DISSERTATION · BITS PILANI WORK INTEGRATED LEARNING PROGRAMME

ThinaiKG

A Knowledge Graph for preserving the intangible cultural heritage of Sangam Tamil literature

Automatic Thinai classification of ancient Tamil verses, semi-supervised extraction of ecological and cultural entities, and a public knowledge-graph platform at **www.sangamtamil.com**

1,891	417	0.82	2,313	21,950
Sangam verses	validated entities	macro F1-score	graph nodes	graph edges

Student	Simon Peter J (2023AC05196)
Supervisor	Srilakshmi Dorairajulu, Director, LSI India R&D Pvt. Ltd. (Broadcom), Bangalore
Additional examiner	Dr. P. Velvizhy, Assistant Professor
Faculty mentor	Parthasarathy P D
Duration	16 weeks · 1 October 2025 – 28 January 2026 · Final viva February 2026

Keywords	Knowledge Graph · NLP · Ontology · Sangam · Tamil · Digital Humanities

Birla Institute of Technology & Science, Pilani · Vidya Vihar, Pilani, Rajasthan 333031 · Report compiled September 2026 from the dissertation record

EXECUTIVE SUMMARY

What was done, and why it matters

Sangam Tamil poetry (c. 300 BCE – 300 CE) conveys meaning indirectly: a hill flower, a seagull or a parched path tells the reader which of five landscapes — the *Thinai* — frames the poem, and with it the mood, season and social world of the verse. That knowledge lives in the poems and in experts' minds, but not in any machine-readable form. ThinaiKG set out to change that.

Over sixteen weeks the project built an end-to-end pipeline that scrapes all seven Akam anthologies from Project Madurai, classifies each of the 1,891 verses into one of the five Thinais, discovers the flora, fauna, occupations and places that characterise each landscape, and organises everything into a formal ontology and knowledge graph. The results are published as an interactive website with a learning game, so that students, researchers and non-Tamil speakers can explore the corpus without reading a graph query language.

Key outcomes. A one-vs-rest Linear SVM trained on Thinai-entity features reaches **82.3% accuracy and a macro F1 of 0.82** on a balanced 500-verse ground truth, up from 0.52 for a conventional TF-IDF baseline. A semi-supervised, human-in-the-loop loop grew the entity lexicon from 176 seed terms to **417 validated entities** with an 83% human acceptance rate. The resulting ThinaiKG contains **2,313 nodes and 21,950 edges**, exported in OWL, GraphML and JSON. Fine-tuning IndicBERT v2 on the same task peaked at 58.6% — confirming that, for a low-resource symbolic corpus, knowledge-guided features beat a transformer.

Contributions at a glance

- **A Thinai classifier** that uses symbolic entity features rather than surface lexical cues, and is interpretable by construction.
- **A semi-supervised entity-discovery pipeline** that mines SVM weights for candidate *Karupporul* and validates them with a human reviewer.
- **The first formal Thinai ontology** (OWL) capturing ecological, cultural and emotional dimensions of Sangam literature.
- **ThinaiKG**, a knowledge graph linking Thinais, verses and entities, with full traceability back to source verses.
- **Open resources**: a cleaned 1,891-verse Akam corpus, a 500-verse ground truth, a 417-entity lexicon with English glosses.
- **sangamtamil.com**: poems with translations, Thinai pages, a knowledge-graph explorer and the ‘Thinai Quest’ game.

CONTENTS

Table of contents

How to read this report. Chapters 1–3 set out the problem, the literature and the design; Chapters 4–6 cover data, method and results; Chapters 7–8 describe the knowledge graph and the public platform; Chapters 9–10 reflect on the journey and conclude. Figures and tables are numbered consecutively and every headline metric is taken from the final dissertation (Ver 7.0, January 2026).

1 Introduction

Ancient literary traditions encode the ecological awareness, social organisation and emotional constructs of their civilisations. Sangam Tamil literature is among the oldest and most sophisticated of these corpora, and its interpretive key — the Thinai system — is an implicit ontology waiting to be made explicit.

1.1 Sangam literature and the Thinai system

Sangam poetry was composed by assemblies of Tamil poets between roughly 300 BCE and 300 CE. Over 2,300 poems survive in the Ettuthogai (Eight Anthologies) and Pathuppattu (Ten Idylls). The tradition is divided into **Akam** — interior themes of love, longing and domestic life — and **Puram**, the exterior world of war, kingship and charity. Unlike narrative-driven literature, Sangam poems convey meaning through carefully chosen references to landscapes, flora, fauna, occupations and human emotions. These references are not ornamental; they are the interpretive code.

