Towards Understanding Bollywood Lyrics

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science
in
Computational Linguistics
by Research

by

G Drushti Apoorva
201225011
drushti.g@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2018

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Towards Understanding Bollywood Lyrics" by G Drushti Apoorva, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Radhika Mamidi

"Throughout the centuries there were men who took first steps down new roads armed with nothing but their own vision. Their goals differed, but they all had this in common: that the step was first, the road new, the vision unborrowed, and the response they received — hatred. The great creators — the thinkers, the artists, the scientists, the inventors — stood alone against the men of their time. Every great new thought was opposed. Every great new invention was denounced. The first motor was considered foolish. The airplane was considered impossible. The power loom was considered vicious. Anesthesia was considered sinful. But the men of unborrowed vision went ahead. They fought, they suffered and they paid. But they won."

- Ayn Rand, The Fountainhead

To you, the reader.

Acknowledgments

I would begin with expressing my gratitude towards my guide, Prof. Radhika Mamidi, who was not just a guide for my research work but also a friend and a mentor throughout my years at IIIT. Thank you ma'am, for being extremely patient with me, being the pillar of support through the thick and thin of times and motivating me whenever I needed it. The freedom you gave me to foray into different topics for research and conduct many failed experiments paved way to the successful results that we finally arrived at. Under your guidance, I, now, not only have a good command on linguistic concepts but have also developed an inquisitive research perspective. Your appreciative nods and smiles on the good days and incessant nudges on the days I slacked off have inculcated in me, perseverance and determination as primary qualities for any task I pick up.

I would like to thank the other professors from the lab for always being there whenever I needed them. Professor Dipti Misra, for emphasising on the importance of resources and data as a strong foundation for research, which would otherwise be neglected by many. Professor Soma Paul, for making me confident about asking questions and doubts, however silly they might seem, which is one of the core qualities needed in a student and for helping me hone my critical awareness about linguistic research problems. I would use this opportunity to also thank Professor Manish Shrivastav, because he never failed to spot me when I was low or troubled and provide a sounding board for my ideas and worries. I would extend my heartfelt gratitude to Late Professor Laxmi Bai as she was instrumental in igniting my interest in linguistics right in the first year of university.

Abundant love to Remya and Akshita for spending sleepless nights with me before conference deadlines and bringing me food. Thank you so much guys for dragging me to the lab every single day and helping me stay motivated. It wouldn't have been possible for me to get any of the results I did or design any of the experiments I used without your presence in the lab. You were the ideal lab-mates. I thank Urvashi for having the super power of doing all these things sitting in her room and also waking from her slumber whenever I needed her without any complaints. She has been by my side through each and every minute of college life.

Thank you, Rajat for inculcating in me the research temperament and teaching me the art of technical writing. I would be ever grateful for all the time you devoted for supporting me in spite of your busy schedule. Arjit and Abhijeet have been ever ready to lend an ear whenever

I whined about research or anything for that matter and helped me vent out the steam. Even through the last couple of semesters they have never once let me stop thinking about my research work and have just been a ping away whenever I wanted to talk about something.

Arwa, Shipra and Vanya have constantly inspired me with their hard work and have proof-read so many of my conference submissions uncomplainingly through every iteration. They have been instrumental in all the good grades I have managed to score in all the courses I took. I would also thank Murthy for forcing me to talk to seniors in CLD and putting me in touch with some of them to get me started with my research work and get over the initial inertia. I acknowledge Kannan, Gaurav, Kritik and Priyansh for brainstorming about the problems at length and trusting me enough to collaborate with me on various projects that helped me shape my final research work.

Finally I'd like to thank my elder brother, Abhishek for having my back always and for never letting me feel any sort of pressure towards any academic step I have taken. My family has been with me at every moment and I feel all the toil is totally worth it when I see their smiles at every achievement of mine.

Abstract

Research in Natural Language Processing is expanding in multiple domains and is seeping into all aspects of life with time. With every advancement, the variety of text that can be processed is growing. One such domain is lyrics processing. Songs are vital to the music and film industry and can be analysed to obtain important information such as genre, theme, mood, etc. of the song and supplement the information gathered by the study of its audio features. Bollywood, the Indian film industry makes a lot of revenue making use of songs. The number of songs churned out by this industry is massive and is a rich source of audio and textual data for Natural Language Processing tasks. It also gives us an opportunity to work on data in Hindi which is a relatively less explored field.

The focus of this thesis is on the textual part of the data. In an attempt to create a data resource for this domain, this work presents a corpus of Bollywood song lyrics and its metadata, annotated with sentiment polarity. We call this BolLy. It contains lyrics of 1055 songs ranging from those composed in the year 1970 to the most recent ones. This dataset is of utmost value as all the annotation is done manually by three annotators and this makes it a very rich dataset for training purposes. In this work, we describe the creation and annotation process, content, and the possible uses of the dataset.

As an experiment, we have built a basic classification system to identify the emotion polarity of the song based solely on the lyrics and this can be used as a baseline algorithm for the same. The lyrics of Hindi songs have been used for their classification as having positive or negative sentiment by extraction of opinions. Some experiments employing subjectivity lexicons and probabilistic approaches were conducted for this on a dataset which was just a subset of 'BolLy'. This motivated us to work with a bigger dataset and focus our efforts towards expanding it. With the complete 'BolLy' dataset, the experiments have been created using different variations of the Naive Bayes Classifier.

There can be a multitude of Natural Language Processing applications on the presented dataset. This thesis work contains one of them explored in detail. Keywords of a document are a representative of its content, and it helps to have meaningful words to facilitate search and organization of documents. Hence, finding methods that can automatically identify keywords in a document is very important as manual processes for this is very cumbersome and error-prone.

If this task is accomplished for song lyrics, it has varied applications such as recommendation systems and digital music library management.

This work proposes and compares methods to identify keywords from lyrics of Bollywood songs. We use a collection of lyrics of 1055 Bollywood songs, all written in the Devanagari script. Experiments include looking at the spatial distribution of the terms, their occurrence in a certain context or position, and using WordNet to generate keywords not present in the document. Validation was done by human annotators by providing a score to each method based on the results obtained on a subset of the data. We also used Latent Dirichlet Allocation and Latent Semantic Indexing to validate the results, as further explained in the paper.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

India is culturally a very rich and diverse country. One major aspect of this is the vast number of languages spoken here which is shown according to the geographical regions in Figure 1.1. This provides a very interesting playground for linguistic researchers. People in India are very interested in the entertainment sector and the film industry here is very successful. There are regional film industries for different languages but the largest is Bollywood for Hindi [79]. With the advent of technology and internet making availability of data easy in various domains, it is natural that researchers turned to movie scripts, song lyrics, screenplays, etc. as a source of data for the Indian languages. This thesis work is a result of such inquisitiveness and looks at the Bollywood song lyrics and their analysis.

1.1 Motivation

Hindi is the most widely spoken language in India, almost 41% of the Indian population being its native speakers. It is also one of the Indian languages with a good repertoire of tools and resources for Natural Language Processing research. Hence it was interesting for us to foray into this field and work on Hindi data available in Devanagari script and combine it with research topics such as sentiment analysis and keyword finding which are picking up now. As no corpus existed for Bollywood lyrics [79] [4], a corpus with 300 song lyrics was created [79] and one with 6529 song lyrics was created by [4], both of these are in Devanagari script without annotations. The latter compares the lyrical data to regular Hindi text which is very insightful.

Music from Bollywood constitutes almost 71% of music sales in India. This is a huge market and it would be very critical to work on better organisation of this data. People choose to listen to different music based on their mood and situations. It is therefore, of aid to users to be able to organise digital music libraries according to their sentiment or use this information in automatic playlist generation tasks as explored in [23]. Keyword generation from lyrics would help identify the sentiment of the song and also in other tasks such as topic detection [36]. It would play an important role in recommendation systems as well which would help us recommend music for

videos, etc. This is an easy task for humans to do, but the real challenge lies in the automation of it which would help tackle the issues of scalability and remove human bias. Some efforts have been made in this direction providing an application to browse music by mood [44], but for English songs.

1.2 Definition of Terms

We give a brief description of different terms used in this thesis that are particular to the domain.

- Rhythm - The flow of pattern in words on the basis of syllables occurring in the text.
- Metre - The rhythmic pattern in poetry or music.
- Sentiment Polarity - Polarity is the classification of something as positive or negative, as two opposites. When this is done on the basis of the sentiment or emotion expressed or invoked by something, it is called sentiment polarity.
- Keyword - A word or phrase that is of significance as it expresses the concept or ideas reflected in a piece of text.
- Bollywood - The film industry in India, that is predominantly Hindi.
- Annotated Dataset - A collection of data that is cleaned and pre-processed and contains valuable information or labelling of some sort.
- Devanagari - The script used to write Hindi.
- Remix - A new version of a musical piece that has something different from the original version.
- SentiWordNet - A lexical resource that is built upon the WordNet, and has sentiment scores assigned to synsets.
- Code Mixing - When two different languages are mixed in terms of text or speech.

1.3 Challenges

One of the biggest challenge in any research approach is to find the right data, both in terms of quality and quantity. A part of this thesis aims at solving this hurdle for the given domain and language. The creation of this data resource involved a number of tricky tasks that were to be solved by us. These involved availability of Hindi data predominantly transliterated in the Roman script. This was not suitable as the transliteration does not follow conventions and also

there are some English words with mean something else in Hindi words that are transliterated such as 'man' (mind in Hindi) or 'hum' (we in Hindi). Also, the inconsistency results in a lot of errors when available transliteration tools are used. We did not want to focus our efforts into the transliteration task and hence decided to go with the data available in Devanagari.

Another challenge that we faced was to identify the uniqueness of lyrics data and that too in Hindi and build our system in a way that these were taken into account for. Lyrics data is dissimilar from other regular text as it does not follow a strict grammar. To incorporate rhythm and metre, grammar is sometimes given a miss. Also, the amount of textual content available in lyrics is lesser than a novel or an article or a research paper, etc. This makes our second part of the thesis, sentiment polarity extraction and finding keywords, compelling research problems to solve for lyrical data. It involved experimentation with different means to extract as many cues as possible to form our hypotheses and shape our inferences.

1.4 Contributions of this Thesis

Any research work is of little significance if it does not have novel contributions to the research community. The contributions of this thesis work are:

- Resources and datasets are a major pillar for any sphere of research. Annotated datasets help validate research work and publicationas and also are vital to gaining insights to form hypotheses and understanding indications towards how to solve a given problem. A major contribution of this work is an annotated dataset of Bollywood song lyrics in Devanagri script.
- In research, it is important to have a reference to measure the value of the proposed methods or systems. This is provided by baseline results. To prove the usefulness of the dataset contributed by us, we also present one application of the dataset using its annotation with the baseline results for it - sentiment polarity detection.
- An in-depth exploration of different systems to finding keywords in song lyrics with novel insights and techniques for solving this problem with significant accuracy for Devanagari script. We also propose an approach to look at words and phrases that were not a part of the original text but are potential keywords.

1.5 Thesis Organisation

This chapter gives an introduction to this thesis work and the rest of the thesis deals with the elaboration of the work presented. The different parts of this thesis work are organised in the following chapters.

- Chapter 2 - Background and Related Work is a detailed description of the problems tackled and how they were conceptualised. This chapter also describes the different publications or the research work done in the specific domains that has been crucial to establish the hypotheses we have worked with or to shape the solutions and experiments we delved into. This chapter makes clear how this work fits into the larger picture of Natural Language Processing research in the field of lyric analysis.

- Chapter 3 - 'BolLy' : Annotated Lyrics Dataset is the very foundation of this thesis work. It elucidates the dataset that has been created by us and also the process of its collection, cleaning and annotation. This chapter establishes the first step towards the subsequent problems explored by putting forth a dataset that can be used for the purposes of lyrics analysis for Hindi in devanagari script.

- Chapter 4 - Sentiment Polarity Detection in Bollywood Lyrics sheds light on an application that demonstrates the usefulness of the dataset created by us, making use of it for a contemporary problem that is being worked on by researchers. This problem is that of analysing the sentiment of text. In this case, it is lyrical text, which is a less explored topic, and we are able to build systems with different approaches for the same and compare them providing a baseline for further investigation in this field..

- Chapter 5 - Finding Keywords in Bollywood Lyrics explores this problem that is very useful for current scenario with the expanding data available to us through the internet. It is a unique research problem to be tackled as finding keywords in lyrics can be very tricky compared to regular text. In this chapter, different methods have been explored for solving this and on the basis of comparison of their accuracies on the dataset presented earlier, the most apt one is considered as the best system for the same.

- Chapter 6 - Conclusions and Future Work is a brief summary of the work presented in this thesis and the plausible future work that can be done in this field. It consists of the conclusions and inferences drawn on the basis of the experiments conducted by us and also the detailed illustration of the future ideas and problems we intend to work on to further this work or to explore tweaks and modifications for making the systems explained more robust and accurate.

1.6 Appendix

Appendix A contains a sample of a song that is annotated as a positive one while Appendix B has the sample of a negatively annotated song. These songs are originally in Devanagari script, so we provide a gloss and a transliterated version as well.

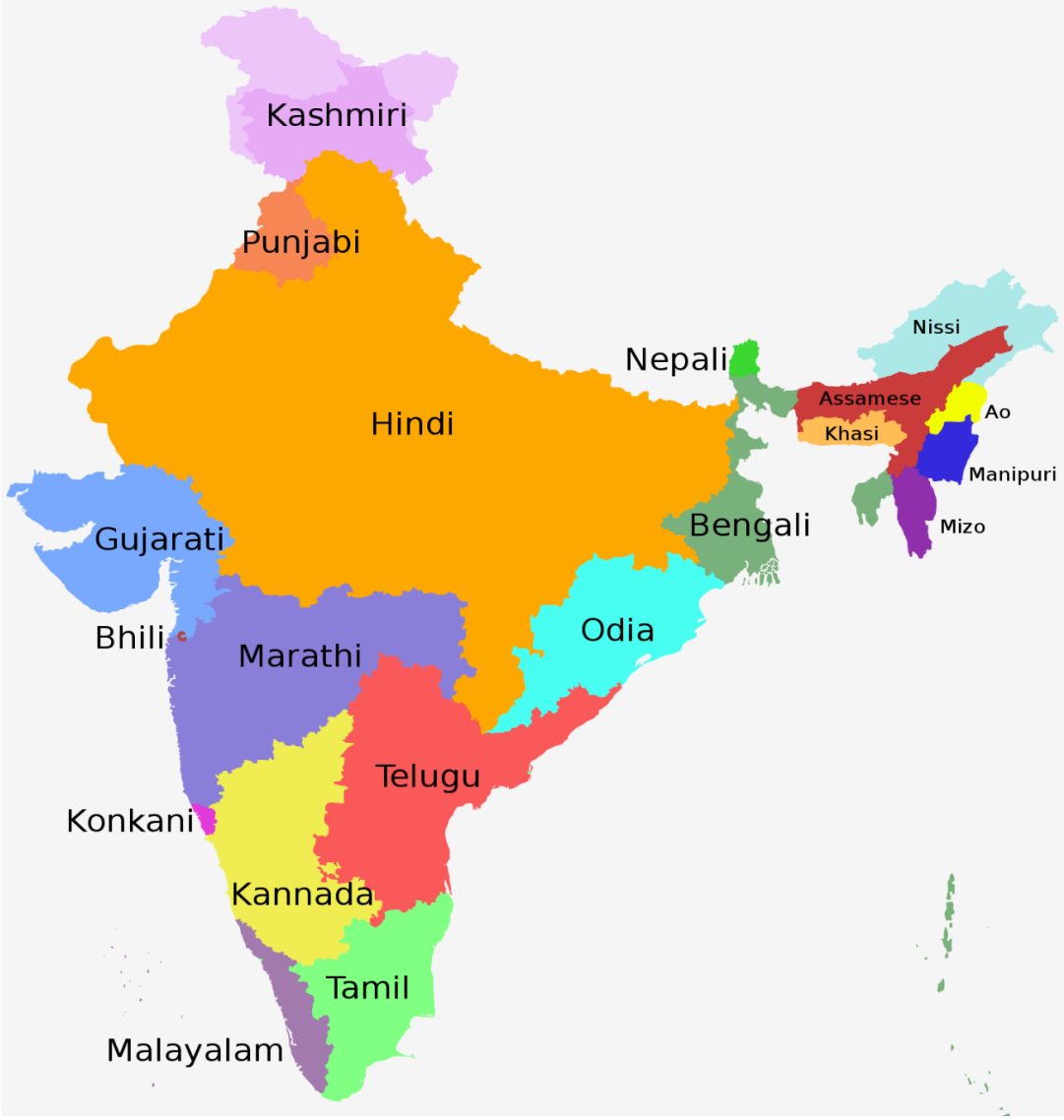

Figure 1.1 Languages spoken in India.

Chapter 2

Background and Related Work

In this chapter, the background study and the literature survey is presented divided into sections on the basis of different areas.

2.1 Sentiment Analysis and Opinion Mining

Sentiment Analysis [9] [50] and Opinion Mining [9] [50] have become much sought after research topics from academic as well as commercial perspectives in Natural Language Processing recently. Notable research has been done in extraction of sentiment and opinion from text such as movie reviews [63], customer reviews [25], tweets [62], product reviews [60], etc. Such tasks require approaches specific to the kind of text under consideration. We will investigate the existing work in identifying sentiment and opinion in music in the next section.

2.2 Lexical Resources

The WordNet [57] is a well-known lexical resource commonly used in Natural Language Processing research. A lot of resources or tools have been built on top of the WordNet framework. WordNet has also been developed for the Hindi language [31]. The subjective lexicon for Hindi [2] depends on the Hindi WordNet [31] and breadth first traversal of graph. It is created on the assumption that the polarity is mainly dependent on the adjectives and the adverbs [5] and that synonyms have the same polarity while antonyms have opposite polarity. The approach for creating a subjective lexicon using bilingual dictionaries [32] does not consider all the words given rise to by the language variations.

SentiWordNet [17] is useful for opinion mining as it assigns scores of positivity, negativity and objectivity to synsets from the WordNet [57]. For Hindi, a SentiWordNet has been created [13] based on the Hindi WordNet [31] using a subjectivity word list made from manually compiled data along with training data that is labelled. WordNet Affect [80] and SenticNet [10] were

merged [69] for obtaining the emotion category along with the polarity score. Sentimantics [15] is a useful resource too for tapping in the contextual information.

2.3 Music Mood Classification

There has been an increase in the interest in music mood classification [27] by researchers working in the domain of music information retrieval [18] [52] [53]. Work done in music classification involves those making use of lyrics [66], audio [52] [22] [64] or a multimodal approach [67] [91] [43] [1]. In this section, we start with looking at the existing literature in music mood classification in general and later narrow down our focus to the same for Hindi language. The first audio mood classification (AMC) task in Music Information Retrieval Evaluation eXchange (MIREX) used a dataset of 600 tracks with five mood clusters [27]. The multilabel classification of audio songs into 6 classes of emotions from 593 songs was done using 72 features [84]. A hybrid approach using theme and mood clustering was proposed based on audio features with 30 second excerpts from a dataste of songs from AllMusic.com and Last.fm [6].

Audio features do not always outperform lyric features for the case of mood classification [26]. Lyric features such as bag of words, function words and part of speech tags were used in a dataset of 5585 English songs to classify them into eighteen mood categories [26]. This was done with the Support Vector Machine classifier [83] using the LibSVM library [11]. An unsupervised model was proposed for classifying Chinese songs based on their lyrics [28] using sentence-level emotion units [90] as an important feature. Fuzzy clustering and ANEW [8] translated into Chinese were used for this system.

Indian songs have received some attention from researchers in the field of music classification. Indian music can be broadly divided as classical music and popular music [87].Furthermore, classical music can be categorised into Hindustani and Carnatic music [66], prevalent in northern and southern India, respectively [66]. Mood classification was forayed into by researchers for Hindustani music using thirteen mood classes [88]. The Carnatic music raagas were classified according to the rasas, or emotions they evoked into ten classes [37].

A mood classification system was built for Hindi and English song and lyric dataset for a comparative analysis [67]. A salient observation from this work, that the mood evoked by lyrics and audio is very different in case of Hindi songs, motivated us to work on Hindi lyric data. A system for based on decision tree classifier [70] built using rhythm, intensity and timbre features from the first 30 second clips of Hindi songs achieved 51.56% accuracy [64]. This system was built on the assumption that emotional associations gave rise to music categories [35]. Another system was built to categorise 250 Hindi songs based on their audio features using fuzzy clustering algorithm into five classes [65]. The most important work in Hindi music from our perspective used sentiment lexicon, stylistic features and n-grams to classify Hindi songs

based on their lyrics [66]. All these works were based on popular Indian music or Bollywood songs [67] [64] [65] [66].

It is worth paying attention to the work that has been carried out in regional Indian languages towards the sentiment analysis of songs [1]. It used Doc2Vec [45] to extract features from the lyrics. The algorithms used for training the lyrics module were Naive Bayes [21] and Support Vector Machine [83] while those for audio features such as timbre, rhytm, tone were Support Vector Machine [83] and Gaussian Mixture Model [72]. It was seen that the audio and lyric features were complementary to each other and a multimodal system performed way better than using just one of them.

2.4 Keyword Generation

Keywords are words that may or may not occur in a document but express the concepts presented in the document. They are condensed form of the summary of a document. It is useful to have keywords for any document as it helps decide the relevance of a document without having to go through it completely. With the growing accessibility to data, it is a very significant and useful feature. Keyword generation is a relatively less explored topic in research in the field of Natural Language Processing. It can be approached as a supervised or an unsupervised learning task, conducted on audio or textual dataset.

Advances in extraction of keywords from audio data have been impressive making use of term frequency - inverse document frequency algorithm with parts of speech, word clustering and salience score of sentences [51]. A graph based approach for this guaged a word's importance according to its relation with other sentences or words. This work was done using the ICSI meeting data [30]. As our work deals with text processing, we will look at some previously conducted research on textual datasets.

We have come a long way since the problem of keyword generation was first presented as a supervised machine learning task using a genetic algorithm [85]. It is explored as a supervised learning task for English text that incorporates semantic context by the means of lexical chains [16], a good representation of lexical cohesion [59], and text features such as the position of occurrence and frequency of the word. The resources that support this work are the ontology from WordNet [57] and the decision tree induction algorithm [70].

Another published work uses a supervised training method but it also takes into consideration syntactic features such as Noun Phrase chunks, n-grams and Part of Speech tags [29]. The dataset that the experiments are carried out is an English text of abstracts of academic papers from the Inspec dataset. A similar algorithm is discussed where they employ the frequency of head noun and noun phrase and the number of words to deduce the keywords [3].

The method proposed which has been instrumental for documents that don't adhere to conventional grammar is Rapid Automatic Keyword Extraction (RAKE) [74]. It focusses on

keyphrases without any punctuation or stop words. The parameteres used for this system include stoplist and phrase and word delimiters and it does the task on a document level with no dependency on other documents in the corpus. This is one of the methods employed by us to get an insight about keyphrases for Hindi text.

Another work on keyword extraction carried out on a document level is presented that shows the importance of keyword extraction for both supervised and unsupervised extractive text summarisation [49]. The keyword extraction task is accomplished using a graph based model as it incorporates the syntactic features much better than the vector space models that are used traditionally. It is common to model them with words or phrases constituting the nodes and the edges joining them representing the syntactic relationship [77]. Some other instances of documents represented as in a graph-based manner can be seen as edges representing the co-occurrence relations connecting words [56] [55] or as semantic relations as edges that connect concepts as nodes [46]. Such methods do not require specific linguistic processing based on the language of the data.

Some research work has been carried out on Chinese text using a sequence labelling method called Conditional Random Fields using an undirected graph model [92]. This makes use of contextual features, both locally and globally.

