After finding the best window size and penalty schemes, we can apply stopping and stemming to see whether they help in correct identification of parallel documents.

To identify the parameter combinations that produce the best results for our European corpus, we experiment with different window sizes and only stop when the SEP values start decreasing, and use opening gaps and extension gaps in the range from 0 to 3 when applicable. We use smaller penalty values since, as discussed in the previous chapter, larger penalty values are not beneficial and lead to negative SEP results.

Table 6.5 shows the combinations of window size, opening gap penalty (OGP), and extension gap penalty (EGP) that produce the highest SEP values for a particular collection. These are different from the optimal parameters found for the Indonesian-English corpus. Except for the substituted English-French collection, the window sizes are bigger than the optimum window size for the Indonesian-English collection. The unsubstituted French-German collection is handled best using OGP of 1 and EGP of 1 while the rest use the same penalty settings as Indonesian-English collection — OGP of 1 and EGP of 0. Except for substituted English-German, of which SEP is significantly worse ($p < 0.001$), all the SEP values produced by the alignment method are significantly better than SEP values produced by the cosine baseline ($p = 0.009$ for substituted English-German and $p < 0.001$ for C_UNSUBSTITUTED).

	C_UNSUBSTITUTED				C_SUBSTITUTED			
	Parameter			MRR	Parameter			MRR
	Window Size	OGP	EGP		Window Size	OGP	EGP	
English-French	12	1	0	0.978 †	24	1	0	1.000
English-German	16	1	0	0.861 †	60	1	0	0.922 †
French-German	28	1	1	0.543 †	—	—	—	—

Table 6.6: The best MRR values for both the unsubstituted and substituted European collections are produced with parameter settings that are different from the optimum parameters for English and Indonesian. OGP is Opening Gap Penalty and EGP is Extension Gap Penalty. The substitution from French to German or German to French is not done. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

Table 6.6 shows corresponding MRR values produced with the parameter setting affording the highest SEP values for the European collection. All MRR values produced by these new optimum settings are also higher than the MRR values produced using the old optimum setting for Indonesian-English collection. All MRR values for C_UNSUBSTITUTED are significantly better than the MRR produced by the cosine baseline ($p = 0.002$ for unsubstituted English-French and $p < 0.001$ for the rest). Our alignment method achieves the perfect MRR of 1 for the substituted English-French collection, which is the same as the MRR of the cosine baseline. Only the MRR produced by the substituted English-German using this new optimum setting is significantly worse than the baseline ($p = 0.038$).

As hypothesised, larger window sizes help in increasing SEP values for both C_SUBSTITUTED and C_UNSUBSTITUTED. Larger window sizes can contain more similar words hence better results. English has more words in common with French than it does with German, therefore aligning English and German documents requires larger window sizes than aligning English and French documents. Moreover, substitution from German to English adds more new words into the substituted text than substitution from French to English. The average number of words per entry for our French to English dictionary is 1.53 and for our German to English is 3.02. For the same reasons, the substituted English-French collection requires a smaller window size of 24, rather than window size of 28 that works well for the substituted English-Indonesian collection. This is because substitution from Indonesian to English introduces

	C_UNSUBSTITUTED			C_SUBSTITUTED	
	E-F	E-G	F-G	E-F	E-G
Old parameter setting	754	638	1979	946	2110
New parameter setting	758	763	749	946	2368

Table 6.7: The average number of words of first document picked up by the alignment methods using the old and new parameter settings. E is English, F is French, and G is German. Numbers are rounded to the nearest integers.

more new words, 3.94 words per entry on average, than substitution from French to English. As for the reason why the window size required to align English and German documents is higher than the window size required to align Indonesian and English documents, a possible reason is because the maximum phrase length from German to English is 13, which is longer than the maximum phrase length from English to Indonesian, which is 5.

A possible reason for why the substituted English-French collection produces higher SEP values than the cosine baseline in contrast to the SEP values for the substituted English-German or substituted Indonesian-English (discussed in Chapter 5) collections, which produce lower SEP values, is the quality of the dictionary. An effective dictionary does not introduce much noise into the substituted text, hence the effect of incorporating the IDF rule of the cosine baseline is not as great as the impact of text with more noise.

Except for the unsubstituted French-German that uses OGP of 1 and EGP of 1, the rest of the collection achieves the best results when using OGP of 1 and EGP of 0. The French and German documents do not have many words in common, and the matches tend to be caused by artefacts such as links to document in different languages and document types. Using an EGP of 1 gives higher penalty values at top row and leftmost column of the traversal matrices in Figure 5.7 and 5.8, therefore reducing the false matches that often occurs for longer documents. Using an EGP of 1 also gives a higher increase of similarity values for matches that occur diagonally, which are more frequent for actual parallel documents, than small matches, which can be artefacts, occurring anywhere in a document. Using a higher EGP is not beneficial because the initial penalty values at the first row and column of the traversal matrix will be too large and affect every document, not only long documents. Using an OGP of 1 for all collections is good because it gives a small penalty value when matches do not occur diagonally; using higher OGP values has similar effects as using higher EGP values.

We initially planned to incorporate document length normalisation to avoid bias toward longer documents. After examining the average number of words for all queries in each collection as shown in Table 6.7, we decided that document length normalisation may not necessarily increase the SEP values. While most of the average document lengths using new parameter setting are higher than the averages using old parameter settings, the SEP values produced by the alignment using new parameter settings are also higher than the SEP values produced using the old parameter settings. We hypothesise that the penalty values can compensate for the effect of document length by punishing smaller matches that do not occur diagonally. Whether document length normalisation is useful merits further investigation.

We now explore the effects of stopping and stemming toward SEP and MRR values on these new parameter settings.

6.3.1 Stopping and stemming

In Section 5.4.3, we reported that stopping and, to a smaller extent, stemming can increase SEP values. In this section, we test whether this is also the case for our European collection.

As for the Indonesian-English collection, there are several permutations of alignment between the queries and the target documents in terms of stopping and stemming for our European languages. For English queries and French and German documents, we can leave them untouched, stop them, stem them, or apply both stopping and stemming. We use the same stopword list compiled by Salton and Buckley and discussed in Section 5.4.3 for our English collection. French and German stopwords are obtained from the Université de Neuchâtel site[8] and can be seen in Appendix K and L. We use the Porter stemmer [Porter, 1980] customised for English, French, and German. [9] As in the previous chapter, we use the English Porter stemmer obtained from `http://tartarus.org/~martin/PorterStemmer`. We obtain the German stemmer from `http://www.cl.uni-heidelberg.de/~esleben/porter.html` and the French stemmer from `http://search.cpan.org/~sdp/Lingua-Stem-Fr-0.02/lib/Lingua/Stem/Fr.pm`.

[8] `http://www.unine.ch/info/clef`

[9] The accuracy figures of Porter stemmers for these languages are not comparable to the accuracy figures of our stemming algorithms. The reported performance of Porter stemmer is usually measured by either its effect in increasing retrieval effectiveness [Krovetz, 1993; Hull, 1996] or from a different angle such as using overstemming and undestemming indexes [Paice, 1994].

	Unstopped, Unstemmed		Stopped		Stemmed		Stopped & Stemmed	
	Cos	Win	Cos	Win	Cos	Win	Cos	Win
Unsubstituted E-F	20.48	48.62 †	37.28	58.40 †	23.40	52.17 †	**38.63**	**59.05** †
Unsubstituted E-G	-2.77	31.57 †	22.85	46.92 †	2.65	34.41 †	**27.88**	**49.11** †
Unsubstituted F-G	-31.08	0.07 †	24.27	**1.89** †	-20.19	0.12 †	**26.53**	1.85 †
Substituted E-F	37.22	41.79 †	**45.87**	**60.24** †	36.54	42.00 †	44.20	59.19 †
Substituted E-G	33.01	17.76 †	**40.55**	**56.16** †	32.50	19.36 †	40.12	53.49 †

Table 6.8: SEP results for European collection with stopping and stemming. Parameters are set to the optimal values with no stopping and stemming. Cos is the cosine baseline, Win is the windowed alignment, E is English, F is French, and G is German. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

For the substituted French collection — where the words in the French documents are substituted into English words — the collection contains both English and French words. Similarly, the substituted German collection contains both English and German words. The substituted French documents can be left untouched, or stopped using English or French stoplists, or stemmed using the Porter stemmer customised for either English or French, or any combination of these. The substituted German collection can also be treated similarly except that it is stopped using English or German stoplists, or stemmed using the Porter stemmer customised for English or German.

There can be different variants of stopping and stemming, we can stop the query but not the document, stop both the query and the document, stop the query and stem the document, and stop the query in one language and stop the documents in another language for C_SUBSTITUTED. In Table 6.8, we show only the variants that produce the best SEP values for stopping, stemming, and combination of stopping and stemming; the corresponding MRR values are shown in Table 6.9. Stopping, stemming, or both stopping and stemming for all collections increases their SEP values to a different extent. Stopping contributes the most in increasing SEP values, especially for C_SUBSTITUTED, where substitution can introduce noise to the collection. Stemming helps to a smaller extent. Except for the stopped and stopped and stemmed unsubstituted French-German and the stemmed substituted English-German collections, all the SEP values of the alignment methods are higher than the corresponding

	Unstopped, Unstemmed		Stopped		Stemmed		Stopped & Stemmed	
	Cos	Win	Cos	Win	Cos	Win	Cos	Win
Unsubstituted E-F	0.838	0.978 †	0.949	**1.000**	0.874	0.982 †	**0.990**	**1.000**
Unsubstituted E-G	0.545	0.861 †	0.800	**0.952** †	0.619	0.899 †	**0.871**	0.948
Unsubstituted F-G	0.332	0.543 †	0.791	**0.912** †	0.405	0.557 †	**0.824**	0.902 †
Substituted E-F	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.990	**1.000**	0.993
Substituted E-G	0.987	0.922 †	**1.000**	**0.993**	0.987	0.955	**1.000**	**0.993**

Table 6.9: MRR results for European collection with stopping and stemming. Parameters are set to the optimal values with no stopping and stemming. Cos is the cosine baseline, Win is the windowed alignment, E is English, F is French, and G is German. The symbol † is used to indicate a statistically significant difference compared to the cosine baseline.

SEP values of the cosine baseline. All differences produced by stopping and stemming are significant ($p = 0.004$ for stemmed substituted English-French and $p < 0.001$ for the rest). A possible reason is that stopping, and to a smaller extent stemming, remove terms that occur frequently, so the impact of using an IDF rule is minimised for the cosine baseline. When compared to the SEP values of the optimum setting (unstopped and unstemmed), only the SEP values of the stemmed unsubstituted French-German ($p = 0.209$) and stemmed substituted English-French ($p = 0.125$) collections are not statistically significant.

Stopping, stemming, or the combination of stopping and stemming generally increases MRR. However, stemming, with or without stopping, decreases the MRR values of the substituted English-French collection, possibly due to an increased number of false matches. All MRR values of our alignment methods are higher than MRR values of the corresponding cosine baseline for C_UNSUBSTITUTED. Only the increases in MRR for the stopped, and stopped stemmed, unsubstituted English-French and for the stopped, and stemmed, unsubstituted English-German are not significant ($p > 0.05$). For C_SUBSTITUTED, the MRR of our alignment methods are either similar or lower than the cosine baseline. None of these decreases are significant ($p > 0.05$). When compared to the MRR values of the optimum setting (unstopped and unstemmed), significant differences are produced by unsubstituted stopped, stemmed, and stopped and stemmed, English-German ($p = 0.005$, $p = 0.005$, and $p = 0.004$ respectively); stopped, and stopped and stemmed, French-German ($p < 0.001$

	C_UNSUBSTITUTED			C_SUBSTITUTED	
	E-F	E-G	F-G	E-F	E-G
Cosine without IDF	20.09	0.85	-15.95	23.78	18.47
Cosine with IDF	20.48	-2.77	-31.08 †	37.22 †	33.01 †

Table 6.10: SEP results for cosine baseline with and without IDF for the European collection. The symbol † is used to indicate a statistically significant difference compared to not using IDF.

	C_UNSUBSTITUTED			C_SUBSTITUTED	
	E-F	E-G	F-G	E-F	E-G
Cosine without IDF	0.940	0.573	0.392	0.990	0.987
Cosine with IDF	0.838 †	0.545	0.332	1.000	0.987

Table 6.11: MRR results for cosine baseline with and without IDF for the European collection. The symbol † is used to indicate a statistically significant difference compared to not using IDF.

for both); and stopped, and stopped and stemmed, English-German ($p = 0.022$ for both) collections.

Similar to the Indonesian-English collection, our European collection benefits more from stopping rather than stemming in increasing SEP and MRR values. The increase in SEP values is more marked for C_SUBSTITUTED, which is expected, since the substitution process introduces more noise to the collection. That is also why the increase is greater for the substituted English-German collection than for the substituted English-French collection. The increase in MRR values from stopping is greater for C_UNSUBSTITUTED than C_SUBSTITUTED when compared to the cosine baseline, which performs quite well for C_SUBSTITUTED.

With high SEP and MRR values after stopping is applied, there is not much room for improvement for our alignment process. As shown in Chapter 5, incorporating IDF to our alignment process may increase SEP and MRR values. We experimented whether incorporating IDF, which is the default of the Zettair setting, increases SEP and MRR for our European corpus. As shown in Tables 6.10 and 6.11, incorporating IDF generally increases SEP and MRR values for C_SUBSTITUTED but decreases SEP and MRR values for C_UNSUBSTITUTED. Similar to the results for the Indonesian-English collection, C_SUBSTITUTED benefits more

from the IDF rule than C_UNSUBSTITUTED does, because the effects of noise introduced by the substitution process are reduced by the IDF rule. The decrease in SEP values for the unsubstituted French-German collection and the increases for C_SUBSTITUTED are significant ($p < 0.001$). Only the decrease in MRR for the unsubstituted English-French collection is significant ($p = 0.002$). Since incorporating IDF increases SEP values for Collection B and C of Indonesian-English as shown in Table 5.14 and for the European substituted collections, it may also help in increasing SEP values for our alignment methods.

We further conjecture that normalisation, especially from French to English [Chen and Gey, 2004], or transliteration of proper nouns [Virga and Khudanpur, 2003] and technical terms [Lindén, 2006] from one language to another may help in increasing SEP and MRR values slightly. This normalisation and transliteration needs to be done carefully since there can be some variations. For example, the common proper noun "Schröder" can be normalised to either "Schroeder" or "Schroder".[10]

To prevent words with various translations from being separated into different windows for substituted documents, we could translate the words on the fly and group different translations of a word into a window, or we could insert special tokens between the translations of one word and those of another. These methods need further investigation as they are not as straightforward as our algorithms. They use variable window sizes that incur additional processing time. We also need to consider special cases such as words with no translation, and noun phrases.

Our algorithms have complexity of $O(N^2)$ where N is the number of windows of words of a document. To optimise our techniques, we can choose to align only certain words such as proper nouns. We can identify proper nouns using the methods described in Section 4.8, which include taking words that are predominantly capitalised when appearing mid-sentence, and capitalised words appearing after titles. We can also use our alignment method as a second-filter in parallel document identification. Rather than processing thousands or even millions of documents, our system could then just measure the similarity of a certain number of documents passed by other systems such as SIMR-cl discussed in Section 5.1.2 to save the processing time.

Since most translations are performed on a sentence-by-sentence basis, the window grouping can be done per one, two, or three sentences to allow more effective alignment. However, as stated in the previous chapter, a sentence in one language can be translated into more

[10] `http://www.antimoon.com/forum/2003/3155.htm` accessed on 24th March 2007.

than one sentence in another language. Thus, sentence-based window grouping needs to take care of such cases. Furthermore, sentence-based window grouping also uses variable window sizes and requires additional processing. We plan to include this method for future work.

