A Case Study in Decompounding for Bengali Information Retrieval

Debasis Ganguly, Johannes Leveling, and Gareth J. F. Jones

CNGL, School of Computing, Dublin City University, Dublin 9, Ireland
{dganguly, jleveling, gjones}@computing.dcu.ie

Abstract. Decompounding has been found to improve information retrieval (IR) effectiveness for compounding languages such as Dutch, German, or Finnish. No previous studies, however, exist on the effect of decomposition of compounds in IR for Indian languages. In this case study, we investigate the effect of decompounding for Bengali, a highly agglutinative Indian language. Some unique characteristics of Bengali compounding are: i) only one constituent may be a valid word in contrast to the stricter requirement of both being so; and ii) the first character of the right constituent can be modified by the rules of *sandhi* in contrast to simple concatenation. While the standard approach of decompounding based on maximization of the total frequency of the constituents formed by candidate split positions has proven beneficial for European languages, our reported experiments in this paper show that such a standard approach does not work particularly well for Bengali IR. As a solution, we firstly propose a more *relaxed decompounding* where a compound word can be decomposed into only one constituent if the other constituent is not a valid word, and secondly we perform *selective decompounding* by employing a co-occurrence threshold to ensure that the constituent often co-occurs with the compound word, which in this case is representative of how related are the constituents with the compound. We perform experiments on Bengali ad-hoc IR collections from FIRE 2008 to 2012. Our experiments show that both the relaxed decomposition and the co-occurrence-based constituent selection proves more effective than the standard frequency-based decomposition. improving MAP up to 2.72% and recall up to 1.8%.

1 Introduction

Vocabulary mismatch between a query and the documents within a collection is an inherent problem in information retrieval (IR), as a result of which documents relevant to a query, but comprising of a set of words different to it, may be retrieved at low ranks. Word compounding is one of the reasons for such a vocabulary mismatch. To illustrate with an example, if a query comprises of the term *land*, a document predominantly containing the term *farmland* may be retrieved at a lower rank, than a document containing the terms *farming* and *land*. Decomposition of the word *farmland* in a document into the constituents

farm and *land* can potentially result in more hits with the query and hence improve its ranking.

Compound splitting has thus become a standard preprocessing step for compounding languages such as Finnish, Dutch or German, where decomposition typically increases IR effectiveness reasonably well [1–3]. While the effect of decompounding has been thoroughly researched for most European languages, there has been comparatively less research in IR on the decompounding of agglutinating Indian languages, such as Bengali and Hindi. In this paper, we explore the effect of decompounding on IR effectiveness for an agglutinating Indian language, namely Bengali.

Existing approaches of decompounding mainly select the splitting position based on the maximum combined frequency of the candidate constituents [4, 5]. While such approaches have proven useful in increasing retrieval effectiveness for European languages [1–3], our reported experiments in this paper show that such approaches do not work particularly well for Bengali IR. The reason, we believe, is due to the very different inherent characteristics of compounding in an Indian language such as Bengali, as compared to a European language. Let us briefly look into the compounding characteristics of the Bengali language.

Compounds can be decomposed into their constituent parts, which are then indexed together with the compound form. For example, assuming a compositional semantics, the German compound *Nasenspitze* (EN: tip of the nose) can be split into *Nase* (nose) and *Spitze* (tip). In Bengali, two words can be concatenated to represent a totally different concept. For example, the words লোক (lok, EN: people)[1] and সভা (sabhA, EN: assembly) can be compounded to form the word লোকসভা (loksabhA, EN: parliament). In this case, therefore, it is not reasonable to split the compound word লোকসভা into the constituents. Note that this is somewhat conceptually similar to phrases in English, where the phrase *House of Commons* represent a different concept than the constituents *house* and *common*, as a result of which an IR system should treat this phrase as one indexing unit instead of two. A frequency based approach of decompounding such as [4], however, can split up the compound word লোকসভা into the constituents লোক and সভা, because both of these constituents are commonly occurring words and thus should have a high frequency in a Bengali news document collection. This in turn can potentially reduce retrieval effectiveness. Thus, a decompounding algorithm has to be selective in its decision making of whether to split a word or not.

