

Morphological and Contextual Meaning Analysis in Colloquial Tamil Using Yarowsky Influenced Modified Apriori Algorithm

Thivaharan.S¹, Srivatsun.G²

Submitted: 03/05/2024 Revised: 16/06/2024 Accepted: 23/06/2024

Abstract: Colloquial Tamil, often referred to as "spoken Tamil" or "conversational Tamil," is the informal variant of the Tamil language used in everyday conversations among native speakers. Colloquial Tamil is rich in idiomatic expressions and colloquialisms. The pronunciation in colloquial Tamil can vary significantly from region to region. The use of colloquial Tamil is influenced by social factors such as age, education level, and urban versus rural upbringing. It serves as a marker of identity and belonging within Tamil-speaking communities. Performing morphological analysis is the need of the hour, especially when regional dialect influenced Tamil content is used across social media. Morphological analysis over the informal Tamil content becomes complex as the language has varying meaning for the same word. To arrive at the actual meaning conveyed by the word, it is imperative to consider the coexisting words in the contextual sentence. These co-existing words roots to the prominent word. In this paper, to support and map the actual contextual meaning of the word under investigation, Yarowsky algorithm influenced modified Apriori algorithm is used. This algorithm along with the generated frequent item sets prove to be a suitable fit for the morphological analyzer in determining the actual contextual meaning. Often the frequent datasets are referred to as associative datasets. As per the literature survey, experimental results demonstrate that Apriori based prediction and time complexity excels over the 2-level grammar based approach. This article concludes with the suggestion that prediction accuracy can be further improved, after performing Yarowsky algorithms initial pruning over the associative datasets.

Keywords: Associative datasets, Prominent root words, 2-Level rule mining, Yarowsky algorithm, Modified apriori algorithm.

1. Introduction

In recent days, the necessity to uncover associated underlying knowledge from the unstructured corpus dataset plays a vital role in natural language processing. The extracted frequent data items followed by the featured pattern mining over the final associative datasets produce reliable prediction accuracy with the textual contents. Prominent root words in any given investigative sentence need to be disambiguated to preserve the specific meaning and their context. But prevailing morphological analyzers struggle to produce high accurate predictions especially with the sentences written in regional language. In this article, a system which can disambiguate and find the association among the prominent root words in Tamil word text is proposed using the Yarowsky influenced Modified Apriori algorithm. At the later part of this article, an analysis is done showing how the algorithm improves the fine tuning accuracy compared to the existing models. The frequent data sets ensemble from this approach can be readily included for the rules categorization phase of a particular language. The same approach can be scaled up to variety of regional languages of similar genre.

Through this article, a model capable of grading the

regional contents is proposed. Content grading system takes regional dialectual contents as input and detects copyright violation, pirated content original content and meaningful valid content. Association Rule Mining, the driving force of Apriori algorithm refers to the algorithm that is used to compute the relationship (association) between multiple objects. The algorithm is used to create rules that associate objects that belong to a particular set.

For a rule $X \rightarrow Y$, where X denotes the priori requirement (antecedent), and Y denotes to posterior presence (consequent). In this context the following metrics are defined:

Support is a measure of how frequently an item set appears in a given data set. Support enables to filter out the items with less frequency.

$$\text{Support} = \frac{\text{frequency}(X,Y)}{N} \quad (1)$$

Confidence is a measure of how often the rule $X \rightarrow Y$ has been found to true in the given data set.

$$\text{Confidence} = \frac{\text{frequency}(X,Y)}{\text{frequency}(X)} \quad (2)$$

Lift is a measure of the strength of any rule.

$$\text{Lift} = \frac{\text{Support}(X,Y)}{\text{Support}(X) * \text{Support}(Y)} \quad (3)$$

¹Assistant Professor (Sr. Gr), CSE Dept, KPR Institute of Engineering and Technology, Coimbatore

²Associate Professor, ECE Dept, PSG College of Technology, Coimbatore

¹thivaharan.s@kpriet.ac.in, ²gsn.ece@psgtech.ac.in

In the above equations X is the antecedent, Y is the consequent and N is the total number of observations in the data set.

Apriori algorithm uses frequent item sets to generate association rules. According to the Apriori algorithm, the subset of a frequent itemset will be a frequent item set. A frequent item set is a one for which Support is greater than the Threshold Support Value.

According to the algorithm, for every subset S of itemset I output the rule,

$$S \rightarrow (I - S) \quad (4)$$

$$\{S \text{ recommends } I - S\} \quad (5)$$

If and only if,

$$\frac{\text{Support}(I)}{\text{Support}(S)} \geq \text{Minimum Confidence Value}. \quad (6)$$

Apriori algorithm is used as it uses large itemset property, it can be easily scaled up and it is easy to implement. In this paper, the Apriori algorithm is used to find the closely associated words in dialectal Tamil language by varying the minimum support and confidence value.

