Sensed-Lexicon based Approach for Identification of Similarity among Punjabi Documents

Jasleen Kaur[1]
School of Engineering
P P Savani University
Kosamba, Gujarat, India

Jatinderkumar R Saini[2]*
Symbiosis Institute of Computer Studies And Research
Symbiosis International (Deemed University)
Pune, India

***Abstract*—Textual similarity among documents often leads to copyright issues. Manual measurement of similarity among documents is a time consuming infeasible activity. In this paper, we proposed a technique for measuring similarity at sensed-lexicon level for documents written in Punjabi language using Gurumukhi script. 50 Punjabi document pairs were manually collected with the help of Punjabi native writers. The proposed technique consisted of major 4 levels. Level 0 consists of data collection phase. Level 1 consists of noise removal and stop word removal sub levels. Extracted tokens were stemmed, lemmatized and synonyms were replaced based on part of speech tagging in level 2. Vector space representation corresponding to each document leads to n-gram generation of documents in level 2. Extracted n-grams were weighted based on term frequency. In level 3, string based token level similarity indexes such as Jaccard Similarity Index (JSI), Cosine Similarity Index (CSI) and Levenshtien Distance Index (LDI) were experimented with weighed tokens. In this work, Human Intelligence Task (HIT) based rating has been utilized for measuring the similarity among documents between 0-100. Results obtained from HIT based rating are compared with results obtained from the proposed technique with various combinations of pre-processing levels. Results revealed that on the basis of majority voting, combination of stop word removal with stemming and 'noun' based synonym replacement leads to the best combination with bi-gram tokens. Statistical analysis indicates strong correlation between CSI and HIT based rating.**

***Keywords*—Cosine Similarity Index (CSI); Jaccard Similarity Index (JSI); Levenshtien Distance Index (LDI); n-gram; Punjabi; similarity checker**

I. Introduction

"It is better to fail in originality than to succeed in imitation" Herman M.

Measuring similarity between words/ terms, sentences, paragraph and document plays an important role in computational linguistics. Similarity measurement is significant component for text classification, search engine, topic modelling, text summarization, legal documents, question answer generation, information retrieval, plagiarism detection and other language related research. Similarity is associated with finding the overlapping index among two documents. This overlapping can be present at sentence level or document level. Similarity among documents can be identified at lexical level and semantic level. In lexical level, words and/or phrases are compared to identify the similarity whereas in semantic level, contextual information associated with words or phrases is extracted and used for comparison.

In general, an automatic document similarity analyzer takes two documents and generates similarity index for them. In this paper, document level similarity is identified at sensed-lexicon level. These documents are written in Punjabi language using Gurumukhi script which adds one more layer of complexity to this task. This work has potential application in plagiarism detection in Punjabi documents. India is the land of languages. Numerous languages and its dialect are being used in spoken as well as written form. Punjabi is one of them. Punjabi falls in Indo –Aryan language category. It is indicated as first language for about 130 million people and is the 10th most spoken language in the world [1-2].

A lot of research has been carried out in area of measuring similarity among documents written in foreign languages, especially English. But this area still needs to be explored in Indian languages. No work has been reported for Punjabi language.

II. Related Work

This section presents different works carried out in area of detecting similarity among documents. Indexes for finding documents similarity are broadly categorized into string based, corpus based and knowledge based measure [5]. String based algorithms perform character level or token level comparison. Corpus based methods detect similarity based on semantic information extracted from large corpus and Knowledge based methods extract semantic similarity based on information extracted from semantic network.

A. Similarity Checking Work in Foreign Languages

Researchers [6] proposed a technique for handling semantically similar words/ paraphrases in Arabic language. Open Source Arabic Corpora (OSAC) was utilized for identifying suspected documents and Word2vec was used for experimentation. Various methods such as Term Frequency-Inverse Document Frequency (TF-IDF), Latent Semantic Analysis (LSA), Latent Dirichlet Allocation (LDA), word2vec, Global Vector Representation (GloVe), and Convolutional Neural Network (CNN) were experimented for paraphrase detection. Another group of researchers [10] used a deep learning based method to detect Arabic paraphrasing. This method consists of pre-processing phase, and word2vec phase. Convolutional Neural Network was used to generate sentence vector. Authors [12] proposed two layer plagiarism

*Corresponding Author

detection method for Arabic documents. This method consists of two layer: Fingerprinting and Word embedding. Documents were weighted using different techniques such as word alignment, POS tags, and inverse document frequency. With recall of 88% and precision of 86%, this method outperformed Plagdet. Different word embedding models were experimented for capturing semantic similarity among sentences. In this work, authors proposed a model (M-MaxLSTM-CNN) for employing multiple sets of word embedding for evaluating sentence similarity. Multi-level comparisons among sentence embedding, generated by multiple word embedding, leads to sentence similarity information. Proposed technique experimented with STS Benchmark dataset and SICK dataset from SemEval and outperform all other existing methods [7].