The second inherent characteristic of the Bengali compounding is that one of the constituents may not be a valid dictionary word (or a very rarely used word, thus less likely to occur in a standard Bengali news collection). For example, the word উপনগর (upanagar, EN: town) have উপ and নগর (nagar, EN: city) as its constituents. The prefix উপ expresses in some sense the equivalent concept of *small* in English, but is not a valid Bengali word. In this case, however, it may help to decompound the word উপনগর (upanagar, EN: town) into one constituent নগর (nagar, EN: city), since these words represent the same concept.

[1] In this paper, for every Bengali word, we provide the transliteration in ITRANS notation followed by its English meaning.

Another challenge in Bengali decompounding arises due to the presence of complex compounding rules in Bengali, known as the *Sandhi*[2] rules. According to the Sandhi rule, the first character symbol of the tail constituent can appear in a modified form in the compound. An illustrative example is সূর্যোদয় (suryoday, EN: sun-rise)= সূর্য (surya, EN: sun) + উদয় (uday, EN: rise), where it can be seen that the first character of the tail constituent, viz. "u" is changed to "o" in the compound word. While it is easy to directly apply a Sandhi rule to the constituents and derive the compound, the reverse direction is more complex because one may need to apply the rules of Sandhi at each candidate split position and then check whether the modified second constituent appears in the dictionary.

In this paper, we propose a decompounding method addressing each of the problems introduced above as follows. To address the first issue, our proposed decompounding approach takes into account of how accurately the concept of the constituents correspond to that of the compound word. To address the second problem, we relax the decompounding process by allowing decomposition of a compound word into constituents when at least one constituent is a valid word. The third issue is taken care of by applying sandhi rules during decompounding. Our experiments show that for Bengali, indexing compounds together with their constituents can improve IR effectiveness considerably.

The rest of this paper is organized as follows: Section 2 presents a brief overview of related work. Section 3 provides a general overview of the compounding process and also introduces our proposed approach to decompounding in Bengali. Section 4 describes and discusses our IR experiments. The paper concludes with Section 5 with directions for future work.

2 Related Work

Compounding is a word formation process joining two (or more) constituent words into a new word, the compound. This process can include the simple concatenation of constituent words, joining constituents together by linking elements, or other modifications. Koehn and Knight [4] proposed a compound splitting approach for decompounding German words to find correct translations of compounds and improve MT quality. They examine all possible candidate splits and select the split with the highest probability, which is estimated by the product of constituent frequencies. They allow a few linking elements between compound constituents, e.g. an additional *"s"* or *"es"* between constituents. Braschler and Ripplinger [2] investigated stemming and decompounding for German IR, comparing different decomposition approaches, from language independent methods to linguistic methods, including freely available and commercial solutions. They found that stemming and careful decomposition boosts IR performance.

Bengali compounding is derived from Sanskrit compounds and the analysis of Bengali compounds has a long history. Dash [6] attempted to capture lexico-

[2] The word *Sandhi* literally means *compounding*.

semantic properties of Bengali compounds to describe syntactic and semantic properties of the compound constituents and their change over time. Decompounding for Bengali IR has not been researched in detail, but there exists previous research on word formation and morphology in Bengali. Dasgupta et al. [7, 8] present a brief overview over morphological analysis of compound words in Bengali. They apply a unification-based morphological analysis to parse and split compound words while resolving ambiguities and handling inflectional variation. Roy [9] explores NLP for Bengali MT and investigates decompounding as a means to increase the coverage for lower resourced languages such as Bengali. He observed that decompounding Bengali can decrease the word error rate and increase the BLEU score for MT.

Deepa et al. [10] generate a lexicon of Hindi compounds for speech synthesis. (Hindi compounding is in fact very similar to Bengali compounding). Their approach involves searching a prefix trie-based dictionary to look for the candidate suffixes that can be appended to the current word to form a potential compound. For example, in order to decompound the word লোকসভা (lok sabhA, EN: the parliament), the algorithm while traversing down the nodes in the trie, discovers that the prefix লোক (lok, EN: people) is a valid word, and that the suffix সভা (sabhA, EN: assembly) also exists in the dictionary as an independent word, thus decompounding the word লোকসভা into the compounds লোক and সভা. However, their approach is relatively simple because they did not consider the rules of Sandhi while splitting up a compound word. According to the rules of sandhi, the first character of the suffix constituent may change in the compounded word, which is not handled by the approach described in [10]. Another major difference of [10] with our work in this paper is that their evaluated was performed only on a small corpus of 50 words by comparing their results with respect to manually decompounded words, whereas our approach is applied for IR in Bengali language and thus evaluated by standard IR metrics, which in turn ensures that firstly our decompounding approach is tested on a much larger vocabulary of words, and secondly we are able to see the effect of the decompounding approach on IR effectiveness.