6.4 Summary

In this chapter, we have shown that our alignment methods work well in separating parallel documents from non-parallel documents not only for an Indonesian-English corpus as discussed in Chapter 5 but also for a corpus of English, French, and German documents, albeit with slightly different parameter settings. The ideal choice of window size and penalty values depend on the number of shared words between the parallel documents, and on the frequency of artefacts, such as links to other documents in different languages, within each collection.

We have shown that the parameter settings of window sizes 12, 16, 24, 60 with OGP of 1 and EGP of 0 are optimum for the unsubstituted English-French and English-German and the substituted English-French and English-German collections, respectively, and the settings of window size 28 with OGP of 1 and EGP of 1 are optimum for the unsubstituted French-German collection. Clearly, the ideal parameter settings vary somewhat between collections. Document length normalisation may not be necessary for our alignment methods as the contribution towards increasing similarity values is penalised by the gap penalties.

The alignment method works well in increasing SEP and MRR values for C_UNSUBSTITUTED. This is beneficial when a substitution dictionary is not available. For C_SUBSTITUTED, the result varies. Stopping the collection increases SEP values, with the increase greater for C_SUBSTITUTED than C_UNSUBSTITUTED. We expect that stopping reduces the level of noise introduced by the substitution process and reduces the benefits of incorporating the IDF rule. Stopping increases the MRR values slightly when compared to the cosine baseline and the unstopped and unstemmed collection. Stemming increases SEP slightly, but may decrease MRR.

Our alignment methods might be improved further by incorporating IDF, normalising, and transliterating the documents. Such investigations are left for future work. To make our algorithm more efficient, we could align only certain words and use our alignment method as a second-filter in parallel document identification.

We have created methods that can generally separate parallel documents from non-parallel documents well, although the results are collection dependent. Our methods are

applicable to languages using the same character set even without transliteration or normalisation. Our methods work well in separating parallel documents from non-parallel documents for both substituted and unsubstituted collections. We have also shown that we could use simple word substitution rather than machine translation for alignment as long as windows of words are used.

In Chapter 7, we conclude our findings and discuss avenues for future work..

Chapter 7

Conclusions and Future Work

In this thesis, we have investigated a range of aspects of Indonesian text retrieval. Our work shows that Indonesian text retrieval can proceed on standard principles, but that achieving good effectiveness requires either: a sufficiently large testbed; a good stemming algorithm; a well-crafted stopword list; tokenisation of each word; accurate identification of proper nouns; or the combination of any of these schemes. We also propose a new algorithm for identifying parallel documents, which can be beneficial for cross-lingual information retrieval. This chapter presents our conclusions and summarises the key contributions made in this thesis, and discusses avenues for future work.

7.1 Effective Indonesian Stemming

In Chapter 3, we investigated five different stemming algorithms for Indonesian and Malay. Malay was chosen because Indonesian and Malay derive from the same Austronesian root. Four of these algorithms, namely S_NA, S_I, S_AS, and S_AYS, use a dictionary, and one of them, S_V does not.

Since there is no testbed available for the consistent evaluation of stemming algorithm performance, we constructed several collections of our own, using manual stemming results as the baseline. The most important collections are C_TR_MAJORITY and C_TE_MAJORITY: they reflect what users perceive as the correct stems, and includes non-unique words, to reflect the skewed distribution of Indonesian words in natural language. It is more important to correctly stem words that occur often than to correctly stem words that are obscure. The rest of the collections are used for comparison.

The algorithms that use a dictionary perform significantly better than the algorithm that does not; the S_NA algorithm performs the best for all collections, producing fewer than two-thirds of the errors of the second best algorithm, S_AYS. This can be attributed to its incorporation of Indonesian morphological rules, allowing it to address the complexity of Indonesian affixes while avoiding most overstemming problems. Failure analysis indicated scope for at least a further 5% improvement. Most of the failure cases are dictionary-related: around 34% of the errors are caused by non-root words being in the dictionary, and 11% by root words not being in the dictionary. Hyphenated words, usually indicating plurals, make up around 16% of the errors, while 18% of the errors are related to the ordering or the absence of particular morphological rules. The remaining errors, such as foreign and misspelt words, are an inherent challenge to natural language processing tasks, and do not relate directly to the stemming algorithm.

We addressed these limitations in several ways. First, we experimented with the use of different dictionaries, and concluded that a good dictionary should contain only root words. The stemming rules are another vital part of the process. Adding new rules to deal with hyphenated words resolved nearly all errors related to hyphenated words, as long as the resulting stems are in the dictionary. We also added new prefix and suffix rules to cover situations not addressed by the S_NA stemmer. This removed nearly all errors, except for informal affixes, caused by incomplete prefix and suffix rules. We analysed all overstemming cases, and discovered that they are caused by certain suffixes being removed before certain prefixes. We reordered these rules; these adjustments remove most overstemming problems and introduce one case of understemming, caused not only by the rule adjustment but also by the ambiguity of the language. We name this stemmer, which incorporates these rule additions and modifications, the CS stemmer. Our experiments show that the CS stemmer is a significant improvement over the S_NA stemmer. The CS stemmer makes less than one error in thirty-eight words, as compared to one error in twenty-one words with the S_NA stemmer.

Future Work

Since most stemming errors are caused by ambiguity, we plan to investigate stemming further by considering the context surrounding a word. The language-independent stemming method suggested by Bacchin et al. [2005] appears promising, and we plan to investigate its applicability for Indonesian. We also plan to investigate the effect of using different types of dictionaries on the performance of stemming algorithms.

Some stemming errors are caused by unnecessary stemming. Such words include proper nouns, compound words, and misspelt words. Another promising avenue for future work on stemming would therefore be the identification of compound and misspelt words, as well as proper nouns. We discuss some of these identification techniques in Section 7.2.

7.2 Techniques for Effective Indonesian Text Retrieval

In Chapter 4, we explore different IR techniques to evaluate their effect on recall and precision for Indonesian IR. The techniques include stemming, stopping, and changing parameter settings for similarity computation.

There is no publicly available testbed to test these techniques for Indonesian, so we constructed our own testbed. For this, we used 3 000 newswire documents crawled from the Kompas web site.[1] We created twenty ad hoc topics and corresponding relevance judgements, following the well-established TREC methodology.

Query length. Topics have three fields: title, description, and narrative. Each field or a combination of fields can be used as a query. We discovered that using only the title generally produces the highest precision. Since the topic title reflects what a typical user might enter during a web search, we focused on such queries for our later experiments.

Similarity measure parameter settings. In IR, similarity measures are used to assess the probability that a collection document is relevant to a query. Two widely used similarity measures are the cosine measure and Okapi BM25. The cosine measure has a document normalisation pivot value p that can be adjusted: a pivot of 0 means there is no document normalisation while a pivot of 1 indicates that the document length normalisation is in full effect. We have empirically shown that a pivot value of 0.95 produces the highest mean average precision (MAP), although this result is not statistically significant compared to the unnormalised version. Similarly, the Okapi BM25 measure has b and k_1 parameters that can be adjusted: b indicates how much document length normalisation is applied, while k_1 indicates the degree of contribution of term frequencies ($f_{d,t}$). We have found that the highest MAP for our Indonesian collection is produced by b of 0.95 and k_1 of 8.4; this is markedly different from the recommended optimum settings for English collections (b=0.75 and k_1=1.2).

Stopping. Having established the topic field and parameter settings to use, we investigated a range of techniques for retrieval of Indonesian text. First, we tested the effect of

[1] `http://www.kompas.com`

stopping, a process to remove words that do not carry specific information in order to reduce the noise in retrieval. Frequency-based stopword lists contain the most frequent n words in a collection; our experiments show that using them as stopwords leads to decreased recall and precision, as they remove some keywords from the queries. Semantic-based stopword lists use words that do not contribute meaningful semantical information; using such a stopword list generally increases recall and precision. From our experiments, we conclude that using semantic-based stopword lists is better than using frequency-based stopword lists.

Stemming. We also tested the effect of different stemming algorithms discussed in Chapter 3 on recall and precision. We discovered that stemming using any of the algorithms increases MAP, R-precision, and recall, but hurts precision@10, although these differences are not significant. Combining stopping and stemming increases precision and recall further, although the increases are not significantly different from no stopping and no stemming.

Tokenisation. Tokenisation — the process of breaking up words into tokens or grams of characters with certain length — can be used for language-independent stemming. We have explored tokenisation for stemming Indonesian. We have found that grams of smaller sizes tend to lead to overstemming, while grams of larger sizes may lead to understemming. Our experiments show that the best results are achieved when using 5-grams and spanning word boundaries.

Dictionary augmentation using n-grams. Approximately 10% of stemming errors in the CS stemmer are caused by misspellings. Since stemming increases precision, we hypothesised that correcting misspelling through n-grams may increase precision further. From experiments with different gram sizes and various methods for finding the closest match of a word in the dictionary, we concluded that using 4-grams and the Q-gram method produces the highest MAP and R-precision. Using smaller grams decreases stemming accuracy; some words that are best left unchanged are changed by smaller grams but not by larger grams. We discovered that some of the misspelling errors can indeed be corrected by dictionary augmentation using n-grams.

Identification of proper nouns. Approximately 13% of stemming errors are caused by proper nouns being stemmed. We conjectured that not stemming proper nouns can increase stemming accuracy and retrieval precision. We approached Indonesian proper noun identification from four different aspects: acronyms; words appearing mid-sentence with their initial letter predominantly capitalised; words likely to be English words; and words appearing after titles. Based on empirical investigation, the highest MAP is achieved when the proper nouns, obtained from combining acronyms, words appearing mid-sentence with

their initially letter predominantly capitalised, and words appearing after titles, are not stemmed. Stopping followed by stemming all words, except for proper nouns identified using the method described earlier, increases all precision and recall values compared to using the unmodified CS stemmer.

Language identification. All the techniques presented above are customised for Indonesian. Applying Indonesian-specific techniques to non-Indonesian text will be counterproductive. Therefore, we need to identify whether a document is in Indonesian. We investigated a simple method of language identification by using word statistics, collecting word occurrence frequencies of Indonesian, English, and Malay documents in our training sets. We discovered that this method can identify whether a document is in English, Indonesian, or Malay with precision ranging from 99.75% and 100.00%.

Identification of compound words. Splitting compound words — also known as decompounding — may increase retrieval effectiveness depending on the language. We investigated whether Indonesian can benefit from decompounding. In our approach, we considered a word to be a compound word if it can be broken into two words that exist in a dictionary. However, this method results in some misclassification of terms, due to factors similar to those that cause stemming to fail: proper nouns, words with certain affixes, misspelt words, and the presence of foreign words. Since Indonesian compound words are not usually written together unless they are prefixed and suffixed, and since the number of compound words correctly identified is less than 1% of the whole collection, we decided that compound word identification and splitting does not merit further investigation.

In many of our text retrieval experiments, we found it difficult to show significance of results; we suspect that this is due to the small number (twenty) of queries that we were able to create. For significant and stable results, it is generally recommended to use at least fifty queries. While preparing topics and relevance judgements is costly in terms of time and resources, any future work should place a high priority on this task as well as on increasing the size of the document collection. Nevertheless, our results are an important contribution to the largely unexplored domain of Indonesian information retrieval, representing the most thorough research on different aspects of Indonesian IR using a published testbed.

Future Work

We plan to use language modelling to measure similarity between queries and documents, and to incorporate query expansion to increase retrieval effectiveness [Abdelali et al., 2007].

We can also investigate query-biased summary techniques [Tombros and Sanderson, 1998] suitable for Indonesian. We have not investigated issues of efficiency; future work can include improving the efficiency of algorithms for Indonesian IR.

7.3 Automatic Identification of Indonesian-English Parallel Documents

In Chapter 5, we proposed an automatic parallel document identification method. As the number of web documents in different languages increases rapidly, we need a cross-lingual information retrieval (CLIR) to reduce the language barriers of obtaining information in different languages. Currently there is no CLIR testbed for Indonesian. One of the simplest ways to build a CLIR testbed is to use a parallel corpus. Such a parallel corpus would also be useful for other NLP tasks including building bilingual dictionaries and correlating synonyms.

We propose a novel method for identifying parallel documents that does not rely on external structures of parallel documents nor make any assumption about the content of the documents. Our alignment methods are based on the global alignment methods of Needleman and Wunsch. The basic premise is that two strings are regarded as a match if they share many symbols in common in the same order. Since two parallel documents are likely to have some words — especially proper nouns — in common, we can try to align them in this way. Since the order of such common words may not be preserved between parallel documents, we relax the ordering constraint by aligning windows of words, instead of aligning the individual words. The alignment algorithm rewards matches of word sequences, especially for matches that occur diagonally, and penalises insertion or deletion.

We chose Indonesian and English documents to test our alignment methods. We use training collections to determine the best parameter settings and test whether these predetermined settings work for the test collection. We also investigated whether simple translation methods have a beneficial impact on our technique using a dictionary to substitute the words in English documents into Indonesian words. There is no context-disambiguation involved; all possible meanings are used during substitution. We refer to this collection as the substituted collection. We can also align parallel documents without any substitution process; we refer to this collection as the unsubstituted collection. As the baseline for our experiments, we used the symmetric cosine similarity measure, where the lengths of both the query and the answer document contribute towards the similarity estimation.

Our experimental results show that the new algorithm is more effective at separating

parallel documents from non-parallel documents compared to a baseline for the unsubstituted collection. Our approach is less successful for the substituted collection. The most likely reason is that the alignment relies heavily on proper nouns that are mostly similar for Indonesian and English without the need of substitution, while the substitution process introduces noise into the matching process. The effects of noise are diminished for the cosine baseline possibly because of its incorporation of inverse document frequency (IDF) rule.

Through failure analysis, we identified four reasons why non-parallel documents may be ranked highly: the presence of synonyms; the presence of misspelt words; the occurrence of proper nouns at different positions; and, the presence of noise introduced by the substitution process. Solutions to the first two problems require deeper understanding of both languages. The third problem can be addressed by altering the window size. Removing stopword noise is straightforward as stopword lists for both languages are available. Stemming may also help in conflating terms referring to a particular topic. We therefore experimented with stopping and stemming on our alignment methods.

Stopping produces separation (SEP) values that are significantly better than the baseline for both the unsubstituted and the substituted collections. As expected, the increase is higher for the substituted collection, as removal of stopwords reduces the number of spurious matches. Stemming increases the SEP values for the unsubstituted collection but decreases the SEP values for the substituted collection. Combining stopping and stemming increases SEP values, largely due to stopping.

Future Work

We suspect that incorporating an IDF rule might be useful in separating parallel documents from non-parallel documents as shown by the baseline. Applying an IDF rule means that the weight of a word is in inverse proportion of its frequency of appearance within the document collection. Such words may be used as the signature of a file. Using an IDF rule for the cosine baseline generally increases SEP and mean reciprocal rank (MRR) values especially for the substituted collection, as the impact of noise is reduced. However, the overall result depends on the collection. The exploration of the impact of incorporating IDF into our alignment algorithms is a promising area for future work. Further areas for future work relating to our alignment approaches are discussed in Section 7.4.

7.4 Automatic Identification of European Parallel Documents

In Chapter 6, we empirically evaluated our alignment methods for identifying parallel documents written in English, German, and French as we hypothesise that our alignment methods work as long as the documents share the same character set.

For the unsubstituted collection, we used English-French, English-German, and French-German document collections. For the substituted collection, we substituted words in French and German documents with their English equivalents, using a simple dictionary lookup process.