Saptono et al. experimented with Vector Space Model (VSM) for detecting plagiarism. In this work, cosine similarity method was used to generate the rank of textual paragraphs from query as well as collection vector. Conditional probability concept was utilized to extract number of words from a paragraph. Results revealed that 54.28% average precision and 100% average recall is achieved with threshold value of 0.3 for the conditional probability and 0.2 for cosine similarity [8]. Authors introduced the project ParaPhraser.ru for collecting of Russian paraphrase corpus and organizing a Paraphrase Detection Shared Task. Different techniques were experimented for finding paraphrases among Russian language. Result revealed that traditional classifiers with linguistics features outperformed other methods [13].

B. Similarity Checking work in Indian Languages

Automatic plagiarism software Maulik was developed to check the plagiarism among Hindi documents. Approach used for detecting plagiarism is based on n-grams and comparison with repository and online documents. Input text was pre-processing using stop word removal and stemming. Different values of n were compared with cosine similarity index to find the best value of n. Accuracy reported was 96.3 which is better as compared to existing techniques [3]. Authors proposed Document Synset Matrix for Marathi (DSMM) technique for measuring among Marathi documents. In this work, proses and verses were used for experimentation. Dataset consists of 1206 proses and verses. Different problems such as sense identification of words, polysemy were handled using proposed technique. Accuracy reported was 80 which was better than existing techniques [4]. In this paper, authors presented fuzzy semantic based and Naïve Bayes model for identifying obfuscated plagiarism in English as well as Marathi Language. Semantic relatedness information was analysed based on part of speech tags and WordNet measures. Results revealed that Naïve Bayes Model performed better as compared to fuzzy method [9]. Authors proposed technique for detecting plagiarism in Urdu documents. Reordering of sentences, and inter-textual similarity among Urdu documents was handled in this work. Proposed technique was evaluated using Support vector machine (SVM) and Naïve Bayes (NB). Performance of this proposed method was better as compared to existing techniques [11]. Author proposed Deep learning based methods for handling paraphrase detection task in Indian languages. Convolutional neural network with word embedding, WordNet score and LSTM based methods were experimented [14].

III. METHODOLOGY

This section provides the detailed architecture of system for finding similarity among Punjabi documents. Fig. 1 presents the architecture of Punjabi document similarity analyzer. Punjabi document similarity analyzer consists of mainly 4 different levels. Each level (except level 0) takes input from previous level and provides some output to the next level. Working and detail about each level is as follows:

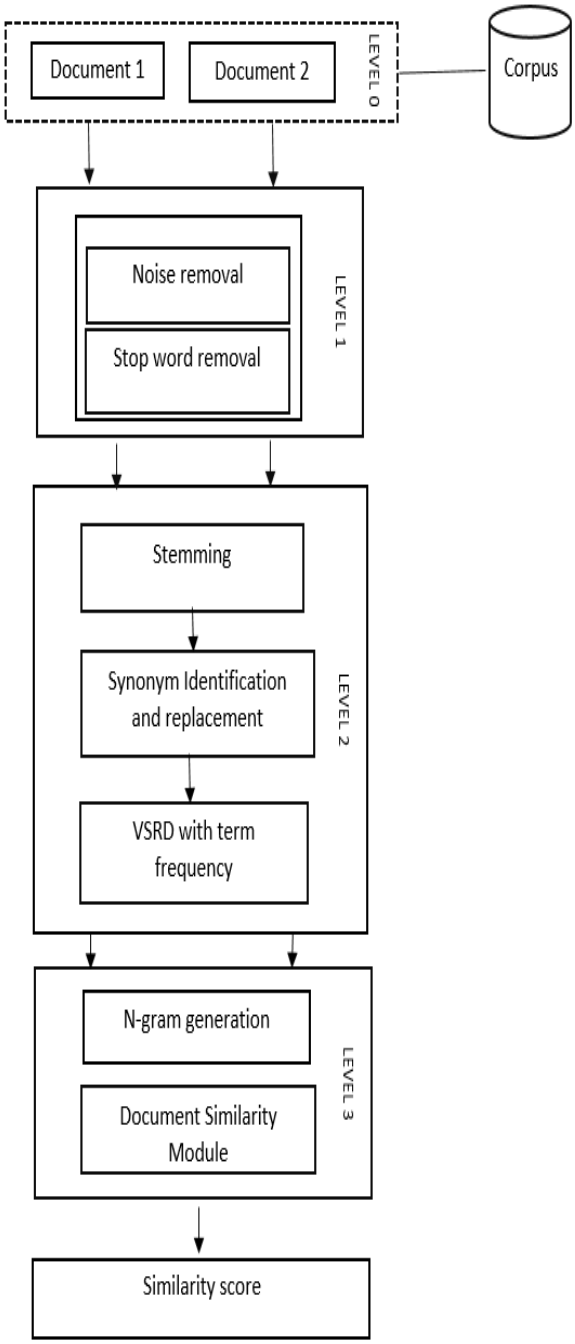

Fig. 1. Architecture of Punjabi Document Similarity Analyzer.

A. Level 0

First step for any kind of analysis is corpus. Due to unavailability of textual similarity corpus in Punjabi, similar document pairs were created. For the creation of these documents, two techniques were followed. In first one, two human annotator (Punjabi native users (writers and speakers)) were requested to write one page on a given topic. As this process was very time consuming, so internet was used as second source for generating similar document pair. Topic selection is versatile from latest topic such as corona virus to festivals of Punjab, from motivational write up to a small story, from real heroes such as Bhagat Singh to real world problems such as pollution, religious gurus to motivational thoughts. Total 50 document pairs (100 documents) written in Punjabi language using Gurumukhi script were collected for further experimentation. Out of 50 document pairs, 26 document pairs were annotated by Punjabi native user whereas, remaining 24 document pairs were generated through internet. Each document pair consists of two documents. So, these two documents D1, and D2 were passed through following phases/levels. Table I provides the statistical details about the dataset.

TABLE I. STATISTICAL DETAILS ABOUT DATASET

Sr. No.	Description	Count
1	Document pair count	50
2	Total documents	100
3	Total token count	12342
4	Unique token count	8976
5	Stop words removed	1679

B. Level 1

D1 and D2 are passed through various pre-processing sublevels. Existing similarity checker for various language (such as English) consists of comparison based on phrases and terms only. Whereas, in this proposed technique, comparison was not just based on exact phrases and but contextual information association with phrases and terms was also checked using IndoWordNet [19]. The purpose of this sub level is to reduce the noise in the input data. So various punctuation marks, symbols were removed from documents (D1 and D2). Stop words were also removed from D1 and D2 [16] [22].

C. Level 2

As mentioned earlier, in this work, document similarity is identified at sensed-lexicon level. Lexical level comprises of lexicons that are being used in both the documents. Lexical features are proven to be effective in Punjabi poetry classification work [17] [21]. Correct sense of these lexicons leads to sensed-lexicon. In next sublevel, remaining words were normalized into their root form. For word normalization, Punjabi stemming rules were used [18]. ND1 and ND 2 (normalized words from document 1 and document 2) were passed to next sublevel. Another important aspect of near copy similarity is synonym replacement. To identify the synonyms replacement among the documents, an algorithm is devised. Detailed steps are presented in algorithm 1. Effect of synonym replacement with stemming is presented separately in the results section. IndoWordNet was utilized for synonym replacement based on Part of Speech (POS) tags [19-20]. In this word, two part of speech tags ('noun' and 'verb') were experimented for identifying the synonym information from document. These normalized words (ND1, ND2) from D1 and D2 represent Vector Space Representation of both documents (VSRD1, VSRD2) [24]. With an intention to give more preference to higher occurring word in document, term frequency (TF) was used to weight the words in D1 and D2. Formula for term frequency is as follows:

$$tf(ND_i, VSRD_i) = count\ of\ ND_i\ appearing\ in\ VSRD_i$$

D. Level 3

In this level, weighted ND1 and ND2 tokens from VSRD1 and VSRD2 were divided into n-grams. Results are presented for n is equal to 1 to 5. Generated n-grams were passed to next sublevel: document similarity level. Lexical similarity between documents was identified through following techniques. Similarity of documents was generated on the basis of scale from 0-100. 0 means no overlapping between the documents and 100 means completely copied document.

Jaccard Similarity Index (JSI): This index was used to measure the similarity between two sets using the formula as given below [24]

$$J(nw-ND_1 \text{ and } \mathrm{nw}-ND_2) = \frac{|nw-ND_1 \cap \mathrm{nw}-ND_2|}{|nw-ND_1 \cup \mathrm{nw}-ND_2|} \quad (1)$$

Where $nw-ND_1$ and $\mathrm{nw}-ND_2$ represents the n-gram representation of weighted ND1 and ND2.

Cosine Similarity Index (CSI): This index was used to measure the similarity based on angle between two vectors [25] where document were represented as vectors.

$$\cos\theta = \frac{\overrightarrow{nw-ND_1}.\overrightarrow{\mathrm{nw}-ND_2}}{\|\overrightarrow{nw-ND_1}\|\|\overrightarrow{\mathrm{nw}-ND_2}\|} \quad (2)$$

Where $nw-ND_1$ and $\mathrm{nw}-ND_2$ represents the n-gram representation of weighted ND1 and ND2

c. Levenshtien Distance index (LDI): This is edit based similarity index. Number of edits in form of insertion, deletion and substitution is calculated. Overall bounded similarity index is generated between 0 and 1 [26].

$$lev_{(nw-ND_1 \text{ and } \mathrm{nw}-ND_2)}(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j)=0 \\ \min\begin{cases} lev_{(nw-ND_1 \text{ and } \mathrm{nw}-ND_2)}(i-1,j)+1 \\ lev_{(nw-ND_1 \text{ and } \mathrm{nw}-ND_2)}(i,j-1)+1 \\ lev_{(nw-ND_1 \text{ and } \mathrm{nw}-ND_2)}(i-1,j-1)+1 \end{cases} & otherwise \end{cases} \quad (3)$$

It measures first i characters and j characters of $nw-ND_1$ and $\mathrm{nw}-ND_2$, respectively.