Indexing compound constituents is a linguistically motivated technique. There are several other approaches which aim at relaxing the requirement that index terms have to be words. McNamee et al. [11] and Leveling et al. [12] performed experiments on indexing character n-grams and subwords for Bengali IR. They found that indexing terms on a subword level, an approach similar to indexing compound constituents, can outperform other approaches based on stemming all words. The morpheme extraction task (MET) at FIRE[3], the Forum for Information Retrieval Evaluation, was introduced in 2011 with an aim to evaluate and compare different IR preprocessing techniques (with a focus on stemming), and to provide the corresponding software tools. The task shows that there is a growing interest in scientific evaluation of Bengali IR.

[3] http://www.isical.ac.in/~fire/morpho/MET.html

There are very few software tools supporting Bengali decomposition. Sandhi splitter[4] is a computational tool which shows all possible splittings of a given Sanskrit string. In addition, PC-Kimmo has been extended to process Bengali compounds [8].

3 Bengali Compounding

In this section, we introduce some of the characteristics of compounding in Bengali. Compounds in Bengali are typically formed by concatenation of two (in rare cases more) constituent words, which can be modified in the compounding process. The compounding rules for Bengali are derived from Sanskrit and are called Sandhi rules. For the experiments described in this paper, we consider hyphens as word delimiters and do not consider decompounding hyphenated words as a problem. In contrast, Roy [9] considers splitting Bengali words at hyphen characters whereas we view hyphens as word delimiters by default.

Let the compound word w be formed of a left constituent (usually called *modifier*), denoted by w_L, and a right constituent, denoted by w_R (usually called *head*). Words are concatenated together (without hyphens), with possible morphological inflections and modification of characters on w_R. Inflections on the constituent w_L are not allowed. In European languages, compounds are predominantly endocentric, i.e. a compound $w = w_L{+}w_R$ denotes a special kind of w_R. For example, $w =$ *"darkroom"* means that w is a special kind of *"room"*. In Indian languages, exocentric compounds (Bahuviri compounds, where $w_L{+}w_R$ denotes a special kind of an unexpressed semantic head) could be more frequent. For example, *"skinhead"* refers to a person (unexpressed).[5] We consider four possible cases when splitting a compound:

- Both w_L and w_R are valid dictionary words.
- w_L is not a valid dictionary word, but w_R is. For example, w_L could be a bound morpheme or a word prefix that does not occur independently in the dictionary.
- w_R is not a valid dictionary word, but w_L is.
- w_L is a valid dictionary word, and the first character of w_R is modified according to *Sandhi* rules. An example Sandhi rule is that if the first character of w_R is an independent vowel (e.g. আ), and the last character of w_L is a consonant, then the independent vowel is changed to a dependent one and is appended after the last character of w_L.

Table 1 shows an example of each case along with the frequencies in the FIRE-2008 document collection[6]. The frequencies in the left-most column of the table show that a high percentage of all words in this Bengali collection can be

[4] `http://tdil-dc.in/san/Sandhi_splitter/index_dit.html`

[5] Our proposed decompounding approach would leave this word unchanged, as *"skinhead"* rarely co-occurs with *"head"*.

[6] `http://www.isical.ac.in/~clia/`

Table 1. Compound examples in Bengali. The frequencies are reported on the FIRE-2008 document collection. Each Bengali word is accompanied with a pair of words, respectively denoting its transliteration in ITRANS and its meaning in English.

Freq.	Conditions	w (Compound)	w_L	w_R
3.35%	$inDict(w_L) \wedge inDict(w_R)$	মূল্যবৃদ্ধি (mulyabriddhi) (EN: price-hike)	মূল্য (mulya) (EN: price)	বৃদ্ধি (briddhi) (EN: hike)
29.83%	$\neg inDict(w_L) \wedge inDict(w_R)$	উপনগর (upanagar) (EN: town)	উপ (upa) (EN: vice)	নগর (nagar) (EN: city)
3.86%	$inDict(w_L) \wedge\neg\ inDict(w_R)$	মশারি (moshAri) (EN: mosquito net)	মশা (moshA) (EN: mosquito)	অরি (ari) (EN: enemy)
2.50%	$inDict(w_L) \wedge$ $inDict(\text{applySandhi}(w_R))$	পূর্বাঞ্চল (purbAnchal) (EN: eastern region)	পূর্ব (purba) (EN: east)	অঞ্চল (anchal) (EN: region)

Table 2. Selected vowel Sandhi types.

