Automated Analysis of Bangla Poetry for Classification and Poet Identification

Geetanjali Rakshit[1,2,3] **Anupam Ghosh**[2]
Pushpak Bhattacharyya[2] **Gholamreza Haffari**[3]
[1]IITB-Monash Research Academy, India, [2]IIT Bombay, India
[3]Monash University, Australia
{geet,pb}@cse.iitb.ac.in, anupam.ghsh@gmail.com
gholamreza.haffari@monash.edu

Abstract

Computational analysis of poetry is a challenging and interesting task in NLP. Human expertise on stylistics and aesthetics of poetry is generally expensive and scarce. In this work, we delve into the data to automatically extract stylistic and linguistic information which are useful for analysis and comparison of poems. We make use of semantic (word) features to perform subject-based classification of Bangla poems, and various stylistic as well as semantic features for poet identification. We have used a Multiclass SVM classifier to classify Tagore's collection of poetry into four categories: devotional, love, nature and nationalism. We identified the most useful word features for each category of poems. The overall accuracy of the classifier was 56.8%, and the analysis led us to conclude that for poetry classification, word features alone do not suffice, due to allusions often being used as a poetic device. We, next, used these features along with stylistic features (syntactic, orthographic and phonemic), for poet identification on a dataset of poems from four poets and achieved a performance of 92.3% using a Multiclass SVM classifier. While content-based and stylometric analysis of prose in Bangla has been done in the past, this is a first such attempt for poetry.

1 Introduction

Poetry is a creative expression of language that often makes use of one or more of the crafts of diction, sound, rhythm, imagery and symbolism. Processing creative writing such as poetry by computers is challenging, as opposed to ordinary everyday text, for computers are efficient in carrying out tasks of a more logical nature, as compared to those involving creativity. The volume of research in automated analysis of poetry has generally been low, and no work has been reported on Bangla poetry. Bangla is the seventh most spoken language in the world and has a rich literary tradition. While work on stylometry for prose (Chakraborty and Bandyopadhyay, 2011) and author identification (Das and Mitra, 2011) has been reported, our work is the first of its kind to analyse Bangla poetry.

The computational analysis of poetry is important, for not only can it lead to a better understanding of what makes rich literature, but it also has applications such as making recommendations to readers based on their literary tastes, as also in the psychological effects of poetry (Stirman and Pennebaker, 2001). Identifying the poet is also important for plagiarism detection.

We explore various kinds of features from Bangla poems to carry out specific analyses. Firstly, we perform a subject-based classification of poems into pre-determined categories from Tagore's poems using semantic features, the categories being *pooja* (devotional), *prem* (love), *prokriti* (nature), and *swadesh* (nationalism). With our experiments, we establish the fact that the words can help only so far, due to frequent use of poetic devices such as allusion and symbolism, which often leave poems open to multiple interpretations. Second, we observe that word features do fairly well for poet identification. The results improve when stylistic features (orthographic, syntactic and phonemic) are also introduced.

The paper is organized as follows. We dis-

cuss the literature in Section 2, and describe our approach in Section 3. The system architecture and its details have been described in 4. The experimental setup and results are covered in sections 5 and 6, respectively. We delve into analysis of the results in Section 7. We conclude our work and discuss scope for future work in Section 8.

2 Related Work

Computational understanding of poetry has been previously studied for languages such as English (Kaplan and Blei, 2007), (Kao and Jurafsky, 2012), Chinese (Voigt and Jurafsky, 2013) and Malay (Jamal et al., 2012).

Kaplan and Blei (2007) analyse American poems in terms of style and visualise them as clusters. Kao and Jurafsky (2012) use various stylistic features to categorise poems into ones written by professional and amateur poets, and establish the importance of *Imagism* in poetry of high-quality. Lou et al. (2015) use of a SVM to classify poems in English into 3 main categories and 9 subcategories by combining tf-idf and Latent Dirichlet Allocation. All this work has been done for English. Voigt and Jurafsky (2013) observed through computational analysis the decline of the classical nature of Chinese poetry. Li et al. (2004) use a technique based on term connections for stylistic analysis of Chinese poetry. Jamal et al. (2012) have used a Support Vector Machine model to classify traditional Malay poetry, called pantun, into various themes.

No work in Bangla poetry has been so far reported in the literature. Chakraborty and Bandyopadhyay (2011) have used low-level, chunk-level and context-level features for semi-supervised detection of stylometry in Bangla prose on the writings of Rabindranath Tagore. Das and Mitra (2011) conducted experiments on author identification of Bangla prose on the works of three authors, namely Rabindranath Tagore, Bankim Chandra Chattopadhyay and Sukanta Bhattacharyay. They have used a Naive Bayes classifier using simple unigram and bigram features.

3 Our Approach

We use both word features and stylistic features that have reported in the literature. In the following section, we briefly describe them and go on to explain how they have been adapted to be used in our system for Bangla. One feature not previously reported for stylometry is *reduplication*, which is common, though not exclusive, to Indian languages. As compared to Indian languages, however, its literary merits in English might be arguable. Usage like *ha ha* doesn't generally act a poetic device.

3.1 Features

The stylometric features used for classification can be broadly classified into three kinds: orthographic, syntactic and phonemic. These are the same categories as reported in (Kaplan and Blei, 2007). Besides these, lexical features have been used.

Orthographic Features: Orthographic features deal with the measurements of various units of the poem. These features include word count, number of lines, number of stanzas, average line length, average word length, and average number of lines per stanza.

Syntactic Features: The syntactic features deal with the frequencies of the various parts of speech (POS) in the poem.

Phonemic Features: Sound plays an important role in poetry. Phonemic features deal with the sound devices used in a poem. Rhyme and metre are essential poetic devices. We make use of the following phonemic features: rhyme scheme, alliteration and reduplication.

Some common kinds of rhyme has been tabulated in Table 1.

Lexical Features: Each word type is a feature and its value is the tf-idf.

4 System Overview

The high-level view of the system is shown in Figure 1. The basic blocks of the system are: Alliteration and Reduplication, Rhyme Scheme Detector, Document Statistics, Shallow Parser and SVM classifier. Each one has been described in the subsequent subsections.

4.1 Alliteration and Reduplication

Alliteration is a poetic device which refers to the repetition of consonant sounds in the beginning of consecutive words. An example for this in Bangla would be অনাদরে অবহেলায়

Rhyme Type	Examples
Identical Rhyme: Identical phoneme sequence	cat-cat, বাঁকে-বাঁকে (*baanke-baanke*)
Perfect Rhyme: Same phoneme sequence from the ultimate stressed vowel onwards, but differing in the previous consonant	cat-rat, বাঁকে-থাকে (*baanke-thaake*)
Semi Rhyme: A perfect rhyme where one word has an additional syllable at the end	stick-picket জবা - অবাক (*joba-obaak*)
Slant Rhyme: Either identical ultimate stressed vowels or identical phoneme sequences following the ultimate stressed vowel, but not both	queen-afternoon কল্লোল - কোলাহল (*kallol-kolahol*)

Table 1: Types of Rhyme

(*anaadore abohelay*). To detect alliteration, we check the beginning sound of each word for every pair of consecutive words in a line.

Reduplication refers to the repetition of any linguistic unit such as a phoneme, morpheme, word, phrase, clause or the utterance as a whole (Chakraborty and Bandyopadhyay, 2010). It is mainly used for emphasis, generality, intensity or to show continuation of an act. It may be partial (খাওয়া দাওয়া *khaawa daawa*) or complete (আকাশে আকাশে *akaashe akaashe*). We check only for complete reduplication. We use a simple algorithm that basically checks if two consecutive words in the poem are identical.

4.2 Rhyme Scheme Detection

A rhyme scheme is the pattern of rhymes at the end of each line of a poem or song. The rhyme scheme of the poem can be determined by looking at the end word in each line of a poem. Various rhyme schemes are used. Ex: *abab*, *aabb*, *ababcc* and so on.

In the event of absence of Bangla Pronunciation Dictionary, we wrote the following algorithm. A character in Indian language scripts

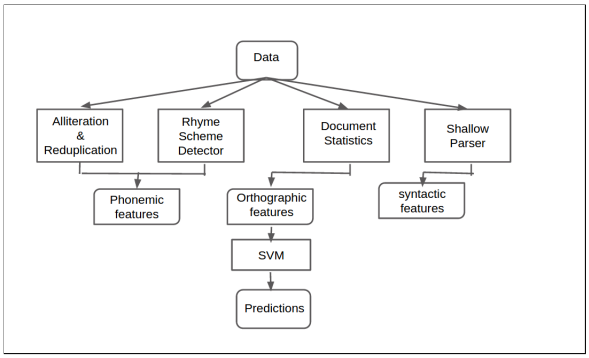

Figure 1: System Overview

is close to a syllable and there is one-to-one correspondence between what is spoken and what is written (Kishore and Black, 2003). In most cases, Bangla words are spoken as they are written. We also accomodate certain non-compliant cases, for instance for the case of হ-ending words, as explained in the subsequent algorithms.