Using the optimum parameter settings for the Indonesian-English collections, our results show that, except for the unsubstituted French-German collection and the substituted English-German collection, the SEP values produced by our algorithm are higher than the cosine baseline. Similar to the Indonesian-English collection, the improvement is more marked for the unsubstituted collection, which relies more on alignment of proper nouns.

An analysis of failure cases showed that the factors described in the previous section also affect the European-language experiments. We further discovered that in the European documents proper nouns are sometimes normalised or transliterated during translation, so aligning them would not produce matches. The presence of artefacts such as links to documents in other languages or other formats, and the presence of words or phrases in a foreign language, can create false matches. The number of meanings introduced by the substitution dictionary also plays an important role: some dictionaries introduce few but accurate meanings into the document collection, while a less effective dictionary — from the perspective of our alignment process — introduces a lot of spurious meanings of a word. Therefore, we explored different window sizes and penalty values that are suitable for each of our European collections.

Our results indicate that optimal parameter settings are language dependent. Most language pairs benefit more from larger window sizes, except for the substituted English-French pair that uses smaller window size than the substituted Indonesian-English pair; this is because they have fewer words in common than the Indonesian-English pair. In contrast, the substitution dictionary used to translate French to English introduces fewer new words, and so a smaller window size is appropriate. With the exception of the substituted English-German collection, the SEP values produced by our alignment with new optimum setting are generally higher than the already-high SEP values produced by the cosine baseline.

Similar to the Indonesian-English collection, stopping increases the SEP values for both

the substituted and unsubstituted collections; stemming increases the SEP values for the unsubstituted collection but decreases the SEP values for the substituted collection.

We conclude that our alignment methods work well in separating parallel documents for non-parallel documents for languages using the same character set. Each collection may need different parameter settings particular to the language pairs used. Our method is more beneficial for aligning unsubstituted parallel documents, where a substitution dictionary is not available and the alignment relies on proper nouns, than for aligning substituted parallel documents.

Future Work

Future research may include experiments on aligning languages that do not use the Latin character set, for example, aligning Chinese documents, using Hànzi character set, with Indonesian documents, using Latin character set. Incorporating normalisation [Chen and Gey, 2004], or transliteration of proper nouns [Virga and Khudanpur, 2003] and technical terms [Lindén, 2006] may also help in matching proper nouns written differently. Tokenisation of words into n-grams and aligning windows of n-grams instead of windows of words is another avenue for future work. This approach may be beneficial for aligning misspelt words or transliterated words that share some n-grams.

We conjecture that our alignment method may benefit from incorporating the IDF rule because stopping helps in increasing the SEP and MRR values for the cosine baseline especially for the substituted collections. For the unsubstituted collections, the likely effect is less clear.

Our alignment methods may also benefit from document length normalisation. The average length of documents placed first by the alignment is lower than the average length of documents in the Indonesian-English collection. In contrast, the alignment favours longer documents for the European collections. Larger penalty values may compensate for the bias toward long documents.

We have discovered that our alignment methods can generally separate parallel documents from non-parallel documents better than the cosine baseline, especially for the unsubstituted collections. However, our method does not actually indicate which document is parallel and which is not. A certain threshold needs to be specified; any document with the similarity value exceeds the threshold is considered as parallel. Future research needs to be done to compute a suitable threshold value. Our alignment method is computationally expensive and may

be more suitable to act as a second-level filter in the identification process. Optimisation to the alignment methods can reduce the resources used. One possibility is to align only proper nouns instead of all words. Another possibility is to do translation on the fly and group different translations of a word into a window, or to insert special tokens between the translations of one word and those of another. To allow more effective alignment, we conclude that we can group one, two, or three sentences together instead of words.

7.5 Final remarks

In this thesis, we have investigated the most effective stemming algorithms for Indonesian, and proposed novel improvements to increase effectiveness. We have also created a testbed for Indonesian text retrieval. We have experimentally identified the techniques that work best in increasing recall and precision for Indonesian text retrieval. We have also created algorithms for identifying proper nouns and for identifying the language of a document. We have discovered a method to identify parallel documents automatically using the words in the documents. Our method works for languages using the same character set and is successful in aligning documents in Indonesian-English, English-French, English-German, and French-German collections.

While substantial further investigation is warranted, this research represents a substantial advance in understanding techniques that can be applied for effective Indonesian text retrieval.

Appendix A

Capitalisation Rules for Indonesian

In this appendix, we explain capitalisation as used in Indonesian documents [Wilujeng, 2002, pages 9–14].

1. At the beginning of a sentence.
   "Saya baru membeli coklat itu." ⟨I have just bought that chocolate.⟩

2. First letter after a quotation mark indicating a direct quote.
   "Budi bertanya, "Kapan kamu datang?"." ⟨Budi asked, "When did you come?".⟩
   " "Saya baru datang," kata Susi, "pagi ini"." ⟨"I have just come," Susi said, "this morning".⟩

3. First letter of words related to God, religious texts, religions and pronouns that relate to them.
   "Tuhan selalu mendengar doa-doa hamba-Nya." ⟨God always hears his follower's prayers⟩ — note that the "Nya" to replace ⟨God's⟩ is also capitalised.

4. First letter of the title for monarchs, government and military officials, and religious leaders when the title is followed by a person's name or a pronoun replacing a person or an institution name or a place name. If the title is not followed by names or pronouns, then it is not capitalised.
   "Sultan Hasanuddin" ⟨Sultan Hasanuddin⟩, "Haji Amir" ⟨Hajj Amir⟩, "Perdana Menteri Blair" ⟨Prime Minister Blair⟩, "Sekretaris Jendral PBB" ⟨UN Secretary General⟩, "Gubernur Bali" ⟨Bali governor⟩ are examples of capitalised titles.
   "Dia adalah putra seorang sultan." ⟨He is the son of a sultan.⟩
   "Dia baru dilantik jadi jenderal." ⟨He/she has just been assigned as a general.⟩

5. First letter of people's names when they are not used as metrics or types.
   For example, "James Watt" and "Rudolph Diesel" are written in capitals while "50 watt" ⟨50 watt⟩ and "mesin diesel" ⟨diesel engine⟩ are not.

6. First letter of names of nations, dialects and languages. If the name is used as an adjective and has at least a prefix or a suffix attached to it, then it is not capitalised.
   Examples of the first case include: "bangsa Indonesia" ⟨Indonesia [as a nation]⟩, "suku Sasak" ⟨Sasak dialect⟩, "bahasa Jerman" ⟨German [as a language]⟩; and for the second case include: "mengindonesiakan pasta" ⟨Indonesianise pasta⟩.

7. First letter of names of day, month, year, historical events, or public holidays. If the historical event is not used as a name, then it is not capitalised.
   Examples of capitalised dates or events are "hari Kamis" ⟨Thursday⟩, "bulan Juli" ⟨July⟩, "hari Paskah" ⟨Easter⟩, and "Perang Teluk" ⟨Gulf War⟩. An example of a non-capitalised event is the sentence "Idealisme dapat menyebabkan perang dunia" ⟨Idealism can lead to a world war⟩.

8. First letter of geographical names used as proper nouns.
   "Asia Selatan" ⟨South Asia⟩ and "Pantai Kuta" ⟨Kuta Beach⟩ are examples of capitalised names, while "pergi ke selatan" ⟨to head towards the south⟩, "berenang di pantai" ⟨swim at the beach⟩, and "kucing siam" ⟨Siamese cat⟩ are examples of non-capitalised names.

9. First letter of the names of countries, government institutions, bills, and laws, except for the first letter of prepositions such as "dan" ⟨and⟩, as long as these words are used as proper nouns. If some of the words are repeated fully, they are still capitalised.
   Capitalised examples are "Republik Indonesia" ⟨Republic of Indonesia⟩; "Departemen Pariwisita, Seni, dan Budaya" ⟨Department of Recreation, Art, and Culture⟩; "Keputusan Presiden Republik Indonesia Nomor 18 Tahun 2003" ⟨Decree of Republic of Indonesia President Number 18 Year 2003⟩; and "Perserikatan Bangsa-Bangsa"[1] ⟨United Nations⟩. A non-capitalised example is "mendirikan sebuah republik baru" ⟨to found a new republic⟩.

10. First letter of all words (including fully repeated words) in the title of a book, a magazine, a newspaper and in any other forms of writing, except for prepositions such as

[1] "Bangsa-bangsa" ⟨nations⟩ is derived from "bangsa" ⟨nation⟩.

"di" ⟨at⟩, "dalam" ⟨in⟩ and conjunctions such as "dan" ⟨and⟩ that are not located at the beginning of the title.
"Saya membaca surat kabar Bali Post." ⟨I read the Bali Post newspaper.⟩
"Annie membeli "Mengelilingi Bumi dalam 80 Hari"." ⟨Annie bought "Around the World in 80 Days".⟩

11. First letter of shortened forms of titles and ranks, for example, "Dr." ⟨Doctor⟩, "Prof." ⟨Professor⟩, and "Ny." ⟨Mrs.⟩

12. First letter of words used to refer to family and relatives as long as they are used in statements or referrals.
"Kemarin saya menelpon Ibu." ⟨Yesterday I called [my] mother.⟩
" "Berapa harga komputer itu, Paman?", tanya Adik." ⟨"How much is that computer, Uncle?" asked [my] younger sibling.⟩

13. First letter of the word "Anda", which is a more polite form of "you" than "engkau" or "kamu".
"Saya telah melihat rumah Anda." ⟨I have seen your house.⟩

Appendix B

Indonesian Grammar

Some aspects of Indonesian grammar are described in this appendix.

B.1 Gender

Most Indonesian words are neutral in terms of gender, and there is no specific gender for nouns. All object names such as "buku" ⟨book⟩, "gunting" ⟨scissor⟩, "meja" ⟨table⟩ are neutral. Personal and possessive pronouns are also neutral [Widyamartaya, 2003, page 49]. "He", "she", "him", and "her" are all replaced by "dia". "His", "her", and "hers" are replaced by the suffix "-nya". "Saya mengembalikan bukunya" is translated into "I return his/her book". We discuss addition of suffixes in Section 2.2.

Most nouns describing a person in Indonesian are also neutral, such as "adik" ⟨younger sibling⟩, "sepupu" ⟨cousin⟩, "anak" ⟨son/daughter⟩, "kepala sekolah" ⟨headmaster/headmistress⟩", and "tukang pos" ⟨postman/postwoman⟩. To denote the gender of these words, words such as "laki-laki"[1] ⟨male⟩ and "perempuan" or "wanita" ⟨female⟩ are used. For example, "anak laki-laki" specifies a son while "anak perempuan" specifies a daughter. There are also certain words in Indonesian that are gender specific. These include "paman" ⟨uncle⟩, "tante" ⟨aunt⟩, "raja" ⟨king⟩, and "ratu" ⟨queen⟩.

B.2 Ordinal Numbers

In Indonesian, ordinal number such as "second", "third", or "fourth" are formed simply by adding a prefix "ke-" in front of the number [White, 1990, page 36]. For example, "kedua"

[1] This is a repeated word that does not imply plural.

⟨second⟩ derives from "ke-" and "dua" ⟨two⟩, "ketiga" ⟨third⟩ from "ke-" and "tiga" ⟨three⟩ "keduabelas" ⟨twelfth⟩ from "ke-" and "duabelas" ⟨twelve⟩. This rule is applicable for all numbers except for the first, of which the ordinal number is "pertama".

For dates, cardinal rather than ordinal numbers are used [White, 1990, page 38]. For example, "2^{nd} March" is written as "tanggal dua Maret" ("tanggal" means "date") instead of "tanggal kedua Maret".

B.3 Negation

Indonesian uses "bukan", "jangan", and "tidak" for negation [Woods et al., 1995, page 21]. "Bukan" is usually used before a noun, for example, "Paul bukan penakut" ⟨Paul is not a coward⟩, "Itu bukan komet Haley" ⟨That is not Haley's comet⟩. "Jangan" and "tidak" are used in front of a verb or an adjective, for example, "Jangan takut!" ⟨Do not be afraid!⟩, "Jangan lari!" ⟨Do not run!⟩, "Paul tidak takut" ⟨Paul is not afraid⟩ and "Susan tidak lari" ⟨Susan does not run⟩.

There is an additional negation "belum" that means "not yet" in Indonesian [White, 1990, pages 18–19]. "Paul belum lari" means "Paul has not run yet".

B.4 Comparative and Superlative

To indicate comparative and superlative, the words "lebih" ⟨more⟩, "kurang" ⟨less⟩, and "[yang] paling" ⟨[the] most⟩ are used [Woods et al., 1995, page 18]. The prefix "ter-" can also be used to indicate ⟨[the] most⟩ instead of "[yang] paling" [White, 1990, page 52].

The comparative and superlative forms for Indonesian are:

"kurang tinggi"	⟨less tall⟩
"tinggi"	⟨tall⟩
"lebih tinggi"	⟨taller⟩
"paling tinggi"="tertinggi"	⟨tallest⟩

For "more" in terms of quantity, BI uses "lebih banyak" [White, 1990, page 51]. The sentence "Pohon Susan punya lebih banyak buah dari pohon Paul" ⟨Susan's tree has more fruit than Paul's tree⟩ ("pohon" ⟨tree⟩, "punya" ⟨has⟩, "buah" ⟨fruit⟩, "dari" ⟨from, than⟩) illustrates this.

B.5 Tenses

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, page 16]. "Saya membaca buku itu kemarin" translates to "I read that book yesterday" ("saya" ⟨I⟩, "membaca" ⟨to read⟩, "buku" ⟨book⟩, "itu" ⟨that⟩); "Saya akan membaca buku itu besok" translates to "I will read that book tomorrow".

Woods et al. [1995, pages 16–17] add that the common words to specify time are "sudah" ⟨already⟩, "sedang" ⟨in the middle of doing something⟩, "akan" ⟨will⟩ for past, present, and future tenses respectively. Examples of the usage of these words are "Saya sudah membaca buku itu" ⟨I have read that book⟩; "Saya sedang membaca buku itu" ⟨I am reading that book⟩; and "Saya akan membaca buku itu" ⟨I will read that book⟩. For things that have just happened, the words "baru saja" are used. "Saya baru saja membaca buku itu" means "I have just read that book".

Appendix C

Indonesian Topics

In this appendix, we show our Indonesian topics used for ad hoc experiments.

<top>
<num> Number: 1
<title> hubungan Indonesia Australia setelah Timor Timur
<desc> Description:
Hubungan Indonesia Australia setelah campur tangan Australia di Timor Timur
<narr> Narrative:
Dokumen yang menggambarkan bagaimana hubungan Indonesia dengan Australia di bidang apapun dan hubungan itu disebabkan campur tangan Australia di Timor Timur dianggap relevan. Dokumen yang hanya menyatakan hubungan antara 2 negara tanpa menyebutkan masalah Timor Timur dianggap tidak relevant.
</top>

<top>
<num> Number: 2
<title> dampak terorisme terhadap penurunan jumlah turis
<desc> Description:
Dokumen harus menyebutkan dampak resiko terorisme terhadap jumlah turis yang datang ke Indonesia.
<narr> Narrative:
Dokumen yang menyatakan negara lain yang melarang penduduknya untuk datang ke In-

donesia karena resiko terorisme dianggap relevan. Dokumen yang hanya menyatakan negara lain yang melarang penduduknya untuk datang ke Indonesia atau penurunan jumlah turis bukan karena terorisme dianggap tidak relevan.
</top>

<top>
<num> Number: 3
<title> kecelakaan pesawat udara Indonesia
<desc> Description:
Dokumen harus menyebutkan segala kecelakaan udara yang terjadi di Indonesia.
<narr> Narrative:
Dokumen harus menggambarkan pesawat jenis apa dan jatuh di mana. Kapan pesawat jatuh bisa relevan tapi tidak perlu. Laporan kesalahan teknis yang terjadi tidak relevan. Kecelakaan yang terjadi sebelum lepas landas dan waktu mendarat tetapi pesawat tidak jatuh dianggap relevan. Kecelakaan dapat terjadi buat pesawat komersial ataupun pesawat jenis lain. Dokumen yang hanya menyebutkan tindakan yang dilakukan setelah kecelakaan seperti evakuasi tidak relevan.
</top>

<top>
<num> Number: 4
<title> pemberantasan narkoba
<desc> Description:
Dokumen harus menggambarkan apakah yang sudah dikerjakan pemerintah untuk memerangi pemakaian dan pengedaran narkoba.
<narr> Narrative:
Dokumen harus menggambarkan apakah yang telah dan akan dikerjakan pemerintah untuk memberantas narkoba. Narkoba menyangkut semua obat terlarang tetapi tidak mencakup rokok maupun alkohol. Pemberantasan oleh pihak bukan pemerintah juga dianggap relevan. Dokument yang hanya menyebut efek-efek narkoba tanpa tindakan untuk mengatasinya dianggap tidak relevan. Penangkapan terhadap penyelundup dianggap tidak relevan tetapi metode untuk menangkap basah penyelundup dianggap relevan. Usaha rehabilitasi juga dianggap relevan.
</top>

<top>
<num> Number: 5
<title> pemilu presiden prancis
<desc>Description:
Dokumen harus menggambarkan situasi pemilu presiden Prancis
<narr> Narrative:
Dokumen yang relevan harus menyebutkan siapakah calon2 presiden (calon siapa saja, tidak terbatas untuk calon tertentu) dan jumlah suara yang mereka dapatkan.
</top>

<top>
<num> Number: 6
<title> ulang tahun megawati sukarnoputri
<desc> Description:
kapankah ulang tahun Megawati Sukarnoputri?
<narr> Narrative:
Dokumen yang relevan harus menyebutkan kapankah ulang tahun Megawati Sukarnoputri dan tidak perlu setelah dia menjadi presiden.
</top>

<top>
<num> Number: 7
<title> situasi banjir jakarta
<desc> Description:
Dokumen menggambarkan situasi Jakarta akibat banjir
<narr> Narrative:
Dokumen harus menggambarkan efek dari banjir yang terjadi di Jakarta (kawasan Jabotabek) dan harus menyebutkan ketinggian air dan dampaknya terhadap penduduk. Dokumen yang hanya menyebutkan penyebab2 banjir dianggap tidak relevan. Tindakan yang dilakukan pemerintah untuk mengatasi banjir maupun sumbangan yang diberikan untuk mengatasi banjir dianggap tidak relevan. Prediksi apakah banjir akan terjadi juga dianggap tidak relevan.
</top>

<top>
<num> Number: 8
<title> duta besar Indonesia
<desc> Description:
Dokumen menyebutkan nama-nama duta besar indonesia
<narr> Narrative:
Sedikitnya nama 1 duta disebutkan dan negara yang sedang berada atau akan dikirim. Dokumen dengan nama bekas duta dan negara dia berada juga dianggap relevan. Nama sebutan yang sering digunakan, bukan nama lengkap tidak apa-apa. Nama calon yang dinominasikan dianggap tidak relevan. Dubes RI untuk PBB dianggap relevan.
</top>

<top>
<num> Number: 9
<title> nama suami megawati
<desc> Description:
Dokumen menyebutkan nama lengkap suami Megawati
<narr> Narrative:
Dengan membaca dokumen yang relevan dapat diketahui siapakah suami Megawati.
</top>

<top>
<num> Number: 10
<title> gejala dan penyebab asma
<desc> Description:
Dokumen harus menggambarkan gejala dan penyebab asma.
<narr> Narrative:
Dokumen harus menyebutkan paling sedikit 1 gejala dan 1 penyebab asma untuk dianggap relevan.
</top>

<top>
<num> Number: 11
<title> pemenang pertandingan piala Thomas jenis apapun asal Indonesia

<desc> Description:
Dokumen harus menyebutkan nama pemenang di segala pertandingan piala Thomas dari Indonesia
<narr> Narrative:
Dokumen harus menyebutkan nama-nama pemenang di perebutan piala Thomas, baik tunggal, ganda maupun beregu dari Indonesia. Kemenangan tidak perlu untuk pertandingan final, dapat untuk segala pertandingan, termasuk babak penyisihan. Pemenang di segala tahun dianggap relevan.
</top>

<top>
<num> Number: 12
<title> nama bos Manchester United
<desc> Description:
Dokumen harus menyebutkan nama bos Manchester United
<narr> Narrative:
Dokumen dianggap relevan asal dapat disimpulkan nama boss Manchester United dari dokumen tersebut.
</top>

<top>
<num> Number: 13
<title> laporan Piala Dunia
<desc> Description:
Dokumen harus menyebutkan laporan score World Cup dan siapa pencetak golnya untuk segala pertandingan di Piala Dunia
<narr> Narrative:
Total jumlah pencetak gol dan gol yang dicetak harus sama dengan score yang dimiliki setiap tim. Contohnya, score Inggris-Brazil adalah 0-1 dan pencetak gol buat Brazil adalah Ronaldo. Perkiraan dan statistik pemain sebelumnya tidak relevan. Bukan nama lengkap tapi sebutan masih dianggap relevan. Laporan sebelum pertandingan selesai dianggap tidak relevan.
</top>

<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: 15
<title> aktor aktris calon atau pemenang Oscar
<desc> Description:
dokumen menyebutkan nama aktor atau aktris calon atau pemenang oscar dan film yang dibintangi mereka
<narr> Narrative:
Dokumen harus menyebutkan setidaknya 1 nama aktor atau aktris dan film yang dibintangi mereka untuk dianggap relevan. Dokumen dengan nama aktor atau aktris pendukung dengan filmnya juga dianggap relevan. Nominasi maupun pemenang dapat terjadi di segala tahun.
</top>

<top>
<num> Number: 16
<title> akibat kenaikan harga BBM
<desc> Description:
akibat kenaikan harga BBM terhadap situasi ekonomi, sosial, politik Indonesia.
<narr> Narrative:
Dokumen cukup menyebutkan salah satu dampak, tidak perlu ketiga-tiganya tapi harus di bidang ekonomi, sosial dan politik, bukan bidang lainnya.
</top>

<top>
<num> Number: 17 <title> susunan kabinet Timor Leste
<desc> Description:
Susunan lengkap kabinet Timor Leste
<narr> Narrative:
Dokumen harus menunjukkan sedikitnya 5 nama dan posisi yang dijabat menteri-menteri Timor Lester.
</top>

<top>
<num> Number: 18
<title> persidangan Tommy Soeharto
<desc> Description:
perkembangan kasus persidangan Tommy Soeharto
<narr> Narrative:
Dokumen harus menceritakan perkembangan persidangan Tommy, apakah ada insiden yang terjadi. Kasus persidangan orang lain yang berhubungan dengan Tommy dianggap tidak relevant tetapi kalau dokumen ada menyebutkan sebab persidangan karena keterlibatan dalam kasus Tommy maka dianggap relevan. Kasus penyuapan saksi dianggap relevan tetapi permohonan grasi tidak resmi dianggap tidak relevan.
</top>

<top>
<num> Number: 19
<title> kunjungan luar negeri Megawati
<desc> Description:
Laporan tentang kunjungan Megawati ke negara lain untuk keperluan resmi negara
<narr> Narrative:
Dokumen harus melaporkan kunjungan resmi kenegaraan Megawati ke negara-negara lain (1 negara sudah cukup) dan tujuan kunjungan tersebut, tanggal kunjungan tidak harus ada. Kunjungan tidak resmi atau pribadi dari Megawati dianggap tidak relevan. Pertemuan dengan masyarakat Indonesia di luar negeri tidak relevan. Permintaan oleh orang lain untuk

berkunjung, rencana, prediksi dan pengumuman kunjungan tidak relevan. Kunjungan tanpa sebutan tujuan juga tidak relevan.
</top>

<top>
<num> Number: 20
<title> masa jabatan Gus Dur sebagai Presiden
<desc> Description:
Tanggal pelantikan dan pemberhentian Gus Dur menjadi presiden
<narr> Narrative:
Dokumen dianggap relevan asalkan berisi tanggal dimulai dan berhentinya Gus Dur menjadi presiden, jika hanya bulan tanpa tanggal juga dianggap relevan.
</top>

Appendix D

English Translation of Indonesian Topics

In this appendix, we show the English translation of Indonesian topics shown in Appendix C.

<top>
<num> Number: 1
<title> Indonesia Australia relationship after East Timor
<desc> Description:
Indonesia and Australia relationship after East Timor independence
<narr> Narrative:
Document describes the impact of Australia's intervention in East Timor case towards Indonesia and Australia relationship in any fields is considered relevant. Document which only describes the relationship between Indonesia and Australia without mentioning East Timor case is not relevant.
</top>

<top>
<num> Number: 2
<title> effect terorism towards tourists decrease
<desc> Description:
The document shall describe the effects of any risk of terorism towards the number of tourists

visiting Indonesia.
<narr> Narrative:
document describing whether any country has travel ban against coming to Indonesia because of risk of terorism is relevant. The mere mention of banning and decreasing of tourism not because of terorism is not relevant
</top>

<top>
<num> Number: 3
<title>air accidents in Indonesia
<desc> Description:
The document describes any air accidents happened in Indonesia.
<narr> Narrative:
The document shall describe what plans were crashed and where and when. The air accident can be commercial plane or other means of air transportation. Document only mentions the action taken after the accident like evacuation is not relevant.
</top>

<top>
<num> Number: 4
<title>war against drugs
<desc> Description:
The document describes what government has done to fight against drug usage and drug dealing.
<narr> Narrative:
The document shall describe what the government has done and planned to do to fight against drugs. Drugs shall include any type of illegal medicine but not including smoke and alcohol. war against drug by non-government can be considered relevant as well. The document mentioning only the effects of drug taking without the action against it is not relevant.
</top>

<top>
<num> Number: 5
<title> french government election <desc>Description:

The document describes the situation of French presidential election.
<narr> Narrative:
A relevant document shall describe who the candidates of the presidential election (any candidates) and the votes they get.
</top>

<top>
<num> Number: 6
<title> megawati sukarnoputri's birthday date
<desc> Description:
When is Megawati Sukarnoputri's birthday?
<narr> Narrative:
The relevant document shall mentions when the birthday of Megawati Sukarnoputri and it is not necessarily when she is the president.
</top>

<top>
<num> Number: 7
<title> jakarta flood situation
<desc> Description:
Document describes the situation in Jakarta because of the flood
<narr> Narrative:
Document shall describe the effects of the flood only in Jakarta and shall mention the height of the water and the effects on the population. Document only describes the causes of the flood is not relevant.
</top>

<top>
<num> Number: 8
<title> indonesian's ambasaddors
<desc> Description:
The document describes names of Indonesian ambasaddors
<narr> Narrative:
The document mentions at least 1 ambasaddor name and the country he/she is/will be sent

to.
</top>

<top>
<num> Number: 9
<title> Megawati's husband's name
<desc> Description:
Document shall mention the full name of Megawati's husband.
<narr> Narrative:
A document is deemed relevant if by reading the document and the reader can conclude who Megawati's husband is.
</top>

<top>
<num> Number: 10
<title> syptoms and causes for asthma
<desc> Description:
Document shall describe the symptoms and causes of asthma.
<narr> Narrative:
Document has to mention at least one each of symptom and cause for astma to be relevant.
</top>

<top>
<num> Number: 11
<title> winners of any Thomas Cup matches from Indonesia
<desc> Description:
Document shall describe winners in any Thomas cup competition from Indonesia
<narr> Narrative:
Document has to mention winners in any match, and it can be for single, double or group players from Indonesia. The winning does not have to be in final rounds but any rounds.
</top>

<top>
<num> Number: 12

<title> name of Manchester United boss
<desc> Description:
Document shall mention the full name of Manchester United's boss.
<narr> Narrative:
As long as the name of Manchester United boss can be derived from the document, the document is relevant.
</top>

<top>
<num> Number: 13
<title> World Cup (soccer) report
<desc> Description:
Document must describe the score in the World Cup and who the goal maker was for any match in the World Cup.
<narr> Narrative:
Total sum of the goals made by the goal maker shall be equal to the sum of the score in each team. For example, the score for England Brazil is 0-1 and the goal maker for Brazil is Ronaldo. Prediction and pass records is not relevant.
</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 USA 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.
</top>

<top>
<num> Number: 15
<title> actors actresses nominees or winners for Oscar
<desc> Description:

document shall mention the name of actors or actresses for nominees and winners of Oscars and the film they starred in to get nominated.
<narr> Narrative:
Document has to mention at least one actor or actress name and the film they starred in to be considered relevant. Winner in any years is OK.
</top>

<top>
<num> Number: 16
<title> the effects of oils (fuels) price hike
<desc> Description:
The effects of oils (fuels) price hike towards economic, social and political situation in Indonesia.
<narr> Narrative:
Document can describe any one of the effects, do not have to be 3 of them. Document describing only other field without any of these 3 is not relevant.
</top>

<top>
<num> Number: 17
<title> Complete names of Timor Leste's ministers
<desc> Description:
Complete names of Timor Leste's ministers and the position they hold
<narr> Narrative:
The document shall show at least 5 names and positions of Timor Lester's ministers.
</top>

<top>
<num> Number: 18
<title> trial of Tommy Soeharto
<desc> Description:
the development of Tommy Soeharto trial case
<narr> Narrative:
The document shall tell how the trial case went, whether any incident happened. The trial

case of any other person related to Tommy is not relevant.
</top>

<top>
<num> Number: 19
<title> Megawati overseas visit
<desc> Description:
Report about Megawati visit to other countries for official business
<narr> Narrative:
The document shall report the official visit of Megawati to other countries and the purpose of the visit. Unofficial or personal visit by Megawati is not relevant.
</top>

<top>
<num> Number: 20
<title> the period of Gus Dur becoming a President
<desc> Description:
The start and end date of Gus Dur becoming a President
<narr> Narrative:
Document is relevant as long as it contains the start and end date of Gus Dur becoming a President
</top>

Appendix E

Top 100 Words in the Indonesian Collection

In this appendix, we show the 100 most frequently occurring words in the training collection C_INDO-TRAINING-SET.

yang	dan	di	itu	dengan
untuk	tidak	dari	dalam	akan
pada	ini	jakarta	tersebut	juga
ke	karena	presiden	katanya	ada
kata	kepada	mengatakan	indonesia	mereka
media	oleh	telah	mpr	sudah
as	saat	sebagai	bisa	saya
para	menjadi	melakukan	pemerintah	dpr
namun	ant	negara	bahwa	ketua
menurut	harus	masih	orang	terhadap
antara	sementara	anggota	lebih	secara
dia	setelah	ol-01	atau	tahun
dua	belum	tim	tni	satu
ia	kami	hal	hanya	masyarakat
seperti	dilakukan	ketika	atas	agar
dunia	sekitar	menyatakan	kita	hari
aceh	dapat	adalah	baru	lain
jika	bagi	terjadi	lalu	masalah
sehingga	serta	kasus	merupakan	megawati
kembali	politik	pihak	partai	besar

Figure E.1: Top 100 most frequent words in C_INDO-TRAINING-SET *used as stopwords (read from left to right).*

Appendix F

VEGA-STOP1 Stopwords

In this appendix, we show an Indonesian stopword list, which consists of 169 words, compiled by Vega [2001].