Sandhi	Rule	Bengali Example / English translation
Dirgha	অ + অ = া	সূর্য + অস্ত = সূর্যাস্ত (sUrja + asta = sUrjAsta)
	(a + a = A)	(EN: sun + set = sunset)
Dirgha	অ + আ = া	মাদক + আসক্ত = মাদকাসক্ত (mAdak + Asakta = mAdakAsakta)
	(a + A = A)	(EN: drug + addicted = drug addict)
Dirgha	আ + আ = া	বিদ্যা + আলয় = বিদ্যালয় (vidyA + Alaya = vidyAlaya)
	(A + A = A)	(EN: education + house = school)
Guna	অ + ই = ে	শ্রবণ + ইন্দ্রিয় = শ্রবণেন্দ্রিয় (shrabaN + indriya = shrabaNendriya)
	(a + i = e)	(EN: hearing + organ = ear)
Guna	অ + উ = ো	সূর্য + উদয় = সূর্যোদয় (sUrja + udaya = sUryodaya)
	(a + u = o)	(EN: sun + rise = sunrise)

compound words (39.54%), out of which $29.83\% + 3.86\% = 33.69\%$ of the words are representative of the cases where only one constituent is a valid dictionary word.

The decomposition process can be complex. Firstly, there may be more than one viable splitting point and the decompounding process has to take into consideration all possible splitting points in a word. Secondly, it has to choose the most likely split between a list of candidate splits. Thirdly, it can be necessary to modify the first character of the constituent w_R by applying the rules of Sandhi. In the next section, we describe our approach of decompounding which considers all of these steps.

3.1 Proposed Decompounding Algorithm

Before describing our proposed algorithm, we first outline its two auxiliary procedures.

- *inDict(w)* is a unary predicate which returns true if the stem of the word parameter w is found in the dictionary. The dictionary, in our case, comprises the vocabulary of the indexed document collection.
- *applySandhi(w_L, w_R)* transforms the first character of the right constituent into another character according to the rules of Sandhi. The *applySandhi* method handles the most frequent Sandhi rules.

Consonant Sandhis occur rarely in the corpus. Examples for the vowel Sandhi rules (Dirgha and Guna Sandhi) are shown in Table 2. We list the steps of our algorithm for splitting a candidate compound word w as follows.

\# initialization
- mw=min. word length # words comprise at least 2 consonants and 1 vowel.
- $splits = \{\}$; $result = \{w\}$

\# generate candidate splits
- FOR $i = mw - 1$ TO $length(w) - mw - 1$
  - Split w into w_L and w_R at position i
  - $w'_R = applySandhi(w_R)$
  - IF $inDict(w_L)$ AND $inDict(w_R)$ THEN $splits = splits \cup (w_L, w_R)$
  - IF $inDict(w_L)$ THEN $splits = splits \cup (w_L)$
  - IF $inDict(w_R)$ THEN $splits = splits \cup (w_R)$
  - IF $inDict(w_L)$ AND $inDict(w'_R)$ THEN $splits = splits \cup (w_L, w'_R)$
- END FOR

\# select best split
- Let w_L and w_R represent the element in $splits$ with the highest value of $cf(w_L) + cf(w_R)$.
- $result = result \cup c$ if $overlap(c, w) > \tau$ (see Equation 1), where $c = w_L, w_R$.
- RETURN result

Our proposed decomposition process is similar to that of [4] and [5] in the sense that we consider all possible candidate splits, and score the candidate splits based on the corpus frequency of compound constituents. However, there are three major differences as follows. The decompounding approach in [4] considered only those decompositions where w_L and w_R are both valid dictionary words. In contrast, due to the linguistic characteristics of Bengali, we needed to consider different cases as described in Section 3.

