What is this Song About?: Identification of Keywords in Bollywood Lyrics

G Drushti Apoorva[1], Kritik Mathur[2], Priyansh Agrawal[1], and Radhika Mamidi[1]

[1] NLP-MT Lab, KCIS, IIIT-Hyderabad
drushti.g@research.iiit.ac.in, priyansh.agrawal@research.iiit.ac.in,
radhika.mamidi@iiit.ac.in
[2] Manipal Institute of Technology, Manipal
krt.mat@gmail.com

Abstract. 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.

1 Introduction

With the growing number of people using the internet, the amount of textual data available to us at our fingertips is also growing tremendously. This necessitates the creation of tools that would help us to sift through many documents quickly. The availability of keywords makes this task so much easier as they can be used to gauge the relevance of the documents or group similar content by its topics. They can also be used to reduce the dimensionality of text to the most important features. Keywords are a set of very significant words or phrases that express the concept, theme or idea of a piece of text. They can be extracted manually or computationally. The former is a cumbersome and a time taking task. Hence, we explore the computational feasibility of this task on specific kind of data.

The music in a song only tells half the story. For music recommendation systems to be on point and feel natural, it is important that it understands what the song is about, and that is where keyword extraction comes in. The presence of keywords along with a song simplifies identification of the songs we need for any particular use. Despite keyword extraction techniques been extensively explored, not a lot of work has been done with lyrics. Also, most of the experiments have been carried out using data in English language. Scarcity of language specific tools and non-availability of data has resulted in the under-representation of resource-poor languages like Hindi. This work explores the task of keyword extraction for lyrics of Bollywood songs written in Devanagari script.

This paper is organized as follows. In Section 2 we explore the work that has been done in this field, Section 3 describes the details of the dataset used for the experiments. The experiments conducted in this work are discussed in detail in Section 4 which has subsections dedicated to each experiment and its intricacies. In Section 5 we explain the validation methods and discuss results of each experiment in Section 6. Finally, Section 7 provides a conclusion to the work presented in this paper along with possible streams of future work.

2 Related Work

Keyword generation is a relatively less explored topic in research in the field of Natural Language Processing. Work has been done for topic detection using non-negative matrix factorization [1], support vector machines [2], conditional random field [3], and sentence level clustering [4] to separate topic from background. Linguistic approaches [5] involve those using word occurrences [6] and lexical chains [7] for this. Approaches using graphs [8] have also been explored. A lot of other methods [9] including some unsupervised [10] ones have been tried as well. But most of this work is done for English and very less for Indian languages.

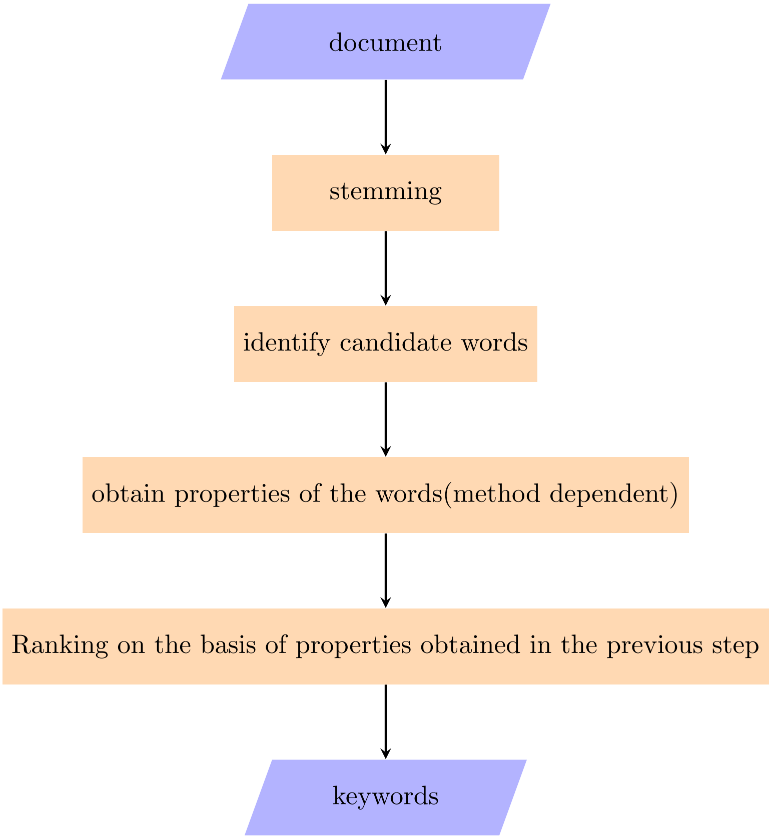
3 Dataset Used

In this paper, we aim to solve the problem of keyword generation in lyrics dataset for Hindi language. For this purpose, we make use of a database containing the lyrics of 1055 unique Bollywood songs composed in the 1970s to the most recent ones. The song lyrics in this dataset is in Devanagari script. The whole of this dataset has been used to extract features that are calculated across documents like tf-idf and for the evaluating criteria making use of document clustering algorithms. Although we only make use of a part of this dataset for manual validation and ranking of appropriateness and accuracy of all the algorithms proposed.

The song lyrics have been pre-processed before using them for keyword generation task. The metadata such as the song's name, movie's name, year of composition, etc. is not considered a part of the document. The punctuation marks are removed from the song lyrics. 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. After this, each song's lyrics was treated as one document for the following experiments.

4 Experiments

Keyword generation task can be mainly divided into three sub-tasks. First, identifying candidate keywords. Second, looking at the properties of all the candidate keywords. Third, ranking them on the basis of properties calculated and selecting the most appropriate ones.


document
stemming
identify candidate words
obtain properties of the words(method dependent)
Ranking on the basis of properties obtained in the previous step
keywords

In this work, we have not manually assigned keywords to the documents. Hence, the algorithms explored do not learn from the labelled dataset, rather generate keywords on their own. The experiments discussed in this section include a baseline experiment that can be used to relatively gauge the performance of three other algorithms that are proposed.

4.1 Baseline Experiment

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, removal of punctuation and stemming is performed on the document. The frequencies, (f_1, f_2, f_3..., f_n) of all unique words, (t_1, t_2, t_3..., t_n) are calculated and ranked in descending order. The top 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.

4.2 Python RAKE

Python provides a library for keyword extraction, called RAKE, Rapid Automatic Keyword Extraction. It works on the observation that keywords may also consist of phrases or words that are not stopwords and have high lexical meaning. They may consist of multiple words but not stop words. RAKE inherently works with English but some changes to the regular expressions that it uses enables its usage with Hindi as well.

An experiment using RAKE was conducted on the dataset. It first tokenises the document into candidate phrases that can potentially be keywords. A score for each of the candidate keyword was computed taking into consideration a lot of significant factors such as its frequency of occurrence, position in the document, phrase length, and similarity to other documents. These scores were then used to rank the keywords and phrases. The top ranking candidate keywords were selected as keywords for the document.

4.3 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 as the distribution of words in a song's lyrics are an important indicator of its importance. For this, all the terms in the document are numbered starting from 1 as per their position. After generating all the unique terms, t_i, with their frequencies, f_i, and a list of their positions of occurrence, P_i, the ones 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), in case t_i occurs n times. 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. Frequency does affect the standard deviation of different words. To avoid this 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. 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.

4.4 Hidden Keywords

This method addresses the fact that lyrics have lesser number of words as compared to other forms of text, such as stories, reviews, novels, etc. This makes it possible that certain words that don't 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 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.

In the proposed method, the aim is to extract candidate keywords that do not exist in the document, using the Hindi WordNet. After removing stopwords, each line 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. 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.

The words that are common to C_i and S_i are added to the 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 Validation

5.1 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. Document clustering tasks are widely applied and this is used as a test to check how well the keywords generated by the different methods represent the documents.

Two well-known document clustering algorithms, LDA (Latent Dirichlet Allocation) and LSI (Latent Semantic Analysis) were implemented and the documents were assigned different categories using them. Once this was done, the documents were replaced by the keywords that our approaches had generated 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.

5.2 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, keyword generation is a very subjective problem. To get a grasp 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.

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.

6 Results

In this section, we present the results of the two tasks performed in the previous section to evaluate the proposed methods of keyword generation in Bollywood song lyrics. Tables 1, 2 and 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 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 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

Table 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 4 shows the percentage overlap of the clusters allocated by LSI and LDA for the documents and their respective set of keywords generated by the four methods proposed in this work.

Table 4. Percentage overlap for classes assigned for the documents and their keywords by LDA and LSI.

Method	LSI	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

7 Conclusions and Future Work

In this paper, we have tried to highlight the applications of a keyword generation system for song lyrics and proposed a baseline method and three other methods to implement this for Bollywood lyrics.

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 were rated even lower than the baseline experiment. One reason for this can be that RAKE extracts phrases from the text itself and not just meaningful words.

When we look at the results from the comparison of documents and their keywords when clustered using LDA and LSI, the Hidden keyword gives best results with LSI but it doesn't perform that well with LDA. On the other hand, RAKE does decently well in both the algorithms. This is because RAKE actually contains chunks of text as it is. 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. 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.

References

1. Kleedorfer, F., Knees, P., Pohle, T.: Oh oh oh whoah! towards automatic topic detection in song lyrics. In: Ninth International Conference on Music Information Retrieval, 2008. (2008) 287–292
2. Zhang, K., Xu, H., Tang, J., Li, J.: Keyword extraction using support vector machine. In: Advances in Web-Age Information Management. (2006) 86–96
3. Zhang, C.: Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems **4(3)** (2008) 1169–1180
4. Wei, B., Zhang, C., Ogihara, M.: Keyword generation for lyrics. In: The International Society of Music Information Retrieval, ISMIR 2007. (2007) 121–122
5. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing '03, Stroudsburg, PA, USA (2003) 216–223
6. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools **13(1)** (2004) 157–169
7. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. Information Processing & Management **43(6)** (2007) 1705–1714
8. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, Stroudsburg, PA, USA (2008) 17–24
9. Rose, S., Engel, D., Cramer, N., Cowley, W. In: Automatic keyword extraction from individual documents. (2010) 1–20
10. Liu, F., Pennell, D., Liu, F., , Liu, Y.: 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, Stroudsburg, PA, USA (2009) 620–628