CHAPTER-1

INTRODUCTION

1.1 Preamble

With the growth of the Internet, the presence of different sections of the society on the World Wide Web has increased rapidly in the recent years. It has led to the availability of huge amount of digital data in multiple languages. It has thus become an important area of interest to identify the challenges related to the efficient handling of these large multi-lingual document bases and to develop additional tools to prepare for those challenges. A lot of language processing tools have been developed for the English language. But for non-English languages these tools are insufficient. The development of language processing tools for non-English languages requires various linguistic resources such as dictionaries, thesaurus, etc. which are either unavailable or incomplete. So, development of language-independent intelligent tools for Natural Language Processing (NLP) and Information Retrieval (IR) has become an active research area.

The different NLP and IR applications require the creation of lexicons or language models from the large collection of documents. But, various morphological variations in the words of the language pose a great difficulty in creation of these vocabularies or language models especially for morphologically rich languages (where a single word can have a large number of morphological variations). Thus, there is a need for development of tools that can match the morphological variants of the language to their root or base word forms. Stemming is one such pre-processing tool that is used to handle all these morphological variations and is widely used in a number of information retrieval, language modeling, and natural language processing applications.

1.2 Morphological Variations in Languages

Morphology is the branch of linguistics that is related to the study of the words, their formation, and internal structure. It analyzes different components of the words such as prefixes, infixes, suffixes, stems, etc. It also considers the parts of speech, intonation, stress, and the different context in which the words are used.
In a language, the different morphological variants are formed from the basic word forms to form grammatically correct sentences. These morphological variants are produced using different linguistic processes such as affixation, compounding, modification, etc. [1-2]. Affixation is one of the most common linguistic processes used for generation of morphological variants from the base forms in which prefixes and/or suffixes are added to the base form according to the language-specific rules. (For e.g. un + like = unlike, like + ly = likely, un + like + ly = unlikely). Compounding is another process that is used to derive new morphological word forms. In this process, two or more base word forms are used to yield a new word form (For e.g. book + store = bookstore). Similarly, modification (man to men), reduplication (bye to bye-bye) are some other processes that are used to form morphological variants from the base word forms.

The morphology of a language can be broadly classified into two main categories: Inflectional morphology and Derivational morphology. Inflectional morphology deals with the variations caused in the word due to the syntax. The inflectional affixes are attached according to the grammatical function of the word so as to make grammatically valid sentences. The various inflectional variations in the English language include changes in nouns (for e.g. plural word forms (boy to boys), possessive forms (Smith to Smith’s)), verbs (for e.g. want to wanted), adjectives (for e.g. comparative and superlative forms (bright to brighter and brightest). The inflectional variations do not affect the part-of-speech or meaning of the word; an adjective still remains an adjective even in comparative or superlative forms.

Derivational morphology, on the other hand, involves changes in the function or the meaning of the word. These variations may or may not affect the part-of-speech of the word. For instance, the word read (verb) can be converted into reader (noun) by attaching a derivational suffix –er whereas the word bachelorhood remains noun even after the addition of the derivational suffix –hood to the noun bachelor. The derivational variations create new lexemes or base forms to which other inflectional or derivational affixes can be attached. For example, the derivational suffix –atic can be attached to the word system (noun) to form new word systematic (adjective) to which further suffixes can be added (systematic to systematical). The inflectional and derivational both can appear in a word in certain definite order. In case of the English language, the inflectional suffixes are followed by derivational suffixes and once the inflectional suffixes are attached then no more derivational
suffixes can be attached. For example, the variant *readers* is produced by first adding the derivational suffix *–er* to the base word *read* and then adding the inflectional suffix *–s* to it.

1.3 Stemming

Stemming is a linguistic process in which the various morphological variant word forms are matched to their stem (base word, root). Stem is the basic word or a part of a word (which may or may not be meaningful) from which new word forms are formed using different linguistic processes. For instance, the variants *reads, reading, reader, readers, readable,* etc. are matched to their stem *read* through stemming. The programs or algorithms that perform text stemming are called stemmers. The development of text stemming algorithms started in the late 1960s [3] and there has been a substantial advancement in the recent years. A large number of stemming techniques for different languages have been proposed in the literature. However, the design of an efficient language-independent unsupervised stemming technique is quite challenging.

Stemming is a widely used mechanism in a number of applications and the researchers in the fields of computational linguistics, language modeling, and information retrieval consider it an important pre-processing step for different reasons [4]. Stemming is found to be more beneficial in highly inflectional and morphologically rich languages where a large number of morphological variants can be formed through a single root word.