The second difference is that since decompounding in [4] was performed to improve MT performance, the decision of whether to split a compound word or not was motivated by comparing the collection frequency of the compound with the sum of the frequencies of its constituents. More specifically, a word w is split into the constituents w_L and w_R only if $cf(w_L) + cf(w_R) > cf(w)$. The reason is that it is more likely to find a translation of a highly frequent word in a corpus parallel to the current one. Thus, if the constituents occur more frequently in the corpus, decompounding a compound word can increase their frequencies even more. In IR however, the highly frequent words, due to high inverse document frequency (*idf*), do not play a vital role in retrieval. It is rather the addition of the high *idf* terms which may boost the retrieval score of

Table 3. Document/Query characteristics.

Data	#Documents	Topics	Avg. #rel	Avg. qry length	
				T	TD
FIRE 2008	123,047	26-75	37.26	3.64	13.44
FIRE 2010	123,047	76-125	10.20	4.84	14.18
FIRE 2011	500,122	126-175	55.50	3.30	9.90
FIRE 2012	500,122	176-225	49.08	3.54	10.14

a document significantly in response to a given query. Thus, a selection rule such as the one proposed in [4] may not be particularly suitable for IR. Our proposed algorithm thus does not involve such a check, and we do decompose a word w into w_L and w_R even if $cf(w_L) + cf(w_R) < cf(w)$.

The third difference is that we attempt to estimate the *relatedness* between each constituent w_L and w_R with that of the compound word w, to avoid the cases where the constituents individually may represent a different concept than the compound word. Some examples in Bengali are ধানবাদ (dhAnbAd, the name of a place) = ধান (dhAn, EN: rice) + বাদ (bAd, EN: kept out), and জলপাই (jalpai, EN: olive) = জল (jal, EN: water) + পাই (pai, EN: get). Adding the constituent words in such cases may be harmful e.g., the retrieval after decompounding can retrieve non-relevant documents on ধানবাদ (Dhanbad, a place) when the added constituent ধান (rice) is a query term. We investigate a co-occurrence based measure to selectively apply the decomposition rules only if the co-occurrence between a constituent and the compound is higher than a particular threshold. The intuition is that if a constituent word co-occurs frequently with the compound word, then they represent related concepts, whereas if the co-occurrence is low, then the constituent word is likely to represent a different concept. In the latter case, the compound should not be split. In the last step of the algorithm, we thus employ co-occurrence check, which adds $w_L(w'_R)$ only if its co-occurrence with w is higher than a threshold τ. The co-occurrence measure used is the *overlap coefficient* between the set of documents $D(c)$ containing the constituent term c, with that of $D(w)$ containing the compound, as defined in Equation 1 [13].

$$overlap(w, c) = \frac{|D(w) \cap D(c)|}{\min\{|D(w)|, |D(c)|\}} \tag{1}$$

The cardinalities of the document lists $D(c)$ and $D(w)$ can differ hugely in which case a standard metric, such as the Jacard coefficient, may be too small and thus difficult to threshold. The overlap coefficient on the other hand determines the ratio of the overlap compared to the minimum of the set sizes and hence is easier to threshold.

4 Experiments and Results

In this section, we describe the evaluation experiments for our proposed decompounding method. We start with a brief description of the dataset and tools, which is followed by a description of the different retrieval settings, and finally we present the results and a comparison between the approaches.

4.1 Dataset and Tools

To test the effectiveness of our proposed decompounding approach, we performed IR evaluations on the FIRE monolingual Bengali data used in ad hoc IR evaluations from 2008 to 2012 (see Table 3). Our IR experiments are performed using SMART[7]. with an extension to support language modelling (LM) with Jelinek Mercer smoothing [14]. The smoothing parameter λ was set to 0.4 by optimizing on the FIRE-2008 data. We employed stopword removal using a list of Bengali stopwords[8]. For stemming, we used our rule-based Bengali stemmer[9] [15], which produced the second best retrieval effectiveness in the morpheme extraction task (MET) in FIRE-2012. Note that stemming was applied prior to decompounding.

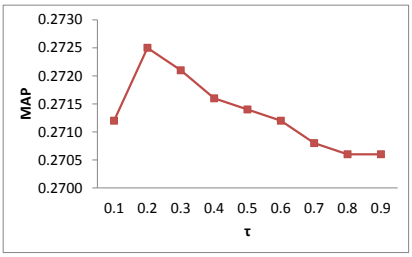

Fig. 1. Optimization of the correlation threshold τ on FIRE-2008.

We chose the topic set 2008 as the training set to optimize the parameter τ, the correlation threshold of Equation 1. The optimal value of $\tau = 0.2$ was set for the other topic sets as well. The variation of MAP with τ for the FIRE-2008 data is shown in Figure 1.