2. Related Works

M Bevilacqua et al [1], discussed methodologies and best practices, noting that top models for English WSD approach or exceed human performance, though the task is still unresolved. This necessitates new, challenging benchmarks and further study of models performance in out-of-domain contexts. Multilingual WSD also needs more comprehensive investigation. Integration with Entity Linking and using WSD in other applications like Machine Translation and Semantic Role Labelling are promising future directions. Additionally, WSD could enhance pre-trained language models by grounding word representations in knowledge bases and integrating with other domains.

Wickramasinghe. I et al. [2], explored the applied implications of Naive Bayes (NB) across different segments of regional dialect spoken in western European countries. Naive Bayes performs exceptionally well compared to other classifiers when its assumptions are met and demonstrates robustness with high accuracy even when assumptions are violated. Empirical research done by this author indicates that NB outperforms other techniques, especially with smaller datasets. Despite natural data often not following NB's assumptions, its robustness makes it a reliable choice. The author also discussed about Variations such as semi-Naive Bayes and weighted Naive Bayes. The article aims to provide readers with both theoretical and practical insights into Naive Bayes and Bayesian methods.

D. Yarowsky et al. [3], analyzed several empirically-observed properties of language, notably the strong tendency for words to exhibit only one sense per collocation and discourse. By modelling a rich diversity of collocational relationships, it utilizes more discriminating information than algorithms that treat documents as bags of words, which ignore relative position and sequence. This sensitivity to a broader range of language details is a key strength. Remarkably, for an unsupervised algorithm, it outperforms Schütze's unsupervised algorithm (96.7% vs 92.2%) on a test of the same four words. It also achieves nearly the same performance as a supervised algorithm with identical training contexts (95.5% vs. 96.1%), and even surpasses it in some cases using the one-sense-per-discourse constraint (96.5% vs. 96.1%). This suggests that a large sense-tagged training corpus may not be necessary for accurate word-sense disambiguation.

Akbar Telikani et al [4], surveyed the use of evolutionary computation (EC) techniques in association rule mining (ARM), highlighting that genetic algorithms (GA) and their variants, such as GNP and MOGA, are the most popular due to their effective implementation, despite their lower efficiency compared to other EC methods. The paper discusses various applications, challenges, and the rising trend of hybrid approaches that combine different algorithms, including fuzzy logic and machine learning, to enhance performance. Recent advancements include applying EC algorithms to big data and real-world applications like business, security, and education. It identifies several research gaps, such as optimizing the use of ant bee colony and cuckoo optimization algorithms, hybrid metaheuristic algorithms, scalability issues, and the need for more studies evaluating the effectiveness of different EC algorithms on benchmark datasets.

Saleem Abuleil et al [5], employed natural language processing to generate rules for understanding and structuring Arabic customer reviews. By focusing on adjectives to highlight key information, the approach tags attribute describing review subjects and associates them with their corresponding values. This paper introduced an NLP-based approach to analyze and structure Arabic customer reviews. Future work will involve applying this method to more data, examining cases where adjectives are missing, and focusing on keywords to improve text analysis.

William Croft et al [6], explored the application of Multidimensional Scaling (MDS), specifically the unfolding model, to identify language universals from grammatical variations across different dialects. MDS offers advantages over traditional semantic maps by providing a mathematical framework that interprets distance and dimensionality within a Euclidean spatial

model. The study finds that greater grammatical regularity emerges from diverse datasets, highlighting that language universals are not direct structures but constraints on grammatical variation. Smaller datasets show less regularity, while larger, more diverse datasets reveal clearer patterns. The paper also proposes that language learning involves developing a low-dimensional model of similarities and differences, which helps children approximate grammatical constructs based on increasing exposure to linguistic expressions.

De Gruyter Mouton et al [7], compared two methods for visualizing typological universals of co-expression: graph structure representations and Euclidean space representations. Euclidean space representations, derived through MDS unfolding, are more effective for larger datasets with many concepts, providing a clearer view of generalizations. A key distinction is that graph structure models assume a discrete conceptual space, making them ideal for identifying specific clusters or communities. Instead, they help identify significant semantic clusters and dimensions that can then be used to build coherent conceptual spaces, with anomalies potentially explained through diachronic or other semantic analyses.

Roberto Navegli et al [8], examined the effectiveness of graph connectivity measures for unsupervised word sense disambiguation (WSD). It compares various local and global measures, finding that local measures, specifically Degree and Page-Rank, perform better. This suggests that denser sense inventories with more connections enhance graph-based WSD algorithms. Future work could focus on automatically enriching dictionaries with relatedness information to further improve performance. The study also notes that the connectivity measures are applicable across different graph-based WSD algorithms and can be used with various lexicons and languages. Results indicate that these measures could also benefit other NLP tasks like summarization, though further research is needed to explore their effectiveness in such contexts.

3. Yarowsky Influenced Apriori Algorithm

This proposed Yarowsky Influenced Apriori (YIA) algorithm primarily disambiguates the contextual coagulation [9] among the similar words with similar morphological pattern. From the lexicon collection, an initial graph $G = (V, E)$ is built considering all the target words that make up the sentence S . YIA algorithm assumes that the investigative sentence is pre-POS tagged and only the core grammatical root words are occurring in the sentence. Lexical morpheme available from TransLexGram (Courtesy: Tamil University, Thanjavur and AU-KBC Research Centre, Anna University) project and CIIL (Central Institute of Indian

Languages) corpus are used for the analysis and prediction.

YIA algorithm generates a graph where all the nodes are the word level contexts and the edge between the vertices are all the word level relationship. Given a graph G , a word in a sentence is represented as $W_i \in S$ and the most appropriate context is depicted by $S_{W_i} \in \text{Context}(W_i)$. The context of word senses are obtained from the TransLexGram and CIIL Corpus [10].

3.1. DATA SET PREPARATION

The data set used is obtained from TransLexGam. The data is scraped and the punctuators and other special symbols have been removed. A sentence $S = (w_1, w_2, \dots, w_n)$ is primed for the analysis follows the below mentioned stages:

1. $R = \sum_{i=1}^n \text{Context}(w_i)$ represents all the related relationships in the sentence.
2. To draft the graph, $R_i \in S$, a breadth first search is done over the TransLexGram – CIIL corpus. The resultant graph has to be in any one of the path either $R_i \leftarrow (\text{Approach to } 0)$ or $(R_i \rightarrow (\text{Approach to } \infty))$. All the relationship R_i are inferred from the Edge set $E = E \cup \{ \{N_0, w_0\}, \{N_1, w_1\}, \dots, \{N_n, w_n\} \}$

The data set consists of two files namely, “words.csv” and “sentences.csv”. Sentences.csv comprises of the sentences that have been scrapped and filtered from the CIIL Corpus and words.csv consists of the unique words used in sentences in the sentences.csv. Structure of Sentence.csv:

```
[x1][x3][x5] ...
[x4][x5][x6] ...
```

Each line of the file represents a sentence scrapped from the website. Here $x_1, x_2, x_3 \dots$ represents the unique words present in the data set. Structure of words.csv:

```
[x1]
[x2]
```

Each line of the file comprises of the unique words from the sentence.csv file. Here $x_1, x_2 \dots$ represent the unique words from sentence.csv file.

3.2. FILTERING AND AUGMENTING

Due to the volume of inflectional availability of Tamil words (Informal sense), the need to associate a context marker becomes inevitable for the accurate morphological tagging and further processing. This section proposes the Context Bootstrap Tagging (CBT) [11] to be associated with every w_i in the sentence set S . To start off with the context bootstrapping, root morphemes are identified and represented as a set R

containing r_i , where “i” assumes values as per the available sentence set S. As per the Yarowsky et al, inclusion of n-dimensional context markers will overshoot the timing complexity. So to simplify the process, without complicating, this research work uses the following two context markers: R_1 : same dimensional context matching words, R_2 : higher dimensional context matching words.

Dimension of the context is decided as per the possible inflectional morpheme attachments. For example: the

root word: avan (Tamil Transliterated) can be further followed by “um”, “aa”, “thaanaa”. So the associated dimension of the root “avan” is 3. The existing Yarowsky framework has no dynamic scaling as the dimension grows. The inherent capability of Apriori along with Yarowsky framework supports the increase in dimensional scale up. Considering the following snippet from TransLexGram-CIIL corpus, an attempt to generate the graphical WSD is done. Figure-1 shows the source snippet:

	text
0	ஏற்றுமதியை நம்பியுள்ள நாடுகளுக்குத்தான் இதனால்...
1	* "ரயிலில் தொங்கிக் கொண்டே சண்டையிடுவது போன்ற ...
2	சௌந்தர்ய லஹரி, லலிதா சகஸ்ரநாமம், குகத்தம், பூ...
3	'கடவுளே... என் சிவனே... எனக்கு நல்ல குருவை அட...
4	'அப்பா' படம் வசனங்களாலேயே நகர்கிறது. காட்சிப்ப...

Fig-1: TransLexGram-CIIL dataset snippet

The above dataset snippet is fed to the following Yarowsky algorithm and the resultant graphical WSD [12] is shown in Figure-2 and Figure-2 as initial state and attainment state respectively.

Algorithm: Generating tree with relationship / Yarowsky framework

Input: Set of $S = \{w_1, w_2, \dots, w_n\}$ and $R = \{S_{w_1}, S_{w_2}, \dots, S_{w_n}\}$

Output: Two-dimensional matrix of tag length and context marker

Def Sample_Context();

For each word(W) in sentence(S):

Transliteration = nltk.word_WSD(word)

Collocate_i = nltk.morphe_tag(Transliteration)

Regex_1 = nltk.RegEx.Parser(Collocate_i)

Parsed_1 = Regex_1.Parser(Transliteration)

Print(Transliteration)

Parsed_1.draw()

Except Exception use:

Print (String(Regex_1))

Process context()

The above algorithm accepts the sentence set $S = \{w_1, w_2, \dots, w_n\}$ and the relationship set $R = \{S_{w_1}, S_{w_2}, \dots, S_{w_n}\}$. This algorithm generated a two dimensional matrix containing the context tag length and the context marker, such that a smooth graphical WSD is generated. To extract and map the morpheme to the appropriate context marker, the algorithm initially transforms the source word to its equivalent transliterated form and performs the collocation and generates the parse tree. Figure-2 illustrates the initial state and position of the context markers, where the context marker R_1 and R_2 are spread across the 2-D plot area. Segmenting and clustering the similar contextual words under the R_1 , R_2 cluster. Figure-3 illustrates the plot for at the end of CIIL corpus length iterations (i.e. dynamically fetched by the NLTK toolkit), here it is evidently clustered as per the contextual marker, though with some island R_i .

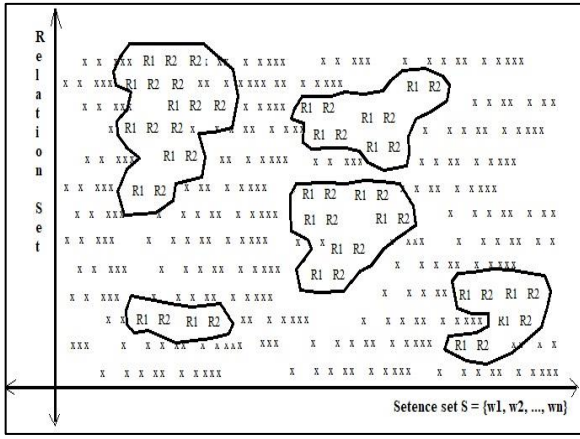


Fig-2: WSD graphical initial state Cluster

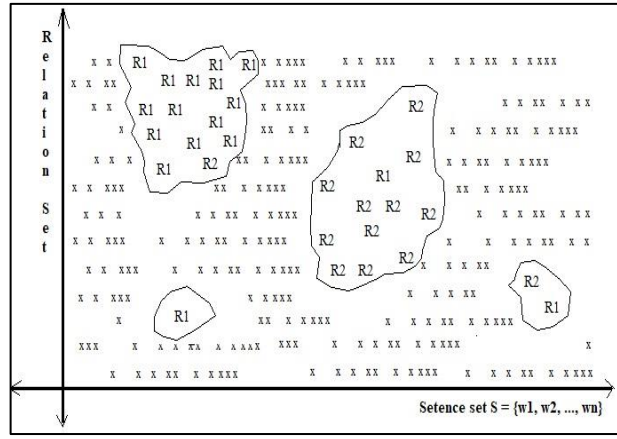


Fig-3: WSD graphical final state cluster

An empirical final decision set [13] is derived for the above mentioned TransLexGram-CIIL dataset and is tabulated in the following Table-1. Here the field values are recorded as vector distribution length, Collocation

word and context marker appropriation. As mentioned the table is a result of log linear iteration run, assuming the vital parameters to its default values.

Table-1: Empirical Final Decision List (Approximated)

Empirical Final Decision List (Approximated)		
Vector Distribution Length	Collocation Word (with Transliterated form) / Informal Sense	Context Marker (Approximated)
24.21	கூட்டம் அலை மோதுது எதுக்காக? (Koottam Alaimothuthu ethukkaaha)	R1
19.86	பாராளுமன்றத்தில் தர்ணா நடக்குது (Paraalumandrathila tharnaa nadakkuthu)	R2
18.64	ஹெல்பேர்ஸ் அங்கேயும் இங்கயுமா நடக்குறாங்க (helpars angayum ingayuma nadakkuraanga)	R1
17.16	நிருபர்களாம் கூட்டமா நிக்கிறாங்க (nirubarhalaam koottamaa nikkiraanga)	R2
17.15	எல்லாரும் கோஷம் போடறாங்க (Ellaarum Gosam Podaraanga)	R2
14.24	கேள்வி நேரத்த டி.வி. ல காட்றாங்க (kelvi Neraththa tivila kaatRaanga)	R2
14.01	நிதியமைச்சர் தொடர்ந்து வாசிக்கறாங்க (nithiyamaichchar thodarndhu vaasikkiraanga)	R2
11.21	எதிர்க்கட்சிய சேர்ந்தவங்க வெளிநடப்பு செய்றங்க (ethirkatchiya serdhthavanga velinadappu seiraanga)	R2
11.05	கூடமா வாக்கு வாதம் நடக்குது (koottama vakuvaatham natakkuthu)	R1
11.01	பார்வையாளர்கள் மேலர்ந்தது பாக்குறாங்க (paarvaiyaalarhal melarndhu paakkuraanga)	R1
09.25	மசோதாவுக்கு வாக்கெடுப்பு நடக்குது (masothaavukku vaakkeduppu nadakkuthu)	R2

4. GRAPHICAL WSD MEASURES

In the context of WSD, it is introduced here the vector length function $V(x,y)$, which by the way deployed for the shortest connecting pairs in the graphical WSD. The

relational distinct pairs are plotted and derived using the function $V(x,y)$ as mentioned below:

$$V(x, y) = \begin{cases} \text{Length of vector in the chosen shortest path,} \\ \text{if } x > y \end{cases} \quad (7)$$

C,
otherwise

In the function above x value is considered and fetched from the R_i set to make inclusive existential path. Otherwise not possible scenarios are enclosed under the conversion constant C, which by the way is decided dynamically case by case, based on the expected outcome threshold.

4.1. DEGREE OF MEDIAN BIAS

As per Wasserman.S et al, maximally connected dense segments should satisfy the degree of median bias. Doing so, confirms the centrality, closeness and the relevance of all other prominent segment in the cluster. The relationship for satisfying the median bias is done as follows:

$$\text{Median}(y) = | \{x, y\} \in R : x \in Y | \quad (8)$$

According to the function, a node with high range in the median outcome is assumed to be in the center of the cluster, thereby providing derivable contextual meaning. The range association of the median value for all the closely related nodes (depiction for the WSSD words) are all brought under the vertex close curve provided context density vector ($C_D(V)$) [14] indication is a positive numeral.

$$C_D(V)) = \frac{\text{Median}(y)}{|V| - 1} \quad (9)$$


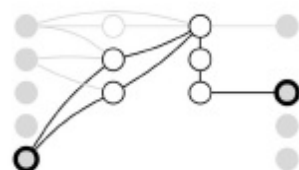
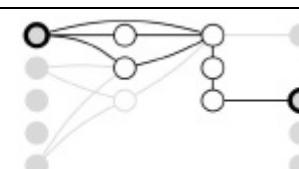
4.2. PAGE RANKING FOR RELEVANCE

Page rank [15] determines the association aptness of the nodes in the graphical WSD. A page rank indicative value is fixed to every edge. One after that, the entire graphical WSD, turns to a dense matrix. To simplify the n^{th} dimensional node relationships, weightage of page rank is done randomly over the pre sampled WSD seeds. Page here refers to the screen of $\langle S, R \rangle$ set, where S denotes the Sentences and R denotes the Relationship. Both of these were fine-tuned by the Yarowsky algorithm. The page rank (PR) is measured using the following relation:

$$\text{Page rank}(S, R) = \frac{(1 - \alpha)}{|V|} + \alpha \sum_{\{x,y\} \in R} \frac{\text{Page rank}(R,S)}{\text{indegree}(x)} \quad (10)$$

In the above equation In-degree refers to the number of incident edges towards a node in the graphical WSD. $|V|$ denotes the vector set, eventually comprising all the S and R. It is evident from the function those repetitive cycles of page rank over the $\langle S, R \rangle$ set tends the decision to zero-in at the position with densely clustered. Table-2 below depicts the interpretations of the median bias, page rank and the cluster density [16].

Table-2: Correlation among Median Bias, Page Rank and Cluster Density

S. No	Graphical WSD	Median Bias	Page Rank	Cluster Density
1		1.85	0.64 (Low)	0.52
2		1.90 (High)	0.92	0.61 (Low)
3		1.72 (Low)	1.02 (High)	0.62

4		1.73	0.88	0.69 (High)
---	--	------	------	----------------

The values above are indicative that maximal numeral is also spread across the rows. Median bias is high for the graphical WSD no.2. Page Rank is maximal for the graphical WSD no.3 and cluster density is high for the graphical WSD no 4.

5. Result Analysis and Complexity Measures

As the continuation to the section 3.1 in this article, the data from the file sentence.csv is read using the csv module. The file consists of 53,632 selected sentences from the TransLexGram-CIIL corpus. Initially the data was cleaned by removing the punctuation marks, numbers and additional white spaces. Then all the sentences were iterated to identify the unique words present in the dataset. It was found that there were 30120 unique words in the data set. A nested list was created in the following format.

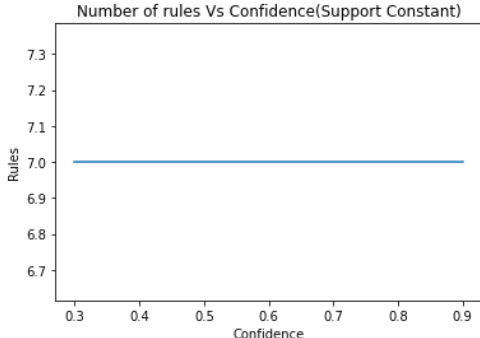
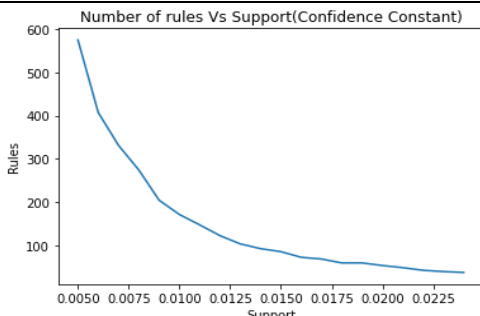
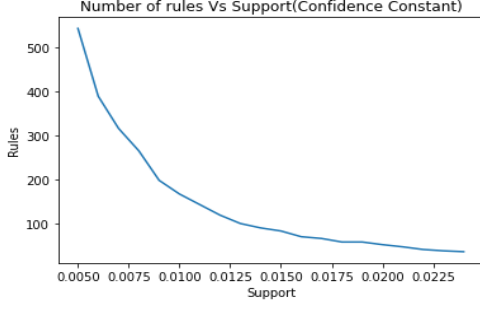
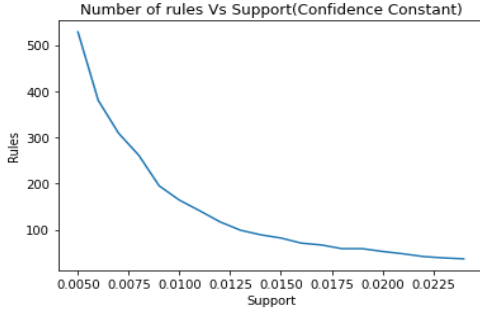
[[word₁, word₂, word₃, ..., word_n], [word₁, word₂, word₃, ..., word_n], ...]

Here the inner lists represent the sentence. Suppose if a word is missing that word is replaced with an empty string. All the sentences after performing cleaning and converting it into the required format were stored in the file data.csv. This nested list created is given as a parameter to the Apriori module that implements the Apriori algorithm to find the closely related words in Tamil language and returns the rules generated as output. The values of minimum support [17] and confidence [18] were varied and the results were observed.

Based on the above analogy three varied trails were done and the outcomes are documented in the following table-3.

Table-3: Train runs for the Support vs. Constant during Apriori estimation

S. No	Trial Description	Rules vs. Confidence	Minimal threshold																
Keeping support constant and varying confidence																			
1	Support = 0.005	<table><thead><tr><th>Minimum Confidence</th><th>Inflectional deviation</th></tr></thead><tbody><tr><td>0.3</td><td>575</td></tr><tr><td>0.4</td><td>557</td></tr><tr><td>0.5</td><td>544</td></tr><tr><td>0.6</td><td>534</td></tr><tr><td>0.7</td><td>528</td></tr><tr><td>0.8</td><td>522</td></tr><tr><td>0.9</td><td>515</td></tr></tbody></table>	Minimum Confidence	Inflectional deviation	0.3	575	0.4	557	0.5	544	0.6	534	0.7	528	0.8	522	0.9	515	
Minimum Confidence	Inflectional deviation																		
0.3	575																		
0.4	557																		
0.5	544																		
0.6	534																		
0.7	528																		
0.8	522																		
0.9	515																		
2	Support = 0.01	<table><thead><tr><th>Minimum Confidence</th><th>Inflectional deviation</th></tr></thead><tbody><tr><td>0.3</td><td>171</td></tr><tr><td>0.4</td><td>169</td></tr><tr><td>0.5</td><td>168</td></tr><tr><td>0.6</td><td>166</td></tr><tr><td>0.7</td><td>164</td></tr><tr><td>0.8</td><td>164</td></tr><tr><td>0.9</td><td>164</td></tr></tbody></table>	Minimum Confidence	Inflectional deviation	0.3	171	0.4	169	0.5	168	0.6	166	0.7	164	0.8	164	0.9	164	
Minimum Confidence	Inflectional deviation																		
0.3	171																		
0.4	169																		
0.5	168																		
0.6	166																		
0.7	164																		
0.8	164																		
0.9	164																		