1.3.1 Basic Concepts

— *Stem:* Stem is a part of a word from which new morphological variant words are formed through linguistic processes. A stem may be a meaningful and valid word (free stem) or it may be a meaningless and invalid word (bound stem) which needs one or more affixes to be added so as to form a linguistically valid word. For example, *hard* is the free stem in case of variant *hardly* whereas *dur* is the bound stem for the variant *durable*.

— *Lemma and Lexeme:* Lemmas are linguistically valid and meaningful components. The lemma is that word form which is present in the dictionary and is conventionally used to represent all the lexemes. Lexemes are the group of morphological variant
words which have the same meaning and lemma is one particular form used for all those lexemes. For example, help, helps, helping, helped are different lexeme forms represented by the lemma help.

— Stemming and Lemmatization: Stemming and lemmatization are both related and are often regarded as sibling processes. Both stemming and lemmatization are used to map the morphological variants to their base forms but the output of both the processes is different. The outcome of stemming procedure is a stem and the output of lemmatization procedure is a lemma.

Stemming is a faster and simpler procedure that does not consider the context or its grammatical category to return the stem of the word whereas lemmatization is a complex procedure which considers both the context and grammatical category of the word to determine its lemma [5]. Lemmatizers carry out complete morphological analysis of the word to return the lemma whereas stemmers remove various morphological variations to return stem which may or may not be linguistically valid. For example, the lemmatizer will map the variants produces, produced, producing, production to the lemma produce whereas the stemmer will map it to the stem produc. But, it cannot be regarded as a limitation of stemmers, as the variant words in different applications are invisibly stemmed for an end user.

Stemmers are designed to handle both inflectional and derivational variations whereas lemmatizers can handle inflectional variations only [6]. Stemmers are task-oriented and are usually designed in the context of a specific application whereas lemmatizers are usually designed independently of any target application. Moreover, stemmers are also semantically-oriented and have the ability to conflate the variants that have similar meanings or contexts. Further, fully unsupervised language-independent stemmers can be developed which do not require any linguistic resource or human intervention, but currently, no lemmatizer has been developed in the literature that can be trained in an unsupervised way.

— Light and Aggressive Stemming: Light and aggressive stemming are the terms used in context with the methods that strip suffixes from variant words to find the stems. It is a measure of how to compare different stemming methods in terms of the amount of truncation involved in the suffix stripping process. Light stemming methods remove simple inflectional variations caused due to changes in gender, number, case, person, mood, and so on. These methods tend to leave the word intact and do not form too short stems in doubtful cases. These methods under-stem the
words and thus favor precision over recall. For example, the word *jumping* is mapped to the word *jumpi* by the light stemming methods. Aggressive stemmers, on the contrary, remove derivational variations also and can detect changes in part-of-speech. These stemmers tend to over-stem the words and remove inflections even at the risk of creating too short stems. For example, the words *generic* and *general* are both mapped to the stem *gene* by an aggressive stemming method.

### 1.4 Classification of Stemming Methods

Text stemming is a well-studied technology and has rich literature. A large number of stemming techniques of different nature and flavors have been proposed in the literature. The current state-of-the-art stemming techniques have been broadly classified into three major categories namely *Rule-Based, Corpus-Based, Hybrid*. Figure 1.1 shows the classification of stemming techniques into different categories.

![Stemming Techniques](image)

Fig. 1.1: Classification of stemming techniques

#### 1.4.1 Rule-Based Stemming Techniques

Rule-based stemmers employ set of pre-defined language-specific rules to determine the stem or base form of a word. So, rule-based stemmers are also known as language-specific stemmers. The set of language-specific rules are manually designed by the linguists or experts of the particular language. The rule-based stemming techniques may vary from simple methods such as deletion of plural and verb forms to complex methods that delete a variety of suffixes. The creation of language-related
rules is a very time-consuming process, and moreover, these methods make use of additional resources such as stem tables, suffix lists, dictionaries, etc. So, for resource-scarce languages, these stemming methods are not preferred. The major advantage of language-specific stemmers is their ease of use. The language-related rules once designed can be employed on any corpus without any further processing. Moreover, these methods can employ complex morphological stemming rules that not only remove suffixes from the variant words but also change the entire word (for e.g., *went* to *go*) [6]. The rule-based stemming methods are further categorized into following categories:

1. **Brute-Force Methods:** These methods maintain a lookup table that stores the base word corresponding to the various inflectional and derivational forms of a word [7]. In order to determine the stem of the word, the lookup table is scanned. If the variant word is formed in the table, then the associated stem is returned. These methods are also called dictionary-based or table lookup methods. These stemming methods are quite easy, simple and straightforward, and moreover, they can stem the exceptional cases where the variant words do not follow the language-related stemming rules appropriately. For example, the suffix stripping algorithms can stem the morphological variant *going* to *go* but it would fail to stem the alternate variant *went*. The major disadvantage of these stemming methods is that they cannot stem the word outside the lookup table and it is not possible to collect and store all the variant word forms of a language. Moreover, these techniques require a lot of storage space to record the relations between the morphological variants and their stems.

2. **Affix Removal Methods:** The suffix and prefix of the word are called affix. These methods strip the prefix and/or suffix from the variant words using prefix and suffix list along with context-sensitive stemming rules to determine the stem of the word [7-8]. A lot of research work has been reported for suffix stripping as compared to prefixes. One notable disadvantage of these methods is that the stems produced by these methods after stripping affixes may or may not be linguistically valid words of the language. For example, a suffix stripping algorithm would stem the variants *generate*, *generated*, *generating*, etc. to the stem *gener* which is not a valid English word. The truncated outputs produced by these stemming methods are sometimes not human understandable and pose great difficulties in some applications. For instance, in information retrieval systems these truncated outputs do not harm because the
variant words in the queries and documents are invisibly stemmed from the end user. The main use of stems in these systems is to increase the term-frequency by mapping all the related morphological variants to a single stem. But in applications such as word sense disambiguation, these truncated outputs are not useful because it is not possible to determine the correct meaning of the word from these truncated stems. Another important limitation of affix removal methods is that they can handle variations caused due to affixation only and cannot stem the variants formed through compounding, conversion, etc.

3. Morphological Stemming Methods: These methods consider the morphology of the language to perform stemming. These methods employ lexicons of language in which the words are grouped according to the syntactic and semantic variations. They consider both inflectional and derivational morphology while stemming. Inflectional analysis identifies variations in number, gender, case, person, tense, voice, or mood. Derivational analysis identifies variations in part-of-speech and can transform various surface forms to their base form [9]. For example, the variant enhancement is mapped to the base word enhance but the variant department is not mapped to depart. Derivational stemmers can identify variations in the categories of the word such as nominalization (a noun derived from another part-of-speech such as intensity from intense), deverbal (a noun or adjective phrase derived from verb such as discovery from discover), deadjectival (a phrase generated from an adjective such as intensify from intense), and denomial (a phrase derived from a noun phrase such as victimize from victim). The morphological stemmers consider both syntax and semantics of the word and thus provide a better output which are linguistically valid stems. Moreover, morphological stemmers can stem exceptional cases and the out-of-vocabulary words by using both lexicon and language-related rules. The word to be stemmed is first searched in the lexicon. If the word is not found in the lexicon, then the language related rules are applied to return the stem of the word. The development of morphological stemmers requires complete expertise of the language and its morphology.

1.4.2 Corpus-Based Stemming Techniques

Corpus-based stemming methods employ semi-supervised or unsupervised learning of stemming rules from a corpus of a particular language. These methods discover set
of morphologically related words from the ambient corpus and thus obviate the need for any linguistic resource or expertise. The corpus-based stemmers perform statistical analysis on the lexicon obtained from the corpus or consider context or co-occurrence of the word in the corpus to return morphologically related words. So, these stemming methods are also called *statistical* or *language-independent* methods. The major advantage of statistical stemmers is that these stemmers can easily be applied to the new language if the language meets the basic presumption of the statistical model (such as morphological variants can be generated through affixation process only). This feature of language independence of statistical models is quite useful particularly for applications related to information retrieval. Moreover, corpus-based stemmers can handle complex morphological variations, sparse data, and infrequent cases in languages. The corpus-based stemmers are good alternatives to rule-based stemmers especially for resource-scarce languages [6, 10, 11, 12, 13]. Some distinguishing features of rule-based and corpus-based stemmers are compared in Table 1.1. The corpus-based stemmers follow three types of approaches:

1. **Lexicon Analysis-Based Stemmers:** These stemming methods analyze the set of words (lexicon) obtained from the corpus to understand the morphology of the language [14]. The morphological variant words are conflated using methods such as computation of frequency of substrings, frequency of suffixes [15-16], string similarity or distance [13, 17], etc. These methods consider each word in the corpus individually to group lexicographically related words.