Implementation of this entire work was done in Python 3.7 [15]. Different packages such as nltk, inltk, sklearn were used in this work.

Algorithm I: Algorithm for finding synonyms of tokens based on part of speech associated

Input: Document1 (D1) and Document2 (D2)

Output: All synonyms replaced in Document2

Step 1: Both documents were tagged based Part of Speech with the help of part of speech tagger.

Step 2: Divide the document D2 into tokens a (t_1 …t_n) and form Bag of Word2 (BOW_2).

Step 3: Extract 'noun'/ 'verb' from document D1 and form Bag of Word1 (BOW_1) with tokens (t_1---t_n).

Step 4: for each token (t_1---t_n) in BOW_1

If token is present in BOW_2

Continue with the next token in Bag of Word1 (BOW_1),

Else

a) Find the synonyms of token using IndoWordNet and search the presence of each synonym in BOW_2

b) If match found in BOW2, replace synonym matched with the original token in BOW_2

c) Goto step3

Step 5: End

IV. Results and Analysis

The purpose of this research work was to find the most suitable similarity index for Punjabi documents. Similarity between the documents can be identified either at Lexical level or at Semantic level. In this work, similarity between Punjabi documents has been measured at lexical level (indicated with 'A' in this work) with different combination of pre-processing techniques. For finding the similarity index, document vectors of TF weighted n-grams have been used. For evaluating the system, results are presented in two sections. Section 1 consists of results by the algorithm and section 2 consists of evaluation results by human linguistic expert through HIT.

A. Results based on Algorithm

In order to find similarity index at lexical level (A), different measures (as specified in previous section) were experimented with different combinations of pre-processing techniques. These combinations have been labelled with characters a to e. Details of these measures with code are presented in Table II. It is notable that these codes have been coined by us for simplicity.

TABLE II. Coined Codes for different Combinations of Pre-Processing and Normalization Techniques

Sr. No.	Coined Codes	Details of different combinations of pre-processing and normalization techniques
1	A.a	Without any pre-processing
2	A.b	Stop words removed from documents
3	A.c	Stop words removed and tokens are stemmed
4	A.d	Stop words are removed, words are stemmed and 'noun' synonym are replaced using IndoWordNet
5	A.e	Stop words are removed, words are stemmed and 'verb' synonym are replaced using IndoWordNet

Each document pair has been evaluated using 5 combinations of pre-processing techniques (as indicated in Table II) in addition with n-gram values from 1 to 5. For a single document pair, 5x5x3 combination have been tested where 5 were the combinations, 5 n-gram values and 3 similarity indexes. In total, 50x5x5x3 combination of experiments have been performed to analyze the result where number of document pairs are 50. For each document pair, each combination from A.a to A.e was tested with value of n - gram used was 1 to 5. Result of each combination (considering only non-zero results for n-grams have been averaged. Results were analyzed based on two valid findings:

1) Finding 1: For more than 38 document pairs, similarity index values have been reported to be 0 for n-gram having value 4 and 5. So, these values were excluded while calculating average.

2) Finding 2: By averaging the n-gram results (as per finding1) obtained in each combination, best combination was selected. Although, combination A.a comes out to be the best combination in all of them. But, A.a results were ignored considering the presence of stop words and so is the maximum overlapping. Detail results are presented in the next subsection.

B. Results based on Human Intelligence Task (HIT)

For this work, each document pair was shared among 10 Punjabi language native speakers. Users selected for this research are from technical background and have sound knowledge about plagiarism and similarity. They were requested to rate the similarity between two documents on the scale of 0-100. Rating value equal to 0 or 100 was ignored considering it as outlier, and such values were not considered while calculating Average Human Intelligence Task (AHIT) rating.

C. Analysis of Similarity Indexes

For each document pair, the best combination is selected on the basis of Average Jaccard Similarity Index (AJSI), Average Cosine Similarity Index (ACSI), and Average Levenshtien Distance Index (ALDI). Table III provides the results obtained with algorithm and index value obtained with AHIT score. Values in column AHIT were averaged and rounded off to 2 decimal points.