a	adalah	agar	akan	aku
anda	andaikata	antara	apa	apakah
apalagi	asal	atas	atau	b
bagaimana	bagaimanakah	bagi	bahkan	bahwa
begitu	begitulah	berkat	biji	bolehkan
bongkah	buah	buat	bungkus	butir
c	d	dalam	dan	dapatkah
dari	daripada	demi	demikian	dengan
di	dia	dimana	dimanakah	e
ekor	f	g	guna	h
hanya	helai	hingga	i	ialah
itu	itulah	itupun	j	jadi
jangan-jangan	janganka n	k	kah	kalau
kalau-kalau	kalaupun	kami	kamu	kapan
kapankah	karena	kau	ke	kecuali
kemudian	kenapa	kepada	ketika	kita
l	lagi	lah	lalu	lembar
m	maka	malah	malahan	melainkan
mengapa	mengapakah	mengenai	menurut	mereka
meskipun	mula	mula-mula	n	namun
o	oleh	orang	p	padahal
pertama-tama	piring	pula	pun	q
r	s	sambil	sampai	sampai-sampai
samping	saya	seakan	seakan-akan	sebab
sebabnya	sebaliknya	sebelum	sebiji	sebongkah
sebuah	sebungkus	sebutir	sedangkan	seekor
sehelai	sehingga	sejak	selagi	selain
selanjutnya	selembar	semenjak	sementara	seolah
seolah-olah	seorang	seperti	sepiring	seraya
serta	seseorang	sesudah	setelah	seterusnya
siapa	siapakah	supaya	t	tanpa
tempat	tentang	terhadap	tetapi	u
untuk	v	w	x	y
yaitu	yakni	yang	z	

Figure F.1: The list of VEGA-STOP1 *stopwords listed alphabetically (from left to right).*

Appendix G

VEGA-STOP2 Stopwords

In this appendix, we show another Indonesian stopword list, which consists of 556 words, compiled by Vega [2001].

a	acuh	ada	adalah	adil
agak	agar	akal	akan	akhir
akhir-akhir	akibat	akibatnya	aku	amat
ambil	anda	antara	antri	anu
apa	apakah	apalagi	apapun	asumsinya
atas	atau	ayo	ayolah	b
bagaimana	bagaimanakah	bagaimanapun	bagian	bagus
bahwa	baik	bakal	banyak	baru
bawah	beberapa	beda	bekas	belakang
belakangan	benar	berbagai	berbeda	bergaul
berguna	berharga	berhubungan	beri	berikut
berikutnya	berlawanan	bermacam	bermacam-macam	berpikir
bersama	berserta	bertanya	bertentangan	berturut-turut
besar	betul	biar	biarkan	biarlah
biarpun	biasa	biasanya	bilang	bisa
boleh	bolehkah	bukan	bukankah	bukannya
c	c.v.	cara	cenderung	coba
cocok	com	contoh	contohnya	cukup
cv	d	dahulu	dalam	dan
dapat	dapatkah	dari	darimana	daripada
data	datang	dekat	delapan	demikian
dengan	deskripsi	deskripsinya	detik	di
dia	diacuhkan	diambil	diambilnya	diantara
diantaranya	diasosiasikan	diatas	dibawah	dibelakang
dibelakangnya	diberi	diberikan	dibolehkan	dicoba
didalam	didapat	didapati	dideskripsikan	digunakan
dihargai	diikuti	diindikasikan	dijelaskan	dikenal
diketahui	dikirim	dilain	dilakukan	dilihat
diluar	dimana	dimanakah	dimanapun	dinyatakan
diperbolehkan	diperoleh	dipertimbangkan	diri	diriku
dirimu	disamping	disebabkan	disebelah	disebut
disebutkan	disekitarnya	disenangi	disimpan	disimpannya
disini	disukai	ditaruh	ditempat	ditengah
ditengah-tengah	ditolong	ditunjukkan	diusulkan	dll
dsb	dua	dulu	dulunya	e
edu	eks	empat	enam	f
g	ganti	guna	h	hai

Figure G.1: The list of VEGA-STOP2 *Part A stopwords listed alphabetically (from left to right).*

hak	halo	hampir	hanya	harap
harga	hargai	harus	hello	heran
hirau	hormat	i	ikut	indikasi
ingin	ini	itu	j	jadi
jahat	jalan	jangan	jarang	jauh
jelas	jelaslah	jelek	jeleknya	jika
juga	k	kadang-kadang	kalau	kalau-kalau
kali	kami	kamu	kanan	kandungan
kapan	kapankah	kapanpun	karena	kasih
kata	katanya	kau	kayak	ke
kebanyakan	kecil	kecuali	kedalam	kedua
keduanya	keempat	keinginan	kelihatan	kelihatannya
kelima	keluar	kemana	kemari	kembali
kemudian	kemungkinan	kenal	kenapa	kepentingan
kepercayaan	keperluan	kepunyaan	kepunyaannya	kesana-kemari
kesana-sini	keseluruhan	kesini	ketiga	ketika
khusus	khususnya	kira-kira	kiri	kirim
kita	konsekuensi	konsekuensinya	kosong	kuat
kurang	l	lagi	lain	lain-lain
lalu	lantaran	lawan	lebih	lihat
lima	lintas	luar	m	maaf
maka	makhluk	malah	malahan	mampu
mana	manakah	mari	marilah	masih
masing-masing	masuk	mati	mau	melainkan
melakukan	melalui	melawan	melebihi	melihat
memadai	memberi	membolehkan	memikir	memiliki
memperbolehkan	memperhatikan	mempertimbangkan	menanyakan	mencoba
mendapat	mendapatkan	mengambil	mengandung	mengapa
mengapakah	mengenai	mengenal	mengetahui	menggunakan
menghargai	menghiraukan	mengikuti	mengirim	mengizinkan
mengusulkan	menjadi	menolong	menuju	menunjukkan
menurut	menyatakan	menyebabkan	menyebutkan	menyediakan
menyenangi	menyenangkan	menyimpan	menyukai	mereka
meski	meskipun	milik	milikku	milikmu
miliknya	minta	moga-moga	mudah-mudahan	mungkin
n	nama	nampak	namun	nanti
nggak	nol	normalnya	novel	o

Figure G.2: The list of VEGA-STOP2 *Part B stopwords listed alphabetically (from left to right).*

oh	ok	okay	oke	oleh
orang	p	p.t.	pada	padam
pantas	pasti	peduli	pengetahuan	penting
penyebab	per	perbedaan	percaya	pergantian
pergi	perkataan	perlu	permintaan	pernah
persis	pertama	perubahan	pikir	plus
pribadi	pt	pula	pun	punya
q	r	relatif	rubah	s
saat	sadis	saja	salam	sama
sambil	sampai	samping	sangat	satu
saya	sayang	sayangnya	sebab	sebagai
sebagian	sebelah	sebelum	sebelumnya	sebenarnya
sebetulnya	sebiji	sebongkah	sebuah	sebungkus
sebut	sebutir	secara	secepatnya	sedang
sedia	sedikit	sedikitnya	seekor	segera
seharusnya	sehelai	sejak	sejauh	sejujurnya
sekali	sekalipun	sekarang	sekeliling	sekitar
selain	selalu	selama	selamat	selanjutnya
selembar	seluruh	semasa	sembilan	semenjak
sementara	semoga	semua	semuanya	senang
senantiasa	sendiri	sepanjang	seperlunya	seperti
sepiring	sering	serius	serta	seseorang
sesuai	sesuatu	sesudah	sesunguhnya	setelah
setiap	setidaknya	sewajarnya	sewaktu-waktu	sial
sialnya	siapa	siapakah	siapapun	simpan
singkat	spesifik	sub	sudah	suka
sungguh	sungguh-sungguh	sungguhpun	t	tahu
tambah	tampak	tanpa	tanya	tapi
telah	teliti	tempat	tentang	tentu
tepat	terakhir	terbaik	terhadap	terima
terjadi	terkenal	terlebih	terlepas	tersedia
tertulis	terus	terutama	tetapi	tidak
tiga	timbang	toh	tolong	tua
tujuh	tunjuk	turun	turut	u
untuk	utama	uucp	v	vs
w	wajar	waktu	walau	walaupun
x	y	ya	yakin	yang
z				

Figure G.3: The list of VEGA-STOP2 *Part C stopwords listed alphabetically (from left to right).*

Appendix H

TALA-STOP Stopwords

In this appendix, we show the Indonesian stopword list compiled by Tala [2003].

ada	adalah	adanya	adapun	agak
agaknya	agar	akan	akankah	akhir
akhiri	akhirnya	aku	akulah	amat
amatlah	anda	andalah	antar	antara
antaranya	apa	apaan	apabila	apakah
apalagi	apatah	artinya	asal	asalkan
atas	atau	ataukah	ataupun	awal
awalnya	bagai	bagaikan	bagaimana	bagaimanakah
bagaimanapun	bagi	bagian	bahkan	bahwa
bahwasanya	baik	bakal	bakalan	balik
banyak	bapak	baru	bawah	beberapa
begini	beginian	beginikah	beginilah	begitu
begitukah	begitulah	begitupun	bekerja	belakang
belakangan	belum	belumlah	benar	benarkah
benarlah	berada	berakhir	berakhirlah	berakhirnya
berapa	berapakah	berapalah	berapapun	berarti
berawal	berbagai	berdatangan	beri	berikan
berikut	berikutnya	berjumlah	berkali-kali	berkata
berkehendak	berkeinginan	berkenaan	berlainan	berlalu
berlangsung	berlebihan	bermacam	bermacam-macam	bermaksud
bermula	bersama	bersama-sama	bersiap	bersiap-siap
bertanya	bertanya-tanya	berturut	berturut-turut	bertutur
berujar	berupa	besar	betul	betulkah
biasa	biasanya	bila	bilakah	bisa
bisakah	boleh	bolehkah	bolehlah	buat
bukan	bukankah	bukanlah	bukannya	bulan
bung	cara	caranya	cukup	cukupkah
cukuplah	cuma	dahulu	dalam	dan
dapat	dari	daripada	datang	dekat
demi	demikian	demikianlah	dengan	depan
di	dia	diakhiri	diakhirinya	dialah
diantara	diantaranya	diberi	diberikan	diberikannya
dibuat	dibuatnya	didapat	didatangkan	digunakan
diibaratkan	diibaratkannya	diingat	diingatkan	diinginkan
dijawab	dijelaskan	dijelaskannya	dikarenakan	dikatakan
dikatakannya	dikerjakan	diketahui	diketahuinya	dikira
dilakukan	dilalui	dilihat	dimaksud	dimaksudkan
dimaksudkannya	dimaksudnya	diminta	dimintai	dimisalkan

Figure H.1: The list of TALA-STOP *stopwords Part A listed alphabetically (from left to right).*

dimulai	dimulailah	dimulainya	dimungkinkan	dini
dipastikan	diperbuat	diperbuatnya	dipergunakan	diperkirakan
diperlihatkan	diperlukan	diperlukannya	dipersoalkan	dipertanyakan
dipunyai	diri	dirinya	disampaikan	disebut
disebutkan	disebutkannya	disini	disinilah	ditambahkan
ditandaskan	ditanya	ditanyai	ditanyakan	ditegaskan
ditujukan	ditunjuk	ditunjuki	ditunjukkan	ditunjukkannya
ditunjuknya	dituturkan	dituturkannya	diucapkan	diucapkannya
diungkapkan	dong	dua	dulu	empat
enggak	enggaknya	entah	entahlah	guna
gunakan	hal	hampir	hanya	hanyalah
hari	harus	haruslah	harusnya	hendak
hendaklah	hendaknya	hingga	ia	ialah
ibarat	ibaratkan	ibaratnya	ibu	ikut
ingat	ingat-ingat	ingin	inginkah	inginkan
ini	inikah	inilah	itu	itukah
itulah	jadi	jadilah	jadinya	jangan
jangankan	janganlah	jauh	jawab	jawaban
jawabnya	jelas	jelaskan	jelaslah	jelasnya
jika	jikalau	juga	jumlah	jumlahnya
justru	kala	kalau	kalaulah	kalaupun
kalian	kami	kamilah	kamu	kamulah
kan	kapan	kapankah	kapanpun	karena
karenanya	kasus	kata	katakan	katakanlah
katanya	ke	keadaan	kebetulan	kecil
kedua	keduanya	keinginan	kelamaan	kelihatan
kelihatannya	kelima	keluar	kembali	kemudian
kemungkinan	kemungkinannya	kenapa	kepada	kepadanya
kesampaian	keseluruhan	keseluruhannya	keterlaluan	ketika
khususnya	kini	kinilah	kira	kira-kira
kiranya	kita	kitalah	kok	kurang
lagi	lagian	lah	lain	lainnya
lalu	lama	lamanya	lanjut	lanjutnya
lebih	lewat	lima	luar	macam
maka	makanya	makin	malah	malahan
mampu	mampukah	mana	manakala	manalagi
masa	masalah	masalahnya	masih	masihkah
masing	masing-masing	mau	maupun	melainkan

Figure H.2: The list of TALA-STOP *stopwords Part B listed alphabetically (from left to right).*

melakukan	melalui	melihat	melihatnya	memang
memastikan	memberi	memberikan	membuat	memerlukan
memihak	meminta	memintakan	memisalkan	memperbuat
mempergunakan	memperkirakan	memperlihatkan	mempersiapkan	mempersoalkan
mempertanyakan	mempunyai	memulai	memungkinkan	menaiki
menambahkan	menandaskan	menanti	menantikan	menanti-nanti
menanya	menanyai	menanyakan	mendapat	mendapatkan
mendatang	mendatangi	mendatangkan	menegaskan	mengakhiri
mengapa	mengatakan	mengatakannya	mengenai	mengerjakan
mengetahui	menggunakan	menghendaki	mengibaratkan	mengibaratkannya
mengingat	mengingatkan	menginginkan	mengira	mengucapkan
mengucapkannya	mengungkapkan	menjadi	menjawab	menjelaskan
menuju	menunjuk	menunjuki	menunjukkan	menunjuknya
menurut	menuturkan	menyampaikan	menyangkut	menyatakan
menyebutkan	menyeluruh	menyiapkan	merasa	mereka
merekalah	merupakan	meski	meskipun	meyakini
meyakinkan	minta	mirip	misal	misalkan
misalnya	mula	mulai	mulailah	mulanya
mungkin	mungkinkah	nah	naik	namun
nanti	nantinya	nyaris	nyatanya	oleh
olehnya	pada	padahal	padanya	pak
paling	panjang	pantas	para	pasti
pastilah	penting	pentingnya	per	percuma
perlu	perlukah	perlunya	pernah	persoalan
pertama	pertama-tama	pertanyaan	pertanyakan	pihak
pihaknya	pukul	pula	pun	punya
rasa	rasanya	rata	rupanya	saat
saatnya	saja	sajalah	saling	sama
sama-sama	sambil	sampai	sampaikan	sampai-sampai
sana	sangat	sangatlah	satu	saya
sayalah	se	sebab	sebabnya	sebagai
sebagaimana	sebagainya	sebagian	sebaik	sebaik-baiknya
sebaiknya	sebaliknya	sebanyak	sebegini	sebegitu
sebelum	sebelumnya	sebenarnya	seberapa	sebesar
sebetulnya	sebisanya	sebuah	sebut	sebutlah
sebutnya	secara	secukupnya	sedang	sedangkan
sedemikian	sedikit	sedikitnya	seenaknya	segala
segalanya	segera	seharusnya	sehingga	seingat
sejak	sejauh	sejenak	sejumlah	sekadar