3	Support = 0.07	<div>Number of rules Vs Confidence(Support Constant)</div> 	<table><tr><th>Minimum Confidence</th><th>Inflectional deviation</th></tr><tr><td>0.3</td><td>7</td></tr><tr><td>0.4</td><td>7</td></tr><tr><td>0.5</td><td>7</td></tr><tr><td>0.6</td><td>7</td></tr><tr><td>0.7</td><td>7</td></tr><tr><td>0.8</td><td>7</td></tr><tr><td>0.9</td><td>7</td></tr></table>	Minimum Confidence	Inflectional deviation	0.3	7	0.4	7	0.5	7	0.6	7	0.7	7	0.8	7	0.9	7
Minimum Confidence	Inflectional deviation																		
0.3	7																		
0.4	7																		
0.5	7																		
0.6	7																		
0.7	7																		
0.8	7																		
0.9	7																		
Keeping confidence constant and varying support																			
4	Confidence=0.3	<div>Number of rules Vs Support(Confidence Constant)</div> 	<table><tr><th>Minimum Support</th><th>Inflectional deviation</th></tr><tr><td>0.005</td><td>575</td></tr><tr><td>0.010</td><td>171</td></tr><tr><td>0.015</td><td>85</td></tr><tr><td>0.020</td><td>53</td></tr><tr><td>0.024</td><td>37</td></tr></table>	Minimum Support	Inflectional deviation	0.005	575	0.010	171	0.015	85	0.020	53	0.024	37				
Minimum Support	Inflectional deviation																		
0.005	575																		
0.010	171																		
0.015	85																		
0.020	53																		
0.024	37																		
5	Confidence=0.5	<div>Number of rules Vs Support(Confidence Constant)</div> 	<table><tr><th>Minimum Support</th><th>Inflectional deviation</th></tr><tr><td>0.005</td><td>544</td></tr><tr><td>0.010</td><td>168</td></tr><tr><td>0.015</td><td>84</td></tr><tr><td>0.020</td><td>55</td></tr><tr><td>0.024</td><td>39</td></tr></table>	Minimum Support	Inflectional deviation	0.005	544	0.010	168	0.015	84	0.020	55	0.024	39				
Minimum Support	Inflectional deviation																		
0.005	544																		
0.010	168																		
0.015	84																		
0.020	55																		
0.024	39																		
6	Confidence=0.7	<div>Number of rules Vs Support(Confidence Constant)</div> 	<table><tr><th>Minimum Support</th><th>Inflectional deviation</th></tr><tr><td>0.005</td><td>528</td></tr><tr><td>0.010</td><td>164</td></tr><tr><td>0.015</td><td>82</td></tr><tr><td>0.020</td><td>53</td></tr><tr><td>0.024</td><td>37</td></tr></table>	Minimum Support	Inflectional deviation	0.005	528	0.010	164	0.015	82	0.020	53	0.024	37				
Minimum Support	Inflectional deviation																		
0.005	528																		
0.010	164																		
0.015	82																		
0.020	53																		
0.024	37																		

On keeping the support constant, and varying the confidence we observe that the number of rules decreases with increase in confidence in the first two trials whereas in third trial the number of rules remains constant. This observation of the third dataset may be due to the nature of words present in the dataset. Thus, there is an inverse relationship between the number of pairs of closely associated words in formal Tamil language and the probability of their occurrence in the dataset.

On keeping the confidence constant, and varying the support we observe that the number of rules decreases with increase in support in all the three trials. Thus, there is an inverse relationship between the number of pairs of closely associated words in formal Tamil language and the frequency of their occurrence in the dataset.

Complexity measurement of this proposed Yarowsky Influenced Apriori algorithm is done in a sentence-by-sentence case basis. While considering the complexity,

disambiguation is not considered to ease the analysis. Considering the graphical WSD components as G_σ boundary marking [19] is measured through the following relation:

$$|G_\sigma| = \sum_{i=1}^n |Context(w)| < \sum_{i=1}^n RelationSet(S) \quad (11)$$

With a local measure, the median bias and page rank calculation requires the entire TransLexGram-CIIL corpus to be utilized. Thus resulting in $O(n)$ timing complexity if Yarowsky algorithm alone is used. By combining the Apriori algorithm with modified confidence and support consideration, the morphological prediction can be obtained in $O(n \cdot \log n)$ complexity. But still exhaustive and exponential data inclusion [20] may disrupt the above mentioned timing complexity.

6. Conclusion

The associative rule mining algorithm along with Yarowsky can be used to figure out the closely associated words/ phrases in Tamil language. The associated words on identification will aid the systems that work on spam detection, and those which work on identifying plagiarized content. The associative rule mining algorithm consumes long time to generate the closely associated words. This is due to the size of the dataset. The speed of generation can be implemented by using parallel processing technique. Also a general rule set of the words in Tamil language can be generated by using wide range of data from books, articles, magazines and websites in Tamil language. This rule sets can be further refined by using the principal component analysis and a corpora of closely associated words can be generated.

It is found that the reference dictionary used has a significant impact on WSD performance. By using a version of TransLexGram-CIIL enriched with thousands of relatedness edges, probabilistic indices has significantly improved as close to 0.78 (where 1 indicates the full closeness). This suggests that graph-based WSD algorithms perform better with more densely connected sense inventories, where each node has more incident edges. A promising future direction would be to explore methods for automatically enhancing universally available lexicon with relatedness information, such as adding edges to the graph for nodes whose distributional similarity surpasses a certain threshold.

ACKNOWLEDGEMENT

The authors thank the efforts and research contributions from various language researchers and the repositories, helping by the way of supplying corpus during the test run. Especially the corpus and detailed description provided from Tamil University, Thanjavur and AU-KBC Research Centre, Anna University.

References

- [1] Bevilacqua, Michele, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. "Recent trends in word sense disambiguation: A survey." In International Joint Conference on Artificial Intelligence, pp. 4330-4338. International Joint Conference on Artificial Intelligence, Inc, 2021.
- [2] Wickramasinghe, Indika, and Harsha Kalutarage. "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation." *Soft Computing* 25, no. 3 (2021): 2277-2293.
- [3] Yarowsky, David. "Unsupervised word sense disambiguation rivaling supervised methods." In 33rd annual meeting of the association for computational linguistics, pp. 189-196. 2015.
- [4] Telikani, Akbar, Amir H. Gandomi, and Asadollah Shahbahrami. "A survey of evolutionary computation for association rule mining." *Information Sciences* 524 (2020): 318-352.
- [5] Abuleil, Saleem, and Khalid Alsamara. "Using NLP approach for analyzing customer reviews." *SOEN—2017* (2017): 117-124.
- [6] Croft, William, and Keith T. Poole. "Inferring universals from grammatical variation: Multidimensional scaling for typological analysis." (2018): 1-37.
- [7] Croft, William. "On two mathematical representations for "semantic maps".*" Zeitschrift für Sprachwissenschaft* 41, no. 1 (2022): 67-87.
- [8] Navigli, Roberto, and Mirella Lapata. "An experimental study of graph connectivity for unsupervised word sense disambiguation." *IEEE transactions on pattern analysis and machine intelligence* 32, no. 4 (2019): 678-692.
- [9] Elkin, Peter L., Sarah Mullin, Jack Mardekian, Christopher Crowner, Sylvester Sakilay, Shyamashree Sinha, Gary Brady et al. "Using artificial intelligence with natural language processing to combine electronic health record's structured and free text data to identify nonvalvular atrial fibrillation to decrease strokes and death: Evaluation and case-control study." *Journal of medical Internet research* 23, no. 11 (2021): e28946.
- [10] Baskaran, S., and S. Thiagarajan. "Tamil WordNet S. Rajendran, S. Arulmozi B. Kumara Shanmugam." In 1st International Global WordNet Conference, January 21-25, 2002: Proceedings, p. 271. Central Institute of Indian Languages, 2020.

- [11] Clark, Stephen, James R. Curran, and Miles Osborne. "Bootstrapping POS-taggers using unlabelled data." In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, pp. 49-55. 2013.
- [12] Sidorov, Grigori, and Francisco Viveros-Jiménez. "One sense per discourse heuristic for improving precision of WSD methods based on lexical intersections with the context." *POLIBITS* 57 (2018): 45-50.
- [13] Lombardi, Michele, Michela Milano, and Andrea Bartolini. "Empirical decision model learning." *Artificial Intelligence* 244 (2017): 343-367.
- [14] Byrd, Mark S., and Navin Khaneja. "Characterization of the positivity of the density matrix in terms of the coherence vector representation." *Physical Review A* 68, no. 6 (2013): 062322.
- [15] Wachsmuth, Henning, Benno Stein, and Yamen Ajjour. "'PageRank' for argument relevance." In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 1117-1127. 2017.
- [16] Hohn, Nicolas, Darryl Veitch, and Patrice Abry. "Cluster processes: a natural language for network traffic." *IEEE Transactions on Signal processing* 51, no. 8 (2021): 2229-2244.
- [17] Dahbi, Azzeddine, Youssef Balouki, and Taoufiq Gadi. "Using multiple minimum support to auto-adjust the threshold of support in apriori algorithm." In International Conference on Soft Computing and Pattern Recognition, pp. 111-119. Cham: Springer International Publishing, 2017.
- [18] Thivaharan, S., K. Hariharan, and R. Christie Jerin Kumar. "content grading system for Tamil based on indexed set weights using PCKimmo." *International journal of engineering research and technology* 8, no. 3 (2019): 177-181.
- [19] Srivatsun, G., and S. Thivaharan. "Modelling a machine learning based multivariate content grading system for YouTube Tamil-post analysis." *Journal of Intelligent & Fuzzy Systems Preprint* (2023): 1-12.
- [20] Dang, Nhan Cach, María N. Moreno-García, and Fernando De la Prieta. "Sentiment analysis based on deep learning: A comparative study." *Electronics* 9, no. 3 (2020): 483.