4.2 Run Description

We investigated four different decompounding variants and compared them to a baseline experiment BL using no decompounding:

1. **CF**: We add the constituents with the highest probability estimate based on the sum of constituent frequencies as in [5]. Here, w is split into w_L and w_R only if $cf(w) < cf(w_L) + cf(w_R)$.

[7] `ftp://ftp.cs.cornell.edu/pub/smart/`
[8] `http://www.isical.ac.in/~fire/data/stopwords_list_ben.txt`
[9] `http://www.computing.dcu.ie/~dganguly/rbs.tar.gz`

Table 4. Results for topic title (T) queries.

Topics	BL		CF		CF_2		DC_0		$DC_{0.2}$	
	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret
2008	.2686	1605	.2699	1619	.2684	1604	.2706	1609	**.2725**	**1624**
2010	.3415	463	.3505	464	.3488	465	.3455	464	**.3508**	**468**
2011	.2410	2257	.2401	2251	.2407	2259	.2452	2253	**.2496**	**2270**
2012	.2026	1438	.2016	1429	.2018	1433	**.2043**	**1441**	.2039	1436

Table 5. Results for topic title and description (TD) queries.

Topics	BL		CF		CF_2		DC_0		$DC_{0.2}$	
	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret	MAP	rel_ret
2008	.3118	1686	.3124	1687	.3111	1687	.3064	1687	**.3148**	**1696**
2010	.4315	**500**	.4348	**500**	.4325	499	**.4352**	498	.4336	498
2011	.3201	2464	.3202	2467	.3194	2474	.3245	2480	**.3279**	**2482**
2012	.2961	1763	.2966	1767	.2975	1765	.2966	1765	**.2985**	**1769**

2. **CF$_2$**: Similar to CF, with the additional constraint that decompounding is done only if two valid constituents are found, i.e. restricting CF to cases where both w_L and w_R are dictionary words. This is the standard decomposition technique for IR on Eurpean languages.
3. **DC$_0$**: Decompose words by the algorithm in Section 3.1 with τ set to 0, i.e. we decompound every word in the most likely splitting point, irrespective of any co-occurrence check. The major difference of this approach with that of CF is that CF does not decompound a word w if $cf(w_L)+cf(w_R) < cf(w)$, whereas DC_0 involves a more aggressive decompounding in the sense that we always decompound the word w. The objective of evaluating this approach is to see whether decompounding a word only to one constituent proves beneficial for retrieval.
4. **DC$_{0.2}$**: Decompose by the algorithm in Section 3.1 with the co-occurrence threshold $\tau = 0.2$ (cf. Figure 1), thus ensuring that a constituent is added only if its overlap coefficient with that of the compound is higher than 0.2.

It is worth emphasizing that Sandhi rules are applied on the tail constituent w_R for all the above approaches described while computing collection frequencies.

4.3 Results

Mean average precision (MAP) and the number of relevant documents retrieved (rel_ret) are reported in Table 4 and Table 5 for the T and TD queries respectively. The results show that decompounding approaches in general can increase effectiveness for Bengali IR, in comparison to the baseline approach of no decompounding (BL). There is a consistent improvement in IR effectiveness when

indexing compounds together with their constituents. The improvements, however, are not statistically significant, as measured by Wilcoxon signed rank test with 95% confidence measure.

The results also show that the standard strategy of decompounding based on the collection frequency estimate, CF does not perform the best for Bengali. This can be seen by the lower MAP values in the second, third and the last row of Table 4 corresponding to title topics of 2010, 2011 and 2012. The fact that DC_0 outperforms CF shows that the aggressive approach of decompounding proves beneficial for Bengali.

Moreover, the strategy of decompounding only if two constituents are available, i.e. CF_2 performs worse than CF, as can be seen by comparing the MAP columns of CF and CF_2 in Table 4 and 5. This suggests that for Bengali, it is beneficial to employ a relaxed decomposition and index at least one constituent (see the second and third rows of Table 1 for examples).

Furthermore, we see that the method of selective decompounding based on the overlap coefficient consistently outperforms the selective decompounding with collection frequencies CF and CF_2, or decompounding without threshold (DC_0). The only two cases where DC_0 outperforms $DC_{0.2}$ are the runs on the T query of FIRE-2012 and the TD query of FIRE-2010.