Figure H.3: The list of TALA-STOP *stopwords Part C listed alphabetically (from left to right).*

sekadarnya	sekali	sekalian	sekaligus	sekali-kali
sekalipun	sekarang	sekarang	sekecil	seketika
sekiranya	sekitar	sekitarnya	sekurang-kurangnya	sekurangnya
sela	selain	selaku	selalu	selama
selama-lamanya	selamanya	selanjutnya	seluruh	seluruhnya
semacam	semakin	semampu	semampunya	semasa
semasih	semata	semata-mata	semaunya	sementara
semisal	semisalnya	sempat	semua	semuanya
semula	sendiri	sendirian	sendirinya	seolah
seolah-olah	seorang	sepanjang	sepantasnya	sepantasnyalah
seperlunya	seperti	sepertinya	sepihak	sering
seringnya	serta	serupa	sesaat	sesama
sesampai	sesegera	sesekali	seseorang	sesuatu
sesuatunya	sesudah	sesudahnya	setelah	setempat
setengah	seterusnya	setiap	setiba	setibanya
setidaknya	setidak-tidaknya	setinggi	seusai	sewaktu
siap	siapa	siapakah	siapapun	sini
sinilah	soal	soalnya	suatu	sudah
sudahkah	sudahlah	supaya	tadi	tadinya
tahu	tahun	tak	tambah	tambahnya
tampak	tampaknya	tandas	tandasnya	tanpa
tanya	tanyakan	tanyanya	tapi	tegas
tegasnya	telah	tempat	tengah	tentang
tentu	tentulah	tentunya	tepat	terakhir
terasa	terbanyak	terdahulu	terdapat	terdiri
terhadap	terhadapnya	teringat	teringat-ingat	terjadi
terjadilah	terjadinya	terkira	terlalu	terlebih
terlihat	termasuk	ternyata	tersampaikan	tersebut
tersebutlah	tertentu	tertuju	terus	terutama
tetap	tetapi	tiap	tiba	tiba-tiba
tidak	tidakkah	tidaklah	tiga	tinggi
toh	tunjuk	turut	tutur	tuturnya
ucap	ucapnya	ujar	ujarnya	umum
umumnya	ungkap	ungkapnya	untuk	usah
usai	waduh	wah	wahai	waktu
waktunya	walau	walaupun	wong	yaitu
yakin	yakni	yang		

Figure H.4: The list of TALA-STOP *stopwords Part D listed alphabetically (from left to right).*

Appendix I

English Stopwords 1

In this appendix, we show an English stopword list compiled by Salton and Buckley obtained from `http://www.lextek.com/manuals/onix/stopwords1.html` accessed on 30th March 2007. This stopword list contains 429 words.

about	above	across	after	again
against	all	almost	alone	along
already	also	although	always	among
an	and	another	any	anybody
anyone	anything	anywhere	are	area
areas	around	as	ask	asked
asking	asks	at	away	back
backed	backing	backs	be	became
because	become	becomes	been	before
began	behind	being	beings	best
better	between	big	both	but
by	came	can	cannot	case
cases	certain	certainly	clear	clearly
come	could	did	differ	different
differently	do	does	done	down
down	downed	downing	downs	during
each	early	either	end	ended
ending	ends	enough	even	evenly
ever	every	everybody	everyone	everything
everywhere	face	faces	fact	facts
far	felt	few	find	finds
first	for	four	from	full
fully	further	furthered	furthering	furthers
gave	general	generally	get	gets
give	given	gives	go	going
good	goods	got	great	greater
greatest	group	grouped	grouping	groups
had	has	have	having	he
her	here	herself	high	high
high	higher	highest	him	himself
his	how	however	if	important
in	interest	interested	interesting	interests
into	is	it	its	itself
just	keep	keeps	kind	knew
know	known	knows	large	largely
last	later	latest	least	less
let	lets	like	likely	long
longer	longest	made	make	making
man	many	may	me	member
members	men	might	more	most

Figure I.1: The list of English stopwords 1 Part A listed alphabetically (from left to right).

mostly	mr	mrs	much	must
my	myself	necessary	need	needed
needing	needs	never	new	new
newer	newest	next	no	nobody
non	noone	not	nothing	now
nowhere	number	numbers	of	off
often	old	older	oldest	on
once	one	only	open	opened
opening	opens	or	order	ordered
ordering	orders	other	others	our
out	over	part	parted	parting
parts	per	perhaps	place	places
point	pointed	pointing	points	possible
present	presented	presenting	presents	problem
problems	put	puts	quite	rather
really	right	right	room	rooms
said	same	saw	say	says
second	seconds	see	seem	seemed
seeming	seems	sees	several	shall
she	should	show	showed	showing
shows	side	sides	since	small
smaller	smallest	so	some	somebody
someone	something	somewhere	state	states
still	still	such	sure	take
taken	than	that	the	their
them	then	there	therefore	these
they	thing	things	think	thinks
this	those	though	thought	thoughts
three	through	thus	to	today
together	too	took	toward	turn
turned	turning	turns	two	under
until	up	upon	us	use
used	uses	very	want	wanted
wanting	wants	was	way	ways
we	well	wells	went	were
what	when	where	whether	which
while	who	whole	whose	why
will	with	within	without	work
worked	working	works	would	year
years	yet	you	young	younger
youngest	your	yours		

Figure I.2: The list of English stopwords 1 Part B listed alphabetically (from left to right).

Appendix J

English stopwords 2

In this appendix, we show another English stopword list compiled by Salton and Buckley obtained from `http://www.lextek.com/manuals/onix/stopwords1.html` accessed on 30th March 2007. This stopword list contains 571 words and is used for the SMART information retrieval system.

able	about	above	according	accordingly
across	actually	after	afterwards	again
against	aint	all	allow	allows
almost	alone	along	already	also
although	always	am	among	amongst
an	and	another	any	anybody
anyhow	anyone	anything	anyway	anyways
anywhere	apart	appear	appreciate	appropriate
are	arent	around	as	as
aside	ask	asking	associated	at
available	away	awfully	be	became
because	become	becomes	becoming	been
before	beforehand	behind	being	believe
below	beside	besides	best	better
between	beyond	both	brief	but
by	came	can	cannot	cant
cant	cause	causes	certain	certainly
changes	clearly	cmon	co	com
come	comes	concerning	consequently	consider
considering	contain	containing	contains	corresponding
could	couldnt	course	cs	currently
definitely	described	despite	did	didnt
different	do	does	doesnt	doing
done	dont	down	downwards	during
each	edu	eg	eight	either
else	elsewhere	enough	entirely	especially
et	etc	even	ever	every
everybody	everyone	everything	everywhere	ex
exactly	example	except	far	few
fifth	first	five	followed	following
follows	for	former	formerly	forth
four	from	further	furthermore	get
gets	getting	given	gives	go
goes	going	gone	got	gotten
greetings	had	hadnt	happens	hardly
has	hasnt	have	havent	having
he	hello	help	hence	her
here	hereafter	hereby	herein	heres
hereupon	hers	herself	hes	hi
him	himself	his	hither	hopefully

Figure J.1: The list of English stopwords 2 Part A listed alphabetically (from left to right).

how	howbeit	however	id	ie
if	ignored	ill	im	immediate
in	inasmuch	inc	indeed	indicate
indicated	indicates	inner	insofar	instead
into	inward	is	isnt	it
itd	itll	its	its	itself
ive	just	keep	keeps	kept
know	known	knows	last	lately
later	latter	latterly	least	less
lest	let	lets	like	liked
likely	little	look	looking	looks
ltd	mainly	many	may	maybe
me	mean	meanwhile	merely	might
more	moreover	most	mostly	much
must	my	myself	name	namely
nd	near	nearly	necessary	need
needs	neither	never	nevertheless	new
next	nine	no	nobody	non
none	noone	nor	normally	not
nothing	novel	now	nowhere	obviously
of	off	often	oh	ok
okay	old	on	once	one
ones	only	onto	or	other
others	otherwise	ought	our	ours
ourselves	out	outside	over	overall
own	particular	particularly	per	perhaps
placed	please	plus	possible	presumably
probably	provides	que	quite	qv
rather	rd	re	really	reasonably
regarding	regardless	regards	relatively	respectively
right	said	same	saw	say
saying	says	second	secondly	see
seeing	seem	seemed	seeming	seems
seen	self	selves	sensible	sent
serious	seriously	seven	several	shall
she	should	shouldnt	since	six
so	some	somebody	somehow	someone
something	sometime	sometimes	somewhat	somewhere
soon	sorry	specified	specify	specifying

Figure J.2: The list of English stopwords 2 Part B listed alphabetically (from left to right).

still	sub	such	sup	sure
take	taken	tell	tends	th
than	thank	thanks	thanx	that
thats	thats	the	their	theirs
them	themselves	then	thence	there
thereafter	thereby	therefore	therein	theres
theres	thereupon	these	they	theyd
theyll	theyre	theyve	think	third
this	thorough	thoroughly	those	though
three	through	throughout	thru	thus
to	together	too	took	toward
towards	tried	tries	truly	try
trying	ts	twice	two	un
under	unfortunately	unless	unlikely	until
unto	up	upon	us	use
used	useful	uses	using	usually
uucp	value	various	very	via
viz	vs	want	wants	was
wasnt	way	we	wed	welcome
well	well	went	were	were
werent	weve	what	whatever	whats
when	whence	whenever	where	whereafter
whereas	whereby	wherein	wheres	whereupon
wherever	whether	which	while	whither
who	whoever	whole	whom	whos
whose	why	will	willing	wish
with	within	without	wonder	wont
would	would	wouldnt	yes	yet
you	youd	youll	your	youre
yours	yourself	yourselves	youve	

Figure J.3: The list of English stopwords 2 Part C listed alphabetically (from left to right).

Appendix K

French stopwords

In this appendix, we show the French stopword list compiled by the Université de Neuchâtel obtained from `http://www.unine.ch/info/clef` accessed on 30th March 2007.

a	á	â	abord	afin
ah	ai	aie	ainsi	allaient
allo	allô	allons	après	assez
attendu	au	aucun	aucune	aujourd
aujourd'hui	auquel	aura	auront	aussi
autre	autres	aux	auxquelles	auxquels
avaient	avais	avait	avant	avec
avoir	ayant	b	bah	beaucoup
bien	bigre	boum	bravo	brrr
c	ça	car	ce	ceci
cela	celle	celleci	cellelà	celles
cellesci	celleslà	celui	celuici	celuilà
cent	cependant	certain	certaine	certaines
certains	certes	ces	cet	cette
ceux	ceuxci	ceuxlà	chacun	chaque
cher	chère	chères	chers	chez
chiche	chut	ci	cinq	cinquantaine
cinquante	cinquantième	cinquième	clac	clic
combien	comme	comment	compris	concernant
contre	couic	crac	d	da
dans	de	debout	dedans	dehors
delà	depuis	derrière	des	dès
désormais	desquelles	desquels	dessous	dessus
deux	deuxième	deuxièmement	devant	devers
devra	différent	différente	différentes	différents
dire	divers	diverse	diverses	dix
dixhuit	dixième	dixneuf	dixsept	doit
doivent	donc	dont	douze	douzième
dring	du	duquel	durant	e
effet	eh	elle	ellemême	elles
ellesmêmes	en	encore	entre	envers
environ	es	ès	est	et
etant	étaient	étais	était	étant
etc	été	etre	être	eu
euh	eux	euxmêmes	excepté	f
façon	fais	faisaient	faisant	fait
feront	fi	flac	floc	font
g	gens	h	ha	hé
hein	hélas	hem	hep	hi
ho	holà	hop	hormis	hors
hou	houp	hue	hui	huit

Figure K.1: The list of French stopwords Part A listed alphabetically (from left to right).

huitième	hum	hurrah	i	il
ils	importe	j	je	jusqu
jusque	k	l	la	là
laquelle	las	le	lequel	les
lès	lesquelles	lesquels	leur	leurs
longtemps	lorsque	lui	luimême	m
ma	maint	mais	malgré	me
même	mêmes	merci	mes	mien
mienne	miennes	miens	mille	mince
moi	moimême	moins	mon	moyennant
n	na	ne	néanmoins	neuf
neuvième	ni	nombreuses	nombreux	non
nos	notre	nôtre	nôtres	nous
nousmêmes	nul	o	o|	ô
oh	ohé	olé	ollé	on
ont	onze	onzième	ore	ou
où	ouf	ouias	oust	ouste
outre	p	paf	pan	par
parmi	partant	particulier	particulière	particulièrement
pas	passé	pendant	personne	peu
peut	peuvent	peux	pff	pfft
pfut	pif	plein	plouf	plus
plusieurs	plutôt	pouah	pour	pourquoi
premier	première	premièrement	près	proche
psitt	puisque	q	qu	quand
quant	quanta	quantàsoi	quarante	quatorze
quatre	quatrevingt	quatrième	quatrièmement	que
quel	quelconque	quelle	quelles	quelque
quelques	quelqu'un	quels	qui	quiconque
quinze	quoi	quoique	r	revoici
revoilà	rien	s	sa	sacrebleu
sans	sapristi	sauf	se	seize
selon	sept	septième	sera	seront
ses	si	sien	sienne	siennes
siens	sinon	six	sixième	soi
soimême	soit	soixante	son	sont
sous	stop	suis	suivant	sur
surtout	t	ta	tac	tant
te	té	tel	telle	tellement

Figure K.2: The list of French stopwords Part B listed alphabetically (from left to right).

telles	tels	tenant	tes	tic
tien	tienne	tiennes	tiens	toc
toi	toimême	ton	touchant	toujours
tous	tout	toute	toutes	treize
trente	très	trois	troisième	troisièmement
trop	tsoin	tsouin	tu	u
un	une	unes	uns	v
va	vais	vas	vé	vers
via	vif	vifs	vingt	vivat
vive	vives	vlan	voici	voilà
vont	vos	votre	vôtre	vôtres
vous	vousmêmes	vu	w	x
y	z	zut		

Figure K.3: The list of French stopwords Part C listed alphabetically (from left to right).

Appendix L

German stopwords

In this appendix, we show the German stopword list compiled by the Université de Neuchâtel obtained from `http://www.unine.ch/info/clef` accessed on 30th March 2007.

a	ab	aber	aber	ach
acht	achte	achten	achter	achtes
ag	alle	allein	allem	allen
aller	allerdings	alles	allgemeinen	als
als	also	am	an	andere
anderen	andern	anders	au	auch
auch	auf	aus	ausser	außer
ausserdem	außerdem	b	bald	bei
beide	beiden	beim	beispiel	bekannt
bereits	besonders	besser	besten	bin
bis	bisher	bist	c	d
da	dabei	dadurch	dafür	dagegen
daher	dahin	dahinter	damals	damit
danach	daneben	dank	dann	daran
darauf	daraus	darf	darfst	darin
darüber	darum	darunter	das	das
dasein	daselbst	dass	daß	dasselbe
davon	davor	dazu	dazwischen	dein
deine	deinem	deiner	dem	dementsprechend
demgegenüber	demgemäss	demgemäß	demselben	demzufolge
den	denen	denn	denn	denselben
der	deren	derjenige	derjenigen	dermassen
dermaßen	derselbe	derselben	des	deshalb
desselben	dessen	deswegen	d.h	dich
die	diejenige	diejenigen	dies	diese
dieselbe	dieselben	diesem	diesen	dieser
dieses	dir	doch	dort	drei
drin	dritte	dritten	dritter	drittes
du	durch	durchaus	dürfen	dürft
durfte	durften	e	eben	ebenso
ehrlich	ei	ei,	ei,	eigen
eigene	eigenen	eigener	eigenes	ein
einander	eine	einem	einen	einer
eines	einige	einigen	einiger	einiges
einmal	einmal	eins	elf	en
ende	endlich	entweder	entweder	er
Ernst	erst	erste	ersten	erster
erstes	es	etwa	etwas	euch
f	früher	fünf	fünfte	fünften
fünfter	fünftes	für	g	gab
ganz	ganze	ganzen	ganzer	ganzes

Figure L.1: The list of German stopwords Part A listed alphabetically (from left to right).

gar	gedurft	gegen	gegenüber	gehabt
gehen	geht	gekannt	gekonnt	gemacht
gemocht	gemusst	genug	gerade	gern
gesagt	gesagt	geschweige	gewesen	gewollt
geworden	gibt	ging	gleich	gott
gross	groß	grosse	große	grossen
großen	grosser	großer	grosses	großes
gut	gute	guter	gutes	h
habe	haben	habt	hast	hat
hatte	hätte	hatten	hätten	heisst
her	heute	hier	hin	hinter
hoch	i	ich	ihm	ihn
ihnen	ihr	ihre	ihrem	ihren
ihrer	ihres	im	im	immer
in	in	indem	infolgedessen	ins
irgend	ist	j	ja	ja
jahr	jahre	jahren	je	jede
jedem	jeden	jeder	jedermann	jedermanns
jedoch	jemand	jemandem	jemanden	jene
jenem	jenen	jener	jenes	jetzt
k	kam	kann	kannst	kaum
kein	keine	keinem	keinen	keiner
kleine	kleinen	kleiner	kleines	kommen
kommt	können	könnt	konnte	könnte
konnten	kurz	l	lang	lange
lange	leicht	leide	lieber	los
m	machen	macht	machte	mag
magst	mahn	man	manche	manchem
manchen	mancher	manches	mann	mehr
mein	meine	meinem	meinen	meiner
meines	mensch	menschen	mich	mir
mit	mittel	mochte	möchte	mochten
mögen	möglich	mögt	morgen	muss
muß	müssen	musst	müsst	musste
mussten	n	na	nach	nachdem
nahm	natürlich	neben	nein	neue
neuen	neun	neunte	neunten	neunter
neuntes	nicht	nicht	nichts	nie
niemand	niemandem	niemanden	noch	nun
nun	nur	o	ob	oben
oder	oder	offen	oft	ohne

Figure L.2: The list of German stopwords Part B listed alphabetically (from left to right).

Ordnung	p	q	r	recht
rechte	rechten	rechter	rechtes	richtig
rund	s	sa	sache	sagt
sagte	sah	satt	schlecht	Schluss
schon	sechs	sechste	sechsten	sechster
sechstes	sehr	sei	seid	seien
sein	seine	seinem	seinen	seiner
seines	seit	seitdem	selbst	selbst
sich	sie	sieben	siebente	siebenten
siebenter	siebentes	sind	so	solang
solche	solchem	solchen	solcher	solches
soll	sollen	sollte	sollten	sondern
sonst	sowie	später	statt	t
tag	tage	tagen	tat	teil
tel	tritt	trotzdem	tun	u
über	überhaupt	übrigens	uhr	um
und	uns	unser	unsere	unserer
unter	v	vergangenen	viel	viele
vielem	vielen	vielleicht	vier	vierte
vierten	vierter	viertes	vom	von
vor	w	wahr?	während	währenddem
währenddessen	wann	war	wäre	waren
wart	warum	was	wegen	weil
weit	weiter	weitere	weiteren	weiteres
welche	welchem	welchen	welcher	welches
wem	wen	wenig	wenige	weniger
weniges	wenigstens	wenn	wenn	wer
werde	werden	werdet	wessen	wie
wie	wieder	will	willst	wir
wird	wirklich	wirst	wo	wohl
wollen	wollt	wollte	wollten	worden
wurde	würde	wurden	würden	x
y	z	z.b	zehn	zehnte
zehnten	zehnter	zehntes	zeit	zu
zuerst	zugleich	zum	zum	zunächst
zur	zurück	zusammen	zwanzig	zwar
zwar	zwei	zweite	zweiten	zweiter
zweites	zwischen	zwölf		

Figure L.3: The list of German stopwords Part C listed alphabetically (from left to right).

Bibliography

A. Abdelali, J. Cowie, and H. S. Soliman. Improving query precision using semantic expansion. *Information Processing & Management*, 43(3):705–716, 2007.

M. Adriani. Evaluating Indonesian online resources for cross-language information retrieval. In *Proceedings of the Cross Language Information Retrieval: A Research Roadmap*, Tampere, Finland, 2002. Workshop at SIGIR-2002. Accessed on 29th March 2007. URL: `http://ucdata.berkeley.edu:7101/sigir-2002/sigir2002CLIR.pdf`.

M. Adriani and I. Wahyu. University of Indonesia's participation in ad hoc at CLEF 2005. In *Working Notes for the Cross-Language Evaluation Forum 2005 Workshop*, Vienna, Austria, 2005. Accessed on 29th March 2007. URL: `http://www.clef-campaign.org/2005/working_notes/workingnotes2005/adriani05.pdf`.

M. Adriani, J. Asian, B. Nazief, S. Tahaghoghi, and H. E. Williams. Stemming Indonesian: A confix-stripping approach. *ACM Transactions on Asian Languages Information Processing*, 2007. To appear.

F. Ahmad, M. Yusoff, and T. M. T. Sembok. Experiments with a Stemming Algorithm for Malay Words. *Journal of the American Society for Information Science*, 47(12):909–918, 1996.

E. Airio. Word normalization and decompounding in mono- and bilingual IR. *Information Retrieval*, 9(3):249–271, 2006.

A. Z. Arifin and A. N. Setiono. 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)*, Surabaya, Indonesia, 2002. Teknik Elektro, Sepuluh Nopember Institute of Technology.

J. Asian, H. E. Williams, and S. M. M. Tahaghoghi. A testbed for Indonesian text retrieval. In *Proceedings of Australian Document Computing Symposium*, pages 55–58, Melbourne, Australia, 2004.

J. Asian, F. Scholer, S. M. M. Tahaghoghi, and J. Zobel. Automatic identification of English and Indonesian parallel documents. In *Proceedings of Australian Document Computing Symposium*, page 89, Sydney, Australia, 2005a. Poster.

J. Asian, H. Williams, and S. M. M. Tahaghoghi. Stemming Indonesian. In *Proceedings of Australasian Computer Science Conference*, pages 307–314, Newcastle, Australia, 2005b. Australian Computer Society, Inc.

M. Bacchin, N. Ferro, and M. Melucci. A probabilistic model for stemmer generation. *Information Processing & Management*, 41(1):121–137, 2005.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley Longman, Reading, Massachusetts, 1999.

D. Bahle, H. E. Williams, and J. Zobel. Efficient phrase querying with an auxiliary index. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 215–221, Tampere, Finland, 2002. ACM Press.

Z. A. Bakar and N. A. Rahman. 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*, volume 2911 of *Lecture Notes in Computer Science*, pages 653–662. Springer, 2003.

Z. A. Bakar, T. M. T. Sembok, and M. Yusoff. An evaluation of retrieval effectiveness using spelling-correction and string-similarity matching methods on Malay texts. *Journal of the American Society for Information Science and Technology*, 51(8):691–706, 2000.

C. L. Barber. *The English language: A historical introduction*. Cambridge University Press, Cambridge, United Kingdom, 1993.

A. C. Baugh and T. Cable. *A history of the English language*. Prentice Hall, Upper Saddle River, New Jersey, fifth edition, 2002.

B. Billerbeck and J. Zobel. Questioning query expansion: an examination of behaviour and parameters. In *Proceedings of the Australasian Database Conference*, volume 27, pages 69–76, Dunedin, New Zealand, 2004. Australian Computer Society, Inc.

W. F. Bolton. The early history of English. In *The English language*. Sphere Books Limited, Penguin Group, London, United Kingdom, 1988.

M. Braschler and C. Peters. Cross-language evaluation forum: Objectives, results, achievements. *Information Retrieval*, 7(1-2):7–31, 2004.

M. Braschler, P. Schäuble, and C. Peters. Cross-language information retrieval (CLIR) track overview. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 25–33, Gaithersburg, Maryland, 1999. NIST Special Publication 500-246.

A. Broder. A taxonomy of web search. *SIGIR forum*, 36(2):3–10, 2002.

E. W. Brown and H. A. Chong. The GURU System in TREC-6. In *Proceedings of the Text Retrieval Conference (TREC)*, Gaithersburg, Maryland, 1997. NIST Special Publication 500-240.

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. Word-sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, pages 265–270, Berkeley, California, 1991. Association for Computational Linguistics.

P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2): 263–311, 1993.

C. Buckley, G. Salton, and J. Allan. Automatic retrieval with locality information using SMART. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 59–72, Gaithersburg, Maryland, 1992. NIST Special Publication 500-207.

W. B. Cavnar and J. M. Trenkle. N-gram-based text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, Nevada, 1994.

A. Chen and F. C. Gey. Multilingual information retrieval using machine translation, relevance feedback and decompounding. *Information Retrieval*, 7(1-2):149–182, 2004.

A. Chen, F. C. Gey, K. Kishida, H. Jiang, and Q. Liang. Comparing multiple methods for Japanese and Japanese-English text retrieval. In *Proceedings of the First NTCIR Workshop*

on Research in Japanese Text Retrieval and Term Recognition, pages 49–58, Tokyo, Japan, 1999. NTCIR.

J. Chen and J. Nie. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 21–28, Seattle, Washington, 2000. Morgan Kaufmann Publishers Inc.

J. Chen, R. Chau, and C. Yeh. Discovering parallel text from the World Wide Web. In *Proceedings of the Second Workshop on Australasian Information Security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 157–161, Dunedin, New Zealand, 2004. Australian Computer Society, Inc.

S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modelling. Technical Report TR-10-98, Harvard University, Cambridge, Massachusetts, 1998.

A. Chowdhury, M. C. McCabe, D. Grossman, and O. Frieder. Document normalization revisited. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 381–382, Tampere, Finland, 2002. ACM Press. Poster.

G. G. Chowdhury. *Introduction to modern information retrieval*. Facet Publishing, London, United Kingdom, second edition, 2004.

G. Churcher, J. Hayes, S. Johnson, and C. Souter. Bigraph and trigraph models for language identification and character recognition. In *Proceedings of the 1994 AISB Workshop on Computational Lingustics for Speech and Handwriting Recognition*, Univesity of Leeds, United Kingdom, 1994.

CICC. Research on Indonesian dictionary. Technical Report 6—CICC—MT53, Center of the International Cooperation for Computerization, Tokyo, Japan, 1994.

N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 250–257, New Orleans, Louisiana, 2001. ACM Press.

F. Crestani, M. Lalmas, C. J. V. Rijsbergen, and I. Campbell. "Is this document relevant?...probably": A survey of probabilistic models in information retrieval. *ACM Computing Surveys*, 30(4):528–552, 1998.

J. R. Curran and S. Clark. Language independent NER using a maximum entropy tagger. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL (Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics) 2003*, pages 164–167, Edmonton, Canada, 2003. Association for Computational Linguistics.

W. W. Daniel. *Applied nonparametric statistics*. PWS-KENT Publishing Company, Boston, Massachusetts, second edition, 1990.

D. Demner-Fushman and D. W. Oard. The effect of bilingual term list size on dictionary-based cross-language information retrieval. Technical Report LAMP-TR-097, CS-TR-4452, UMIACS-TR-2003-22, University of Maryland, College Park, Maryland, 2003.

Dewan Bahasa dan Pustaka. *Kamus Dewan*. Dewan Bahasa dan Pustaka, Kuala Lumpur, Malaysia, 1991.

T. Dunning. Statistical identification of language. Technical Report MCCS-94-273, New Mexico State University, New Mexico, 1994.

G. Dwipayana. *Sari Kata Bahasa Indonesia*. Terbit Terang, Surabaya, Indonesia, 2001.

J. M. Echols and H. Shadily. *Kamus Indonesia-Inggris: An Indonesian-English dictionary*. Percetakan PT Gramedia, Jakarta, Indonesia, third edition, 1998.

I. Fahmi, 17 March 2004. Personal Communication.

C. Faloutsos. Access methods for text. *ACM Computing Surveys*, 17(1):49–74, 1985.

B. A. Fennell. *A history of English: A sociolinguistic approach*. Blackwell Publishing, Oxford, United Kingdom, 2001.

C. G. Figuerola, R. Gómez, A. F. Z. Rodríguez, and J. L. A. Berrocal. Spanish monolingual track: The impact of stemming on retrieval. In *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001*, Lecture Notes in Computer Science, pages 253–261, Darmstadt, Germany, 2002. Springer.

C. Fluhr. Multilingual information retrieval. In *Survey of the State of the Art in Human Language Technology*, pages 291–305. Center for Spoken Language Understanding, Oregon Graduate Institute, 1995.

A. S. Fraenkel, S. T. Klein, Y. Choueka, and E. Segal. Improved hierarchical bit-vector compression in document retrieval systems. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 88–96, Pisa, Italy, 1986. ACM Press.

W. Frakes. Stemming algorithms. In *Information Retrieval: Data Structures and Algorithms*, chapter 8, pages 131–160. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.

T. Gaustad and G. Bouma. Accurate stemming of Dutch for text classification. *Language and Computers*, 45(1):104–117, 2002.

D. Grossman and O. Frieder. *Information Retrieval: Algorithms and Heuristics*. Springer, Dordrecht, The Netherlands, second edition, 2004.

P. A. V. Hall and G. R. Dowling. Approximate string matching. *ACM Computing Surveys*, 12(4):381–402, 1980.

D. K. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15, 1991.

D. K. Harman. Overview of the first text retrieval conference (TREC-1). In *Proceedings of the Text Retrieval Conference (TREC)*, pages 1–20, Gaithersburg, Maryland, 1992. NIST Special Publication 500-207.

D. K. Harman. Towards interactive query expansion. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 321–331, Grenoble, France, 1988. ACM Press.

T. Hedlund. Compounds in dictionary-based cross-language information retrieval. *Information Research*, 7(2), 2001. Accessed on 29th March 2007.
URL: `http://informationr.net/ir/7-2/paper128.html`.

T. Hedlund, H. Keskustalo, A. Pirkola, E. Airio, and K. Järvelin. Utaclir @ CLEF 2001: Effects of compound splitting and n-gram techniques. In *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum, CLEF 2001, Darmstadt, Germany, September 3-4, 2001. Revised Papers*, volume 2406 of *Lecture Notes in Computer Science*, pages 118–136. Springer, 2002.

S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22): 10915–10919, 1992.

D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, Netherlands, 2001.

D. T. Hill and K. Sen. *The Internet in Indonesia's new democracy*. Routledge, London, United Kingdom, 2005.

T. Hoad and J. Zobel. Methods for identifying versioned and plagiarised documents. *Journal of the American Society for Information Science and Technology*, 54(3):203–215, 2003.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, 1999. ACM Press.

M. Höhl, S. Kurtz, and E. Ohlebusch. Efficient multiple genome alignment. *Bioinformatics*, 18:312–320, 2002.

V. Hollink, J. Kamps, C. Monz, and M. de Rijke. Monolingual document retrieval for European languages. *Information Retrieval*, 7(1-2):33–52, 2004.

D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 329–338, Pittsburgh, Pennsylvania, 1993. ACM Press.

D. Hull. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84, 1996.

D. Hull and G. Grefenstette. Querying across languages: a dictionary-based approach to multilingual information retrieval. In *Proceedings of the ACM-SIGIR International Conference*

on Research and Development in Information Retrieval, pages 49–57, Zurich, Switzerland, 1996. ACM Press.

N. Idris. Automated essay grading system using nearest neighbour technique in information retrieval. Master's thesis, University of Malaya, Malaysia, 2001.