After discussing the different keyword extraction tasks done for various kinds of datasets, we discuss one important work done in the field of lyrics for the same as that is what our work is about. Natural Language Processing for lyrics is very different from regular text [78]. It was tackled using the DigiTrad folksong dataset in English using sentence similarity and relation links [89] that were obtained from WordNet. We explore keyword extraction from lyrics for text in Hindi written in Devanagari script in our work discussed in Chapter 5.

Chapter 2 presented the literature survey to establish the context of the work presented in this thesis. Chapter 3 will describe a data resource which is one of the major contributions of this thesis.

Chapter 3

BolLy: Annotated Lyrics Dataset

Bollywood, the Hindi film industry churns out numerous songs and movies every year. On an average, around 10 Bollywood movies are released monthly and each of them have around 7-8 songs. This amounts up to a big approximate sum of 850 songs that come into the market on an annual basis which makes Bollywood one of the most popular forms of music in India [12]. This is a rich source of textual data if we look at the scripts of the movies and song lyrics. They can be used for creation of data resources that are valuable for a variety of applications. These applications can be based on the audio or the textual aspect of the songs. Some research has also explored multimodal approaches.

Lyrics, in particular aid in tasks related to music information retrieval, genre classification, sentiment analysis [9] [50], opinion mining [9] [50], code mixing [61], etc. This motivated the creation of an annotated dataset for Bollywood lyrics. A dataset is helpful in stating a research question, testing the hypotheses and evaluating the outcome. It is always important to clean and pre-process the data before starting with any experiments. Also, some experiments on the data make available the insights on its suitability. That is how the work presented here, on Bollywood lyrics, took its course. This chapter comprises of the collection, cleaning and annotation of the lyrics which gave rise to the annotated Bollywood lyrics dataset: 'BolLy'.

The market of Bollywood music is massive with a customer base of 14 million people and with the advancement in technology, digitisation of music is inevitable. Bollywood music and lyrics are being made available online with a greater speed and outreach with every passing year. Bollywood lyrics can be easily extracted from many websites. They are made available in different formats, i.e., transliterated in Roman script or in the original form, in the Devanagari script. For simplicity of processing, we extracted them from a source [1] where Hindi words were available in Devanagari script, i.e. consisting of utf-8 characters. The lyrics of the songs were extracted with their metadata including the movie or album they belong to, the singers and the year of release.

[1] www.hinditracks.in

3.1 First Attempt at Dataset Creation

Our efforts started with the collection of a smaller dataset to understand the nature of the data by executing some experiments related to sentiment polarity [15] extraction, which has been explained in the next chapter. This led us to realise that, for any computational linguistic application, only a substantially big enough dataset would be useful for obtaining meaningful results for the purpose of research.

Our very first endeavour was a foray into research concerning Bollywood lyrics, which resulted in manually selected Bollywood lyrics comprising of only Hindi lyrics without any code mixing. The songs were selected from the period between 1980s to early 2000s. The manual selection gave rise to biasing as the selector was also one of the annotators. The dataset created thus, comprised of 200 unique Bollywood song lyrics obtained from an online source[2]. Out of these 200 Bollywood songs, 100 were annotated as having positive polarity and 100 as having negative polarity (hereafter referred to as positive and negative songs respectively) by the annotators.

Following are the features of this dataset:

- procured 200 song lyrics.
- 100 songs annotated as positive in the dataset.
- 100 songs annotated as negative in the dataset.
- average number of tokens in a song are rounded off to 207.
- positive songs have a total of 3524 unique tokens.
- negative ones have 2953 unique tokens.
- 1126 tokens are common to both the classes.

The data has been annotated by three annotators following the same guidelines and principles as explained in Section 3.3. Each song has been annotated with the tag given by two or more, which constitutes a majority, of the annotators. The annotators were given guidelines for annotation based on the taxonomy proposed by Russell's Circumplex Model of 28 affect words [75] as shown in Figure 3.1 which is based on eight basic affects : arousal, excitement, contentment, pleasure, sleepiness, depression, misery and distress [75]. Based on this model, all the songs evoking emotions with positive valence are to be categorised as positive and those evoking emotions with negative valence are to be categorised as negative. The Fleiss' kappa [19] achieved for the annotations (as shown in Table 3.2) is 0.81 which corresponds to an Almost Perfect Agreement [40] according to the interpretation shown in Table 3.1. Fleiss' kappa is a

[2]www.hinditracks.in

Table 3.1 Interpretation of Fleiss' kappa values for inter-annotator agreement [40].

κ	Interpretation
< 0	Poor agreement
0.01 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 1.00	Almost perfect agreement

score that measures the agreement between annotators [19]. In the categorisation of data points into given categories by a fixed number of annotators, Fleiss' kappa measures the reliability of the rating agreement statistically [19].

Annotator	Positive Tags	Negative Tags
1	101	99
2	102	98
3	100	100

Table 3.2 Number of songs annotated as positive and negative by the three annotators for the initial dataset of 200 song lyrics.

A few basic experiments were carried out on this dataset for identifying sentiment polarity. These experiments gave us an insight as to how this size of the dataset was not enough for carrying out more detailed experiments. This is discussed in detail in Chapter 4.

3.2 Creation of the Final Dataset

The work described in the previous section paved way to the creation of a bigger new dataset that we named 'BolLy'. This has been created taking into account the disparity in the factors such as length, the number of words, etc. in positive and negative songs. Also, a bigger dataset includes a better distribution of songs of different genres, moods, singers, composers, etc. After the collection of data, it was cleaned to further reduce pre-processing required. Repetitions of lines or words in the lyrics were represented using numbers, for example, a line was followed by 'X2' if it was to be repeated twice. In certain cases, these numbers were in Devanagari. All such representations were removed and the line or word in question were copied as many times mentioned.

In Bollywood, a lot of songs are remixed into different versions. Sometimes the same songs are sung by different singers. There are quite a few instances wherein the same song occurs in different moods, such as both happy and sad, in a movie. All such songs appeared multiple

times in the dataset. These songs differ in terms of audio features and evoke varied emotions when listened to. As there was no difference in their lyrics and our work focuses solely on the emotions evoked by song lyrics, the multiple occurrences of such songs were removed from the dataset, and only one file was retained.

Following are the features of the dataset:

- originally procured 1,082 song lyrics.
- 1,055 song lyrics in the final dataset, after removal of duplicates.
- song lyrics/files with metadata amounting to 2.6 MB data.
- 712 songs annotated as positive in the final dataset.
- 343 songs annotated as negative in the final dataset.
- total number of tokens in the dataset are 2,17,285.
- average number of tokens in a song are rounded off to 211.
- total number of tokens in positive songs are 1,51,362.
- average number of tokens in a positive song can be rounded off to 218.
- total number of tokens in negative songs are 65,923.
- average number of tokens in a negative song can be rounded off to 196.

3.3 Annotation

3.3.1 Taxonomy for Mood Classification

There have been a few well-known mood classification taxonomies. In this subsection, we will briefly mention some of them and describe the method of arriving at our annotations from them. Musical parameters such as mode, tempo, pitch, rhythm, harmony and melody were used to devise categories based on the correlation of these parameters with the evoking of a mood [24]. Also, a Circumplex with moods placed on two axes of arousal and valence as shown in Figure 3.1 was put forth by another work [75]. This was used by a lot of researchers for music mood classification [64] [34] [91]. Another taxonomy was created based on stress and energy as the two parameters to define 4 clusters of moods [82]. These were some of our guiding taxonomies that we used for some experiments.

The 8 mood classes in Hevner's model [24] was very intuitive but only regarding the music of a song. For this, the audio played a major role and the classification wasn't very useful in our work as we only took into account the lyrics. Some other works have also made use of Russells'

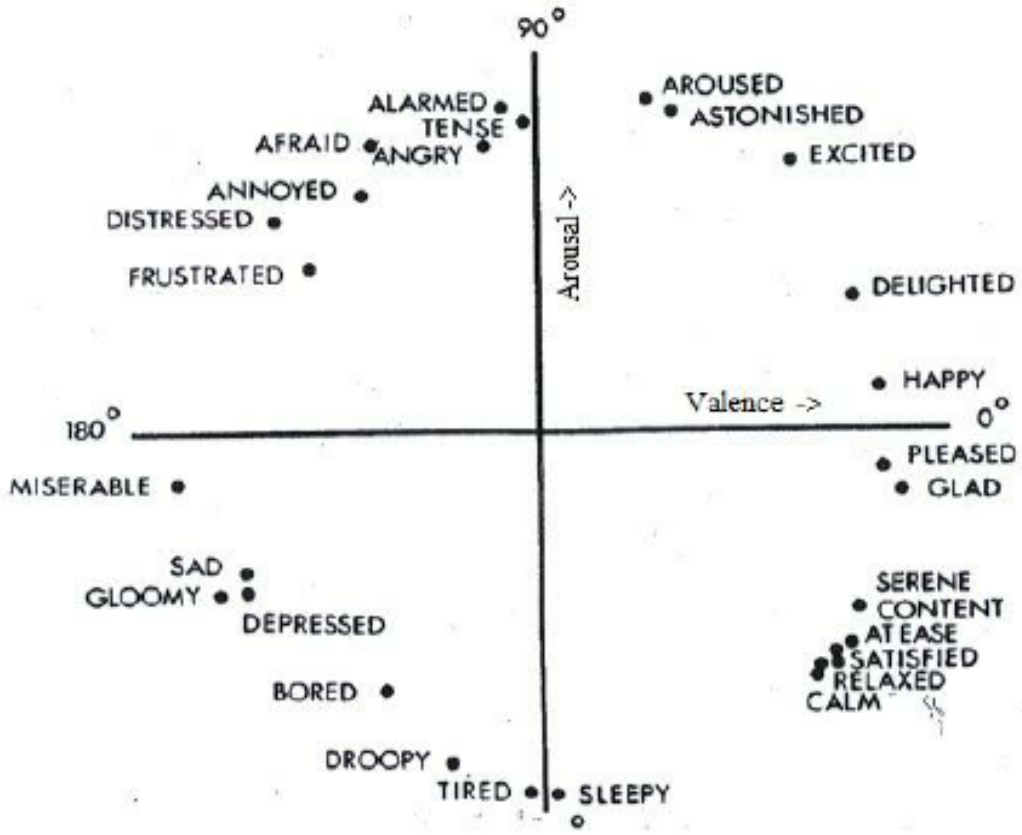

Figure 3.1 Russell's Circumplex Model [75] classifying 28 Affect words on the basis of positive and negative valence and arousal.

Circumplex [75] to arrive at a music mood classification taxonomy. It was used to propose a classification that grouped some moods from the Circumplex [75] in close proximity into five different classes, with e central mood of calm, happy, angry, sad and excited [64].

Using this mood taxonomy [64] led to some confusion. Some experiments in annotations showed that the annotators were confused between classes with similar valence or arousal, such as angry-excited, calm-sad, etc. This was an issue that we faced in annotations and decided to work on getting a clearer taxonomy. The next step was to combine moods towards the ends of the axes but even that led to similar confusions because it had conflicting emotions in the same group. So the idea of having finer categories was dropped and we decided to just have two classes, positive and negative, which is explained further in the next section.