TABLE III. RESULTS OBTAINED WITH ALGORITHM AND HIT SCORING

Sr. No.	Document Pair	Best Code	AJSI	ACSI	ALDI	AHIT
1	DP-1	A.e	0.068	0.192	0.07	0.17
		A.d	0.001	0.189	0.021	0.17
2	DP-2	A.d	0.121	0.314	0.132	0.34
3	DP-3	A.e	0.078	0.321	0.046	0.35
4	DP-4	A.d	0.068	0.412	0.063	0.45
5	DP-5	A.d	0.023	0.342	0.021	0.33
6	DP-6	A.e	0.064	0.286	0.053	0.21
7	DP-7	A.d	0.053	0.332	0.083	0.28
8	DP-8	A.c	0.058	0.409	0.049	0.3
9	DP-9	A.c	0.055	0.234	0.038	0.19
10	DP-10	A.c	0.055	0.291	0.07	0.24
11	DP-11	A.d	0.084	0.324	0.113	0.31
		A.c	0.068	0.356	0.07	0.28
12	DP-12	A.c	0.043	0.215	0.04	0.17
		A.d	0.041	0.231	0.042	0.31
13	DP-13	A.d	0.52	0.142	0.034	0.29
14	DP-14	A.d	0.064	0.231	0.062	0.25
15	DP-15	A.c	0.014	0.45	0.023	0.27
		A.b	0.012	0.213	0.14	0.19
16	DP-16	A.d	0.06	0.256	0.071	0.17
17	DP-17	A.d	0.031	0.134	0.012	0.19
18	DP-18	A.d	0.075	0.309	0.053	0.18
19	DP-19	A.d	0.05	0.154	0.038	0.34
20	DP-20	A.d	0.054	0.254	0.041	0.27
21	DP-21	A.c	0.1	0.578	0.149	0.53
22	DP-22	A.e	0.046	0.287	0.08	0.26
23	DP-23	A.d	0.042	0.456	0.04	0.46
24	DP-24	A.d	0.09	0.422	0.078	0.44
25	DP-25	A.d	0.09	0.422	0.078	0.43
26	DP-26	A.c	0.046	0.142	0.034	0.18
27	DP-27	A.d	0.082	0.409	0.079	0.43
28	DP-28	A.e	0.119	0.219	0.123	0.17
29	DP-29	A.c	0.134	0.234	0.098	0.23
30	DP-30	A.d	0.054	0.209	0.041	0.18
31	DP-31	A.d	0.123	0.381	0.14	0.39
32	DP-32	A.d	0.057	0.19	0.069	0.19
33	DP-33	A.c	0.123	0.667	0.149	0.21
34	DP-34	A.d	0.041	0.212	0.056	0.23
35	DP-35	A.b	0.139	0.183	0.045	0.19
36	DP-36	A.c	0.062	0.267	0.068	0.17
37	DP-37	A.d	0.087	0.414	0.078	0.42
38	DP-38	A.b	0.084	0.398	0.113	0.34
39	DP-39	A.d	0.023	0.234	0.012	0.24
40	DP-40	A.e	0.058	0.177	0.049	0.21
41	DP-41	A.d	0.045	0.167	0.038	0.29
42	DP-42	A.d	0.021	0.335	0.067	0.39
43	DP-43	A.d	0.075	0.341	0.054	0.37
44	DP-44	A.e	0.074	0.47	0.0123	0.12
45	DP-45	A.d	0.021	0.127	0.049	0.21
46	DP-46	A.d	0.038	0.177	0.043	0.16
47	DP-47	A.d	0.021	0.532	0.099	0.52
48	DP-48	A.d	0.023	0.452	0.113	0.47
49	DP-49	A.c	0.033	0.145	0.043	0.12
50	DP-50	A.d	0.083	0.318	0.128	0.39

From Table III, Table IV is derived based on the frequency count of each combination. From Table IV, it can be observed that combination A.c is proven to be the best combination so-far on the basis of majority voting mechanism. Result of combination A.a is ignored as stated in finding 1. *Total value reflected in Table IV is 54 because in DP-1, DP-11, DP-12 and DP-15, two combinations comes out to be the best instead of one.

In second phase of experimentation, all the results for combination A.c were compared for checking the existence of correlation with AHIT obtained. For finding the correlation among these values, distribution of data was identified.

Distribution details were presented in Fig. 2. As it can be observed from Fig. 2, data is not normally distributed, so spearman correlation coefficient method was used for finding the correlation between values obtained by algorithm and human score [23]. Correlation strength values lies between -1 and 1. Table V presents the different strength values.

TABLE IV. FREQUENCY DISTRIBUTION FOR COMBINATIONS

Sr. No.	Combination Code	Frequency Count
1.	A.b	3
2.	A.c	12
3.	A.d	32
4.	A.e	7
Total		54*

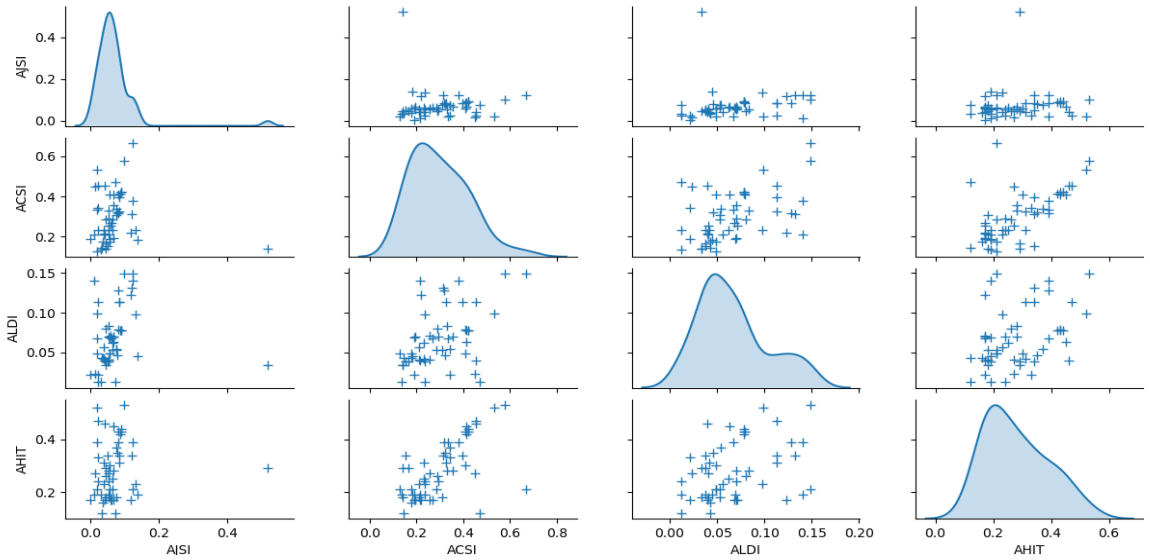

Fig. 2. Distribution of AJSI, ACSI, ALDI and AHIT.

TABLE V. STRENGTH VALUES FOR CORRELATION

Sr. No.	Coefficient Value	Interpretation
1.	0.00-0.19	Very Weak
2.	0.20-0.39	Weak
3.	0.40-0.59	Moderate
4.	0.60-0.79	Strong
5.	0.80-1.00	Very Strong

TABLE VI. COEFFICIENT SCORE

Sr. No.	Correlation Between	Coefficient Value
1.	AHIT and AJSI	0.184
2.	AHIT and ACSI	0.621
3.	AHIT and ALDI	0.351

Spearman correlation coefficient was obtained between 3 similarity index values and average HIT score. Table VI presents the coefficient values. From Fig. 3, it can be observed that highest coefficient value is 0.621 with p-value >0.05. So, AHIT score is more correlated with average cosine similarity index value. So, ACSI values obtained with algorithm has strong association with AHIT (as indicated from Table V values).

D. Analysis of n-gram

In this section, n-gram effect on similarity task is studied. For this work, value of n is taken from 1 to 5. As per assumption specified in result section, results are taken into consideration for n equal to 4 and 5. Analysis is carried out on unigram (n=1), bigram (n=2) and trigram (n=3). Table VII presents the results obtained for 50 document pairs for these n-grams.

For n-gram analysis, n-gram wise result for each combination (A.a to A.e) are averaged. Value for trigrams in document pair 4 and 10 are ignored and are not considered while calculating column average. It can be observed from the Table VII and Fig. 4 that bigram (n=2) gives the best result whereas as n is increased to 3, index values have been reduced.

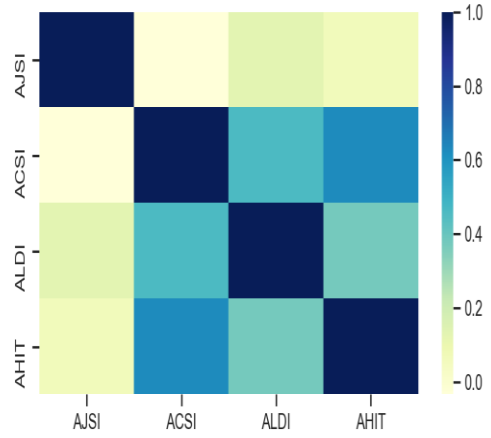

Fig. 3. Correlation Coefficient between Similarity Index Values and Average HIT.

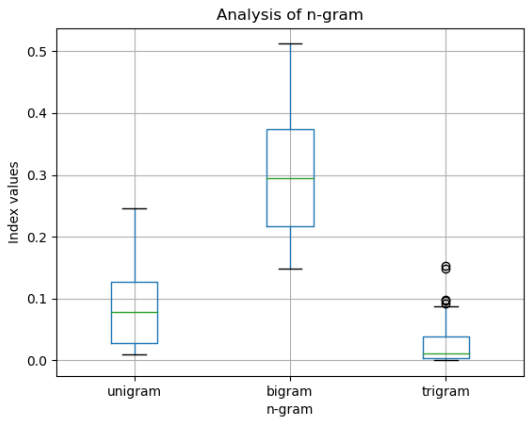

Fig. 4. Analysis of n-gram Index Values.

TABLE VII. ANALYSIS OF N-GRAM VALUES FOR 50 DOCUMENT PAIRS

Sr. No.	Document pair	Unigram (n=1)	Bigram (n=2)	Trigram (n=3)
1.	DP-1	0.127	0.343	0.01275
2.	DP-2	0.123	0.343	0.015
3.	DP-3	0.087	0.334	0.043
4.	DP-4	0.016	0.221	0
5.	DP-5	0.065	0.328	0.008
6.	DP-6	0.197	0.383	0.098
7.	DP-7	0.040	0.316	0.010
8.	DP-8	0.040	0.278	0.002
9.	DP-9	0.015	0.185	0.001
10.	DP-10	0.009	0.166	0
11.	DP-11	0.060	0.313	0.012
12.	DP-12	0.022	0.213	0.014
13.	DP-13	0.013	0.189	0.001
14.	DP-14	0.021	0.181	0.003
15.	DP-15	0.090	0.273	0.008
16.	DP-16	0.026	0.154	0.005
17.	DP-17	0.031	0.240	0.003
18.	DP-18	0.075	0.279	0.003
19.	DP-19	0.130	0.429	0.038
20.	DP-20	0.079	0.386	0.019
21.	DP-21	0.080	0.398	0.012
22.	DP-22	0.246	0.450	0.153
23.	DP-23	0.177	0.442	0.065
24.	DP-24	0.201	0.434	0.046
25.	DP-25	0.202	0.444	0.034
26.	DP-26	0.011	0.176	0.002
27.	DP-27	0.080	0.271	0.007
28.	DP-28	0.024	0.149	0.001
29.	DP-29	0.035	0.242	0.005
30.	DP-30	0.081	0.273	0.002
31.	DP-31	0.145	0.299	0.038
32.	DP-32	0.078	0.375	0.018
33.	DP-33	0.078	0.478	0.010
34.	DP-34	0.232	0.512	0.148
35.	DP-35	0.185	0.374	0.097
36.	DP-36	0.026	0.154	0.005
37.	DP-37	0.011	0.176	0.002
38.	DP-38	0.128	0.328	0.016
39.	DP-39	0.085	0.312	0.042
40.	DP-40	0.145	0.299	0.038
41.	DP-41	0.078	0.375	0.018
42.	DP-42	0.172	0.283	0.092
43.	DP-43	0.017	0.178	0.005
44.	DP-44	0.145	0.299	0.038
45.	DP-45	0.078	0.375	0.018
46.	DP-46	0.118	0.289	0.088
47.	DP-47	0.080	0.271	0.006
48.	DP-48	0.022	0.148	0.001
49.	DP-49	0.029	0.243	0.004
50	DP-50	0.029	0.165	0.006
Average		0.085	0.295	0.027

V. CONCLUSION

As the Punjabi textual content is increasing day by day on web, there is a need to check many of such documents for similarity. Manually detecting the similarity is a tedious task. So, the main objective of this work was to automate the similarity detection process. As there was unavailability of similarity textual corpus, it was created manually through human annotators. 50 document pairs were collected for further experimentation. Each document pair consists of information about the same topic. These document pairs were passed through various pre-processing techniques such as stop word removal, stemming, part of speech based synonym replacement with the help of IndoWordNet. Different combinations of these techniques were tested with n-gram with value of n from 1 to 5. JSI, CSI, LDI and HIT based rating have been used for evaluation. Results indicated that combination of pre-processing technique (stop word removal with root word conversion using stemming and synonym replacement with 'noun' based part of speech tag) proven to be the best combination so-far for detecting similarity among Punjabi documents. Out of the 3 indexes used for experimentation, values obtained for CSI are highly correlated with HIT based rating.

REFERENCES

[1] Punjabi language accessed from https://simple.wikipedia.org/wiki/Punjabi_language in Jan 2020.

[2] S. Jatinderkumar, and K. Jasleen, "Kāvi: An Annotated Corpus of Punjabi Poetry with Emotion Detection Based on 'Navrasa'". Proc. Comp. Sci., vol. 167, pp. 1220-1229, March 2020.

[3] G. Urvashi, and G. Vishal, "Maulik: A Plagiarism Detection Tool for Hindi Documents" Ind. J. of Sci. and Tech., vol. 9, no.12, pp. 1-6, March 2016.

[4] B. Prafulla, and R. S. Jatinderkumar, "Marathi Document: Similarity Measurement using Semantics-based Dimension Reduction Technique" Int. J. of Adv. Comp. Sci. and App., vol. 11, no. 4, pp. 138-143, 2020.

