IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 24-28
www.iosrjournals.org

Measuring Author Impression Using Cosine Similarity Algorithm

Maheshkumar B.Landge[1], Ramesh R.Naik [2], C. Namrata Mahender[3]
[1](Dept. of Computer Science & IT, Dr.B.A.M.University Aurangabad, India (MS))
[1](maheshkumar.landge@gmail.com)
[2](Dept. of Computer Science & IT, Dr.B.A.M.University Aurangabad, India (MS))
[2](ramesh.naik31@yahoo.com)
[3](Dept. of Computer Science & IT, Dr.B.A.M.University Aurangabad, India (MS))
[3](nam.mah@gmail.com)

Abstract: *In this research we are focused on measuring the impact of main author on dependent author/followers. When dependent author use the work of main author, the dependent author may carry some impression of main author in his writing. Text similarity is comparison between query text and text from main document. This comparison gives most similar documents to user. Text similarity is important in the classifying the text as well as document. Using cosine similarity algorithm we can measure similarity between two different documents and also we can find impression of main author on dependent author. Up till now this work is not available for Marathi and English language it's the first attempt in this direction where significance of author. This will also be applied in detection of plagiarism.*

Keywords - *cosine similarity, lexical similarity, semantic similarity, text frequency, inverse document frequency.*

I. Introduction

We have studied literature of Kabir Panth. From that we come to know, the collection of Kabir's poems were found in Adi Granth or Guru Granth Sahib which is holy book of Sikh community and was collected in A.D. 1604 by the orders of sixth Guru, Guru Arjun. The Granth also contains speech of praise of various Gurus and religious songs or bhajans of several devotees such as Kabir, Namdev whose teachings were supposed to corroborate that of the Sikh Gurus **[**1]. This literature shows impact of Kabir's poems was found in Guru Granth Sahib.

In this research we come with new feature i.e. "author impression". We want to extract with this feature, the impact of main author that can be seen on dependent authors when they use main author work. This work tries to find those impacts of well-known writers on dependent authors. We have used cosine similarity algorithm to find impression of author on dependent authors. Author impression can be used in plagiarism detection as well as in text similarity detection. Plagiarism is act of stealing another person's original work and ideas as created by own [2]. The word plagiarism is derived from the Latin words 'plagiaries', means snatcher. In recently, a large amount of data is easily available on the internet. It is very easy to copy the material from Google search engine. It has increased the plagiarism [2].

Similarity is a process by which we can determine relationship between extracted texts. Text similarity is
Having two types: i) Lexical similarity ii) Semantic similarity

1.1. Lexical similarity: It offers similarity based on matching a character or string. For e.g. 'book' and 'cook' are having lexical similarity.

1.2. Semantic Similarity: It provides similarity based on meaning of the words. For e.g. 'happy' and 'pleasure' are having semantically similar to each other. Text similarity is having various applications like question answering system, machine translation, information retrieval, text sum marization, clustering and text classification [3].

Following are statistical measures used in cosine similarity algorithm [14].

i) Term frequency:
Term frequency is the number of times a term (word) occurs in a document.

ii) Normalized term frequency:
NTF=Term frequency/Total number of terms

iii) Inverse document frequency:
IDF=1 + loge (Total Number of Documents / Number of Documents with term)

iv) TF * IDF:
Product of term frequency and inverse document frequency.

v) Cosine similarity: Cosine Similarity (d1, d2) = Dot product (d1, d2) / ||d1|| * ||d2||We have created database of summary on two Marathi poems by six different authors. We have calculated term frequency, normalized term frequency, inverse document frequency, and cosine similarity ratio in cosine similarity algorithm. Using cosine similarity algorithm, we can find the impact of main author on dependent authors.

II. Literature Survey

This literature review shows lexical similarity measure [4]. It is classified in two types of similarities: 1) character based similarity and 2) statement based similarity.

2.1 Character based similarity:

In character based similarity four algorithms are used namely:

i) LCS similarity ii) N-gram similarity iii) Levenshtein similarity iv) Jaro similarity.

2.1.1 LCS (Longest Common Subsequence) similarity: It is a frequently used technique to compute the comparison between two strings. It calculates the longest substring from all the matched substring among two strings [5].

2.1.2 N-gram similarity: Using N-gram similarity method we can find the similarity of sub-sequence of n objects. In this similarity we compute the similarity on the basis of distance between each character in two strings [6].

2.1.3 Levenshtein similarity: It is a technique which uses the distance factor to calculate the similarity between given two strings [7].