3.3.2 Principles of Annotation

Three levels of granularity have been described for existing methods of sentiment analysis [50]. On the basis of the level defined, the task is to identify if positive or negative sentiment

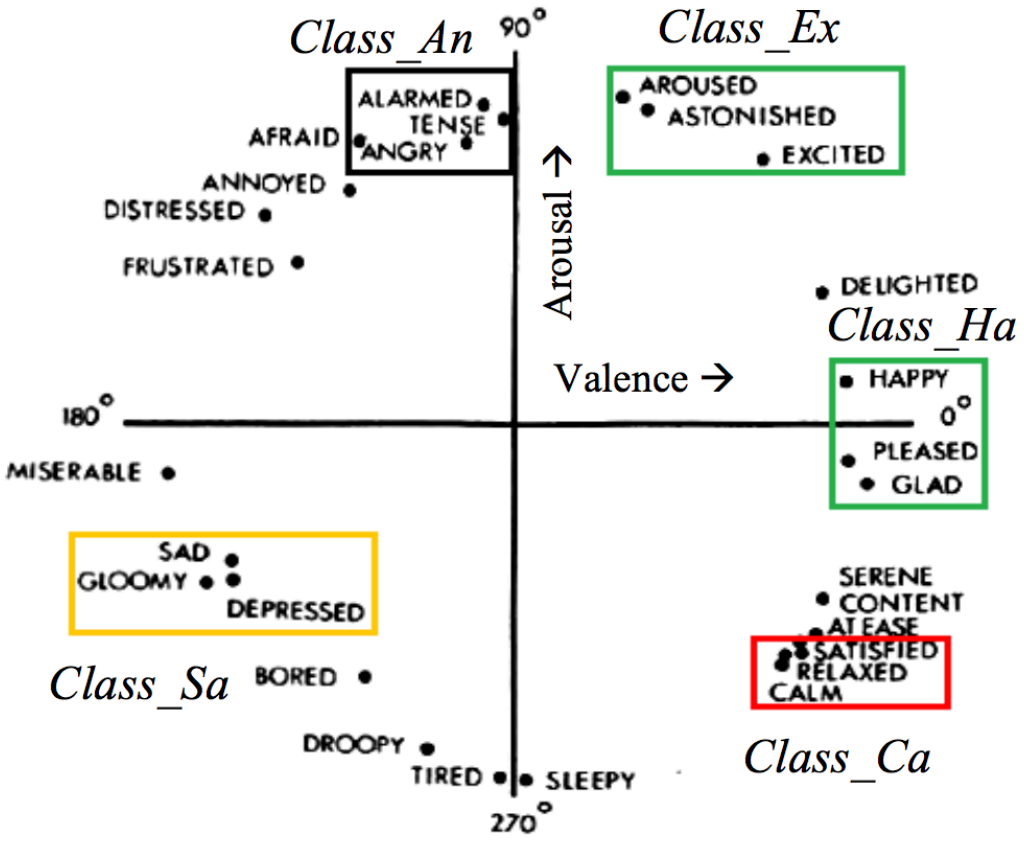

Figure 3.2 Mood classification taxonomy proposed by Patra et al using Russell's Circumplex Model

is expressed at that level. They can be carried out at the level of the whole document [86], or at sentence level or at the level of entities and aspects [25], [81]. It is possible that different smaller parts of a song's lyrics evoke different emotions but in the current task, we aim to identify whether the whole song's lyrics evoke a positive or a negative emotion. Hence, it is best for us to look at document level classification. The annotators were asked to go through the whole song before tagging them. This gives us the tag corresponding to the polarity of the general mood evoked by it.

3.3.3 Annotation Process

Each song in the dataset was annotated as positive or negative by three annotators, all of whom were university students in the age group 20-24 and were native speakers of Hindi. Songs evoke a certain emotion or mood, and these can be classified as those with positive or negative valence according to Russell's Circumplex Model [75], as shown in Figure 3.1. The songs that evoke emotions ranging from 'aroused' to 'sleepy' including 'calm', 'satisfied', 'delighted',

Annotator	Positive Tags	Negative Tags
1	721	334
2	728	327
3	710	345

Table 3.3 Number of positive and negative tags given by each annotator in the 'BolLy' dataset.

'excited', etc. are to be tagged as positive. Negative tags are to be given to songs evoking moods such as 'angry', 'annoyed', 'miserable', 'depressed', etc., all of them spanning from 'alarmed' to 'tired'. Each song is annotated with the tag given by majority of the annotators for it.

The annotations were carried out in a controlled environment in which the annotators were not allowed to listen to the audio of the song presented. Hence, the annotation is solely on the basis of lyrics. Also, the songs were presented to the annotators without any of the metadata associated with it to prevent any preconditioning. The number of positive and negative tags given by each annotator can be seen in Table 3.3.

From the original dataset of 1,082 songs, 27 were removed as they were duplicates. The annotation for these 27 songs were used to check for consistency of the individual annotators. While annotators 1 and 2 had annotated the duplicate instances of only 1 song out of 27 differently, the third annotator's tagging was inconsistent for 2 songs. These small numbers can be ignored in such a large dataset as they show that the annotators were rarely inconsistent in their task. By the end of annotation of the final dataset of 1055 songs, 712 are tagged as positive while the rest 343 are annotated as negative.

3.3.4 Inter-Annotator Agreement

Inter-annotator agreement is a measure of how well the annotators can make the same annotation decision for the same category. Given the task in hand, it is fair to assume that annotation of the songs based on the emotions evoked by reading the lyrics is a very subjective opinion. Thus, inter-annotator agreement becomes an important factor in validating the annotators' work.

There are many statistical measures that can be used for measuring the reliability of agreement between the annotators given the number of data points, number of annotators and the number of tags they can be given. Cohen's kappa, Fleiss' kappa and Scott's pi are some of them. Out of these, Cohen's kappa and Scott's pi work only for cases where there are two annotators. Fleiss' kappa, a generalisation of Scott's pi statistic, is used to measure the agreement among a consistent number of raters and this measure is unweighted [19]. Thus is most suited for our work.

κ	Interpretation
< 0	Poor agreement
0.01 - 0.20	Slight agreement
0.21 - 0.40	Fair agreement
0.41 - 0.60	Moderate agreement
0.61 - 0.80	Substantial agreement
0.81 - 1.00	Almost perfect agreement

Table 3.4 Interpretation of Fleiss' kappa values for inter-annotator agreement [40].

The Fleiss' kappa obtained for the annotations for our dataset is 0.79. This corresponds to 'substantial agreement' [40] according to the interpretation of Fleiss' kappa shown in Table 3.4 which is a positive result for the dataset we created.

3.3.5 Conclusion

In this chapter, we walked you through the process of creation of 'BolLy' dataset. All the challenges faced at every step right from collection of the data through cleaning and finally its annotation on the basis of sentiment polarity have been touched upon. The step by step refining of the annotation procedure is also delved into supported by the ideas and observations that helped shape the final annotation scheme. The final dataset created includes 1055 songs tagged by three annotators. This dataset proves to be a valuable resource for research in the field of Natural Language Applications for Bollywood lyrics. The agreement calculated is considered to be reliable and substantial ensuring that the annotation is consistent.

The dataset available for Bollywood song lyrics [66] has a total of 461 song lyrics classified as positive or negative based on certain moods in each class. Another work presented uses another dataset for Bollywood song lyrics that contains lyrics and the metadata of 300 songs composed between the years 1995 and 2015 with details about the usage of foreign words in different decades [79]. Thus, to the best of our knowledge, the annotated dataset presented by us, BolLy, is the largest of its kind amongst other resources for the language and the only one that contains instances of code-mixing.

The work described in this chapter is a start in the direction of creating resources for languages with less resources in a domain that is less explored. This work has multiple directions that it can be taken in, in future. We will touch upon some of them now. We intend to increase the size of this corpus so that it is more fruitful for learning purposes. Also, similar corpus can be created for other resource poor Indian languages because the Indian film industry also produces numerous regional movies with songs in regional languages. This can be tapped in for further efforts towards resource creation.

The sentiment polarity annotation process can also be made more granular with more classes and specific sentiment identification. The metadata can be put to use for labelling tasks. Other types of annotation can be explored for tasks like genre identification, author profiling, topic identification, code-mixing, etc. Collaborations with speech processing can give rise to multimodal approaches for all the mentioned tasks. Speech processing can also be used to establish correlations between the music and the lyrics of a song. Trends can be compared to the work already done in western music in similar domains.

In the further chapters, we shall dive deep into the different applications that this dataset could be put to use for and some experiments we conducted on it.

Chapter 4

Sentiment Polarity Detection in Bollywood Lyrics

Songs have two major components, the lyrics and the music. The component of the song that mainly expresses the emotion and the meaning is the lyrics. We chose to experiment with the task of identifying the sentiment polarity of the song using the dataset described in the previous chapter. This aligned with our research interest as our work was about analysis of song lyrics. In this chapter we start with looking into some preliminary experiments that we conducted. These were carried out during the collection of the dataset, with a smaller sample, so that we gained some validation about the usefulness of the dataset. Further, we will describe detailed experiments conducted with some concrete results.

People listen to different songs based on their moods. Hence, sentiment polarity identification is very useful. Research has progressed in this direction concerning the music and the audio of songs. The audio representation of songs can be utilised to decode a lot of finer qualities such as tone, timbre, etc. to gain some insight about the sentiments they express. Although lyrics play a major role in this, research in lyrics has been limited. Since we created a lyric dataset for Bollywood songs, we thought this would be an interesting opportunity to tackle this research problem.

4.1 Preliminary Experiments

This section talks about the experiments that were conducted on the preliminary dataset of smaller size, i.e., 200 song lyrics. These experiments make use of corpus specific and generic subjectivity lexicons [2] and also word frequencies. We classify songs as positive, those evoking positive sentiment and negative, those evoking negative sentiment.

4.1.1 Methodology

We have looked at approaches making use of word counts and subjectivity lexicons [2] and then proceeded to incorporate these into probabilistic models. The first experiment makes use

of the tagged dataset to create two word lists L_1 and L_2. L_1 contains all the unique tokens from the positive songs while L_2 contains the unique tokens from negative songs. The tokens common to L_1 and L_2 are appended to the list C, along with their frequencies of occurrence in both the lists, $freq_1$ and $freq_2$. Tokens from C are removed from L_1 and L_2 to obtain l_1 and l_2 respectively. The songs with a greater number of tokens belonging to l_1 are classified as positive while those with more tokens belonging to l_2 are classified as negative.

A similar method has been carried out using the Hindi SentiWordNet [13] [14] [15] as a resource. In place of using l_1 and l_2, this method checks if the tokens in the given song are tagged as expressing positive or negative sentiment in the subjectivity lexicon built using the Hindi SentiWordNet [13] [14] [15] and assigns a tag to the song in a similar way. These two experiments were conducted to see if a corpus specific subjectivity lexicon performs better than a generic one.

The 'Prevalent In One' method uses a threshold, t and the list C mentioned earlier in this section. For each word in a song, if its occurrences in the positive list divided by its occurrences in the negative list exceeds t, it contributes to the positivity of the song, else to its negativity. Hence, if $freq1/freq2 > t$, the score of the song is incremented and it is decremented when $freq1/freq2 < t$. Thus the final score of the song is used to tag it as positive or negative.

The traditional approach to classification problem, Naive Bayes model [21] has been used in this final approach.

$$P(sentiment|lyrics) = P(lyrics|sentiment) * P(sentiment)/P(lyrics)$$

As we are not taking into account sentiment across multiple lyrics, $P(lyrics)$ would not affect our results. Also, we do not look at the context of occurrence of the words which makes the final probability for each sentiment to be

$$P(sentiment|lyrics) = \prod_{i=1}^{n} P(w_i|sentiment).$$

The song is tagged as positive or negative depending on which sentiment shows a higher probability.

4.1.2 Experiments

Experiments were conducted for all the previously explained approaches. The first approach was carried out in two different settings, one being 10 fold cross validation wherein the whole dataset was split into training and testing set in the ratio of 9:1 while the second used 5 fold cross validation with the data split into training and testing set in the ratio 4:1. The second method employing the subjectivity lexicon from Hindi SentiWordNet [13] [14] [15] was conducted for

the complete dataset of 200 songs and the tag for each was checked against the tag given to it by the annotators.

The Prevalent In One algorithm requires t to be the input. The optimum t was decided upon by conducting the experiment using 5 fold cross validation dividing the dataset into training and testing set in the ratio 4:1. The parameter t was varied from 0.1 to 2.5, incrementing it by 0.1 in each iteration. Figure 4.1 and Figure 4.2 show the accuracies achieved at different thresholds. The best results were obtained when t was set to 1.2. The Naive Bayes [21] implementation also uses a 5 fold cross validation by dividing the whole dataset into training and testing set in the ratio of 4:1.