2. **Corpus Analysis-Based Stemmers:** These stemming methods conflate morphologically related words in the corpus by analyzing their co-occurrence or context with the other words in the corpus. These methods use different statistical analysis techniques such as co-occurrence strength [11-12], distributional similarity [18], expected maximization [19], etc. These methods are based on the principle that the words which co-occur in the corpus are better members to be grouped together in a class than the words which do not co-occur. Corpus analysis-based methods require relatively larger corpora as compared to lexicon analysis-based methods so as to increase the reliability of co-occurrence or contextual data in the corpus [12].
3. Character N-Gram Based: These methods analyze the n-grams obtained from the corpus to learn morphology of the languages. The morphological variant words are conflated using the probabilities or frequencies of n-grams. These methods can handle variations caused due to compounding, spelling exceptions along with other inflectional and derivational variations. These methods are found to be quite suitable for alphabetic languages [11].

<table>
<thead>
<tr>
<th>Features</th>
<th>Rule-Based Stemmers</th>
<th>Corpus-Based (Lexicon, Corpus, N-Gram Based)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brute-Force</td>
<td>Affix Removal</td>
</tr>
<tr>
<td>Use of rule and suffix</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Use of dictionary/stem</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>tables/context</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Use of Corpus</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Knowledge of Language</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Involve Computations</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Execution Time</td>
<td>Fast</td>
<td>Medium</td>
</tr>
</tbody>
</table>

1.4.3 Hybrid Stemming Techniques

Hybrid stemming techniques perform stemming by combining several methods. The combination of different methods usually helps in enhancing the performance of the stemmer. Hybrid stemmers can be developed using different combination strategies such as combining various rule-based methods or combining a rule-based and a corpus-based approach. For instance, the effectiveness of a rule-based affix stripping algorithm can be further enhanced by combining with a table lookup algorithm for unusual inflections such as go to went. So, the input word to be stemmed is first scanned in the lookup table. If the word is found, then the stem is returned. Otherwise, the word is stemmed using stemming rules. Similarly, the equivalence classes produced by a rule-based algorithm can be further refined by using corpus-specific knowledge such as co-occurrence or context statistics.
1.5 Stemming Algorithms: Purpose

The stemming process has been perceived in different prospects and has been employed in linguistic processing and information retrieval applications for different reasons. The stemming techniques are usually developed for a particular application and they perform different functions in different type of applications. In the subsequent sub-sections, we describe the purpose of stemming in different natural language processing and information retrieval applications.

1.5.1 Stemming in Information Retrieval

Information Retrieval is a branch that aims to represent, store, organize and retrieve various information resources [8]. These resources can be text documents, catalogs, web pages or multimedia items. Earlier the scope of IR was limited to index the text or for searching the useful documents. Nowadays IR is widely used for classification, modeling, visualization of the text, web search, filtering, user interfaces, etc.

The information retrieval users have different needs of information and of different complexity. So the needs of the user must be first converted into query containing the relevant keywords and then the IR system is used to find all the relevant information to the user query with least amount of non-relevant information.

IR systems make use of keywords or index terms to index documents and the user queries. These keywords can be a single word or group of words that define the documents. So when the user writes a query, it is first parsed and the keywords of the query are generated. The keywords of the query are matched with the documents keywords from the document collection. The documents so retrieved are ranked according to different models of IR system and the relevant documents are displayed in the decreasing order of the ranks.

Stemming is employed at an early stage in the text pre-processing chain of information retrieval systems as shown in Figure 1.2. Stemming in information retrieval serves two major purposes. Firstly, stemming matches the morphological variant word forms in the documents and queries to the same stem, thereby solving the issue of vocabulary mismatch at the time of indexing and searching. So, stemming in IR not only improves recall but also improves precision by promoting relevant documents at higher ranks [19]. The performance of stemming algorithms in information retrieval increases with the increase in the morphological complexity of
the language. Thus, stemming can be considered as a part of a user interface that allows the user to use any morphological variant of the word while formulating the query.

Secondly, the grouping of variant word forms to the same stem reduces the size of the lexicon. The storage space required to store the different data structures for the index terms is reduced considerably with the help of stemming. For highly inflectional languages, stemming is found to be so advantageous that the size of the index file is half as compared to the original index size.

