Integration of Self-Attention Model with Intralingual Word Embedding for Contextual Semantic Analysis of Thirukkural Text.

[*]Shanthi Murugan[1], [*]Kaviyarasu[1], Balasundaram S R[2]
[1]R.M.K. Engineering College, Thiruvallur, Tamilnadu, India
[2]National Institute of Technology,Tiruchirappalli, Tamilnadu, India
[1]{msi,kavi22022}.ad@rmkec.ac.in
[2]blsundar@nitt.edu

Abstract

Thirukkural, one of the ancient works of Tamil Literature, is popular worldwide due to the moral values and practices it teaches to the society. Understanding the verses with meaning, especially context, is important. In this regard, this paper introduces a system designed to generate contextualized word meanings for the couplets of the Thirukkural, tailored to assist school children in understanding the text more effectively. Unlike traditional methods that provide detailed explanations in paragraph form, our method focuses on word-by-word interpretation, based on context through an integrated self-attention model. By combining the self-attention mechanism with FastText embeddings, our approach achieves improved performance over state-of-the-art models such as Word2Vec and standalone FastText. We evaluate the semantic understanding of the Thirukkural text using metrics as manual scoring. Tamil Thirukkural Agarathi serves as the gold-standard dataset for evaluation, demonstrating the effectiveness of our approach in capturing the nuanced semantics of the Thirukkural.

1 Introduction

Tamil, an ancient language with a rich cultural heritage, continues to thrive in the modern world. Tamil is spoken by approximately 83 million people worldwide, with 79 million first-language speakers (27th edition of Ethnologue published). Tamil is globally renowned for its rich literature and ancient traditions that spans thousands of years. Through classical poetry, art, philosophy and its adaptive nature, Tamil continues to inspire and influence global cultures, making it a vital part of the world's linguistic and cultural diversity. Thirukkural written by Thiruvalluvar, is one of the cornerstones of Tamil language, a classic literature, written in 300 BCE. It has been widely classified into three sections namely Aram, Porul and Inbam[1]. Thirukkural comprises 1330 couplets which are organized into three sections and 133 chapters[2].Each chapter consists of ten kurals under unique topics. Each kural consists of two lines, known as couplets, and contains only seven words. It provides all possible solutions to lead a successful and a peaceful life fitting any generation. It has been translated into 82 global languages[3] including 143 different English versions, allowing its profound insights to reach a global audience.

The popularity of Thirukkural has grown to such an extent that quoting it in speeches, and various other contexts has been happening on various occasions. Thirukkural is now part of the school curriculum across all grades in Tamil Nadu, ensuring that students are exposed to its moral and ethical values from an early age. However, despite its widespread presence, students often struggle to fully grasp the subtle meaning of kural. Because, classical Tamil used in the Thirukkural text differs significantly from contemporary Tamil. To address this issue (difficult to understand the Classical Thirukkural), the couplets are given with paragraph explanations in textbooks.

Due to the widespread popularity of Thirukkural worldwide, computational processing is available in the form of tools, websites and applications. In existing systems, the computational works done on Thirukkural vary on different objectives. Retrieval of specific kural based on the user query has been done using the Multinomial Naive Bayes classifier (Ramalingam et al., 2022). Arjun et al., (2024) focus on retrieving a given English query or keyword using GPT4 and Retrieval-Augmented Generation (Website of KaniTamil24 conference). Use of Rhetorical Structure Theory has been proposed for discourse relation between phrases(Anita and Subalalitha, 2019).

In general the above approaches focus on generating the meaning for the entire couplets. Whereas, word-by-word interpretation (in Modern Tamil) is mainly necessary for a deep understanding of the text especially for children or new readers.

*Both Authors contributed equally to this work

[1]https://en.wikipedia.org/wiki/Kural
[2]https://www.britannica.com/topic/Tirukkural
[3]https://en.wikipedia.org/wiki/Tirukkural_translations

To address this challenge, our work proposes a new approach to make Thirukkural more understandable (i.e. word by word with context) for students, learners and the public. The core design of the work involves two important stages.

1. Intra Lingual Thirukkural Embedding : To get the dense vector representation for Tamil words which includes Thirukkural vocabulary.

2. Self-Attention: To convert static embedding to contextual and dynamic embedding for the query.

2 Related Work

To retrieve word semantics for various downstream tasks, many state-of-the-art systems rely on word embedding models. Among these, highly efficient systems include Word2vec, introduced by Mikolov et al. (2013), is a neural network-based model for learning distributed representations of words. It comes in two architectures: Continuous Bag-of-Words (CBOW) and Skip-gram. Both use a shallow neural network with one hidden layer. Word2vec's key feature is its ability to capture semantic and syntactic relationships between words in vector space, where similar words cluster together. Next, FastText architecture introduced by Bojanowski et al., (2017) is an efficient text classification and word representation model that extends word2vec by treating words as bags of character n-grams. FastText's key innovations are Hierarchical softmax and N-gram embeddings. This approach incorporates subword information, enabling the generation of embeddings for out-of-vocabulary words. These features make FastText particularly effective for languages with rich morphology and texts containing numerous rare words or neologisms, as it can generate meaningful representations for words not encountered during training. However, a key limitation of both Word2vec and FastText is that they extract the meaning of words without considering the context in which the word is used. This results in static word embeddings, where each word has a single vector representation, regardless of the varying meanings it may take in different contexts. This lack of contextual understanding reduces their ability to capture the nuances of meaning that arise from the word's surrounding text.

In other direction, Vulic and Moens (2015) introduce cross-lingual information which is embedded within a single vector space, allowing two different languages to be represented and handled together. This enables the model to map words from different languages into a common semantic space, making it possible to extract meaning across languages and facilitate tasks like cross-lingual information retrieval and translation. By aligning the representations of different languages, the framework allows for more efficient and accurate meaning transfer from one language to another. Whereas our work focuses on different varieties of text within the Tamil language, specifically classical and modern Tamil words. This approach aims to capture the semantic relationships and nuances present in these distinct forms of Tamil, allowing for a more comprehensive understanding and representation of the language's rich linguistic heritage.

Transformer architecture, introduced by Vaswani et al. (2017), deals with the attention mechanism. Self-attention allows the model to weigh the importance of different parts of the sequence when processing each element. The multi-head attention allows the model to attend to information from different representation subspaces. Position encodings are added to input embeddings to retain sequence order information. These methods capture the contextual information of the token.

One of the first discourse parsers for Thirukkural, utilizing Rhetorical Structure Theory (RST) has been introduced to uncover coherence between text spans through discourse relations (Anita and Subalalitha, 2019). Analyzing Thirukkural poses additional challenges due to its irregular syntactic and semantic patterns, rich morphological variations, and unique poetic structure, making it more complex than prose. This approach enhances semantic indexing and retrieval, surpassing simple keyword searches for Thirukkural content.

The work by Ramalingam et al (2022) classified Thirukkural into ten new classes named superclasses, further narrowed down by classifying each superclass into two subclasses. Multinomial naive bayes classifier has been used to retrieve the meaningful thirukkural for a query in the search system. Use of text-to-speech conversion in Tamil especially for Tirukuural is available (Rama et al., 2002).

An e-learning based speech recognition system to recognize and retrieve the meaning of kurals is designed to display meaning with Chapter details (Bharathi et al., 2017) and to synthesize the meaning as speech output. The work mentions challenges such as lack of a standard Thirukkural speech database and building acoustic models for entire kurals rather than words/phonemes. The system is positioned as a tool to help students, visually challenged people, and others. Aalamaram, the largest tree bank in Tamil with nearly 10,000 sentences annotated, is available for the tasks of POS tagging, NER, Morphological Parsing, and Dependency Parsing (Abirami et al.,2024).

In the realm of Thirukkural literature, various tools and applications have been developed to enhance access to the couplets and their meanings. Numerous mobile applications provide the complete text along with translations and interpretations. Additionally, several websites showcase the couplets alongside scholarly commentaries, while digital libraries like Project Madurai and Tamil Virtual Academy offer translations and annotations to enrich users' understanding of this classical work. The aforementioned tools and applications significantly enhance the retrieval of Thirukkural couplets and their explanations; however, they often lack a detailed word-by-word analysis, which is essential for a deeper understanding of the text.

3 Contextualized Semantic Learning for Thirukkural Words using Unsupervised Learning

The vocabulary of the Thirukkural is rooted in classical Tamil, making it challenging for contemporary learners, including both students and people knowing Tamil, to grasp its meaning. Much of the analytical methodologies for language processing are seen for English language and very few exist towards Tamil especially for Thirukkural(Anita et al., 2021). The primary objective of this work is to identify modern, easily understandable Tamil words that correspond to the original terms in Thirukkural, thereby facilitating better comprehension. To achieve this, we employ an unsupervised learning algorithm rather than relying on traditional rule-based approaches or dictionary lookups.

Several state-of-the-art systems exist for retrieving word meanings , utilizing unsupervised learning algorithms such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017). FastText, in particular, has shown enhanced performance in capturing word semantics when compared to both Word2Vec and GloVe. In our proposed approach, we employ FastText word embeddings for Thirukkural word meaning retrieval.

3.1 Training the Intralingual FastText Embedding Model

Thirukkural couplets (Classic Tamil) and their explanations (Contemporary Tamil) with Tamil Indic corpus (Kunchukuttan et al., 2020) have been used to train the FastText embedding model. In general, the FastText model has been trained based on its context (surrounded word relations) to represent the distribution of words in the vector space. Classic Tamil words from literary works often appear infrequently within Tamil corpora and do not easily correlate with the semantic meanings that contemporary Tamil speakers understand. Random shuffling of Thirukkural (classic) and its explanation (modern) has been done to make the model understand and map words from classic Tamil to modern Tamil in vector space.

3.2 Intralingual Representation(Thirukkural and Explanations)

Inspired from the work of Ivan Vulić et al. (2015) in shared interlingual embedding space, we propose the novel work of intralingual embedding space for contextual analysis. We introduce the framework of intralingual representation as follows.

Let C $\{C_1,C_2,C_3,\ldots,C_{1330}\}$ be the Thirukkural couplets and M $\{M_1,M_2,M_3,\ldots,M_{1330}\}$ be their explanations. We define T $\{T_1, T_2, T_3, \ldots, T_{1330}\}$ as corpus, Where T = $\{(C_1,M_1),(C_2,M_2),(C_3,M_3), \ldots ,(C_{1330},M_{1330})\}$. Here, T_i (i.e.) (C_i,M_i) denotes a pair of Couplet(C_i) and its Meaning(M_i). The goal is to learn word embeddings and cluster the similar words which will be semantically coherent over the vector space. We denote Wci as words in Thirukkural couplets and Wmi as words corresponding to

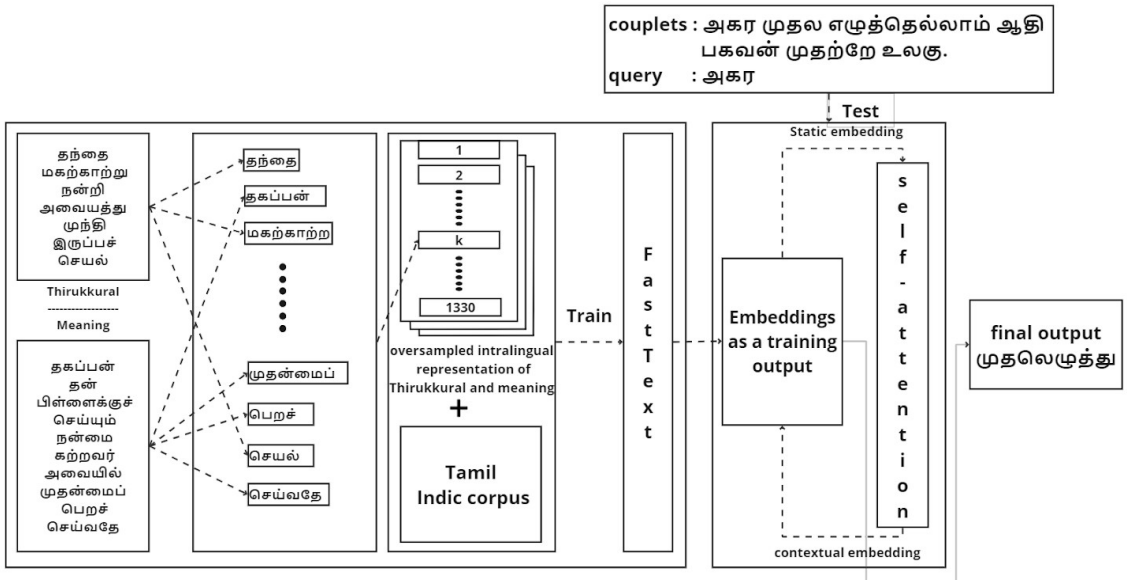

Figure 1: Intralingual Static Embedding with Self Attention to find the Contextual Meaning of Thirukkural Words.

their meanings. In the first step, we merge W_{ci} and W_{mi}. Following that, we randomly shuffle words of the newly constructed sequence without changing the word order in W_{ci} and W_{mi} which is depicted in Figure 1.

3.3 Oversampling

By combining Kural and Porul in a random manner, we generate a unique sequence for each instance. This oversampled dataset enhances the model's robustness, leading to improved performance. Although fastText effectively processes classic Tamil texts using n-grams, it generates only static embeddings, which capture the general characteristics of words rather than their contextual nuances. To address this limitation, we propose the use of self-attention.

3.4 Integrating Self Attention Layer on Top of FastText Model

Self-attention, a key component of transformer architectures (Vaswani et al., 2017), It operates on an input of n tokens, each with an embedding dimension d. Three main components of self-attention include Query (Q), Key (K), and Value (V), Each of Q, K, and V is represented as an [n x d] matrix. The process as presented in Algorithm 1 involves computing attention scores by taking the dot product of Q and K` (Transpose of key), resulting in an [n x n] matrix Attention Score (S). These scores are then scaled by $1/\sqrt{d}$ to mitigate issues in high-dimensional spaces.

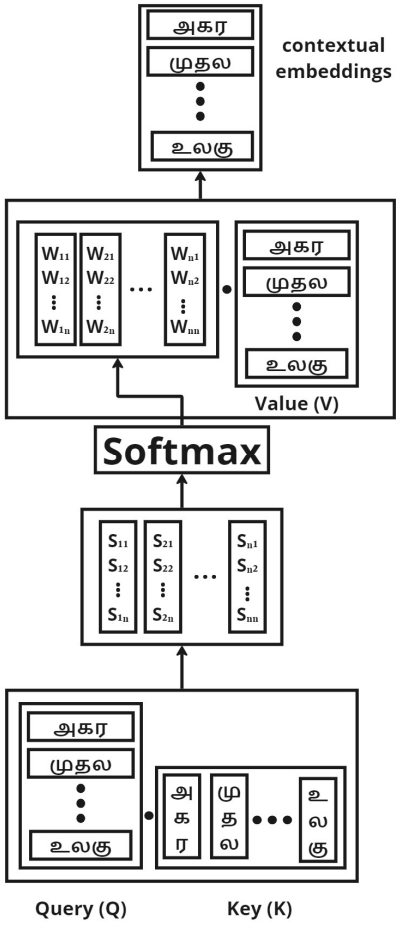

Figure 2 : Self Attention Mechanism to retrieve contextual information. Input is a static embedding and the output will be contextual embeddings

Chapter	Kural No	Query Word	Gold standard	FastText	FastText With self Attention	Score(S)
1	1	அகர	அ என்னும் எழுத்து	அகரம்	முதலெழுத்து	5
54	534	அரண்	காவல், பாதுகாப்பு	கோட்டை	பாதுகாப்பு	5
39	381	அரண்	கோட்டை மதில்	கோட்டை	கோட்டை	4
3	25	அகல்விசும்பு	விரித்த வானம், அகன்ற வானம்	ஆற்றல்மிகு	கல்விமான்	0
3	22	இறந்தாரை	செத்தவரை	இறந்தார்	உயிர்துறந்தாரை	5
13	122	காக்க	பாதுகாக்க	காப்பாற்று	காப்பாற்றுக	3
13	127	காக்க	அடக்குக	காப்பாற்று	காப்பாற்றுவான்	2
79	781	செயற்கரிய	செய்து கொள்வதற்கு கடினமான	செயற்கரிய	செய்யார்	2

Table 1 : Semantic Meaning of Thirukkural Words based on FastText with Self Attention

The scaled scores undergo SoftMax normalization to produce weights (W), The final output, also an [n x d] matrix, is calculated as the matrix product of W and V, providing a contextual embedding for each token. This is mathematically represented in Eq(1).

$$Attention(Q,K,V) = SoftMax(\frac{Q.K^T}{\sqrt{d}}).V \text{–Eq(1)}$$

We are taking Q, K and V from fastText embeddings. this mechanism calculate the contextual embedding of words in Thirukkural using only static embedding.

This approach allows us to capture contextual relationships between words in Thirukkural based on their static representations, The resulting contextual embeddings reflect the importance of each word in relation to others within the same couplet, based on their semantic similarities as captured by embedding model. This has been pictorially represented in Figure2.

Algorithm 1: Procedure for Self-Attention

```
procedure SELF_ATTENTION(Q, K, V, word)
begin
    attention_scores = []
    for key in keys:
        score = dot_product(Q, K) / sqrt(dim(K))
        attention_scores.append(score)
    weights = softmax(attention_scores)
    weighted_sum = initialize with zeros of dim(V[0])
    for i in range(len(weights)):
        weighted_sum += weights[i] * V[i]
    return weighted_sum[wordindex]
end
```

This hybrid approach of self-attention on top of FastText not only preserves the integrity of the classical text but also enhances its applicability and understanding in contemporary Tamil, leveraging the Tamil Indic corpus and self-attention models to enrich the contextual interpretation of Thirukkural words.

4 Experiments

4.1 Datasets

In order to get the meaning for classic Tamil words in Thirukkural, a large enough vocabulary of modern Tamil is required. Along with Thirukkural, the Tamil Indic corpus dataset (reference) has been used (approximately 5 Gb) to train the model.

To understand the context of Thirukkural, it was represented along with its meaning. The parallel representation of couplets and their corresponding summary in a line did not support well to map the classic Tamil words with modern Tamil meaning in vector space. Random merge representation of Tirukkural and summary without changing word order overcomes this problem. Oversampling with variant random merges (20 times), increases the word occurrence to satisfy minimum word count and each random merge gives us a new sequence that will enable the model to understand better.

4.2 Baseline Systems

The Word2Vec and FastText models have been trained on the dataset mentioned above. However, the Word2Vec model faces significant challenges with out-of-vocabulary (OOV) words when different forms of Thirukkural words are used as queries. In contrast, FastText is more robust due to its sub word tokenization approach, allowing it to handle OOV issues more effectively. Despite this, FastText lacks the ability to provide contextual meanings.

Algorithm 2: Procedure to Find Context Word

```
1: FIND_CONTEXT_WORD( input_couplets, word)
2: begin
3:   X=[]
4:   for words in input_couplets do
5:      if word in vocabulary then
6:         get the vector and append to X
7:      else then
8:         tokenize then
9:            get vector for tokens and append to X
10:C=SELF_ATTENTION(Q=X,K=X,V=X ,word)
11:    find meaning for C using cosine similarity
12:  display meaning
13:end
```

4.3 Proposed System

The FastText model can be enhanced with a self attention mechanism to outperform the baseline models, addressing both out-of-vocabulary (OOV) challenges and improving contextual understanding. The art of self attention layer is to convert the static embedding into dynamic and contextual embedding, The FastText model, implemented using the Gensim library, was trained with the following custom parameters: embedding dimension (d) of 300, minimum word frequency (minCount) of 20, context window size (ws) of 7, number of worker threads (n) set to 4, and 10 training epochs (epoch). The embeddings are taken as a key feature. The dimension of the self attention layer varies with the d. This combination allows for better subword tokenization and, when combined with self attention mechanism offers more meaningful and contextual output. SoftMax Normalization, Dropout layers, and cross-validation during training have attempted to mitigate overfitting. The overall procedure of this proposed system is defined in Algorithm 2.

5. Results

Since no publicly available WordNet or word analogy dataset exists for Thirukkural, we conducted a manual evaluation using a ranking system from 0 to 5. The Thirukkural contains approximately 9310 words, with around 6250 unique words (vary with specific printed edition).

$$Accuracy = \frac{count(S \geq 3)}{count(S \geq 3) + count(S < 3)} \dots\dots Eq(2)$$

Size of the testset/ algorithm	Word2Vec	FastText	FastText with Self-Attention.
100 words	31%	41%	70%
200 words	36%	40%	71%
300 words	39%	56%	74%

Table 2: Performance Analysis of Thirukkural Words

We selected 100, 200, 300 words (by generating random numbers for fairness) for testing model performance alongside their corresponding Thirukkural verses. With the help of five Tamil linguists and the Thirukkural Agarathi, a gold standard dataset was created. The evaluation metric involved assigning manual scores (S) to predictions from word2vec, fastText and fastText

enhanced with self-attention. Scores ranged from 0 (entirely irrelevant) to 5 (accurate and highly relevant), FastText with self-attention models, are presented in Table 2.

5.1 Error Analysis

The percentage of unrecognized or incorrectly identified words in the proposed model ranges between 26% and 30%. The reason for increased error rate includes reversed order of verses and meaning, complex literary style of handling words (agglutinative and decomposed nature), and open-ended meanings.

In some cases, the explanations provided by various literary experts do not strictly follow the original order of the verses. Instead, they reverse the sequence or present the second phrase before the first. This inconsistency creates ambiguity for the model, making it difficult to cluster the correct set of words, even when represented as an intralingual merged representation. The instance of such Kural has been presented in Figure 3.

Kural

ஈன்ற பொழுதின் பெரிதுவக்கும் தன்மகனைச் சான்றோன் எனக்கேட்ட தாய்

Meaning 1: நல்ல மகனைப் பெற்றெடுத்தவள் என்று ஊரார் பாராட்டும் பொழுது அவனைப் பெற்றபொழுது அடைந்த மகிழ்ச்சியைவிட அதிக மகிழ்ச்சியை அந்தத் தாய் அடைவாள்
Meaning 2: தன் மகனை நற்பண்பு நிறைந்தவன் என பிறர் சொல்லக் கேள்வியுற்ற தாய் தான் அவனை பெற்றக் காலத்தில் உற்ற மகிழ்ச்சியை விடப் பெரிதும் மகிழ்வாள்
Meaning 3: தம் மகனைக் கல்வி ஒழுக்கங்களால் நிறைந்தவன் என்று அறிவுடையோர் கூற அதைக் கேட்ட தாய் அவனைப் பெற்ற பொழுதைக் காட்டிலும் மிகுதியாக மகிழ்வாள்

Figure 3: Example for the changed order of Kural and Meaning

Context is ambiguous for some polysemy words (same word represent different meaning based on the context), the proportion of retrieval of appropriate meaning in such case is less compare to clear context. The variation has been shown in Table 1 and Table 3. perhaps Multi Head Attention robustly handles ambiguous context and extending this as a future work. Instances of incorrect clustering of words present challenges in handling ambiguous and complex phrases. This issue has been addressed through Parimelalhagar's literary work on the Thirukkural, where such words were manually tokenized and meaningfully recombined to ensure accurate interpretation and resolve syntactic ambiguities. The system integrates this as a unique feature to effectively mitigate the identified challenge.

S.No	Kural No	word	Meaning
1	10	இறை	கடவுள்
2	388	இறை	தலைவன்
3	432	இறை	தலைமை
4	541	இறை	நடுவுநிலைமை
5	733	இறை, இறை	அரசு, வரி வகைகள்

Table 3: Examples for polysemy

Instances of incorrect clustering of words present challenges in handling ambiguous and complex phrases. This issue has been addressed through Parimelalhagar's literary work on the Thirukkural, where such words were manually tokenized and meaningfully recombined to ensure accurate interpretation and resolve syntactic ambiguities. The system integrates this as a unique feature to effectively mitigate the identified challenge.

We aimed to derive contextual words from the summaries themselves, but it proved challenging to extract exact similar words for some difficult words and rare words where the frequency is very less in classical as well as in contemporary Tamil.

6. Conclusion and Future Work

The proposed computational work on Tamil mainly on Thirukkural incorporates the self-attention mechanism with word embedding, which plays a vital role in analysing how each word within the Thirukkural couplets interacts with others to derive comprehensive meanings for each word rather than phrases or couplets.

By employing Multi-Head Attention (MHA), as seen in BERT, the system can be tuned to examine various linguistic facets simultaneously. The integration of MHA into this architecture aims to further enhance the model's ability to provide subtle and contextual meaning for classical Tamil words present in Thirukkural. As a next step extending the current work to other classical text and non-literary domains by training models on domain-specific corpora that include classical and modern Tamil for intralingual representation. This methodology not only aids in better

comprehension of the Thirukkural but also serves as a model for preserving and appreciating cultural heritage, moral values, and historical narratives of Tamil.

References

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, C. GokulN., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. ArXiv, abs/2005.00085.

R. Anita, C.N. Subalalitha. 2019. Building Discourse Parser for Thirukkural. *International Conference on Natural Language Processing, Hyderabad, India., pages 18–25.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia polosukhin. 2017. Attention Is All You Need. *31st Conference on Neural Information Processing Systems (NeurIPS).*

Dr. B. Bharathi, Sridevi G, Varshitha G J. 2017. Recognising and Retrieving the Meaning of Thirukkural from Speech Utterances. *4th International Conference on Signal Processing, Communications and Networking (ICSCN).*

A M Abirami, Wei Qi Leong, Hamsawardhini Rengarajan, D Anitha, R Suganya, Himanshu Singh, Kengatharaiyer Sarveswaran, William Chandra Tjhi, Rajiv Ratn Shah. Aalamaram. 2024. A Large-Scale Linguistically Annotated Treebank for the Tamil Language. *ELRA Language Resource Association. Page 73-83.*

Anita, Ramalingam, Subalalitha Chinnaudayar Navaneethakrishnan. 2021. A Discourse-Based Information Retrieval for Tamil Literary Texts. *Journal of Information and Communication Technology 2021. Page 353-389.*

Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations (ICLR).*

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Associa tion for Computational Linguistics, 5:135–14Jeffrey Pennington, Richard Socher, Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. Conference on Empirical Methods in Natural Language *Processing, 2014: 1532–1543.*

Ramalingam, Anita and Navaneethakrishnan, Subalalitha Chinnaudayar. 2022. A Novel Classification Framework for the Thirukkural for Building an Efficient Search System. *Journal of Intelligent & Fuzzy Systems. page 2397 – 2408.*

G. L. Jayavardhana Rama, A. G. Ramakrishnan, R. Muralishankar and R. Prathibha. 2002. "A complete text-to-speech synthesis system in Tamil," Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002., Santa Monica, CA, USA, 2002, pp. 191-194.

Ivan Vuli´ c and Marie-Francine Moens Department of Computer Science KU Leuven, Belgium. 2015. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. SIGIR’15.