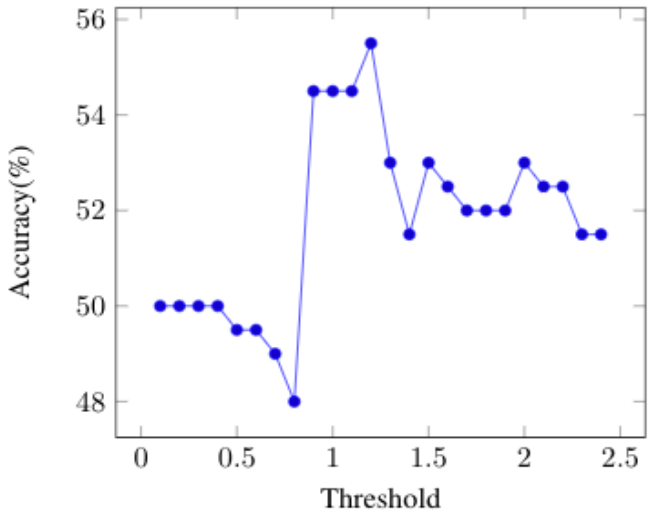

Figure 4.1 Average accuracies for complete testing dataset using Prevalent In One approach for different thresholds.

In all the methods except the last one, each song of the testing set is assigned an initial score of '0'. In the experiments, by making use of subjectivity lexicon and word lists, this score is incremented or decremented accordingly when a positive or a negative token is encountered. The same applies for the Prevalent In One method as well, except that the basis for deciding if a word contributes to the song's positivity or negativity is different. This is decided using the threshold t. Thus, the final score of the song is checked and it is tagged as positive for a score greater than '0' and negative for a score lesser than '0'. Naive Bayes [21] method decides the positivity and negativity of the song by including probabilities of each word in the song as being positive or negative.

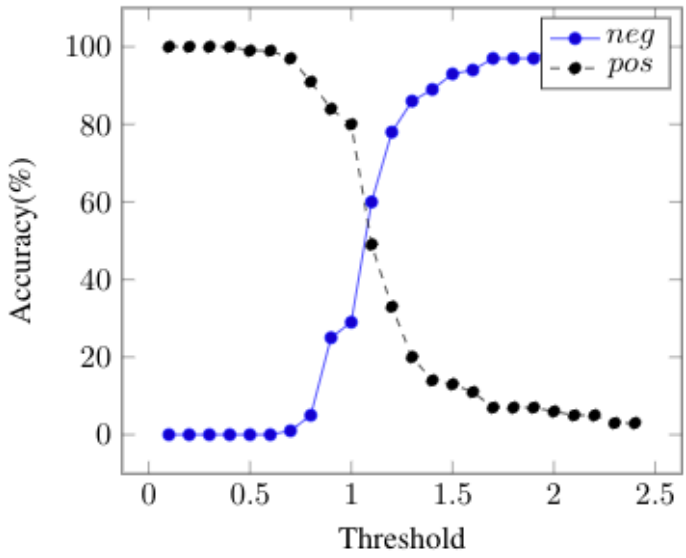

Figure 4.2 Average accuracies of negative and positive lyrics compared with each other for Prevalent In One approach, given different thresholds.

4.1.3 Results and Discussion

The average accuracy achieved for the first experiment conducted using a 10 fold cross validation, was 98.5%. The accuracies were either 100% or 95% (due to misclassification of one of the test cases). The second setting with 5 fold cross validation resulted in an average accuracy of 97% while the individual accuracies for each of the runs varied between 92.5% to 100%.

In this experiment, it is seen that a high level of accuracy is achieved and most of the misclassifications have been with positive songs. One of the drawbacks of this experiment is that it takes into account all the words that have occurred in one class of songs with equal weightage and does not differentiate on the basis of its frequency of occurrence or the subjectivity. This approach is, in a way, similar to employing a corpus specific subjectivity lexicon.

	Recall	Precision
Positive Songs	0.62	0.55
Negative Songs	0.43	0.60

Table 4.1 Results for the experiment using Hindi SentiWordNet [13] [14] [15].

f

The method using Hndi SentiWordNet [13] [14] [15] performs very poorly for the negative songs as compared to the positive ones. It is claimed that the subjectivity lexicon should

be corpus-specific, or created based on statistics from the corpus on which it will be used as this gives better results [?] [?]. On comparing the accuracies given by our corpus specific subjectivity lexicon and the generic lists obtained from Hindi SentiWordNet [13] [14] [15], this can be confirmed as the former gives an accuracy of 97% while the latter performs poorly with an accuracy of 53%.

In the Prevalent In One method, the average accuracy over all the thresholds for the whole testing dataset was found to be 51.75% with the minimum being 48%(t=0.8) and the maximum being 55.5%(t=1.2). An interesting observation is that for positive songs, the accuracy goes up with increasing threshold whereas the opposite is true for negative songs. The average accuracy over all the thresholds is 56.3% for negative songs and 47.2% for positive songs. Using these observations, we decided on the ideal value of t to be 1.2 to ensure better results.

Method Used	Avg Accuracy
Word List (10 fold cross validation)	98.5%
Word List (5 fold cross validation)	97%
Using Hindi SentiWordNet	53%
Prevalent In One method (t=1.2)	55.5%
Using Naive Bayes Model	58.5%

Table 4.2 A comparison of results obtained by all the experiments conducted.

The accuracy varied between 42.5% to 70% with an average of 58.5% while using the Naive Bayes [21] method. It is seen that the algorithm performs way better for positive songs than for negative songs with average accuracies being 77% and 40% respectively which shows a great disparity and proves that this method works better for positive songs.

4.2 Experiments with 'BolLy', the complete dataset

4.2.1 Theory

We conducted a few experiments to classify the song lyrics to extract sentiment polarity expressed by them. This can be used as a baseline experiment for sentiment analysis tasks for Hindi song lyrics. We make use of Naive Bayes [21] and Support Vector Machine [83] classifiers for these experiments. These are discussed here briefly.

Bayesian classifiers have been widely studied and applied for classification tasks [21] [20] [76] [41]. They have also been applied for tasks dealing with text classification [73] [47] [76] [33] [42] [38]. Naive Bayes classifier is a supervised learning algorithm based on the Bayes' assumption that every pairs of features are independent of each other in a given context. This can be translated

to the classification rule

$$P(y|x_1...x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{4.1}$$

given a class variable y and dependent feature vector x_1 through x_n.

The multivariate Bernoulli model [21] is a Bayesian Network that has no dependencies between the occurrence or non-occurrence, i.e., binary word features. It does not have dependencies between words either [42] [38]. The probability of a document given its class can be represented as

$$P(d_i|c_j;\theta) = \prod_{t=1}^{|V|}(B_{it}P(w_t|c_j;\theta) + (1B_{it})(1P(w_t|c_j;\theta))) \tag{4.2}$$

where V is the vocabulary and $t\epsilon 1, ..., |V|$ representing each dimention of the space for the word w_t from V. B_{it} represents t for document d_i and is 0 or 1 depending on the occurrence of w_t in d_i. $c_j \epsilon C c_1, ..., c_{|C|}$ is a component of the mixture model and θ parameterises the mixture model.

The integral word counts based unigram language model is used for the multinomial Naive Bayes model [48] [58]. The multinomial distribution can be shown as

$$P(d_i|c_j;\theta) = P(|d_i|)|d_i|!\prod_{t=1}^{|V|}\frac{P(wt|cj;\theta)^{N_{it}}}{N_{it}!} \tag{4.3}$$

where d_i, c_j, θ, V ,w_t hold the same meaning as in Equation 4.2. N_{it} is the frequency of w_t in d_i

Another supervised learning algorithm used commonly for text classification tasks is SVM or Support Vector Machine [83]. This makes use of a hyperplane to identify the decision boundary. Given a dataset of n points of the form $(\vec{x_1}, y_1)....(\vec{x_n}, y_n)$ where y_i is either 1 or -1 depending on which class the data point $\vec{x_i}$ belongs to. The hyperplane can be defined as the set of points $\vec{x}$ satisfying $\vec{w}.\vec{x} - b = 0$ where $\vec{w}$ is the normal vector to the hyperplane and parameter $\frac{b}{||\vec{w}||}$ expresses the offset of this hyperplane along $\vec{w}$ from the origin.

4.2.2 Methodology

The song lyrics were available as files which were converted into separate lists of lyrics and their corresponding tags for positive and negative songs, called PLyr and PTags for positive songs and NLyr and NTags for negative songs, respectively. For splitting the dataset into training and testing set, we create lists TrainLyr and TestLyr with their respective tags in the lists TrainTags and TestTags. This split is in the ratio 9:1. TrainLyr and TrainTags are used to train the models for Multinomial Naive Bayes (MultinomialNB) [21], Bernoulli's Naive Bayes (BernoulliNB) [21] and Support Vector Machine (SVM) [83]. The latter is run over 5 folds. These trained models are used to predict the sentiment polarity of the data points in TestLyr and the predicted tags are stored in PredTags, which are then compared with TestTags for evaluation.

Classifier used	Accuracy (%)
MultinomialNB	69.61
BernoulliNB	71.57
SVM	75.49

Table 4.3 Accuracies obtained for the classifiers

Classifier	Class	Precision	Recall	F1 Score (%)
MultinomialNB	Positive	0.80	0.62	0.70
	Negative	0.46	0.67	0.54
BernoulliNB	Positive	0.80	0.77	0.79
	Negative	0.56	0.61	0.58
SVM	Positive	0.74	0.99	0.84
	Negative	0.90	0.27	0.42

Table 4.4 MultinomialNB Classifier Scores

4.2.3 Experiments Conducted

The method explained above was conducted on the 'BolLy' dataset. The dataset was split in the ratio of 9:1 for training and testing purposes. As the class is imbalanced and there are more number of positively tagged songs, we set the class prior according to the class distribution. Without this, the classifier would have tagged all the test data points as positive as that would give a better result but that is not our aim. Specifying the class priors helps the classifier to take into account the features given and predict the classes for test data accordingly.

These experiments were carried out using an open source Python library, 'scikit-learn' [68] which is easy to use with very less dependencies. The features used for our experiment included bag of words, term frequency and term frequency-inverse document frequency which were all extracted using the modules from this library. The evaluation metrics such as accuracy, precision, recall and F1 measure were extracted. The SVM classifier [71] was run for 10 folds over the dataset.

4.2.4 Results and Discussion

The average accuracies for the three classifiers with the data split in the ratio 9:1 are shown in the Table 4.3. In this, SVM [83] performed the best with an accuracy of 75.49%. Table 4.4 shows the other evaluation metrics for each of the classifiers used. It compares how the different approaches fared on scores of precision, recall and F1 for positive and negative classes individually.

These results show that Support Vector Machine [83] works the best for this experiment. These experiments can be used as a baseline as the features used are very basic and not tuned specific to the dataset or the classification problem. Incorporating these would definitely give better results which is why the results shown here would work well as a baseline. Precision can be looked at as a classifier's exactness while recall would give a measure of its completeness. If we look at all the evaluation metrics, we see that barring two cases, the models give better precision and recall for the positive class.

4.2.5 Conclusion

In the first set of experiments, it can be seen that corpus specific subjectivity lexicons perform better for a domain specific task as compared to a generic subjectivity lexicon [2]. It is also seen that probabilistic approaches based on the word counts and frequencies alone do not yield satisfactory results. These experiments were conducted to test the usability of the corpus that was under construction. It proved that 200 songs was a very small corpus for achieving significant results for any application. So we continued working on the corpus and built 'BolLy'. The next section describes experiments carried out on the bigger dataset.

Experiments carried out on the complete dataset are valuable to the larger research community as they form a good set of baseline experiments for research advancements in the field of lyric analysis. They also show how different variants of the traditional Naive Bayes [21] model performs on the dataset for this task. The approaches described here can be extended and combined with feature engineering for the task to come up with much more efficient and accurate models for this task.

After establishing the worth of the dataset presented with some preliminary experiments on it, the next chapter will run you through the building of a system to find keywords from Hindi song lyrics.

Chapter 5

Finding Keywords in Bollywood Lyrics

With the widespread use of internet, a vast amount of textual data is available online. It would be highly beneficial if there was some kind of indexing that would allow users to sift through the massive collection of textual data easily and quickly being able to sort out the relevant documents. One approach to do this is to identify keywords for the documents. Keywords are a representative of the document's content and aid in searching and organising them. Means for automatic identification of keywords is a less cumbersome, faster and less error-prone alternative to manually assigning keywords to documents.