In our system, we check for perfect rhyme and identical rhyme only. We grouped similar sounding vowels and consonants into groups, to allow for similar sounds to rhyme in case of perfect rhyme. This grouping was done as shown in Table 2. A detailed study of Bangla phonemics can be found in (Barman, 2011). The algorithm to detect the rhyme scheme is shown in Algorithm 1. The algorithm to check for rhyming words is described in Algorithm 2.

অ	আ	ই,ঈ	উ, ঊ	এ, ঐ
ও, ঔ	ক, খ	গ, ঘ	চ, ছ	জ, ঝ, য
ট, ঠ	ড, ঢ	ত, থ, ৎ	দ, ধ	ণ, ন
প, ফ	ব, ভ	ম	র, ড়	ল
শ, ষ, স	হ	য়		

Table 2: Sound Groupings

The *Find-rhyme-scheme* algorithm takes a poem and the length of a stanza in the poem as input, and returns the rhyme scheme for the poem. We first initialise a string variable *r*hyme_scheme to a sequence of consecutive English alphabets, which denotes the rhyme scheme. Next, we pick the end word for the first line and check if it rhymes with the end word of the next line (by calling *Check-rhyme()*). We keep checking until the last line, or until, a rhyming line is found. We then update the rhyme_scheme variable and check

Algorithm 1: Find-rhyme-scheme

Input: poem, len_of_stanza
Output: rhyme_scheme

Initialise: $rhyme\_scheme$ = "$abcdefgh..$"
1. for i in range(0, $len\_of\_stanza - 1$)
2. Read line and pick last word $word[i]$
3. for j in range($i + 1$, len_of_stanza)
4. Read line and pick last word $word[j]$
5. Check-rhyme($word[i]$, $word[j]$)
6. If true
7. $rhyme\_scheme[j] = rhyme\_scheme[i]$
8. break
9. return rhyme_scheme height

from the next line onwards, and repeat the process until the last but one line is processed.

Algorithm 2: Check-rhyme

Input: $word_1$, $word_2$
Output: flag
V denotes vowel, C denotes consonant

Initialise: flag = 0
1. Pick last character z_1 and z_2 of $word_1$ and $word_2$, respectively
2. if similar_sounding(z_1, z_2) or if either of z_1 or z_2 is C while the other is 'ো'
3. pick the last but one character y_1 and y_2 of $word_1$ and $word_2$, respectively
4. if both y_1 and y_2 are V or both y_1 and y_2 are C
5. if similar_sounding(y_1, y_2)
6. flag = 1
7. if both z_1 and z_2 are C
8. if y_1 and y_2 are C
9. flag = 1
10. if y_1 and y_2 are V
11. if similar_sounding(y_1, y_2)
12. flag = 1
13. return flag

The *Check-rhyme* algorithm takes as input two Bangla words, and returns whether or not they rhyme. It basically compares the last two characters of both words. The last two characters should either be identical to each other, or should be similar sounding, based on the groupings we made in Table 2. Thus words like মাঝে(*maajhe*) and লাজে (*laaje*)would rhyme.

Also, there is the special case of handling 'ো' (the vowel *o*). In most cases, when the last character in a Bangla word is a consonant, they have an implied *o* sound. This is kind of the reverse of the *inherent vowel suppression* in Hindi (Kishore and Black, 2003). Hence, words like দেহো (*deho*) and কেহ (*keho*) would rhyme. Thus, if one of the last character is a consonant, we need to check if the other word ends in 'ো'.

4.3 Document Statistics

The Document Statistics module basically takes as input a poem, and returns its orthographic features by counting the number of characters, words and stanzas. It also returns the tf-idf scores of the words.

4.4 Shallow Parser

The shallow parser gives the analysis of a sentence in terms of morphology, POS tagging, chunking, etc. We use the POS tags as features in our classification. The shallow parser for Bangla from IIIT Hyderabad has been used.

4.5 SVM

A Support Vector Machine (SVM) classifier was used for classification (Vapnik, 1998). based on the idea of learning a linear hyperplane from the training set that separates positive examples from negative examples. The hyperplane must be at the maximum distance possible from data instances of either class in order to obtain the best generalization. The SVM implementation of SVMLight was used for our experiments (Joachims, 1999).

5 Experimental Setup

For classification of poems into various categories, a bag of words model was trained using only lexical features. Five-fold cross-validation was done on 1341 poems, for training and testing. A linear kernel was used.

We crawled data from the website of *The Complete Works of Tagore* [1] to collect poems by Rabindranath Tagore in four categories: *pooja, prem, prokriti*, and *swadesh*. The data statistics are shown in Table 3.

For the poet identification task, we crawled data from the website of *Bangla Kobita* [2] by

[1] http://tagoreweb.in/
[2] http://www.bangla-kobita.com/

Category	Number of Poems
Pooja (Devotional)	617
Prem (Love)	395
Prokriti (Nature)	283
Swadesh (Nationalism)	46
Total	1341

Table 3: Data

four poets: *Rabindranath Tagore, Jibanananda Das, Kazi Narul Islam*, and *Sukumar Roy*. The data statistics are shown in Table 4.

Poet	Number of Poems
Rabindranath Tagore	382
Jibanananda Das	348
Kazi Nazrul Islam	198
Sukumar Roy	130
Total	1058

Table 4: Data

We trained a Multiclass SVM classifier with a linear kernel for poet identification, using just lexical features (Model-lex) and using lexical as well as stylometric features (Model-lex+style). Five-fold cross-validation was done on the 1058 poems.

6 Results

The results for subject-based poem classification have been tabulated in Table 5 in terms of Precision, Recall and F-measure. The class *pooja* has the best score, and lowest score is for *swadesh*. The confusion matrix has been shown in Table 6. The precision for *swadesh* is high, but the recall is very low, which means instances of *swadesh* are often predicted to be of some other class. The overall performance is 56.8%.

The results for poet identification are shown in Table 7. We compare the results from the SVM classifier, with a Naive Bayes Classifier, in terms of lexical as well as stylistic features. The SVM trained on both lexical and stylistic features was found to have the best performance. When using a Multiclass SVM for classification, introducing stylistic features helped improve the overall performance by 2.2%.

Class	P	R	F-measure
pooja	73.6	84.3	78.6
prem	58.9	55.4	57.1
prokriti	61.9	53.3	57.3
swadesh	83.3	21.7	34.4

Table 5: Results for Poem Classification

	pooja	prem	prokriti	swadesh
pooja	521	71	24	2
prem	110	219	66	0
prokriti	56	76	151	0
swadesh	27	6	3	10

Table 6: Confusion Matrix

Model	P	R	F-measure
Naive-Bayes-lex	90.3	89.2	89.5
Naive-Bayes lex+style	91.0	90.1	90.4
SVM-lex	92.0	87.9	89.9
SVM-lex+style	91.4	93.2	92.3

Table 7: Results for Poet Identification

In Table 8, we tabulate the effect of using various types of stylistic features on the prediction task. The syntactic features alone helped increase the performance by 1.2% over lexical features. Introducing orthographic and phonemic features further increased the performance slightly, by 0.5% and 0.7%, respectively.

7 Analysis and Discussion

From the confusion matrix in poem classification (Table 6), we observe that *swadesh* is often confused with *pooja*. A closer inspection of the poems from the category *swadesh* reveals that the presence of words like জপমালা (*japmala*), পবিত্র (*pobitro*), তীর্থ (*teertho*), etc., which mean *rosary*, *holy*, *pilgrimage*, respectively, might have caused the misclassification. One might note that in these poems, the words of worship such as pilgrimage, rosary, etc., have been used in the context of worship of one's motherland, and hence actually belong to the category *swadesh* or nationalism. On the other hand, in poems from *pooja* misclas-

Features	P	R	F-measure
lex+syn	91.2	92	91.1
lex+syn+orth	91.4	92.5	92
lex+syn+orth +phonemic	91.4	93.2	92.3

Table 8: Effect of various stylistic features

sified as *swadesh*, words like আঘাত (*aaghaat*), ,ভয় (*bhoy*), which mean *hit* and *fear*, respectively, might have caused the misclassification. Similarly, *prem* is most often confused with *pooja*, while *prokriti* is most often confused with *prem*.

Category	Most useful words
Pooja (Devotional)	*hridoy*(heart), *jibon*(life), *gobhir*(deep), *anando*(joy), *alo*(light), *alok*(light), *dhulo*(dust)
Prem (Love)	*sokhi*(friend), *hridoy*(heart), *pran*(life), *haashi*(smile), *madhur*(sweet), *nayan*(eyes), *aakulo*(eager), *aakhi*(eyes)
Prokriti (Nature)	*akash*(sky), *megh*(cloud), *hawa*(breeze), *phool*(flower), *baanshi*(flute), *gagan*(sky), *chhaaya*(shadow)
Swadesh (Nationalism)	*poth*(road), *bangla*(Bangla), *jaagrat*(awake), *bhai*(brother), *bharat*(India)

Table 9: Most distinguishing words from each category

Table 9 shows the most useful word features in identifying each category of poems.

It is observed that lexical features are very useful for poet identification, as poets often have a tendency to use the same set of or similar words. Stylistic features help only to a small extent, particularly, orthographic and phonemic features vary a lot across poems by the same poet, and hence are not much of a distinguishing feature in identifying the poet.

8 Conclusion and Future Work

We conducted what we presume to be the first reported computational analysis of Bangla poetry. With some preliminary investigation, we observed that words alone aren't always sufficient for classifying poems into categories, because of poets often resorting to symbolism. It would be interesting to further investigate if this problem could be helped with Word Sense Disambiguation (WSD). We were able to determine the poet correctly 92.3% of the the time using the SVM classifier. The set of lexical and stylometric features could also be used to categorise poems into ones written by professional and amateur poets, which could throw some light on poetry appreciation, like (Kao and Jurafsky, 2012). The phonemic features could be further enhanced with checking of presence of rhyming words in the same line as also checking for style where each line in a poem begins with the same word. For example: অনেক কীর্তি, অনেক মূর্তি, অনেক দেবালয (*onek keerti, onek smriti, onek debalaya*). The phonemic features may also be extended to detect metre and prosody (Dastidar, 2013), involving syllabification of the verse.

References

Binoy Barman. 2011. A contrastive analysis of English and Bangla phonemics. *Dhaka University Journal of Linguistics*, 2(4): 19–42.

Tanmoy Chakraborty and Sivaji Bandyopadhyay. 2011. Inference of Fine-grained Attributes of Bengali Corpus for Stylometry Detection. *Polibits*, Vol(44): 79–83. Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo.

Tanmoy Chakraborty and Sivaji Bandyopadhyay. 2010. Identification of reduplication in Bengali corpus and their semantic analysis: A rule-based approach. *NAACL Workshop on Computational Linguistics for Literature*.

Suprabhat Das and Pabitra Mitra. 2011. Author Identification in Bengali Literary Works. *Pattern Recognition and Machine Intelligence*, Vol(6744): 220–226. Springer Berlin Heidelberg.

Rimi Ghosh Dastidar. 2013. Symmetry in Prosodic Pattern of Rhyme and Daily Speech: Pragmatics of Perception. *Mining Intelligence and Knowledge Exploration*, 823–830. Springer.

Noraini Jamal, Masnizah Mohd and Shahrul Azman Noah. 2012. Poetry classification using support vector machines. *Journal of Computer Science*, Vol 8(9):1441.

T. Joachims. 1999. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, 169–184 MIT Press. Cambridge, MA.

David M. Kaplan and David M. Blei. 2007. A computational approach to style in american poetry. *In Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on.* 553–558. IEEE.

Justine Kao and Dan Jurafsky. 2012. A computational analysis of style, affect, and imagery in contemporary poetry. *In Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature. Montreal, Canada*, 8–17.

S. Prahallad Kishore and Alan W. Black. 2003. Unit size in unit selection speech synthesis. *INTERSPEECH.*

Noraini Jamal, Masnizah Mohd and Shahrul Azman Noah. 2004. Poetry stylistic analysis technique based on term connections. *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, Vol(5):2713–2718. IEEE.

Andres Lou, Diana Inkpen, and Chris Tanasescu. 2015. Multilabel Subject-Based Classification of Poetry. *The Twenty-Eighth International Flairs Conference.*

Shannon Wiltsey Stirman and James W. Pennebaker. 2001. Word use in the poetry of suicidal and nonsuicidal poets *Psychosomatic Medicine*, Vol 63(4):517–522. LWW.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. 1998. *Statistical learning theory*, Vol 1. Wiley New York.

Rob Voigt and Dan Jurafsky. 2013. Tradition and Modernity in 20th Century Chinese Poetry. *23rd International Conference on Computational Linguistics.*

Bengali Shallow Parser http://ltrc.iiit.ac.in/analyzer/bengali/. *LTRC, IIIT Hyderabad.*