2.1.4 Jaro similarity: This technique defines the comparison between two strings on the basis of common character and it is used in duplicate detection [8].

2.2 Statement based similarity: In statement based similarity three different algorithms are described namely: i) cosine similarity ii) Centroid based similarity iii) Web Jaccard Similarity

2.2.1 Cosine similarity: It is broadly used method to find the similarity between two documents. Each text document is denoted in the form of vector [9].

2.2.2 Centroid based similarity: It is a statement based similarity in which each statement is considered as vector form of a document [10].

2.2.3 Web Jaccard Similarity: It is a count based co-occurrence measure technique. This technique is used to find the similarity between words. [11, 12].

III. Development of text Corpus

We have created the text corpus using summary of 3 Marathi Poems from text book of Marathi second standard, which contains 15 text document files. We have taken summary of three poems written by five different authors. We have taken five authors summary on one Marathi poem written by poet K. Narkhede.

IV. Cosine Similarity Algorithms

Cosine similarity is a degree of similarity between two vectors of n dimensions by finding the cosine of the angle between them. The cosine similarity is a method of normalizing length of document during comparison [13].

There are some steps to calculate cosine similarity between two documents.

4.1 Term frequency: It is called as TF. It measures the number of occurences of a word in a text document [14]. We have taken six authors summary from poem. In below table1 there are six specific words and their frequencies in each document are shown.

Table 1. Term frequency of poem "Anandane Nachuya"

Words	Au-thor 1	Au-thor2	Au-thor 3	Author4	Au-thor 5	Au-thor6
आनंदाने	1	1	1	2	0	1
भुंगे	1	1	0	1	0	1
रंग	0	1	1	1	0	1
बाग	1	1	0	1	0	0
गाणी	0	1	0	2	0	0
फुले	1	1	1	2	0	1

In general each document is not having similar size. In large document the frequency of the words will be more as compare to smaller documents. So we require normalizing the document on the basis of size.

For example: In Author4 the word "आनंदाने" occurs two times. The total numbers of words in document Author4 are 94. So the normalized

frequency is 2/94=0.021.

NTF=no. of occurences of word/ total no. of words in document

In below table 2 the normalized term frequencies for six documents are shown.

Table 2. Normalized term frequencies of six documents.

Words	Au-thor 1	Au-thor2	Au-thor3	Au-thor4	Au-thor 5	Au-thor6
आनंदाने	0.02 6	0.025	0.033	0.021	0	0.022
भुंगे	0.02 6	0.025	0	0.010	0	0.022
रंग	0	0.025	0.033	0.010	0	0.022
बाग	0.02 6	0.025	0	0.010	0	0
गाणी	0	0.025	0	0.021	0	0
फुले	0.02 6	0.025	0.033	0.021	0	0.022

a. **Inverse Document Frequency:** The important objective of doing a search is to find similar documents, which are matched with query. At first level all words are have equal importance. Some terms which are coming repeatedly, that terms have little power in determining the importance. We require a way to consider the effects of mostly occurring terms. The words that are having minimum occurrence, those words can be more important. We require a way to consider the effects of less frequently occurring terms [14].

IDF=1+ loge (Number of total documents/number of documents with term frequency). In below table3 inverse document frequency of six words from each document is calculated.

Table 3. Inverse document frequency of terms which are occurring in all six documents.

TERM	IDF
आनंदाने	1.0791
भुंगे	1.1760
रंग	1.1760
बाग	1.3010
गाणी	1.4771
फुले	1.0791

b. TF* IDF

We are finding related documents for the query: **आनंदाने भुंगे**

For each term in the query multiply its normalized term frequency with its IDF on each document.
In Author1 for the word **"आनंदाने"**

the normalized term frequency is 0.026 and its IDF is 1.0791. Multiplying them together we get 0.0280566 (0.026 * 1.0791). Given below table 4a and table 4b product of term frequency and inverse document frequency i.e. TF * IDF, calculations for term **"आनंदाने"**

and **"भुंगे"**

in all the documents are calculated respectively [14].