We have explored this in the domain of Bollywood song lyrics as the music of a song only tells half of the story. This is a step towards analysis of music lyrics in Hindi, which is a language that is less explored. The dataset we have is in Devanagari script. Also, keywords would aid in tapping the contribution of lyrics of a songs for applications such as recommendation systems, digital music library management, music therapy, TV and radio programmes, etc..

5.1 Dataset Used

For the task of finding keywords from Bollywood song lyrics, the dataset we used is the same as explained in Chapter 3. We used the 'BolLy' dataset without the annotations. This dataset consists of 1055 Bollywood song lyrics in Devanagari script. It includes songs composed in the 1970s to the most recent ones. Only a part of the dataset was used for manual validation and ranking the proposed algorithms for keyword finding on the basis of accuracy and appropriateness.

5.2 Experiments

Keyword generation task can be mainly divided into three sub-tasks. Figure 5.1 shows the steps involved in this process.

- First, identifying candidate keywords from the complete document after pruning the words that do not contribute majorly towards describing the topic of the document.
- Second, deciding the important properties required for selecting a keyword and elucidating these properties for all the candidate keywords and looking at them closely.
- Third, ranking the candidate keywords on the basis of properties calculated for the given method of keyword selection and deciding on the most appropriate ones as keywords for the document.

In this work, we have not manually selected keywords from the documents. Hence, the algorithms explained here are not machine learning algorithms, as they do not learn from the labelled dataset, rather they generate or extract keywords from the document on their own depending on the method chosen. The experiments conducted include a baseline experiment that can be used to relatively gauge the performance of three other algorithms that are proposed. The algorithms are explained in detail in this section followed by a comparison of all the methods.

5.2.1 Baseline Experiment

The baseline experiment described here relies solely on the frequency of occurrence of the words, which is a very basic and a standard measure of the importance of a word in a document. Stopwords occur very frequently in any text and may not be of much importance in terms of the content of the text, hence they are removed from the document as this method is based on the frequencies of the terms. After this, all use of punctuation is removed as well as they don't contribute to the content of the document either. Stemming is performed on the terms used in the document to capture the higher frequency of the words occurring in different forms as their meaning is the same.

The notation f_i is used for frequency and t_i for term. The frequencies, (f_1, f_2, f_3..., f_n) of all unique terms used in the document, (t_1, t_2, t_3..., t_n) are calculated. These terms are ranked in descending order of their frequencies. The top most frequent words are chosen as the keywords for the given document. This is a very simple experiment that does not take into consideration any complex features but just uses the term frequencies. This can be used to measure the performance of other methods.

5.2.2 Python RAKE modified for Hindi

Python provides a library for keyword extraction, called RAKE [74], Rapid Automatic Keyword Extraction. It works on the observation that keywords may also consist of phrases or words. This method only looks at keywords that have a high lexical meaning and do not include stopwords. they may consist of multiple words. RAKE [74] inherently works with English

but some changes to the regular expressions that it uses enables its usage with Hindi as well. These changes are on the basis of the sentence structure for Hindi and the strings used in Devanagari.

An experiment using RAKE [74] was conducted on the dataset. It first tokenises the document into candidate phrases that can potentially be keywords using the modified regular expressions provided by us. A score for each of the candidate keyword was computed taking into consideration significant factors such as:

- frequency of occurrence
- position in the document
- phrase length
- similarity to other documents.

These scores were then used to rank the candidate keywords and phrases in descending order of their scores. The top ranking candidate keywords were selected as keywords for the document.

5.2.3 Statistical Approach Using Spatial Distribution

This is a corpus-independent and language-independent method to generate keywords. It takes into account the spatial distribution of the terms along with their frequency. This is done as the distribution of words in a song's lyrics are an important indicator of its importance. In this method, all the terms in the document are numbered starting from 1 as per their position. All the unique terms, t_i, with their frequencies, f_i, and a list of their positions of occurrence, P_i are generated. The terms with low frequencies are eliminated. P_i consists of the positions at which t_i occurs in the document, (p_1, p_2, p_3..., p_n), when t_i occurs n times. After this, the next nearest neighbour distance series, D_i, is populated for each t_i. This consists of a series of differences in the positions of consecutive occurrences of t_i. Hence, D_i would be the list, (p_2-p_1, p_3-p_2..., p_n-p_{n-1}) wherein (p_1, p_2, p_3..., p_n) belong to P_i.

Once D_i is generated for all t_i, mean, μ_i, and standard deviation, σ_i, is calculated for each of them. The σ_i calculated for the series, D_i, is normalized by dividing it by its corresponding mean, μ_i and this normalized standard deviation is σ_i'. This is done to ensure that the frequency doesn't affect the standard deviation of different words. The important words in a document show less random occurrence or a more noticeable clustering. This is conveyed by a higher value of σ_i'. So all the unique terms can be ranked on the basis of decreasing value of σ_i' and the ones with higher values can be chosen as keywords for the document in question.

5.2.4 Hidden Keywords

Lyrical text is different from other forms of text, such as stories, reviews, novels, etc in the sense that it's grammar is different, the sentence structure is not the one used in regular text. Another major distinction between lyrics and regular text is in the number of words used. Lyrics, in general, have fewer words. The method described in this section addresses this fact.

It is possible that certain words that do not occur in a song's lyrics might be very insightful keywords for the songs, when looked at semantically. These can be referred to as hidden keywords. WordNet [57] is a lexical database that has words grouped into synsets, which are sets of cognitive synonyms. Each synset expresses a distinct concept. Links between synsets are defined on the basis of conceptual-semantic and lexical relations. WordNet [57] can prove to be a powerful resource to try and find the hidden keywords in a text.

In the proposed method, the aim is to extract candidate keywords that do not exist in the document, using the Hindi WordNet [31]. After removing stopwords, each line from the document is treated as the context for each word that it consists of. For each term, a context bag, C_i and a sense bag, S_i is created. C_i consists of all the terms occurring in the context of t_i. All the synset members of t_i are extracted from the WordNet [31]. From these synset members, all the content words occurring in their noun senses and their examples are selected to populate S_i. Only noun senses are taken into account because they are observed to be of higher lexical value than verbs, adverbs and adjectives in terms of expressing the concept or idea of a document's content.

The words that are common to C_i and S_i are used to the create a list of candidate keywords along with their frequencies. Once we have a comprehensive list of all unique candidate keywords with their frequencies, the most frequent ones are chosen as keywords for the specific document.

5.3 Validation

For getting deeper insights into the accuracy of the methods proposed, we came up with two different methods of validation of the results obtained from the proposed methods. One of them looks into how accurately the keywords represent the document under consideration while the other one uses human annotation to get ratings for the keywords obtained by different methods. This section looks into these two processes of validation in detail.

5.3.1 Representation of the actual document

This method of validation is conducted based on the assumption that keywords must represent the document as accurately as possible because keywords, are, in a way, a very concise enumeration of the different ideas and topics conveyed in a document. Topic modelling tasks are widely applied to identify the topics that are expressed by a document. These algorithms

can be used to group documents on the basis of their content. This is used as a test to check how well the keywords generated by the different methods represent the documents.

Latent Dirichlet Allocation (LDA) [7] is a generative probabilistic model for a corpus. It assumes that each document is a mixture of topics and each topic is a distribution of words. Latent Semantic Analysis (LSA) [39] is an algorithm that computes concepts in a document on the basis of relationships in documents and terms. These two topic-modelling algorithms were implemented using the software framework Gensim [71] and the documents were classified into different categories using them. Once this was done, the documents were replaced by the keywords that our approaches had generated, one approach at a time, and these algorithms were employed to again categorise them. The categories allocated to the original documents were compared to those allocated to their keywords. If a document is represented accurately by the keywords, both of them must lie in the same class. The score of number of documents and their keywords lying in the same class shows how efficient that particular method of keyword generation is.

5.3.2 Manual evaluation

The dataset used for the experiments in this work does not have keywords assigned manually to the documents. This makes it difficult to objectively evaluate the performance of these algorithms. Also, identification of keyword is a very subjective problem. To get an insight about the relative performance of the four different algorithms discussed in this paper, we collected results generated from each of the methods explained in section 5 and asked a group of 3 participants to rate each of them out of 10 with 1 for the set of keywords least relevant to the document and 10 for the set of keywords that are accurate and very relevant to the document. Each participant was shown the original document along with four sets of keywords. They were not given any information about the method used to generate the keywords. They were also not allowed to look at the metadata of the song lyrics to prevent biasing.

This task was performed only for 100 song lyrics out of the complete dataset of 1055 songs. These 100 songs were carefully selected ensuring that a good mix of songs was presented to the participants, on the basis of when it was composed, genre of songs, music composers, etc. Also, all the participants selected for the task were University students who were native speakers of Hindi and were of the age group 20-25. They were presented with a short reading material explaining the concept of keywords containing examples of pieces of text such as research paper, medical article, newspaper article and a story with keywords. This was done so that they had a clear understanding of the concept of keywords before they started the task. This means of validation would give us a fairly good idea about how the different methods rank as per human evaluation.

Table 5.1 Scores given by Participant 1 averaged over 100 selected data samples.

Method	Avg. Scores
Baseline Experiment	3.46
Python RAKE	2.83
Statistical Approach using Spatial Distribution	7.04
Hidden Keyword	5.0

Table 5.2 Scores given by Participant 2 averaged over 100 selected data samples.

Method	Avg. Scores
Baseline Experiment	3.42
Python RAKE	2.58
Statistical Approach using Spatial Distribution	6.17
Hidden Keyword	4.88

5.4 Results

In this section, we present the results of the two tasks - manual evaluation and document representation - performed in the previous section to evaluate the proposed methods of keyword generation in Bollywood song lyrics. This also helps us evaluate the performance of the three proposed methods as compared to the baseline experiment. Tables 5.1, 5.2 and 5.3 show the average of the scores given by the three participants on the basis of the relevance of the keywords generated by the different methods.

Table 5.5 shows the percentage overlap of the document clusters allocated by LSA [39] and LDA [7] for the documents and their respective set of keywords generated by the four methods described in this work.

Table 5.3 Scores given by Participant 3 averaged over 100 selected data samples.

Method	Avg. Scores
Baseline Experiment	4.92
Python RAKE	3.08
Statistical Approach using Spatial Distribution	8.54
Hidden Keyword	6.38

Table 5.4 Average scores given by the 3 Participants for 100 selected data samples.

Method	Avg. Scores
Baseline Experiment	3.93
Python RAKE	2.83
Statistical Approach using Spatial Distribution	7.25
Hidden Keyword	5.42

Table 5.5 Percentage overlap for classes assigned for the documents and their keywords by LDA [7] and LSA [39].

Method	LSA	LDA
Baseline Experiment	66.2	71.6
Python RAKE	84.5	84.5
Statistical Approach using Spatial Distribution	77.7	82.4
Hidden Keyword	92.8	81.6

5.5 Conclusion

This chapter looks at the task of identification of keywords in a song lyric document in Devanagari script. We look at one baseline experiment for the same followed by three methods proposed for it including one that also looks at candidate keywords that might not occur in the document's text.

All the manual participants in the validation task were consistent with their ratings and all of them found the Statistical Approach using Spatial Distribution to be the best while the keywords generated by Python RAKE [74] were rated even lower than the baseline experiment. One reason for this can be that RAKE [74] extracts phrases from the text itself and not just meaningful words. Sometimes the keywords extracted might not be grammatically correct phrases. This shows that the use of Python RAKE [74] is not a very good means of extracting keywords from Bollywood song lyrics in Hindi.

When we look at the results from the comparison of documents and their keywords when clustered using LDA [7] and LSA [39], the results obtained for LDA [7] and LSA [39] are not very similar. Hidden keyword gives best results with LSA [39] but it does not perform that well with LDA [7]. On the other hand, RAKE [74] does decently well in both the algorithms. This is because RAKE [74] actually contains chunks of text as it is, so there is a higher chance of overlap with the original document. These results show that with some improvement Hidden Keyword and Statistical Approach using Spatial Distribution can be combined to make a good system for keyword generation.

This work can be continued further by narrowing down the search for keywords using very specific features. Also, exploring methods to build language independent tools for the same would be a challenge that would prove very useful for research in this field for resource-poor languages. Once a robust system for keyword generation is created, it can be used to organise digital music libraries and also as tools for improving recommendation systems.

This chapter presented a major contribution of this thesis work, a system to find keywords from Hindi song lyrics. The next chapter concludes the work presented in this thesis and looks into the future aspects of continuing this work.

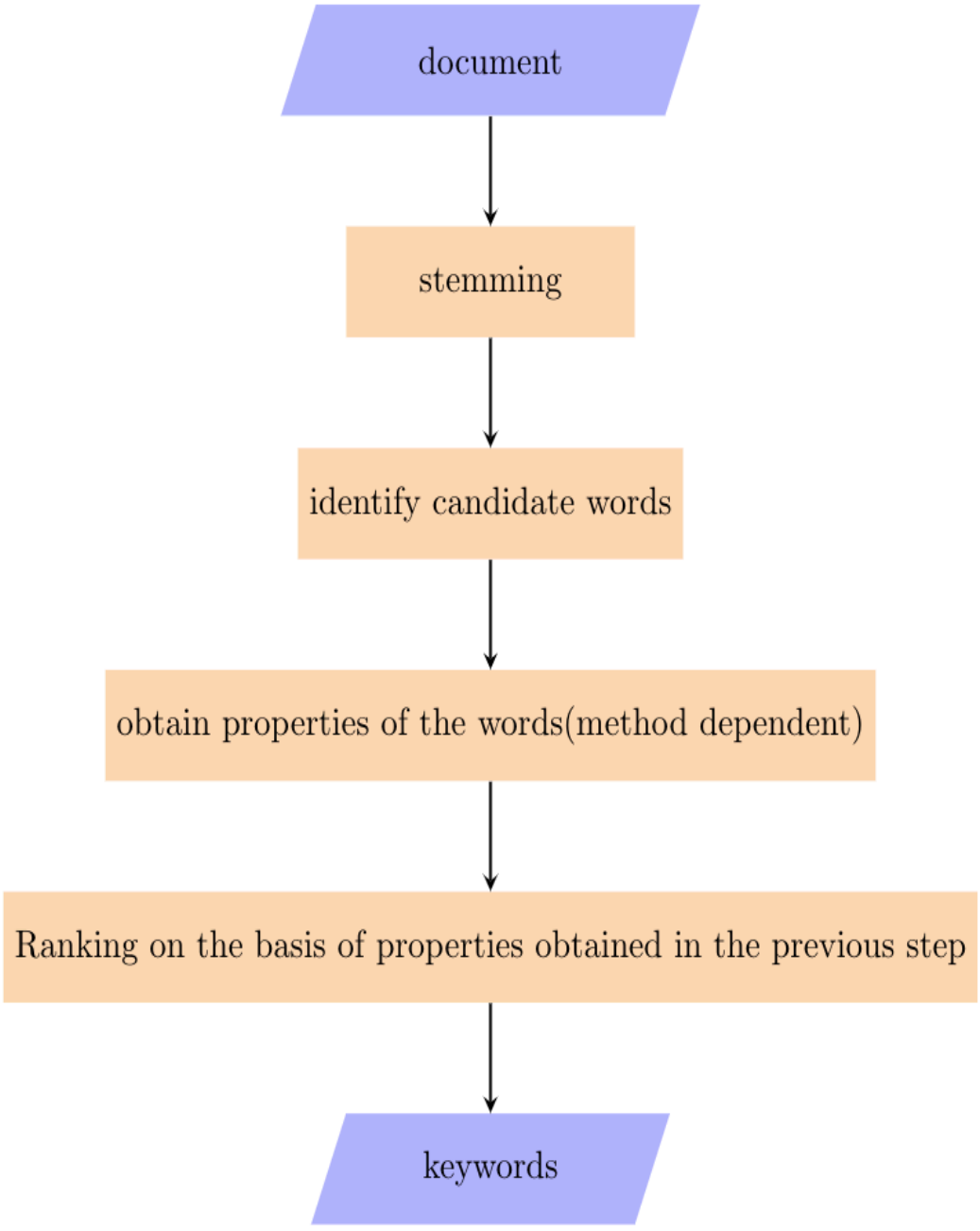

Figure 5.1 Flowchart showing the steps involved in finding keywords.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

This thesis has three major parts: a dataset of Hindi song lyrics annotated with sentiment polarity, experiments for opinion extraction and a system for finding keywords. All these three parts are done on Hindi song lyrics from Bollywood in Devanagari script. All in all, this thesis work is a stride towards enabling the Natural Language Processing research community to better analyse lyrical data, specially in Hindi.

The dataset presented is called 'BolLy'. It comprises of 1055 song lyrics belonging to Bollywood movies collected from early 1970s onwards. They range over a variety of genres and singers and were collected from an online source where it was available in Devanagari script. Of these 712 songs were annotated as positive and 343 as negative on the basis of sentiment polarity expressed by their lyrics only. This data amounts up to 2.6 MB of data along with the metadata.

In the experiments conducted on the 'BolLy' dataset, we start with a preliminary approach for opinion extraction [50] [9] from Hindi song lyrics. We have attempted it by various approaches making use of word dictionaries, subjectivity lexicon [2] and Naive Bayes [21] approach to gain an insight as to which approach performs better. It has been observed that corpus specific subjectivity lexicon performs better than a generic one. With the bigger dataset, we explore Multinomial Naive Bayes [21] [54], Bernoulli Naive Bayes [21] [54] and Support Vector Machine [83] with the latter achieving an accuracy of 75.49%.

The system that is proposed for finding keywords from song lyrics explores the usage of keywords that might not occur in the data. This is a novel technique and gives high percentage overlap for text representation using LSI [39] amounting to 92.8%. Also, this method and Statistical Approach using Spatial Distribution are rated highly by human annotators as 5.42 and 7.25 respectively on a scale of 10. As this is a subjective task, it is important to take into account the ratings given by human annotators, and thus it is proved that Statistical Approach using Spatial Distribution and Hidden Keywords are very useful techniques for this.

6.2 Future Work

It would be great if we could exploit models built for transliteration from Roman script to Devanagari script as most freely available online sources have Bollywood song lyrics in the Roman script. This would enable us to expand the data resource by a substantial amount. Also further annotation can be greatly beneficial. Code-mixing [61] is a popular field of research that is picking up heat and Bollywood songs provide many instances of these in a new context. Annotation of code mixed parts would be a very interesting way to make this dataset richer. It would also be useful if we annotated the existing dataset with finer levels of mood or sentiment.

In future, we would like to handle other linguistic aspects such as negation and thwarting. This would also help in better classification of songs that express a positive emotion about a negative subject or vice versa or if the song that has mood variations. Our work encourages further research towards building a robust system for mood classification of Hindi songs solely based on lyrics as we provide the baseline experiments for this task.

The system proposed for finding keywords in Bollywood song lyrics can be worked upon to make it more robust and language independent. The language specific experiments gave us useful insights about this task but it would be a great advancement if we are successful in building a language independent system for this. Also dependency on tools and resources for languages is also a constraint for resource-poor languages that can be worked upon.

Appendix A

Sample Song Lyrics from Dataset Tagged as Positive

This appendix consists of a sample from the dataset, 'BolLy' of a song lyric tagged as positive.

A.1 Data sample as appearing in 'BolLy' annotated as positive

, ,

A.2 Transliteration in Roman

Tu hai mera ye sansaar saara
Main aur mera pyar saara
Tere hi liye hai
Tu hai, jag mein hai rang jaise
Rut mein hai tarang jaise
Tu hai toh, tu hai toh

Gagan gagan lehar lehar
Bahe ye chandni
O dhara pe jaagi jyoti hai teri
Nayan nayan ghuli hui hai kaamna koi

Nahi nahi koi tujhsa hai hi nahi

Tu hai mera ye sansaar saara
Main aur mera pyar saara
Tere hi liye hai

Chalte chalte kisi dagar mein
Jaise achanak mod aata hai
Yunhi koi ek hi pal mein
Sab kuch piche chhod aata hai

Chamka jo tera mera mann banjara
Choome re
Prem bhare dhun mere mann ne li jo sunn
Jhoome re

Paas aake bhi kyun moun hai tu
Ye toh keh de meri kaun hai tu

Bolte hain nayan moun hoon main
Apne naino se sun kaun hoon main

Tu hai mera ye sansar sara
Main aur mera pyar sara
Tere hi liye hai
Tu hai mera ye sansar sara
Main aur mera pyar sara
Tere hi liye hai

A.3 English gloss for the data sample

You are my universe
Me and all my love
Are just for you
You're like the colour in the world
There's melody in the breeze

Because of you, because of you

In the skies like waves
Moonlight is flowing
Your shining light is present on the Earth
There's a melting desire in my eyes
There's no one like you

You are my universe
Me and all my love
Are just for You

While walking on the road
Suddenly there is a turn
In just a single moment
One leaves everything behind

As the star is shining, my crazy heart
Is rejoicing
If you listen to the love-filled tune of my heart
You'll also dance

Even after coming near, why are you still silent
Tell me this at least, who are you to me?

My eyes are speaking even though I am silent
Listen to me using your eyes and to know who I am

You are my universe
Me and all my love,
Are just for You
You are my universe
Me and all my love,
Are just for You

Appendix B

Sample Song Lyrics from Dataset Tagged as Negative

This appendix consists of a sample from the dataset, 'BolLy' of a song lyric tagged as negative.

B.1 Data sample as appearing in 'BolLy' annotated as negative

, , , ,

, , , ,
, ,

, , , ,

B.2 Transliteration in Roman

Meri raahon mein pade
Tere pairon ke nishan
Ne kahan ne kahan
Teri saanson se judi
Meri saanson ki wafa
Ne kahan ne kahan

Girte un ansuon mein
Kuch toh tujhsa lage hai
In ashqon mein main na hota

Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum

Paon ko the miley zameen ki tarah
Ankhon mein kyu hue nami ki tarah
Dil ko tum kehte the khuda ka hai ghar
Chod ke kyu gaye ajnabi ki tarah

Girte un ansuon mein
Kuch toh tujhsa lage hai
In ashqon mein main na hota

Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum

Bin tere dekhun main zara sa lagoon
Gham se hi aaj kal bhara sa lagoon
Chod de saath na zindagi me
Soch ke baat yeh dara sa lagoon

Girte un ansuon mein
Kuch toh tujhsa lage hai
In ashqon mein main na khota

Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum
Agar tu hota toh, na rote hum

B.3 English gloss for the data sample

That fell in my path
Your footsteps
Said this, said this
Holding our breath together
The faith that is
Said this, said this

Falling along with the teardrops
Something feels like you
I wouldn't be crying in this way

If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying

You were like the ground for my feet
Then why did you become the moisture in my eyes
You called my heart as the abode of God
Then why did You leave it and go away like a stranger

Falling along with the teardrops
Something feels like you
I wouldn't be crying in this way

If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying

I feel as if I'm nothing without You
I'm full of grief nowadays
May life not leave my side
On thinking about this I feel scared

Falling along with the teardrops
Something feels like you

I wouldn't be crying in this way

If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying
If You were here, I wouldn't be crying

Related Publications

- Drushti Apoorva and Radhika Mamidi. BolLy: Annotation of Sentiment Polarity in Bollywood Lyrics Dataset. In Conference of the Pacific Association for Computational Linguistics (PACLING), 2017, pages 41-50. DBLP, 2017. Received Best Student Paper Award.
- Drushti Apoorva, Kritik Mathur, Priyansh Agrawal and Radhika Mamidi. What is this Song About?: Identification of Keywords in Bollywood Lyrics. In 19th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING), 2018.

Bibliography

[1] H. Abburi, E. S. A. Akkireddy, S. V. Gangashetty, and R. Mamidi. Multimodal sentiment analysis of telugu songs. In Proceedings of the 4th Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2016), pages 48–52, 2016.