C. Jacquemin, J. L. Klavans, and E. Tzoukermann. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proceedings of the 35th Annual Meeting on Association for Computational Linguistics*, pages 24–31, Madrid, Spain, 1997. Association for Computational Linguistics.

B. J. Jansen and A. Spink. How are we searching the World Wide Web?: A comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1):248–263, 2006.

B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, 36(2):207–227, 2000.

W. Kraaij, J. Nie, and M. Simard. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3):381–419, 2003.

R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 191–202, Pittsburgh, Pennsylvania, 1993. ACM Press.

R. Krovetz and W. B. Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10(2):115–141, 1992.

J. Kupiec. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pages 17–22, Columbus, Ohio, 1993. Association for Computational Linguistics.

T. K. Landauer and M. L. Littman. Fully automatic cross-language document retrieval using latent semantic indexing. In *Proceedings of the Sixth Annual Conference of the UW Centre for the New Oxford English Dictionary and Text Research*, pages 31–38, Ontario, Canada, 1990. UW Centre for the New OED and Text Research.

L. S. Larkey, L. Ballesteros, and M. E. Connell. 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*, pages 275–282, Tampere, Finland, 2002. ACM Press.

D. L. Lee, H. Chuang, and K. Seamons. Document ranking and the vector-space model. *IEEE Software*, 14(2):67–75, 1997.

J. H. Lee and J. S. Ahn. Using n-grams for Korean text retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 216–224, Zurich, Switzerland, 1996. ACM Press.

M. Liberman and C. Cieri. The creation, distribution and use of linguistic data. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 159–164, Granada, Spain, 1998. European Language Resources Association.

K. Lindén. Multilingual modeling of cross-lingual spelling variants. *Information Retrieval*, 9 (3):295–310, 2006.

R. D. Lins and P. Gonçalves. Automatic language identification of written texts. In *SAC'04: Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 1128–1133, Nicosia, Cyprus, 2004. ACM Press.

J. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computation*, 11(1-2):22–31, 1968.

K. Lunde. *CJKV Information Processing: Chinese, Japanese, Korean Vietnamese Computing*. O'Reilly, Sebastopol, California, 1999.

T. Mandl and C. Womser-Hacker. The effect of named entities on effectiveness in cross-language information retrieval evaluation. In *SAC'05: Proceedings of the 2005 ACM Symposium on Applied computing*, pages 1059–1064, Santa Fe, New Mexico, 2005. ACM Press.

G. S. Mann. Fine-grained proper noun ontologies for question answering. In *COLING-02 on SEMANET: Building and Using Semantic Networks*, pages 1–7, Taipei, Taiwan, 2002. Association for Computational Linguistics.

J. Mayfield and P. McNamee. Single n-gram stemming. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 415–416, Toronto, Canada, 2003. ACM Press.

P. McNamee and J. Mayfield. JHU/APL experiments in tokenization and non-word translation. In *Comparative Evaluation of Multilingual Information Access Systems: 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003, Trondheim, Norway, August 21-22, 2003, Revised Selected Papers*, volume 3237, pages 85–97. Springer, 2004a.

P. McNamee and J. Mayfield. Character n-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1-2):73–97, 2004b.

P. McNamee, J. Mayfield, and C. D. Piatko. The HAIRCUT system at TREC-9. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 273–279, Gaithersburg, Maryland, 2000. NIST Special Publication 500-249.

C. T. Meadow, B. R. Boyce, and D. H. Kraft. *Text information retrieval systems*. Academic Press, San Diego, California, second edition, 2000.

I. D. Melamed. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, 2000.

A. M. Moeliono and S. Dardjowidjojo. *Tata Bahasa Baku Bahasa Indonesia*. Departemen Pendidikan dan Kebudayaan, Republik Indonesia, Jakarta, Indonesia, 1988.

R. Nallapati, B. Croft, and J. Allan. Relevant query feedback in statistical language modeling. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 560–563, New Orleans, Louisiana, 2003. ACM Press.

B. A. A. Nazief and M. Adriani. Confix-stripping: Approach to stemming algorithm for Bahasa Indonesia. Internal publication, Faculty of Computer Science, University of Indonesia, Depok, Jakarta, 1996.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

C. Ng, R. Wilkinson, and J. Zobel. Experiments in spoken document retrieval using phonetic n-grams. *Speech Communication*, 32(1–2):61–77, 2000. Special issue on Accessing Information in Spoken Audio.

J. Nie and J. Chen. Exploiting the Web as parallel corpora for cross-language information retrieval. In *Web Intelligence*, pages 218–239. Springer, 2002.

J. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 74–81, Berkeley, California, 1999.

A. F. A. Nwesri, S. M. M. Tahaghoghi, and F. Scholer. Capturing out-of-vocabulary words in Arabic text. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 258–266, Sydney, Australia, 2006. Association for Computational Linguistics.

D. W. Oard and B. J. Dorr. A survey of multilingual text retrieval. Technical Report CS-TR-3615, University of Maryland, College Park, Maryland, 1996.

Y. Ogawa and T. Matsuda. Overlapping statistical word indexing: a new indexing method for Japanese text. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 226–234, New York, New York, 1997. ACM Press.

P. Ogilvie and J. Callan. Combining document representations for known-item search. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 143–150, Toronto, Canada, 2003. ACM Press.

C. Orăsan, V. Pekar, and L. Hasler. A comparison of summarisation methods based on term specificity estimation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC2004)*, pages 1037–1041, Lisbon, Portugal, 2004. European Language Resources Association.

A. Othman. Pengantar perkataan Melayu untuk sistem capaian dokumen. Master's thesis, University Kebangsaan Malaysia, 1993.

M. Padró and L. Padró. Comparing methods for language identification. *Procesamiento del Lenguaje Natural*, 33:155–162, 2004.

C. D. Paice. An evaluation method for stemming algorithms. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 42–50, Dublin, Ireland, 1994. Springer-Verlag New York, Inc.

C. D. Paice. Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science*, 47(8):632–649, 1996.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, New York, 2001.

T. E. Payne. *Describing Morphosyntax: A Guide for Field Linguists*. Cambridge University Press, Cambridge, United Kingdom, 1997.

C. Peters. What happened in CLEF 2005. In *Accessing Multilingual Information Repositories: 6th Workshop of the Cross-Language Evalution Forum, CLEF 2005, Vienna, Austria, 21-23 September, 2005, Revised Selected Papers*, volume 4022 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2006.

C. Pike and I. D. Melamed. An automatic filter for non-parallel texts. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 114–117, Barcelona, Spain, 2004. Association for Computational Linguistics.

A. Pirkola. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 55–63, Melbourne, Australia, 1998. ACM Press.

A. Pirkola, T. Hedlund, H. Keskustalo, and K. Järvelin. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information Retrieval*, 4 (3-4):209–230, 2001.

M. Popovič and P. Willett. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5): 384–390, 1992.

M. Porter. An algorithm for suffix stripping. *Program*, 13(3):130–137, 1980.

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

P. Resnik and N. A. Smith. The Web as a parallel corpus. *Computational Linguistics*, 29 (3):349–380, 2003.

S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(3): 294–304, 1977.

S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

S. E. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 151–162, Gaithersburg, Maryland, 1999. NIST Special Publication 500-246.

S. E. Robertson, S. Walker, M. M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 21–30, Gaithersburg, Maryland, 1992. NIST Special Publication 500-207.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 109–126, Gaithersburg, Maryland, 1994. NIST Special Publication 500-226.

S. Robson. *Instant Indonesian: How to Express 1,000 Different Ideas with just 100 Key Words and Phrases*. Tuttle Publishing, Boston, Massachusetts, 2004.

R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88:1270–1278, 2000.

F. Sadat, H. Déjean, and E. Gaussier. A combination of models for bilingual lexicon extraction from comparable corpora. In *Proceedings of Papillon 2002 Seminar*, pages 16–18, Tokyo, Japan, 2002. National Institute of Informatics.

W. Sahanaya and A. Tan. *The Oxford study Indonesian dictionary*. Oxford University Press, South Melbourne, Australia, 2001.

G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 162–169, Salvador, Brazil, 2005. ACM Press.

M. Sato, H. Ito, and N. Noguchi. NTCIR experiments at Matsushita: Ad-hoc and CLIR task. In *Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition*, pages 71–81, Tokyo, Japan, 1999. NTCIR.

J. Savoy. A stemming procedure and stopword list for general French corpora. *Journal of the American Society for Information Science*, 50(10):944–952, 1999.

J. Savoy. Stemming of French words based on grammatical categories. *Journal of the American Society for Information Science*, 44(1):1–9, 1993.

P. Sheridan and J. P. Ballerini. Experiments in multilingual information retrieval using the SPIDER system. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 58–65, Zurich, Switzerland, 1996. ACM Press.

D. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press LLC, Boca Raton, Florida, 1997.

A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 21–29, Zurich, Switzerland, 1996. ACM Press.

J. Slocum. Machine translation: its history, current status, and future prospects. In *Proceedings of the 10th International Conference on Computational linguistics*, pages 546–561, Stanford, California, 1984. Association for Computational Linguistics.

J. N. Sneddon. *Indonesian: A Comprehensive Grammar*. Routledge, London and New York, 1996.

C. Soanes, A. Stevenson, and S. Hawker. *Concise Oxford English Dictionary*. Oxford University Press, New York, eleventh edition, 2004.

K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing & Management*, 36(6):779–840, 2000.

A. Spink, D. Wolfram, M. B. J. Jansen, and T. Saracevic. Searching the Web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234, 2001.

I. Syu, S. D. Lang, and N. Deo. Incorporating latent semantic indexing into a neural network model for information retrieval. In *Proceedings of the International Conference on Information and Knowledge Management*, pages 145–153, Rockville, Maryland, 1996. ACM Press.

K. Takeda. The application of bioinformatics to network intrusion detection. In *Proceedings of the 39th Annual IEEE International Carnahan Conference on Security Technology*, pages 130–132, Canary Islands, Spain, 2005. IEEE.

F. Tala. A study of stemming effects on information retrieval in Bahasa Indonesia. Master's thesis, University of Amsterdam, Netherlands, 2003.

C. Tang, S. Dwarkadas, and Z. Xu. On scaling latent semantic indexing for large peer-to-peer systems. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 112–121, Sheffield, United Kingdom, 2004. ACM Press.

P. Thompson and C. Dozier. Name searching and information retrieval. In *Proceedings of the Second Conference on Empirical Methods on Natural Language Processing*, pages 134–140, Providence, Rhode Island, 1997. Association for Computational Linguistics.

A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 2–10, Melbourne, Australia, 1998. ACM Press.

E. Ukkonen. Approximate string-matching with q-grams and maximal matches. *Theoretical Computer Science*, 92(1):191–211, 1992.

V. B. Vega. Continuous naive Bayesian classifications. In *Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access*, volume 2911 of *Lecture Notes in Computer Science*, pages 279–289. Springer, 2003.

V. B. Vega. Information retrieval for the Indonesian language. Master's thesis, National University of Singapore, 2001.

V. B. Vega and S. Bressan. Continuous-learning weighted-trigram approach for Indonesian language distinction: A preliminary study. In *Proceedings of 19th International Conference on Computer Processing of Oriental Languages*, Seoul, Korea, 2001.

P. Virga and S. Khudanpur. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 57–64, Sapporo, Japan, 2003. Association for Computational Linguistics.

E. M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing & Management*, 36(5):697–716, 2000.

E. M. Voorhees. Overview of the thirteenth TREC conference (TREC-13). In *Proceedings of the Text Retrieval Conference (TREC)*, Gaithersburg, Maryland, 2004. NIST Special Publication 500-261.

E. M. Voorhees. The TREC-8 question answering track report. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 77–82, Gaithersburg, Maryland, 1999. NIST Special Publication 500-246.

E. M. Voorhees. Overview of TREC 2003. In *Proceedings of the Text Retrieval Conference (TREC)*, pages 1–13, Gaithersburg, Maryland, 2003. NIST Special Publication 500-255.

E. M. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 316–323, Tampere, Finland, 2002. ACM Press.

E. M. Voorhees and D. K. Harman. Overview of the ninth TREC conference (TREC-9). In *Proceedings of the Text Retrieval Conference (TREC)*, pages 1–14, Gaithersburg, Maryland, 2000. NIST Special Publication 500-249.

E. M. Voorhees and D. K. Harman. Overview of the sixth TREC conference (TREC-6). In *Proceedings of the Text Retrieval Conference (TREC)*, pages 1–24, Gaithersburg, Maryland, 1997. NIST Special Publication 500-240.

E. M. Voorhees and D. K. Harman. Overview of the eighth TREC conference (TREC-8). In *Proceedings of the Text Retrieval Conference (TREC)*, pages 1–23, Gaithersburg, Maryland, 1999. NIST Special Publication 500-246.

T. Wakao, R. J. Gaizauskas, and Y. Wilks. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 418–423, Copenhagen, Denmark, 1996. Association for Computational Linguistics.

J. J. Webster and C. Kit. Tokenization as the initial phase in NLP. In *Proceedings of the 14th International Conference on Computational linguistics*, pages 1106–1110, Nantes, France, 1992. Association for Computational Linguistics.

I. J. White. *Bahasa Tetanggaku: a notional-functional course in Bahasa Indonesia*. Longman Cheshire, Melbourne, Australia, 1990.

D. Widdows, B. Dorow, and C. Chan. Using parallel corpora to enrich multilingual lexical resources. In *International Conference on Language Resources and Evaluation*, pages 240–245, Las Palmas, Spain, 2002. European Language Resources Association.

A. Widyamartaya. *Seni Menerjemahkan*. Kanisius, Yogyakarta, Indonesia, thirteenth edition, 2003.

H. Williams and J. Zobel. Searchable words on the web. *International Journal of Digital Libraries*, 5(2):99–105, 2005.

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

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, San Francisco, California, second edition, 1999.

J. Woodier. Perning in the Gyre: Indonesia, the globalised media and the 'war on terror'. In *Conflict, Terrorism and the Media in Asia*. Routledge, New York, New York, 2006.

P. Woods, K. S. Rini, and M. Meinhold. *Indonesian Phrasebook*. Lonely Planet Publications, Hawthorn, Australia, third edition, 1995.

Z. Wu and G. Tseng. Chinese text segmentation for text retrieval: Achievements and problems. *Journal of the American Society for Information Science and Technology*, 44(9): 532–542, 1993.

J. Xu and W. B. Croft. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems*, 16(1):61–81, 1998.

C. C. Yang and K. W. Li. Building parallel corpora by automatic title alignment using length-based and text-based approaches. *Information Processing & Management*, 40(6): 939–955, 2004.

C. Zhai. Fast statistical parsing of noun phrases for document indexing. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 312–319, Washington, D.C., 1997. Morgan Kaufmann Publishers Inc.

A. V. Zhdanova. Automatic identification of European languages. In *Natural Language Processing and Information Systems: 6th International Conference on Applications of Natural Language to Information Systems, NLDB 2002, Stockholm, Sweden, June 27-28, 2002. Revised Papers*, volume 2553 of *Lecture Notes in Computer Science*, pages 76–84. Springer, 2002.

J. Zobel. 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*, pages 307–314, Melbourne, Australia, 1998. ACM Press.

J. Zobel and P. Dart. Phonetic string matching: Lessons from information retrieval. In *Proceedings of the ACM-SIGIR International Conference on Research and Development in Information Retrieval*, pages 166–173, Zurich, Switzerland, 1996. ACM Press.

J. Zobel and A. Moffat. Inverted files for text search engines. *Computing Surveys*, 38(2): 1–56, 2006.

J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems*, 23(4):453–490, 1998.