The best percentual improvement in MAP is 2.72% (on FIRE 2010 title queries) using the $DC_{0.2}$ approach, which is lower than what has been achieved for Dutch or German. For comparison, Monz et al. report 6.1% and 9.6% improvement for Dutch and German, respectively [3].

Our experiments show some promising results so far. Clearly, simply using approaches that have been proven successful for languages such as Dutch or German and applying them to Bengali does not produce the same improvements (see the results CF_2 in Tables 4 and 5). In summary, the standard collection frequency based decompounding approach can yield some improvement in MAP. However, our proposed approach of selective decompounding shows a more consistent and typically higher improvement in the experiments, due to the more careful choice of decompounding a word using the degree of co-occurrence of the constituents with that of the compound.

5 Conclusions and Future Work

In this paper, we investigated the effect of decompounding on IR effectiveness for a relatively less researched Indian language, namely Bengali. This paper reviewed compounding characteristics of Bengali and differences compared to European languages. The major differences in compounding characteristics arise firstly due to the rules of Sandhi where the first character of the second constituent appear in a modified form in the compound, and secondly due to the fact that constituents may not be valid dictionary words.

Due to the very different characteristics of Bengali compounding, we proposed a selective decomposition method based on the co-occurence of the constituents and the compound. We observe that for Bengali, selective decompound-

ing with a co-occurrence threshold works best, improving MAP up to 2.72%. We also find that a relaxation of the decomposition process, i.e. allowing decomposition even if only one constituent is a valid word, proves beneficial to improve retrieval quality.

As part of future work, we want to investigate the effect of compounding in other Indian languages, such as Hindi and Marathi. We also want to investigate the effect of our co-occurrence based constituent selection approach for non-Indian languages such as Dutch or German.

Acknowledgments

This research is supported by the Science Foundation of Ireland (grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (`http://www.cngl.ie/`).

References

1. Alfonseca, E., Bilac, S., Pharies, S.: Decompounding query keywords from compounding languages. In: ACL/HLT '08. HLT-Short '08 (2008) 253–256
2. Braschler, M., Ripplinger, B.: Stemming and decompounding for German text retrieval. In: ECIR-03. (2003) 177–192
3. Monz, C., de Rijke, M.: Shallow morphological analysis in monolingual information retrieval for Dutch, German, and Italian. In: CLEF 2001. LNCS (2002) 262–277
4. Koehn, P., Knight, K.: Empirical methods for compound splitting. In: EACL '03, Stroudsburg, PA, USA, ACL (2003) 187–193
5. Chen, A., Gey, F.C.: Multilingual information retrieval using machine translation, relevance feedback and decompounding. Inf. Retr. **7**(1–2) (2004) 149–182
6. Dash, N.S.: The morphodynamics of Bengali compounds – decomposing them for lexical processing. Language in India **6** (2006)
7. Dasgupta, S., Khan, M.: Morphological parsing of Bangla words using PC-KIMMO. In: ICCIT 2004. (2004)
8. Dasgupta, S., Ng, V.: High-performance, language-independent morphological segmentation. In Sidner, C.L., Schultz, T., Stone, M., Zhai, C., eds.: Proceedings of NAACL HLT 2007, April 22–27, 2007, Rochester, NY, USA, ACL (2007) 155–163
9. Roy, M.: Approaches to handle scarce resources for Bengali statistical machine translation. PhD thesis, School of Computing, Simon Fraser University (2010)
10. Deepa, S.R., Bali, K., Ramakrishnan, A.G., Talukdar, P.P.: Automatic generation of compound word lexicon for Hindi speech synthesis. In: LREC'04. (2004)
11. McNamee, P.: N-gram tokenization for Indian language text retrieval. In: FIRE 2008, Kolkata, India (2008)
12. Leveling, J., Jones, G.J.F.: Sub-word indexing and blind relevance feedback for English, Bengali, Hindi, and Marathi IR. TALIP **9**(3) (September 2010)
13. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)
14. Hiemstra, D.: Using Language Models for Information Retrieval. PhD thesis, Center of Telematics and Information Technology, AE Enschede, The Netherlands (2000)
15. Ganguly, D., Leveling, J., Jones, G.J.F.: DCU@FIRE 2012: Rule-based stemmers for Bengali and Hindi. In: FIRE 2012, Kolkata, India, ISI (2012) 37–42