[2] A. Bakliwal, P. Arora, and V. Varma. Hindi subjective lexicon: A lexical resource for hindi polarity classification. In 24th International Conference on Computational Linguistics (CICLing), 2012.

[3] K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In Conference of the Canadian Society for Computational Studies of Intelligence, pages 40–52. Springer, 2000.

[4] A. Behl and M. Choudhury. A corpus linguistic study of bollywood song lyrics in the framework of complex network theory. In Proceedings of ICON-2011: 9th International Conference on Natural Language Processing Macmillan Publishers, India, 2011.

[5] F. Benamara, C. Cesarano, A. Picariello, D. R. Recupero, and V. S. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In ICWSM. Citeseer, 2007.

[6] K. Bischoff, C. S. Firan, R. Paiu, W. Nejdl, C. Laurier, and M. Sordo. Music mood and theme classification-a hybrid approach. In International Society of Music Information Retrieval, 2009.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022, 2003.

[8] M. M. Bradley and P. J. Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Citeseer, 1999.

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. IEEE Intelligent Systems, 2013.

[10] E. Cambria, R. Speer, C. Havasi, and A. Hussain. Senticnet: A publicly available semantic resource for opinion mining. In AAAI fall symposium: commonsense knowledge, volume 10, 2010.

[11] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3):27, 2011.

[12] M. Choudhury, R. Bhagwan, and K. Bali. The use of melodic scales in bollywood music: An empirical study. In ISMIR, pages 59–64, 2013.

[13] A. Das and S. Bandyopadhyay. Sentiwordnet for indian languages. Asian Federation for Natural Language Processing, China, 2010.

[14] A. Das and S. Bandyopadhyay. Dr sentiment knows everything! In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies: systems demonstrations, 2011.

[15] A. Das and B. Gambäck. Sentimantics: conceptual spaces for lexical sentiment polarity representation with contextuality. In Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, 2012.

[16] G. Ercan and I. Cicekli. Using lexical chains for keyword extraction. Information Processing & Management, 43(6):1705–1714, 2007.

[17] A. Esuli and F. Sebastiani. Sentiwordnet: a high-coverage lexical resource for opinion mining. Evaluation, 17:1–26, 2007.

[18] Y. Feng, Y. Zhuang, and Y. Pan. Popular music retrieval by detecting mood. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pages 375–376. ACM, 2003.

[19] J. L. Fleiss. Measuring nominal scale agreement among many raters. Psychological bulletin, 76(5):378, 1971.

[20] J. H. Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. Data mining and knowledge discovery, 1(1):55–77, 1997.

[21] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. Machine learning, 29(2-3):131–163, 1997.

[22] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. A survey of audio-based music classification and annotation. IEEE Transactions on Multimedia, 2011.

[23] K. Gajjar and S. Shah. Article: Mood based playlist generation for hindi popular music: A proposed model. International Journal of Computer Applications, 2015.

[24] K. Hevner. Experimental studies of the elements of expression in music. The American Journal of Psychology, 48(2):246–268, 1936.

[25] M. Hu and B. Liu. Mining and summarizing customer reviews. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM, 2004.

[26] X. Hu, J. S. Downie, and A. F. Ehmann. Lyric text mining in music mood classification. American music, 183(5,049):2–209, 2009.

[27] X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. F. Ehmann. The 2007 mirex audio mood classification task: Lessons learned. In In Proceedings of the 9th International Conference on Music Information Retrieval, 2008.

[28] Y. Hu, X. Chen, and D. Yang. Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In International Conference on Computational Linguistics, 2009.

[29] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing '03, pages 216–223, Stroudsburg, PA, USA, 2003.

[30] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, et al. The icsi meeting corpus. In Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on, volume 1, pages I–I. IEEE, 2003.

[31] S. Jha, D. Narayan, P. Pande, and P. Bhattacharyya. A wordnet for hindi. In International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India, 2001.

[32] A. Joshi, A. Balamurali, and P. Bhattacharyya. A fall-back strategy for sentiment analysis in hindi: a case study. Proceedings of the 8th ICON, 2010.

[33] T. Kalt and W. Croft. A new probabilistic model of text classification and retrieval. Technical report, Technical Report IR-78, University of Massachusetts Center for Intelligent Information Retrieval, 1996.

[34] H. Katayose, M. Imai, and S. Inokuchi. Sentiment extraction in music. In Pattern Recognition, 1988., 9th International Conference on, 1988.

[35] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull. Music emotion recognition: A state of the art review. In Proc. ISMIR, pages 255–266. Citeseer, 2010.

[36] F. Kleedorfer, P. Knees, and T. Pohle. Oh oh oh whoah! towards automatic topic detection in song lyrics. In Ninth International Conference on Music Information Retrieval, 2008, pages 287–292, 2008.

[37] G. K. Koduri and B. Indurkhya. A behavioral study of emotions in south indian classical music andits implications in music recommendation systems. In Proceedings of the 2010 ACM workshop on Social, adaptive and personalized multimedia interaction and access, pages 55–60. ACM, 2010.

[38] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab, 1997.

[39] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. Discourse processes, 25(2-3):259–284, 1998.

[40] J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. biometrics, pages 159–174, 1977.

[41] P. Langley, W. Iba, K. Thompson, et al. An analysis of bayesian classifiers. In Aaai, volume 90, pages 223–228, 1992.

[42] L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, pages 289–297. ACM, 1996.

[43] C. Laurier, J. Grivolla, and P. Herrera. Multimodal music mood classification using audio and lyrics. In Machine Learning and Applications, 2008. ICMLA'08. Seventh International Conference on, pages 688–693. IEEE, 2008.

[44] C. Laurier, M. Sordo, and P. Herrera. Mood cloud 2.0: Music mood browsing based on social networks. In Proceedings of the 10th International Society for Music Information Conference (ISMIR 2009), Kobe, Japan, 2009.

[45] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In International Conference on Machine Learning, pages 1188–1196, 2014.

[46] J. Leskovec, M. Grobelnik, and N. Milic-Frayling. Learning semantic graph mapping for document summarization. In Proceedings of ECML/PKDD-2004 Workshop on Knowledge Discovery and Ontologies. Citeseer, 2004.

[47] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, pages 37–50. ACM, 1992.

[48] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 3–12. Springer-Verlag New York, Inc., 1994.

[49] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pages 17–24, Stroudsburg, PA, USA, 2008.

[50] B. Liu. Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1):1–167, 2012.

[51] F. Liu, D. Pennell, F. Liu, , and Y. Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 620–628, Stroudsburg, PA, USA, 2009.

[52] L. Lu, D. Liu, and H.-J. Zhang. Automatic mood detection and tracking of music audio signals. IEEE Transactions on audio, speech, and language processing, 14(1):5–18, 2006.

[53] M. I. Mandel, G. E. Poliner, and D. P. Ellis. Support vector machine active learning for music retrieval. Multimedia systems, 12(1):3–13, 2006.

[54] A. McCallum, K. Nigam, et al. A comparison of event models for naive bayes text classification. In Association for the Advancement of Artificial Intelligence-1998 workshop on learning for text categorization, volume 752, pages 41–48. Citeseer, 1998.

[55] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions, page 20. Association for Computational Linguistics, 2004.

[56] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[57] G. Miller. WordNet: An electronic lexical database. MIT press, 1998.

[58] T. M. Mitchell et al. Machine learning. wcb, 1997.

[59] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational linguistics, 17(1):21–48, 1991.

[60] S. Mukherjee and P. Bhattacharyya. Feature specific sentiment analysis for product reviews. In Part 1, Lecture Notes in Computer Science, Springer 7181:475– 487. 2012.

[61] P. Muysken, C. P. Díaz, P. C. Muysken, et al. Bilingual speech: A typology of code-mixing, volume 11. Cambridge University Press, 2000.

[62] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls : Linking text sentiment to public opinion time series. In Proceedings of International AAAI Conference on Web and Social Media, 2010.

[63] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In In Proceedings Of Conference on Empirical Methods in Natural Language Processing, 2002.

[64] B. G. Patra, D. Das, and S. Bandyopadhyay. Automatic music mood classification of hindi songs. In Proceedings of the 3rd Workshop on Sentiment Analysis where AI meets Psychology, 2013.

[65] B. G. Patra, D. Das, and S. Bandyopadhyay. Unsupervised approach to hindi music mood classification. In Mining Intelligence and Knowledge Exploration. 2013.

[66] B. G. Patra, D. Das, and S. Bandyopadhyay. Mood classification of hindi songs based on lyrics. In Proceedings of the 12th International Conference on Natural Language Processing (ICON-2015), 2015.

[67] B. G. Patra, D. Das, and S. Bandyopadhyay. Multimodal mood classification - a case study of differences in hindi and western songs. In International Conference on Computational Linguistics, 2016.

[68] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12(October):2825–2830, 2011.

[69] S. Poria, A. F. Gelbukh, A. Hussain, N. Howard, D. Das, and S. Bandyopadhyay. Enhanced senticnet with affective labels for concept-based opinion mining. IEEE Intelligent Systems, 2013.

[70] J. R. Quinlan. C4. 5: programs for machine learning. Elsevier, 2014.

[71] R. Rehurek and P. Sojka. Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Citeseer, 2010.

[72] D. Reynolds. Gaussian mixture models. Encyclopedia of biometrics, pages 827–832, 2015.

[73] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. Journal of the American Society for Information science, 27(3):129–146, 1976.

[74] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents, pages 1–20. 2010.

[75] J. A. Russell. A circumplex model of affect. Journal of Personality and Social Psychology, 1980.

[76] M. Sahami. Learning limited dependence bayesian classifiers. In KDD, volume 96, pages 335–338, 1996.

[77] A. Schenker. Graph-theoretic techniques for web content mining, volume 62. World Scientific, 2005.

[78] S. Scott and S. Matwin. Text classification using wordnet hypernyms. Usage of WordNet in Natural Language Processing Systems, 1998.

[79] A. A. Shakoor, W. B. Sahebdin, and S. Pudaruth. Exploring the evolutionary change in bollywood lyrics over the last two decades. In The Second International Conference on Data Mining, Internet Computing, and Big Data (BigData2015), page 46, 2015.

[80] C. Strapparava, A. Valitutti, et al. Wordnet affect: an affective extension of wordnet. In Lrec, volume 4, pages 1083–1086. Citeseer, 2004.

[81] M. K. Szabó, V. Vincze, K. I. Simkó, V. Varga, and V. Hangya. A hungarian sentiment corpus manually annotated at aspect level. 2016.

[82] R. E. Thayer. The biopsychology of mood and arousal. Oxford University Press, 1989.

[83] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. Journal of machine learning research, 2(Nov):45–66, 2001.

[84] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas. Multi-label classification of music into emotions. In International Society of Music Information Retrieval, volume 8, pages 325–330, 2008.

[85] P. D. Turney. Learning algorithms for keyphrase extraction. Information retrieval, 2(4):303–336, 2000.

[86] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 417–424. Association for Computational Linguistics, 2002.

[87] A. M. Ujlambkar and V. Z. Attar. Mood classification of indian popular music. In Proceedings of the CUBE International Information Technology Conference, pages 278–283. ACM, 2012.

[88] M. Velankar and H. Sahasrabuddhe. A pilot study of hindustani music sentiments. In Proceedings of the 2nd Workshop on Sentiment Analysis where AI meets Psychology, 2012.

[89] B. Wei, C. Zhang, and M. Ogihara. Keyword generation for lyrics. In The International Society of Music Information Retrieval, ISMIR 2007, pages 121–122, 2007.

[90] Y. Xia, L. Wang, and K.-F. Wong. Sentiment vector space model for lyric-based song sentiment classification. International Journal of Computer Processing Of Languages, 21(04):309–330, 2008.

[91] Y.-H. Yang, Y.-C. Lin, H. tze Cheng, I. bin Liao, Y. chin Ho, and H. H. Chen. Toward multi-modal music emotion classification. In In Proceedings of Pacific Rim Conference on Multimedia (PCM'08), 2008.

[92] C. Zhang. Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems, 4(3):1169–1180, 2008.