[5] H. G. Wael, and A. F. Aly, "A Survey of Text Similarity Approaches" Int. J. of Comp. Appl., vol. 68, no. 13, pp. 13-18, 2013.

[6] A. Mahmoud, and M. Zrigui, "Similar Meaning Analysis for Original Documents Identification in Arabic Language" In International Conference on Computational Collective Intelligence, pp. 193-206. Springer, Cham, 2019.

[7] N. H. Tien, M. N. Le, Y. Tomohiro, and I. Tatsuya, "Sentence modeling via multiple word embeddings and multi-level comparison for semantic textual similarity" Inf. Process. & Manage., vol.56 no. 6, pp. 1-10, 2019.

[8] R. Saptono, H. Prasetyo, and A. Irawan, "Combination of Cosine similarity method and conditional probability for plagiarism detection in the thesis documents vector space model" J. of Telecomm., Elect. and Comp. Engi., vol. 10 no. (2–4), pp. 139–143, 2018.

[9] N. Shenoy, and M. A. Potey, "Semantic similarity search model for obfuscated plagiarism detection in Marathi language using Fuzzy and Naïve Bayes approaches" IOSR J. of Comp. Engi., vol.18 no.3, pp. 83–88, 2016.

[10] A. Mahmou, A. Zrigui, and M. Zrigui, "A text semantic similarity approach for Arabic paraphrase detection" In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2017. Lecture Notes in Computer Science, 10762. Springer, Cham. pp. 338-349, 2018.

[11] W. Ali, T. Ahmed, Z. Rehman, A. Rehman, and M. Slaman, "Detection of Plagiarism in Urdu Text Documents" 14th International Conference on Emerging Technologies (ICET), Islamabad, pp. 1-6, 2018, doi: 10.1109/ICET.2018.8603616.

[12] B. Nagoudi, A. Khorsi, H. Cherroun, and D. Schwab, "A two-level plagiarism detection system for Arabic document" Cyber. and Info. Tech., vol. 18 no.1, pp. 1–18, 2018.

[13] L. Pivovarova, E. Pronoza, E. Yagunova, and A. Pronoza, "ParaPhraser: Russian paraphrase Corpus and shared task" In: Filchenkov A., Pivovarova L., Žižka J. (eds) Artificial Intelligence and Natural Language. Communications in Computer and Information Science, Springer, 789, pp. 211–225, 2018.

[14] B. Rupal, S. Gargi, and S. Yashvardhan, "Deep Paraphrase Detection in Indian Languages" In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp.1152–1159, 2017.

[15] S. Bird, L. Edward and K. Ewan, Natural Language Processing with Python. O'Reilly Media Inc, 2009.

[16] K. Jasleen, and S. Jatinderkumar. Punjabi Stop Words: A Gurmukhi, Shahmukhi and Roman Scripted Chronicle. In Proceeding of ACM Symposium WIR'16, 32-37. DOI: 10.1145/2909067.2909073. 2016.

[17] K. Jasleen, and S. Jatinderkumar, "Designing Punjabi Poetry classifiers using machine learning and different textual features" Int. Arab J. of Info. Tech., vol.17, no. 1, pp. 38-44, 2020.

[18] V. Gupta, "Automatic Stemming of Words for Punjabi Language," Advances in Signal Processing and Intelligent Recognition Systems, vol. 264, pp. 73-84, 2014.

[19] B. Pushpak, IndoWordNet, Lexical Resources Engineering Conference 2010 (LREC 2010), Malta, May, 2010.

[20] Punjabi Part of Speech Tagger available at http://punjabipos.learnpunjabi.org/.

[21] K. Jasleen, and S. Jatinderkumar, "Punjabi Poetry Classification: The Test of 10 Machine Learning Algorithms", International Conference on machine learning and computing, Singapore, February 24-26, 2017, pp. 1–5, https://doi.org/10.1145/3055635.3056589.

[22] K. Jasleen and S Jatinderkumar, "Automatic Punjabi Poetry Classification Using Machine Learning Algorithms with Reduced Feature Set" Int J. of Art. Int. and Soft comp. Inderscience Publishers. vol 5, no 4, pp 311-319. DOI: 10.1504/IJAISC.2016.10002239.

[23] M. Jerome, and W. Arnold, Research Design and Statistical Analysis (2nd ed.). Lawrence Erlbaum. pp. 508. ISBN 978-0-8058-4037-7, 2003.

[24] M. Melucci, "Vector-Space Model". In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA, 2009.

[25] L. Michael, and D. Winter, "Distance between sets" Nature, vol. 234 no.5, pp. 34–35, 1971.

[26] A. Singhal, "Modern Information Retrieval: A Brief Overview" Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 24 no.4, pp. 35–43, 2001.