Table 4 a. Product of Term frequency and Inverse document frequency for the term **"आनंदाने"**

Term	Au-thor 1	Au-thor2	Au-thor 3	Au-thor4	Au-thor 5	Au-thor 6
आनंदा ने	0.02 80	0.0269	o.o35 6	0.0226	0	0.023 7

Table 4 b. Product of Term frequency and Inverse document frequency for the term "भुंगे"

Term	Au-thor1	Au-thor2	Au-thor3	Au-thor4	Au-thor5	Au-thor6
भुंगे	0.030576	0.0294	0	0.01176	0	0.025872

4.4 Cosine Similarity:

In cosine similarity we have taken five documents. In that we compared Author1 document with all four documents. we can compute similarity between two documents by using given formula. The cosine value is having range between -1 and 1. The cosine of small angle is near to 1 and cosine value of large angle near 180 degrees is close to -1.

Cosine Similarity (d1, d2) = Dot product (d1, d2) / ||d1|| * ||d2||

Dot product (d1, d2) = d1 [0] * d2 [0] + d1 [1] * d2 [1] * … * d1[n] * d2[n]

||d1|| = square root (d1 [0]2 + d1 [1]2 + ... + d1 [n]2)

||d2|| = square root (d2[0]2 + d2[1]2 + ... + d2[n]2) [14]. Table 5 shows that cosine similarity ratio for all documents. In author 1 and author 2 documents cosine similarity ratio is 1, which shows the impact of main author on dependent authors.

Table 5 Cosine similarity for all documents

	Author1	Author2	Author3	Author4	Author5
Cosine Similarity	1.00	1.00	19.16	0	0.03412

V. Conclusion

In this research we are mainly focused on author impression of main author on dependent authors. We have taken summary of one poem written by five authors. We have considered six specific words from poem. We have studied literature review of text similarity algorithms. There are two types of similarity algorithms namely: character based similarity algorithms and statement based similarity algorithms. From statement based similarity algorithms we have selected cosine similarity algorithm to find author impression. In cosine similarity, we have calculated term frequency, inverse document frequency, product of term frequency and inverse document frequency and cosine similarity ratio. If there is more similarity between two documents, the cosine similarity value is near to 1 otherwise it is near to -1 or 0. We have found main author's impression in two documents out of five documents. In future, we will work on similarity detection using synonyms which will be useful to detect paraphrasing plagiarism.

Acknowledgements

We are thankful to the Computational and Psycho-linguistic Research Lab, Department of Computer Science & Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad (MS) for providing the facility for carrying out the research.

References

[1] Retrieved August 31, 2016, from shodhganga.inflibnet,http://shodhganga.inflibnet.ac.in/bitstream/10603/28514/11/11_chapter%203.pdf.

[2] Chong, M. Y. M. (2013). A study on plagiarism detection and plagiarism direction identification using natural language processing techniques.

[3] Pradhan, N., Gyanchandani, M., & Wadhvani, R. (2015). A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications, 120(9).

[4] Chapman, S. (2006). Simmetrics: java & c #. net library of similarity metrics.

[5] Irving, R. W., & Fraser, C. B. (1992, April). Two algorithms for the longest common subsequence of three (or more) strings. In Annual Symposium on Combinatorial Pattern Matching (pp. 214-229). Springer Berlin Heidelberg.

[6] Barrón-Cedeno, A., Rosso, P., Agirre, E., & Labaka, G. (2010, August). Plagiarism detection across distant language pairs. In Proceedings of the 23rd International Conference on Computational Linguistics (pp. 37-45). Association for Computational Linguistics.

[7] Navarro, G. (2001). A guided tour to approximate string matching. ACM computing surveys (CSUR), 33(1), 31-88.

[8] Cohen, W., Ravikumar, P., & Fienberg, S. (2003, August). A comparison of string metrics for matching names and records. In Kdd workshop on data cleaning and object consolidation (Vol. 3, pp. 73-78).

[9] Qian, G., Sural, S., Gu, Y., & Pramanik, S. (2004, March). Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In Proceedings of the 2004 ACM symposium on Applied computing (pp. 1232-1237). ACM.

[10] Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. Information Processing & Management, 40(6), 919-938.

[11] Manusnanth, P., & Arj-in, S. (2009). Document clustering results on the semantic web search. In Proceedings of The 5th National Conference on Computing and Information Technology.

[12] Jaccard, P. (1901). Etude comparative de la distribution florale dans une portion des Alpes et du Jura. Impr. Corbaz.
[13] Sree, K. P. N. V. S., & Murthy, J. (2012). Clustering based on cosine similarity measure. International Journal of Engineering Science & Advanced Technology, 2(3), 1-2.
[14] Vembunarayanan, J. (2013, October 28). Tf-Idf and Cosine similarity. Retrieved August 31, 2016, from janav.wordpress.com, https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity.