Word Sense Disambiguation of Bengali Words using FP-Growth Algorithm

Mohammad Shibli Kaysar*¶, Md. Asif Bin Khaled†‖, Mahady Hasan‡‖, and Mohammad Ibrahim Khan§¶
¶Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Chittagong, Bangladesh
‖Department of Computer Science and Engineering
Independent University, Dhaka, Bangladesh
*shibli.kaysar@gmail.com, †mdasifbinkhaled@gmail.com, ‡mahady@iub.edu.bd, §muhammad_ikhancuet@yahoo.com

***Abstract*—Word Sense Disambiguation (WSD) is the task of determining the specific meaning of an ambiguous word according to the context. In the realm of natural language processing, WSD is an open problem, and its development can significantly assist in human-level machine translation. In this paper, we have proposed a system for Bengali Word Sense Disambiguation through the FP-Growth algorithm. Moreover, we have also implemented the Apriori algorithm and presented an analysis on both of them to explain how the FP-Growth algorithm outperforms the Apriori algorithm performing Bengali Word Sense Disambiguation. Furthermore, In the testing phase we have found that for 80% of the test sentences, our proposed method can retrieve the exact meaning of an ambiguous word.**

***Index Terms*—natural language processing; word sense disambiguation; fp-growth algorithm; apriori algorithm; association rule**

I. Introduction

One of the significant problems of Bengali language is its ambiguity. A distinct Bengali word can have different meaning depending on the context of the sentences, for example, "তিনি সমাজ এর মাথা" In this sentence the word "মাথা" means"প্রধান" and it translates to English as "main" but "মাথা" in the sentence "তার অঙ্কে মাথা ভালো" means "দক্ষ" and this translates to English as "skilled." As a result, it's tough for a machine to understand the actual meaning of a word in a particular sentence. In order to abolish this problem, many word sense disambiguation approach has been adopted such as knowledge-based, semi-supervised, supervised and unsupervised. Fundamentally, word sense disambiguation is a method of extracting the original meaning of a word according to the context by analyzing the sentence [1]. Comprehending the meaning of an ambiguous word by analyzing a sentence through its context is inherent to human beings, but at the same time, it is difficult for a machine to do such analysis. Moreover, the complexity in the grammatical structure of the Bengali language made it more challenging for the traditional methods used in other languages like English. English has a vibrant online lexical reference system like WordNet [2] which made it convenient to perform natural language processing on it, however in the Bengali language it challenging to find such abundant resources. In the field of data mining frequent pattern mining is a significant approach to figure out which items are often seen together in the dataset [3]. WSD was done using the Apriori algorithm in [4] to figure out the semantics of an ambiguous word according to the context. In our model, we used the Frequent Pattern Growth (FP-Growth) algorithm for the first time to perform WSD. We have developed a Bengali dataset, and in each instance of the dataset, we have appended the semantic of the ambiguous word of that instance. During the training phase, the FP-Growth algorithm creates a frequent pattern tree (FP-Tree) and eventually establishes some association rules. Subsequently, in the testing phase, we have tested our model with test sentences and tried to find out the meaning of the ambiguous word in those test sentences.

II. Related Work

Word Sense Disambiguation on the Bengali language has not yet been research that adequately. However, researchers are showing their growing interest to work with the Bengali language. Besides, in the past few years, a couple of resources for the Bengali language have been developed for instance IndoNet [4], a multilingual lexical knowledge base, which has seized the attention of researchers even further. Pal et al. [5] used a Naive Bayes probabilistic classifier to classify sentences. They used nouns for their experiment, but they did not apply the classifier for verbs. In another paper, Pal et al. [6] also used the Naive Bayes approach but here they subdivided their task into two stages. In the first stage, they did disambiguation using a Bengali corpus, and that gave them around 80% accuracy and the second stage gave them an accuracy of 85%. Moreover, both the training and the test data were lemmatized. A dictionary based approach has been demonstrated to perform WSD on the Bengali language by Haque et al. [7], where they showed two significant steps for their model which are parsing and

978-1-5386-9111-3/19/$31.00

detection. However, they only used nouns, adjectives, and verbs in their experiment. Saiba et al. [8] used the Naive Bayes algorithm and Artificial Neural Network to perform WSD on Bengali sentences, where they used a statistical approach. We have used a data mining approach like Kaysar et al. [9], where they used the Apriori algorithm to apply word sense disambiguation in the Bengali language to remove ambiguity.

III. Methodology

The central objective of our system is to devise an approach that can recognize the meaning of an ambiguous Bengali word in sentences according to the context. Below we provide the flow diagram of our proposed word sense disambiguation system in Fig. 1

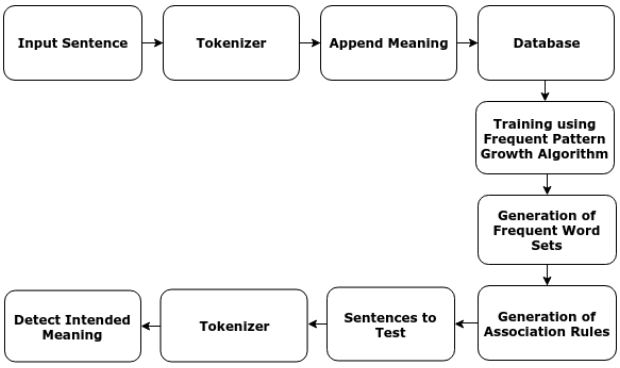

Fig. 1. Flow Diagram of Our System

A. Tokenization

The tokenizer program takes a sentence as an input and then it splits the line into multiple tokens. Each sentence that we are taking as input has been labeled, with the ambiguous word of that sentence along with its meaning according to the context. For example, if "তিনি সমাজ এর মাথা" is an input sentence then the tokenizer program will split this line into "তিনি", "সমাজ", "এর" and "মাথা". When we generated all of the tokens, we appended the meaning of the ambiguous word along with the tokens and then inserted all of the tokens in the database as a separate line. This process was performed for all of the input sentences.

B. FP-Growth Algorithm for WSD

The FP-Growth algorithm identifies compelling links in a massive amount of datasets in the form of frequent items, which generally appear together and association rules, which symbolizes that there exists a strong connection between two set of items [10]. The database that we have prepared after the tokenization process contains all the tokens in a comma-separated manner. From the dataset, we have picked all the rows that have been labeled with identical meaning and ran the FP growth algorithm on each portion separately. A small part with the same meaning is shown in table I below. The second column enlists all the tokens generated from the sentences. The third column represents the ambiguous word, and the last column renders the intended meaning.

TABLE I
SAMPLE DATASET

Serial	Tokens	Ambiguous Word	Intended Meaning
1	আমি,আমার,দেয়া,কথা,রাখবো	কথা	অঙ্গীকার
2	তুমি,আমাকে,দেয়া,কথা,রেখেছো	কথা	অঙ্গীকার
3	কথা,রাখার,জন্যে,তোমাকে,ধন্যবাদ	কথা	অঙ্গীকার
4	সে,আমাকে,দেয়া,কথা,রাখবে	কথা	অঙ্গীকার
5	তুমি,আমার,কথা,রাখবে	কথা	অঙ্গীকার
6	সে,কথা,দিয়ে,কথা,রাখে	কথা	অঙ্গীকার
7	তুমি,আমাকে,দেয়া,কথা,রাখোনি	কথা	অঙ্গীকার
8	আমি,তোমাকে,কথা,দিলাম	কথা	অঙ্গীকার
9	তুমি,কথা,দিয়ে,কথা,রাখোনি	কথা	অঙ্গীকার
10	সে,আমাকে, দেয়া,কথা,রাখতে,পারলো,না	কথা	অঙ্গীকার

Before starting the FP-Growth algorithm, we have sorted all of the tokens of each sentence. After the sorting procedure, we ran the FP-Growth algorithm where firstly the algorithm generated the frequency table of all the candidate items. We will refer to this table as the candidate itemset. The candidate frequency table is shown in table II.

TABLE II
CANDIDATE FREQUENCY

Candidate Items	Support	Candidate Items	Support
অঙ্গীকার	10	রাখার	1
আমার	2	রাখবে	2
আমি	2	সে	3
কথা	10	দিয়ে	2
দেয়া	5	রাখে	1
রাখবো	1	রাখোনি	2
আমাকে	4	দিলাম	1
তুমি	4	না	1
রেখেছো	1	পারলো	1
জন্যে	1	রাখতে	1
তোমাকে	2	সেটি	2
ধন্যবাদ	1		

We have taken the minimum support threshold as 5 for the sample data, hence only "কথা", "অঙ্গীকার" and "দেয়া" are taken as frequent items because only they have frequencies equal or more than 5. Selected items are shown in table III below.

TABLE III
FREQUENT ITEMS FREQUENCY TABLE

Frequent Items	Support
কথা	10
অঙ্গীকার	10
দেয়া	5

Subsequent to the process as mentioned earlier, we have built an FP-Tree where tokens of many sentences can use an identical path if they follow the same order. Moreover, in that way, we can also count each of the words accurately. Therefore, we have sorted the frequent words in descending

order so that the probability of overlapping prefix increases [11]. Below in Fig. 2 we show the FP-Tree that was created during the process.

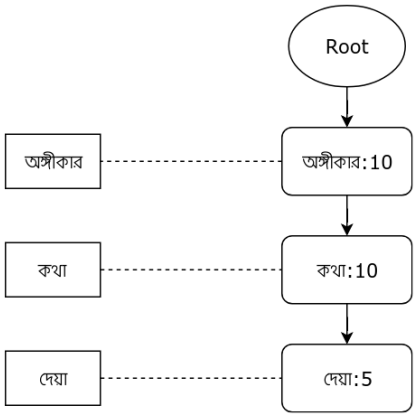

Fig. 2. FP-Tree From The Current Sample Data

When we eventually managed to build the FP-Tree, we started mining the tree by producing all the conditional FP-Trees which was in turns used to find all the frequent word sets. When we managed to generate all the words set, we calculated their support measure, which indicates a sets percentage of availability in the dataset. The formula 1 for the support measure of X concerning the dataset D is,

$$Support\ Measure(X) = \frac{|X \in D|}{|D|} \tag{1}$$

All the frequent word sets along with its support measure are given in table IV.

TABLE IV
SUPPORT MEASURE OF FREQUENT WORD SETS

Frequent Word Set	Support Measure (%)
দেয়া	50
কথা, দেয়া	50
অঙ্গীকার, কথা, দেয়া	50
অঙ্গীকার, দেয়া	50
কথা	100
অঙ্গীকার, কথা	100
অঙ্গীকার	100

After generating all of the frequent word sets along with their support measure, we also produced all of the association rules along with their confidence score. To get strongly associated rules, we have taken a confidence score threshold as high as 90% in our case. The following formula 2 finds the confidence value of a rule $X \rightarrow Y$ concerning the dataset D,

$$Confidence\ Score(X \rightarrow Y) = \frac{Support\ Measure(X \cup Y)}{Support\ Measure(X)} \tag{2}$$

All of the association rules along with their confidence score is given in V.

TABLE V
ASSOCIATION RULES ALONG WITH THEIR CONFIDENCE SCORE

X	Y	Confidence Score (%)
দেয়া	কথা	100
দেয়া	অঙ্গীকার, কথা	100
অঙ্গীকার, দেয়া	কথা	100
কথা, দেয়া	অঙ্গীকার	100
দেয়া	অঙ্গীকার	100
অঙ্গীকার	কথা	100
কথা	অঙ্গীকার	100

Eventually, when we obtained all of the association rules we wrote a script that particularly chooses only those rules where the ambiguous word exists in the premise (X), and the intended meaning exists in conclusion (Y). Hence, the association rules are shortlisted to only two rules which are enlisted below in table VI,

TABLE VI
SHORTLISTED ASSOCIATION RULES ALONG WITH ITS CONFIDENCE SCORE

X	Y	Confidence Score (%)
কথা, দেয়া	অঙ্গীকার	100
কথা	অঙ্গীকার	100

From the information as mentioned earlier, we understand that in the case of new sentences if it contains "কথা" and "দেয়া" then the meaning of "কথা" indicates to "অঙ্গীকার". Furthermore, according to our analysis, we have found out that a more significant number of words on the premise make the rule more suitable for making a decision.

C. Testing Phase

In the Bengali language, we can find a lot of words that have multiple definitions; however, the amount of meaning is limited. The number can be more than two, but it is likely that it is less than twenty. We trained our system for a particular word কথা which generally translates to English as "saying". Although depending on the context, it can have six other meanings. The meaning of the word is dependent on the formation of the words in the sentence. We trained our system with all the seven groups of sentences. Based on that training, the system generates some association rules to define the that indicates the intended meaning of the ambiguous word. In the testing phase, for each unseen sentence, our testing program tokenizes the sentence and generates a powerset of tokens holding no non-empty set. Following that, the program compares each of the subsets with premises of the association rules that we have generated in the training phase. If a subset matches with the association rule then we can conclude that the intended meaning is the conclusion of the association rule. Since we are dealing with only a distinct word which has seven meanings so

using 800 entries gave us a very accurate result. Sample input for the training system is given in table I and sample outcome of the training system is given in table VI. For the testing phase the sample input is given in table VII and the sample output is given in Fig. 3.

TABLE VII
TESTING SAMPLES

Serial	Testing Sample
1	ইচ্ছা করে তুমি তোমার দেয়া কথা রাখোনি
2	মানুষটি সমাজ কে দেয়া তার কথা রাখতে পেরেছে
3	জনগণের কাছে দেয়া কথা জনপ্রতিনিধিদের রাখতে হবে

```
########################################################################
Testing Phase
########################################################################
Sentence Number:  1
['ইচ্ছা', 'কথা', 'করে', 'তুমি', 'তোমার', 'দেয়া', 'রাখোনি']
----------------------------
Detected Rule(s)
['কথা', 'দেয়া'] > ['অঙ্গীকার'] 100.0
----------------------------
Sentence Number:  2
['কথা', 'কে', 'তার', 'দেয়া', 'পেরেছে', 'মানুষটি', 'রাখতে', 'সমাজ']
----------------------------
Detected Rule(s)
['কথা', 'দেয়া'] > ['অঙ্গীকার'] 100.0
----------------------------
Sentence Number:  3
['কথা', 'কাছে', 'জনগণের', 'জনপ্রতিনিধিদের', 'দেয়া', 'রাখতে', 'হবে']
----------------------------
Detected Rule(s)
['কথা', 'দেয়া'] > ['অঙ্গীকার'] 100.0
----------------------------
```

Fig. 3. Output

IV. EVALUATION & RESULT

To calculate how efficient and precise our system is we have split our dataset into two separate sections. We kept 800 out of 1000 sentences in the training file and the remaining 200 sentences in the testing file. Next, we ran the FP-Growth algorithm along with our written script to ascertain the final association rules. Eventually, we tested our system with the testing data. In table VIII we have shown a comparison between both the algorithms.

TABLE VIII
COMPARISON BETWEEN APRIORI ALGORITHM & FP-GROWTH ALGORITHM

Number of Sentences	Apriori Algorithm (s)	FP-Growth Algorithm (s)
100	2	1
200	2.3	1.3
300	2.5	1.6
400	3.3	2
500	4	2.5
600	6	3
700	8.5	4.2
800	12	5

From the table given above, we see that the execution time of the FP-Growth Algorithm increases in a linear manner, however, in the case of the Apriori Algorithm, we see that the increment is exponential [12].

The FP-Growth algorithm uses an FP-Tree where it first stores all sentences by traversing through the entire database however usually it's size is considerably much smaller than the original database [11]. Since there are

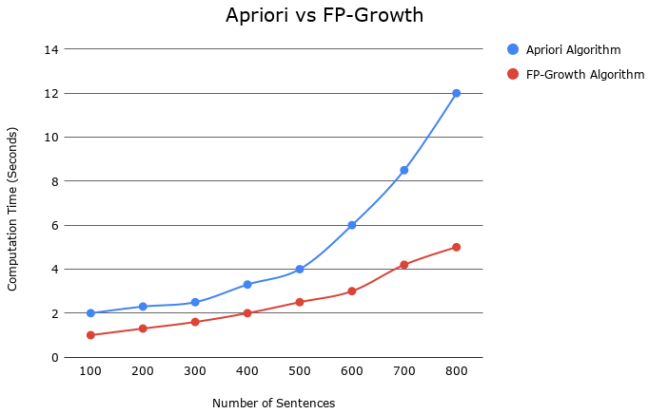

Fig. 4. Comparison Between the Apriori and the FP-growth Algorithm

multiple overlapping patterns in sentences when they are sorted, the tree gets compressed. Nevertheless, the Apriori algorithm generates multiple redundant word sets, which take up a lot of space. During the testing phase, for around 80 percent test sentences, the system could retrieve the intended meaning of the ambiguous word কথা according to context. Both the Apriori and FP-Growth algorithm produces the same rules, but we found out that the FP-Growth algorithm is much quicker.

V. CONCLUSION

In this paper, we have proposed a relatively newer approach to perform Word Sense Disambiguation, which does not depend on lexical or syntactic data. Our system is trained using the FP Growth algorithm which is much faster and memory efficient than the Apriori algorithm. While the accuracy of our system depends on the proportion of the dataset, a higher accuracy rate can be achieved by increasing the number of sentences in the dataset. We are developing a database, which will contain all adequate number of sentences for all the ambiguous words in the Bengali language.

REFERENCES

[1] R. Mitkov, *The Oxford Handbook of Computational Linguistics*, ser. Oxford Handbooks Series. OUP Oxford, 2004. [Online]. Available: https://books.google.com.bd/books?id=yl6AnaKtVAkC

[2] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[3] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[4] B. Bhatt, L. Poddar, and P. Bhattacharyya, "Indonet: A multilingual lexical knowledge network for indian languages," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2013, pp. 268–272.

[5] A. R. Pal, D. Saha, and N. S. Dash, "Automatic classification of bengali sentences based on sense definitions present in bengali wordnet," *arXiv preprint arXiv:1508.01349*, 2015.

[6] A. R. Pal, D. Saha, S. Naskar, and N. S. Dash, "Word sense disambiguation in bengali: A lemmatized system increases the accuracy of the result," in *Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on*. IEEE, 2015, pp. 342–346.

[7] A. Haque and M. M. Hoque, "Bangla word sense disambiguation system using dictionary based approach," 2016.

[8] S. Nazah, M. M. Hoque, and M. R. Hossain, "Word sense disambiguation of bangla sentences using statistical approach," in *Electrical Information and Communication Technology (EICT), 2017 3rd International Conference on*. IEEE, 2017, pp. 1–6.

[9] M. S. Kaysar and M. I. Khan, "Word sense disambiguation for bangla words using apriori algorithm," *International Conference on Recent Advances in Mathematical and Physical Sciences*, p. 61, 2018.

[10] P. Harrington, *Machine Learning in Action*. Greenwich, CT, USA: Manning Publications Co., 2012.

[11] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[12] M. Mythili and A. M. Shanavas, "Performance evaluation of apriori and fp-growth algorithms," *International Journal of Computer Applications*, vol. 79, no. 10, 2013.