1.5.2 Stemming in Natural Language Processing

In natural language processing applications, text stemming has been employed at the pre-processing stage to reduce the size of the word vocabulary, and hence the dimensionality of the feature sets or training data for statistical models. The use of stemming in different natural language processing applications is discussed below.

— Text Classification and Clustering: Text classification systems aim at labeling the texts in natural languages with the pre-defined genres or categories. These systems are quite useful in a number of applications like email and spam filtering, automatic
document indexing, classification of web documents, text filtering, etc. In these text classification systems, the documents or text words are represented as features in the classification model. So, grouping various morphologically related words in the documents can reduce the dimensionality of the classification model. Several studies [18, 20] have reported that stemming helps in increasing the accuracy of the classification systems. Like text classification, stemming is also found to be quite advantageous for text clustering systems where the similar texts or documents are grouped together but are not assigned to the pre-defined labels. Researchers have been employing stemming [21-23] at the text pre-processing step of the text clustering systems to enhance the quality and accuracy of clusters by reducing the word variants to their stems.

— *Machine Translation:* Machine translation, an important subfield of natural language processing employs computer software to translate the speech or text in one natural language to another natural language. Stemming helps in improving the quality and complexity of the machine translation systems by reducing the variant word forms. Stemming lowers the sparsity of the translation tables and language models thereby improving the performance of the machine translation system [24]. Lavie et al. [25] experimentally verified that the stemming is useful in enhancing not only simple machine translation metrics like precision and recall but also extensively used machine translation metrics like BLEU and NIST. Goldwater and McClosky [26] demonstrated that stemming helps in an appropriate approximation of word-to-word alignment counts in statistical machine translation systems. Several studies in the literature [27-29] has verified that for languages with complex morphology, stemming not only reduces the training data size but also lowers the out-of-vocabulary rates and the lexical granularities of the source and target language.

— *Automatic Text Summarization Systems:* Automatic text summarization is quite a challenging and demanding task in human communication technology that involves producing a compact representation of the text while retaining the important and essential information in the text. The major challenge in this field is to produce summaries that are quite close to human-generated summaries. Stemming helps in enhancing the performance and complexity of automatic statistical summarization models by reducing the size of training data and feature sets. The experiments performed by Méndez-Cruz et al. [30] and Louis and Nenkova [31] have revealed that
the stemming employed in the pre-processing step of text summarizations systems helps in improving the performance of these systems.

— **Question Answering Systems:** Question answering systems aim to determine the relevant answers to the natural language questions posed by the user from a large document collection. Text stemming approaches are quite useful in question answering systems as it resolves the issue of vocabulary mismatch between the questions and answers at the time of indexing and searching. Various studies [32-34] have demonstrated that stemming significantly increases the performance of question answering systems both in terms of precision and recall if employed at the time of indexing and retrieval.

— **Part-of-Speech Tagging Systems:** Stemming techniques have been applied to part-of-speech (POS) tagging systems also, which assigns the best syntactic category to the input word of the text according to the context of the word. In POS tagging systems, stemming maps the variant word forms to their stem during the learning phase which helps in enhancing the probability of finding the correct choice for the input word. Shrivastava and Bhattacharya [35] experimentally demonstrated that efficient POS tagging systems can be developed by using stemming and without using any other linguistic tool such as a morphological analyzer, structured lexicon, etc. Al Shamsi and Guessoum [36] and Khoja [37] also reported a positive impact of stemming in POS tagging systems.

— **Word Sense Disambiguation:** Word sense disambiguation systems determine the correct meaning or sense of the input word according to its occurrence or context in the sentence. The different morphological variations not only represent relationships between the words but also represent the relation between the word senses [9]. The stemming algorithm can utilize all these relationships to disambiguate the meaning of the word. For example, stemmer can predict the correct sense of the input word in the sentence with the knowledge of irregular morphology or part-of-speech [38].

— **Text Searching:** The major issue in text searching systems is that the user must identify the correct variant form of the word that must be used in the query so as to provide the optimal search results. Stemming allows the user to formulate the query with any variant word form without bothering about the most appropriate form. Context sensitive stemming performed at the document and the query side helps in improving the performance of the text searching systems. Context-based stemming algorithms do not blindly stem the query terms rather they consider the context of the
words in the query and provide those variants that are related to the original intent or thematically coherent with the query terms. [12, 39, 40]. Stemming on the document side performs context-based matching of the documents and retrieves the documents that match with the expanded query terms. Context-sensitive stemming on the document and query side not only improves the performance of text searching in terms of precision and recall but also reduces the computational load and query traffic [39].

— Miscellaneous Applications: Besides the above discussed application areas, stemming is found to be quite useful in a number of other applications like spell checking, named entity recognition [41], keyword extraction [42], sentiment analysis [43], opinion mining [44], email and SMS spam filtering [45], etc. In fact, stemming can be employed in any area where document or text processing is needed to handle the issue of vocabulary mismatch, reduction in dimensionality of the feature set or training data, and hence improving the performance of the system.

1.6 Stemming Evaluation Metrics

The effectiveness and performance of stemming methods have been computed through a variety of methods and has always been a debated affair. The different stemming evaluation metrics proposed in the literature can be broadly divided into two major categories: Direct and Indirect evaluation methods. In the following subsections, we describe the direct and indirect methods of evaluation of stemming algorithms.

1.6.1 Direct Evaluation Metrics

Direct stemming evaluation methods test the effectiveness of stemming algorithm directly on a set of test words without any target application. These methods evaluate the performance of stemming methods in terms of certain features such as conflation or compression rate, error percentage, statistical significance tests, etc. These direct evaluation methods are quite complex as they require a lot of manual labor to develop the various test sets in advance. The popular direct evaluation methods proposed in the literature are discussed below.

— **Under-Stemming Errors**: Under-stemming refers to the case when the stemming algorithm does not remove the suffix up to the expected level. In other words, under-stemming is the case in which the word-pairs that share the same stem are not conflated together by the stemming algorithm. For instance, the well-known Porter stemming algorithm [47] does not conflate the morphologically related words *adhesion* and *adhere* to the same stem. The different methods such as partial matching [3, 48] are quite useful in handling these errors [49]. Paice [46] provided a metric called Under-stemming Index (UI) to estimate the under-stemming rate of the stemmer. The Under-stemming Index is computed as $UI = 1 - CI$; where CI denote Conflation Index and is defined as the ratio of the number of words in the lexicon that are correctly conflated together is to the total number of words.

— **Over-Stemming Errors**: Over-stemming to the case when the stemming algorithm performs more terminations thereby treating valid word endings as suffixes. In other words, over-stemming tends to truncate those parts of the input word which belong to the stem. Over-stemming errors tend to conflate unrelated morphological variants together in the same group. For instance, the words *illegible* and *illegal* are conflated together as they are stemmed to the same root *illeg*. Like under-stemming errors, over-stemming errors also lower the performance of the stemming algorithm as two unrelated words are assigned the same stem. These kinds of stemming errors are reduced by imposing certain constraints on the output of the stemming algorithm such as minimum length of the output stem. Paice [46] provided a metric called Over-stemming Index (OI) to estimate the over-stemming rate of the stemmer. The Over-stemming Index is computed as $OI = 1 - DI$; where DI denote Distinctness Index and is defined as the ratio of the number of morphologically related word pairs not grouped together is to the total number of words in the lexicon. The ratio of under-stemming index is to the over-stemming index is defined as Stemmer Weight (SW). A high score of SW denotes a strong stemming algorithm whereas a low score indicate a weak or light stemming algorithm.

2. **Hull Evaluation Mechanism**: Hull [50] suggested that if the performance difference between the stemming procedures is quite small, then it is very difficult to judge that whether the difference is significant or not. In such cases, statistical testing helps in providing useful information regarding the experimental results. The statistical methods of testing are quite important and useful in the evaluation of stemming
techniques because the test samples such as queries used for testing of stemming techniques are just a small sample of the complete query sets. Hull [50] recommended statistical methods such as t-tests, analysis of variance (ANOVA) that can be employed in case of continuous, large and normally dispersed samples.

3. Accuracy: Accuracy or stemmer strength is another method of direct evaluation of stemming techniques. In this method of direct evaluation, the output stems generated by the stemmer are compared with the required desired outputs. The stemmer strength or accuracy is thus estimated as the proportion of stems correctly generated. Frakes and Fox [51] suggested a number of metrics to measure the accuracy of the stemmer which are described as follows:

— **Conflation rate / Index Compression Factor (ICF):** Stemming maps a number of morphologically related word forms to the same stem thereby reducing the index size. ICF or conflation rate is thus a metric that measures decrease in the index size through stemming. It is computed as the number of distinct words present in the lexicon before stemming is to the number of unique words after stemming. Several experimental studies [46, 51, 52, 53] have employed this metric to evaluate the performance of stemmers and reported that higher value of this metric indicates a stronger stemmer.

— **Average number of words per conflation class:** It means an average number of words that are conflated together in the same group. In other words, it refers to an average number of words in the lexicon that have the same stem. A high value of this metric denotes better performance of the stemmer.

— **Average number of characters removed:** It refers to an average number of characters that are truncated from the input words. This metric does not take into account the variations made in the stems and is thus not too useful.

— **Difference between average number of stems and words:** Some input words are themselves stem and are not changed by the stemming algorithm. The accuracy of the stemmer can also be estimated by measuring the proportion of the input words modified by the stemmer.

### 1.6.2 Indirect Evaluation Metrics

The indirect methods of stemmer evaluation measure the performance by employing the stemmer as a pre-processor of a particular application such as information
retrieval, machine translation, text classification, etc. These methods use automated programs or tools to measure the stemming performance and do not need manually annotated data or manual labor as in direct evaluation methods. But the indirect evaluation methods need different resources such as standard text collections, queries, etc. In information retrieval application, following metrics indirectly evaluate the performance of stemmers:

— **Precision**: It refers to the proportion of relevant documents retrieved out of total documents retrieved. The high precision score means better stemming performance as the total number of relevant documents in the retrieved set is more than the irrelevant documents. Precision is computed by considering all the relevant documents in the retrieved set, but it can also be computed by considering the retrieved set up to a particular rank which is termed as precision at n (P@n).

— **Recall**: It refers to the proportion of relevant documents retrieved out of total relevant documents for the collection. High recall score indicates that result set contains more number of relevant documents.

Stemming in IR enhances both precision and recall. This is due to the reason that stemming reduces the number of variant words to the same stem thereby enhancing the term frequency. Thus, more number of relevant documents are retrieved at higher ranks in the result set.

— **F-Score**: It refers to the weighted mean of recall and precision scores. The F-score metric considers both recall and precision and is used in a number of retrieval applications such as question answering systems, text classification and clustering, query classification, etc.

— **R-Precision(R-P)**: It is defined as the precision at the Rth rank in an ordered list of retrieved documents for a collection which has R relevant documents.

— **Average Precision (AvP)**: It is defined as the average of precision scores at all recall points in an ordered ranked set of retrieved documents. In an ordered result set, precision is computed at each point and then the average precision is computed as the average of P(r) from r 1 to n.

— **Mean Average Precision (MAP)**: It is the mean of average precision of all the queries of the test collection i.e. it is the sum of average precision scores of all the queries in the test set is to the total number of queries.
1.7 Motivation

The research in the area of text stemming started in the late 1960s. There have been rapid advancements in the terms, ideas, and concepts related to text stemming which are being used interchangeably in the stemming literature. This has led to an increase in difficulty for the users to decide which technique is most relevant for a specific application. The current state-of-the-art stemming techniques are developed using language-specific rules or corpus-based techniques. As already described in the Chapter, the rule-based stemming techniques require various linguistic resources and are quite time-consuming. For resource-scarce languages, the rule-based stemming techniques are either not available or lack complete coverage. The unsupervised corpus-based stemmers are thus good substitutes as they obviate the requirement of any linguistic expertise or resources. Thus, major objective of our research work is to design and develop efficient unsupervised language-independent corpus-based stemming technique. But, the unsupervised text stemming has certain open issues that need attention. Some of the open issues and challenges in the area of text stemming are described below.

1. **Efficiency of unsupervised stemming techniques:** The efficiency of the unsupervised stemming in a particular application is a major concern that needs to be addressed. The existing corpus-based stemming approaches perform many errors such as grouping of unrelated morphologically variant word forms, treating valid word endings as suffixes, etc. All such errors degrade the performance of the unsupervised stemming. For instance, most of the corpus-based stemmers remove the learnt suffixes from the corpus on the basis of longest match principle whereas the well-known Lovins [3] or Porter stemmer [47] rule-based stemmers impose context-sensitive conditions or use stemming rules that decide when and how to stem the input word. Therefore, methods of unsupervised stemming must be explored that resolves all such errors and increase the efficiency of unsupervised stemming.

2. **Identification of a wide range of morphological variants:** One major challenge related to unsupervised text stemming is that most of the corpus-based techniques are developed for suffixing languages. Most of the existing stemming methods [10, 14, 15, 16, 54, 55, 56] strip off suffixes learnt from the corpus. But, there are number of morphological variations other than suffixation such as compounding of words,
conversion of words, duplication, etc. The rules that cause these morphological variations also occur frequently in large text collection of languages. Thus, methods need to be developed that identify these variants from the text collection and hence the stemming rules for the unsupervised stemming need to be refined. This refinement of stemming rules could correct many stemming errors or bad conflation and thus enhance the performance of unsupervised stemming.

3. Identifying advanced semantic relations from corpus: Another important challenge in unsupervised text stemming is to develop methods to learn advanced semantic relations from the text corpus so as to group semantically related variant word forms from the corpus. The existing unsupervised text stemming techniques [14, 55] group many unrelated word forms by treating valid word endings as suffixes. These stemming techniques remove the learnt affixes from the variant word forms in each case. But, it is important for the stemming techniques to decide that when to remove these affixes and when to not. For example, grouping of words “eat” and “eating” by removing ending “ing” is a correct stemming decision but grouping of words “shin” and “shining” in the same group by removing “ing” is an incorrect action. All these errors can be resolved if semantic knowledge along with the lexical features between the words is considered. Thus, the research in the area of text stemming must focus in incorporating semantic relations such as co-occurrence, context, distribution similarity, semantic spaces, etc. along with lexical relationships between the word pairs [6].

4. Size and nature of corpus used during training: Another important issue that is related to corpus-based stemmers is the size and type of document collection used during training of these stemmers. The corpus-based stemmers generally involve a lot of computations. The complex text mining methods like clustering, graph-based procedures, etc. employed to learn morphologically related words from the ambient corpus are computationally quite intensive. An appropriate size of the training data can reduce the amount of calculations and hence the time required for the development of the stemmer. The researchers should test the performance of the stemming techniques with different size of the training data so as to reduce the computational load. Brychcin and Konopik [6] tested the performance of stemming algorithm with different training data ranging from 50,000 to 15,000,000 tokens and
showed that good results in different applications can be achieved even if 50,000 tokens are used during training. Moreover, the type of documents used during training of stemmers can also affect the performance of the stemmer. A collection that contains text from different subjects such as religion, medicine, news, engineering, fiction, non-fiction, etc. can enrich the vocabulary of the training corpus. Hence, analysis of different size and type of documents during training phase is an important issue that needs to be addressed.

5. **Type of training**: The retrieval effectiveness of the stemming approaches is generally evaluated by using the same data collection for the training of stemmers that is to be indexed. But the data in real situations is increasing continuously as the new data emerge quite frequently. Thus, the stemmers require retraining in order to index the new document collections. The retraining of the stemmers generally takes a lot of time and is computationally intensive. In these situations, stemming techniques which do not require retraining and can manage the unseen data efficiently are preferred.

6. **Parameter tuning for unsupervised stemming methods**: Most of the corpus-based stemmers proposed in the literature are quite sensitive and dependent on parameters due to which the stemming process is not fully unsupervised. The tuning of parameters depends upon a number of factors such as the nature of corpus, type of the language involved, the type of similarity or association metric used, type of clustering mechanism employed, etc. Therefore, unsupervised methods of stemming must be explored where the thresholds are either static or unsupervised threshold selection criteria is employed.

1.8 **Organization of Thesis**

In Chapter 1, concepts related to text stemming have been introduced. Various methods of development of text stemmers have been discussed. The purpose of text stemming in IR and NLP applications is also presented.

In Chapter 2, an exhaustive and comprehensive review of literature for different text stemming techniques have been presented. Work done in the area of unsupervised text stemming has been highlighted and the various unsupervised stemming techniques have been compared in terms of their pros, cons and experimental results.
Chapter 3 presents two lexicon analysis based unsupervised stemming approaches that improve retrieval accuracy at a low computational cost. The working of the proposed approaches has also been illustrated through the formation of classes of morphologically related words in different languages.

Chapter 4 presents a lexicon and co-occurrence statistics based unsupervised stemming approach that improves the quality of clusters of morphologically related words using both lexical and co-occurrence features. The morphological class formation for different languages using the proposed algorithm has also been illustrated.

In Chapter 5, the proposed approaches have been evaluated and compared with the strong baseline stemmers in five different test scenarios namely information retrieval, text classification, part-of-speech tagging, inflection removal, and stemmer strength. The summarization and analysis of results obtained in each test scenario have also been covered.

In Chapter 6, the research work has been concluded and summarized. Also, some future directions in the area of unsupervised corpus-based text stemming have been highlighted.