Within Akam poetry the organising principle is the **Thinai**: five ecological landscapes, each bound to a characteristic emotion, time of day, season, deity, set of flowers, animals and people.

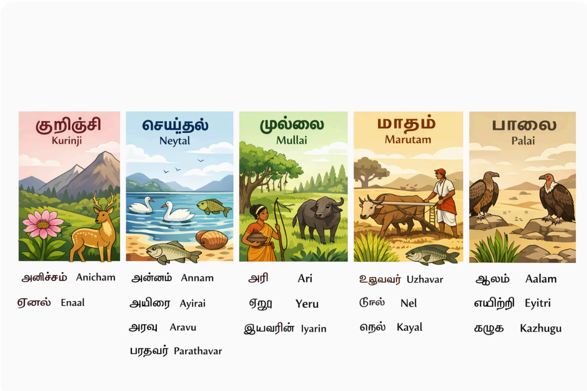

Figure 1 – The five Thinai landscapes and a sample of their characteristic flora, fauna and people

Thinai	Landscape	Associated emotion	Typical entities
Kurinji	Mountains	Union and secret love	Kurinji flower, peacock, elephant, hill tribes (Kuravar), millet
Mullai	Forest and pasture	Patient waiting	Jasmine, deer, cowherds, rainy season, evening
Marutham	Farmland and river	Domestic life and quarrel	Lotus, buffalo, farmers, paddy, ponds
Neytal	Seashore	Longing and separation	Water lily, seagull, fishermen, boats, salt
Palai	Arid wasteland	Hardship and elopement	Vulture, dry trees, bandits, scorching sun

An example

செங்களம்படக் கொன்று அவுணர்த் தேய்த்த
செங்கோல் அம்பின் செங்கோட்டி யானைக்
கழல் தொடிச் சேஎய் குன்றம்,
குருதிப் பூவின் குலைக் காந்தட்டே

Kurunthogai 1 (Ettuthogai)

The hero sends flowers to the heroine through her friend. But it is a common hill flower found everywhere on the mountains ruled by the god Murugan, so she refuses it — telling him that her friend is far more precious than this and his approach is in vain.

The landscape is never named, yet *kunram* (hill), *kanthal* (glory lily, a mountain flower), the deity Murugan and the elephant imagery all point unambiguously to **Kurinji**. A classifier must infer the landscape from these symbolic cues alone.

1.2 Problem statement

> "There is no system today that can automatically identify Thinai, or grasp and represent the ecological and cultural knowledge of Sangam literature."

Existing digital resources present the poems as unstructured text usable only by readers with deep domain expertise. Modern Tamil NLP tools are trained on news and social media; archaic vocabulary, agglutinative compounding and metaphor defeat them — and, as the project verified, even the latest large language models misread Thinai cues. UNESCO's guidance on intangible cultural heritage stresses that preservation means capturing the underlying knowledge structures, not merely digitising text.

1.3 Objectives

- Analyse and extract cultural and ecological knowledge embedded in Sangam texts using NLP techniques.
- Design and construct ThinaiKG — an ontology and knowledge graph capturing relationships among flora, fauna, deities, people, places and emotions in each Thinai.
- Develop automated pipelines for preprocessing, entity recognition and relation extraction from the ancient Tamil corpus to populate the graph.
- Enable semantic querying and reasoning through a user interface so that anyone can explore the interlinked concepts.
- Contribute reusable datasets and ontology resources to digital-heritage and low-resource NLP research.

1.4 Scope and limitations

The work focuses on the core Sangam Akam corpus and its five Thinais. Poetic meter, phonetics, authorship attribution and historical chronology are out of scope. Classical deep-learning approaches were deliberately limited because of data scarcity and the need for interpretability; the emphasis is on hybrid, ontology-guided machine learning that balances performance with semantic transparency.

2 Background and related work

The literature survey spanned Tamil NLP, computational processing of classical texts, low-resource language research, ontology engineering, knowledge graphs for cultural heritage and the representation of intangible heritage. Its purpose was to establish what could be reused, what would fail, and where the gap lay.

2.1 Tamil natural language processing

Early Tamil NLP concentrated on morphological analysis, tokenisation and part-of-speech tagging, which are challenging even for modern Tamil because of its agglutinative morphology [2]. Later work extended to parsing, named-entity recognition and sentiment analysis, but almost always on contemporary news, conversational or social-media data. Applied to Sangam-era texts these systems degrade severely: vocabulary mismatch, archaic grammar and poetic compounding render standard pipelines ineffective [3]. Research targeting classical Tamil directly is sparse and mostly concerned with word segmentation and lexical normalisation [4], [5]; it treats the poems as linguistic artefacts rather than carriers of structured cultural knowledge.

2.2 Semantic analysis and the low-resource problem

Metaphor detection and emotion classification have been attempted for Tamil literary texts [6], [7], but they rely on explicit sentiment markers that Sangam poetry does not use — longing is signalled by coastal imagery, not by emotional vocabulary. Ancient Tamil is a textbook low-resource language: no large annotated datasets, no pretrained models, no standard ontology. Surveys argue that data-driven methods alone are insufficient here and that linguistic and domain knowledge must be injected into models [8], [9]. This directly motivates ThinaiKG's hybrid design.

2.3 Knowledge graphs for cultural heritage

Semantic-web technologies have proven their value for interoperable, machine-readable heritage data [11], [12], and recent work on 'intelligent heritage digital twins' places knowledge graphs at the centre of AI-enabled cultural systems [10]. Large initiatives such as Europeana and CIDOC-CRM, however, are artefact-centric; they model objects and archival metadata rather than literary symbolism. Ontology-guided extraction from heritage texts improves retrieval and accessibility [13], [14], but existing systems address narrative prose, not poetry with implicit, ecology-driven semantics. Ontologies for traditional ecological knowledge and mythological narratives [17], [18] show that symbolic relationships can be modelled formally — yet none covers the Thinai system.

Research gap. No prior work explicitly models the Thinai system as an ontology or knowledge graph [21]; no automated Thinai classifier handles its implicit, metaphorical semantics; and generic LLMs hallucinate on culturally grounded symbols when domain knowledge is absent [22]. ThinaiKG integrates ancient-Tamil NLP, a Thinai-specific ontology and semi-supervised, human-validated extraction into a single framework.

3 System design

ThinaiKG is a knowledge-centric architecture rather than a purely data-driven pipeline. Domain knowledge enters through Thinai-specific entity dictionaries, ontology constraints and human-in-the-loop validation — a deliberate response to the symbolic, metaphorical and low-resource nature of the texts.

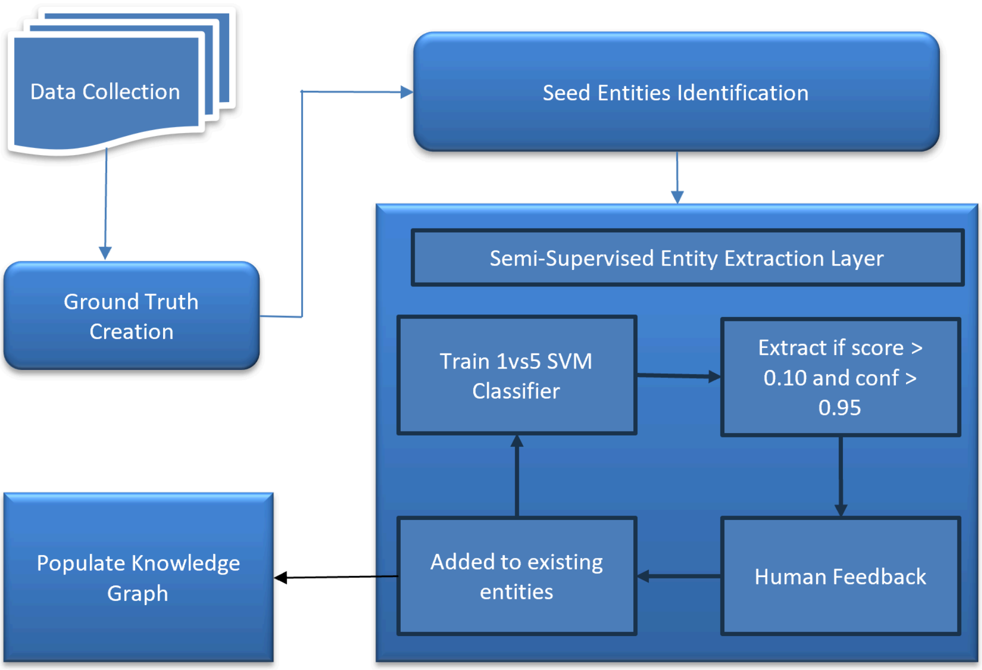

Figure 2 – Methodology: data collection, ground truth, seed entities, then the semi-supervised extraction loop that feeds the knowledge graph

3.1 Layered architecture

Layer	Responsibility
Data acquisition & preprocessing	Scrape Project Madurai, strip metadata and commentary, normalise Unicode Tamil, keep only verse text
Thinai classification	One-vs-rest Linear SVM over entity-based features assigns each verse to one of five Thinais
Entity extraction & expansion	Mine SVM weight vectors on high-confidence verses for candidate entities; score, filter and send for review
Ontology design	Lightweight OWL schema: Thinai, Verse, Entity, Category and their relationships
Knowledge graph construction	Populate the ontology with verses, entities and Thinais; export OWL, GraphML, JSON
Query, visualisation & game	Website with poems, translations, KG explorer and the Thinai Quest game

Each layer operates independently and can be improved without redesigning the rest. The pipeline runs in **refinement cycles**: classify with the current feature set, keep high-confidence predictions, analyse feature weights, validate candidate entities with a human, expand the dictionaries, retrain. Cultural knowledge extraction is not a one-shot process; the loop makes expert feedback a first-class part of the system.

3.2 Design assumptions and ethical considerations

Three assumptions underpin the design: Thinai can be inferred from the presence of symbolic entities; cultural semantics can be partially captured in curated ontologies; and human validation is necessary for semantic correctness. Misclassifying a Thinai could distort cultural understanding, so ThinaiKG emphasises transparency and traceability — every entity and relationship can be traced back to the verses it came from, enabling verification and correction rather than imposing external interpretive bias.

3.3 Implementation environment

All experiments ran on a Dell G15 5520 laptop (RTX 3050, 16 GB RAM) in a Miniconda Python environment using scikit-learn, BeautifulSoup, NetworkX and Owlready-style OWL export. The SVM work is light; the BERT experiments used CUDA and took roughly two days per run. The code base is organised as numbered scripts — download, preprocess, extract verses, analyse unique words, train the SVM, classify the full corpus, generate the OWL file and generate the website — with entity lexicons and knowledge-graph builds kept under explicit version folders (v1.0 … v16.0, vv1.0 … vv1.2).

4 Data

4.1 Sources

The corpus was built exclusively from the Akam tradition, aggregating all seven Akam texts to represent the full range of poetic styles and meters. Texts were downloaded as Unicode HTML from Project Madurai (e.g. projectmadurai.org/pm_etexts/utf8/pmuni0028.html), parsed with BeautifulSoup and segmented into individual verses with regular expressions that detect the start of each poem.

Text	Verses
Ainkurunuru	498
Kuruntokai	401
Natrinai	399
Akananuru	396
Kalittokai	149
Kurinjipattu (single poem, split)	27
Mullaippattu (single poem, split)	21
Total	**1,891**


Project Madurai
மதுரை தமிழ் இலக்கிய மின்தொகுப்புத் திட்டம்

aingurunUru of kUdalur kizhAr
one of "eTTutokai" anthology
(in tamil script, unicode/UTF-8 format)

கூடலூர் கிழார் அருளிய எட்டுத்தொகை
நூல்களில் ஒன்றான "ஐங்குறு நூறு"

Acknowledgement:
Our Sincere thanks go to Dr. Ram Ravindran, Indianpolis, USA for his assistance in the preparation of this work.
This webpage presents the Etext in Tamil script, in Unicode encoding (utfi-8 format).

© Project Madurai, 1998-2021.
Project Madurai is an open, voluntary, worldwide initiative devoted to preparation
of electronic texts of tamil literary works and to distribute them free on the Internet.
Details of Project Madurai are available at the website
https://www.projectmadurai.org/
You are welcome to freely distribute this file, provided this header page is kept intact.

கூடலூர் கிழார் அருளிய எட்டுத்தொகை
நூல்களில் ஒன்றான "ஐங்குறு நூறு"

aingurunUru : One of "eTTutokai" anthology of 500 short poems (two are missing),
composed by kUdalUr kizhAr at the instance of Chera King "yAnaikkatcEy mAntaran cEral irumporai"

(வித்துவான் எம்.நாராயண வேலுப்பிள்ளை அவர்களால் தொகுக்கப்பட்டது)

1.
வாழி ஆதன் வாழி அவினி
நெற்பல பொலிக பொன்பெரிது சிறக்க
எனவேட் டோ ளே யாயே யாமே
நனைய காஞ்சிச் சினைய சிறுமீன்
யாணர் ஊரன் வாழ்க
பாணனும் வாழ்க எனவேட் டேமே.

2.
வாழி ஆதன் வாழி அவினி
விளைக வயலே வருக இரவலர்
எனவேட் டோ ளே யாயே யாமே

Figure 3 – A typical Project Madurai page: editorial notes precede the verses

4.2 Preprocessing

Preprocessing retained only the raw verses: metadata, numbering and commentary were removed, special characters and formatting artefacts eliminated, and Unicode Tamil standardised. Aggressive normalisation and lemmatisation were intentionally avoided — the original poetic word forms carry the symbolic Thinai indicators, and stripping inflections would erase them.

4.3 Class distribution and ground truth

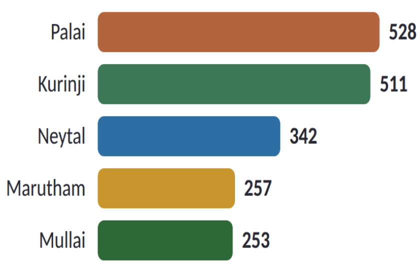

Figure 4 – Verses per Thinai (Palai 27.9%, Kurinji 27.0%, Neytal 18.1%, Marutham 13.6%, Mullai 13.4%)

Unlike earlier studies that used balanced subsets, the full corpus was kept to reflect the natural distribution of themes. Palai and Kurinji dominate; Marutham and Mullai are minority classes, which later shaped the targeted entity expansion.

From the corpus, **500 verses were manually verified** as ground truth — 100 per Thinai — giving a balanced set for supervised training and evaluation. A 176-entity seed lexicon of citation-verified flora, fauna, occupations and food terms from scholarly sources was curated alongside it.

5 Methodology

5.1 The turning point: features that mirror how a scholar reads

The first classifier was conventional: TF-IDF over 10,000 uni-, bi- and tri-gram features (91.6% sparse), SMOTE oversampling to balance classes, and an RBF-kernel SVM. It reached a macro F1 of only **0.52**; the majority class scored well while the minority Kurinji collapsed to F1 0.32. A hypothesis that the 'confusing' Marutham class was the culprit was tested by dropping it — and the four-class model did no better (0.52). The problem was the features, not a class.

The breakthrough came from literary theory. Tolkappiyam already tells us what defines a landscape: its *Karupporul* — the flora, fauna, land, occupations and food native to it. Keeping **only** these entity words in each verse and discarding everything else aligned the feature space with the way a scholar actually reads a poem. With 145 curated entities the macro F1 rose to 0.68; with iterative expansion it reached **0.82**. As a bonus the model became explainable: every decision can be traced to named cultural entities.

5.2 One-vs-rest Linear SVM

Five binary Linear SVMs (scikit-learn) are trained, each separating one Thinai from the other four by minimising hinge loss with L2 regularisation. At prediction time the decision scores of all five are computed and the verse is assigned to the Thinai with the maximum score. Linear SVMs were chosen for their robustness in high-dimensional sparse spaces, suitability for small datasets, and — crucially — their interpretable weight vectors, which double as an entity-discovery tool. A prediction is accepted for entity extraction only when both the decision score and the confidence exceed predefined thresholds.

5.3 Semi-supervised, human-in-the-loop entity discovery

Manual curation does not scale, so the model itself was used to propose new entities:

1. **Train the baseline** one-vs-rest Linear SVM on the current lexicon (TF-IDF over entity tokens).
2. **Filter high-confidence verses** — unlabelled verses whose predicted Thinai has confidence above 0.95.
3. **Score candidate terms** with Entity Score = SVM coefficient × Σ TF-IDF; keep terms with coefficient > 0.7 and frequency ≥ 3.
4. **Human validation** of each candidate for cultural and literary relevance, correct Thinai association, and being a genuine entity rather than a suffix, verb or tokenizer fragment.
5. **Expand and retrain**: approved entities join the lexicon; the cycle repeats.

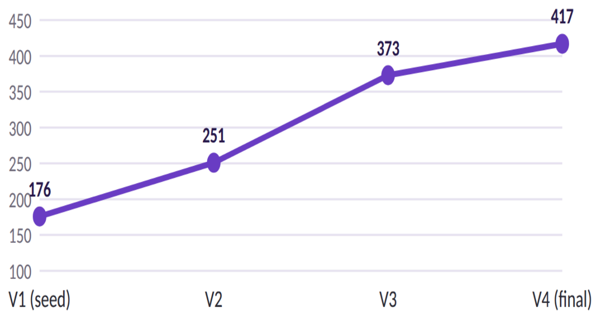

Figure 5 – Growth of the validated entity set across the four milestone versions

Across iterations the entity set grew from **176 to 417**, the growth rate averaged about 50% per cycle, and **83%** of automatically proposed entities were accepted by the human reviewer — evidence that weight-driven extraction surfaces meaningful symbols rather than lexical noise. Classification confidence rose from 0.57 to 0.82 over the same period.

Behind the four milestones sit sixteen internal lexicon versions: frequency-exclusivity analysis, a Thinai-affinity score, stop-word and verb heuristics, a strict linguistic filter that removed partial words and suffixes (cutting a 1,000-term set to 237 with under 1% accuracy loss), chi-square selection, and targeted expansions to lift the lagging Marutham and Mullai classes.

5.4 Evaluation protocol

Classification is evaluated on the balanced 500-verse ground truth with per-class precision, recall and F1 plus overall accuracy. Because entity extraction is semi-supervised and culturally sensitive, two further measures are reported: **entity growth rate** per iteration and **human acceptance rate**. Knowledge-graph quality is assessed with structural metrics — node and edge counts, density and degree statistics.

6 Results and analysis

6.1 Thinai classification

82.3%	0.83	0.81	0.82
Accuracy	Precision	Recall	Macro F1

The one-vs-rest Linear SVM on entity features reaches an overall accuracy of 82.28% and F1 of 0.82 (Table I). Balanced precision and recall show the model avoids both excessive false positives and false negatives — important in a domain where misclassifying a Thinai means misreading the poem. The rise from the 0.52 baseline confirms the value of ontology-guided feature engineering and iterative entity expansion.

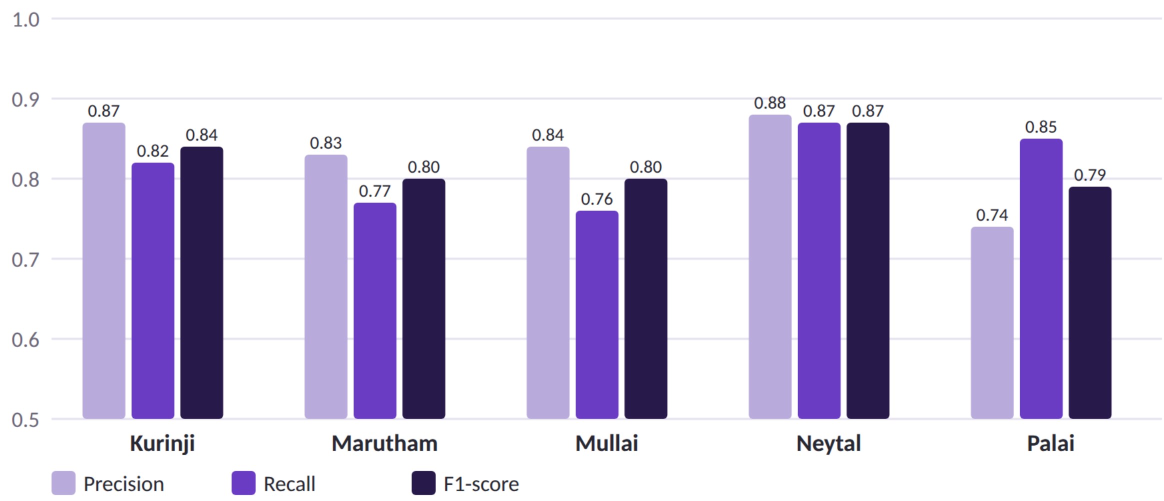

Figure 6 – Per-Thinai precision, recall and F1 on the 500-verse ground truth (Table II)

Thinai	Precision	Recall	F1-score
Kurinji	0.87	0.82	0.84
Marutham	0.83	0.77	0.80
Mullai	0.84	0.76	0.80
Neytal	0.88	0.87	**0.87**
Palai	0.74	0.85	0.79

Neytal is easiest (F1 0.87): sea, fishermen, boats and marine life are distinct, consistent cues. **Kurinji** benefits from characteristic mountain flora, fauna and terrain. **Marutham** and **Mullai** share overlapping agricultural and pastoral imagery yet remain balanced at 0.80. **Palai** is hardest (0.79, precision 0.74): literary scholarship describes it not as a natural region but as any landscape transformed by drought, so it shares symbols with the others — the model recalls 85% of Palai verses but admits some false positives. The error profile matches the intrinsic semantics of the corpus.

6.2 Entity extraction

The average entity growth rate of roughly 50% per cycle and the 83% human acceptance rate show that the feature-weight-driven strategy discovers culturally valid indicators well beyond the manually curated seed set, while the review step prevents semantic drift and preserves the cultural integrity of the lexicon.

6.3 Transformer experiments

To test whether a pretrained language model could do better, IndicBERT v2 (masked model) was fine-tuned in four configurations on the same task, each taking about two days on the RTX 3050.

Figure 7 – Test accuracy of BERT configurations versus the entity-based SVM (Table IV)

The best transformer variant — BERT embeddings (768) concatenated with 519 KG features and a classification layer — reached only 58.6%; a frozen encoder with a linear head managed 53.6%. With 1,891 verses there is simply not enough data to re-teach a 12-layer encoder an archaic vocabulary that its tokenizer fragments into meaningless sub-pieces, and a transformer's weights offer none of the entity-discovery interpretability the SVM provides. The project therefore kept the entity-based SVM and recorded BERT as future work contingent on a larger annotated corpus.

6.4 Summary of findings

ThinaiKG demonstrates that combining machine learning, ontology modelling and human validation can interpret Sangam Tamil literature computationally. Strong classification, robust and culturally valid entity extraction, and a richly connected knowledge graph together validate the knowledge-driven design — and establish a foundation for semantic preservation of the corpus.

7 Ontology and knowledge graph

7.1 Ontology design

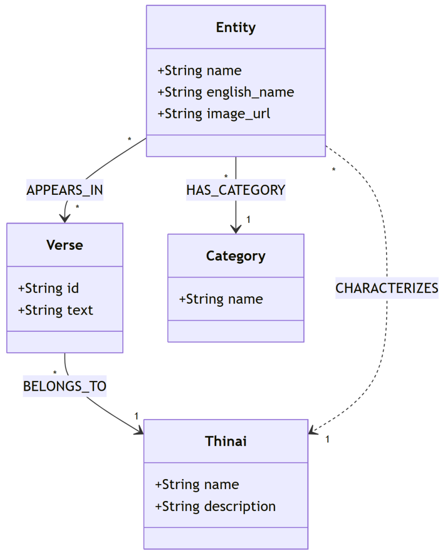

Figure 8 – ThinaiKG ontology classes and relationships

The ontology is deliberately lightweight — clear, extensible and expressive enough to guide graph construction and keep extracted data consistent. It defines four classes:

- **Thinai** — one of the five landscapes, with name and description.
- **Verse** — an individual poem or verse with identifier and Tamil text; the contextual container for entities.
- **Entity** — a culturally significant term with Tamil name, English gloss and image.
- **Category** — the entity type: flora, fauna, occupation, land, food, and so on.

and four relationships: **APPEARS_IN** (Entity → Verse), **BELONGS_TO** (Verse → Thinai), **HAS_CATEGORY** (Entity → Category) and the derived **CHARACTERIZES** (Entity → Thinai), asserted when an entity co-occurs with high-confidence verses of a Thinai. The design principles are minimalism, extensibility to further literary genres, and interpretability — transparency between textual evidence and semantic representation.

7.2 Graph construction and metrics

The construction layer populates the ontology with instances from the processed corpus. Verses are first-class nodes, so every claim in the graph is traceable to the poem it came from; entities were added only where there was double confirmation — a high-confidence Thinai classification and a significant SVM weight.

Attribute	Value
Thinai nodes	5
Entity nodes	417
Verse nodes	1,891
Total nodes	**2,313**
Verse → Thinai edges	1,891
Entity → Thinai edges	885
Entity → Verse edges	19,174
Total edges	**21,950**
Graph density	0.0082
Average degree	18.98
Maximum degree	1,117

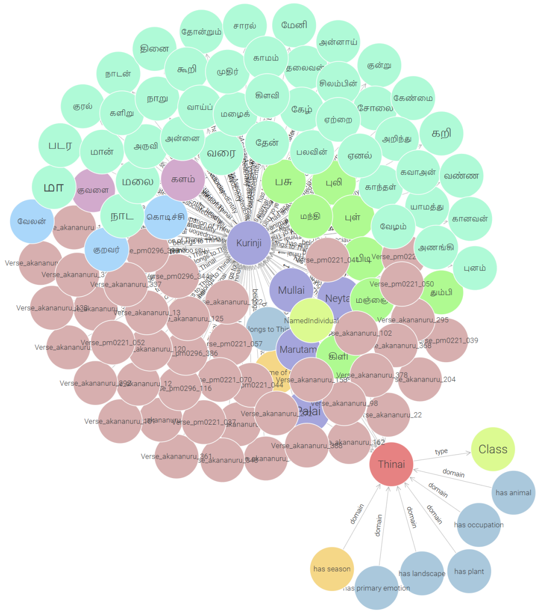

Figure 9 – A rendering of ThinaiKG: Thinai hubs surrounded by their entities and verses

The high count of entity–verse edges indicates rich contextual grounding: each entity is supported by many textual occurrences. Low density is characteristic of semantic graphs, where sparsity reflects selective, meaningful relationships. The maximum degree of 1,117 reveals hub entities — recurring cultural symbols that play a central role in Thinai semantics. The graph is exported as OWL, GraphML and JSON for reuse.

8 The sangamtamil.com platform

With the faculty mentor's encouragement the project was gamified and published online, so that its benefit reaches beyond the examination room. The site presents the knowledge graph, background on Sangam poetry, the verses themselves with English translations, and the ‘Thinai Quest’ game.

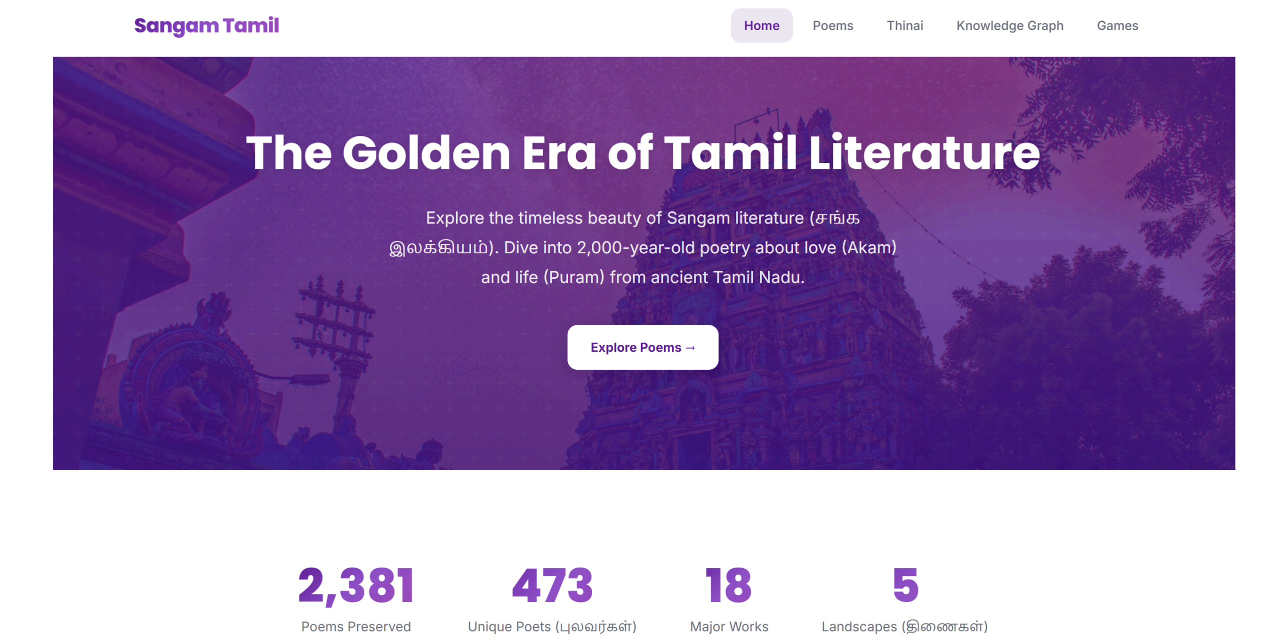

Figure 10 – Home page: 2,381 poems preserved, 473 unique poets, 18 major works, 5 landscapes

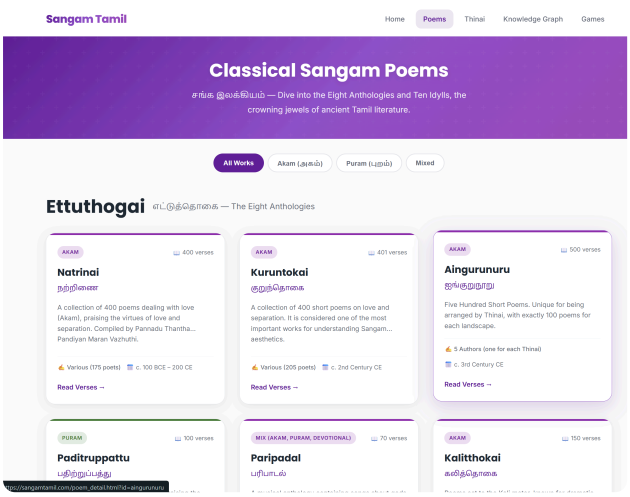

Figure 11 – Poems section: browse every anthology and idyll

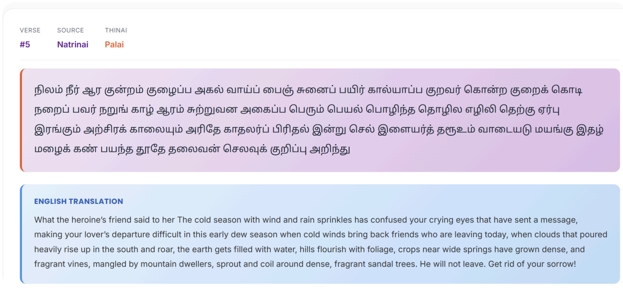

Figure 12 – A verse page: source, predicted Thinai, Tamil text and translation

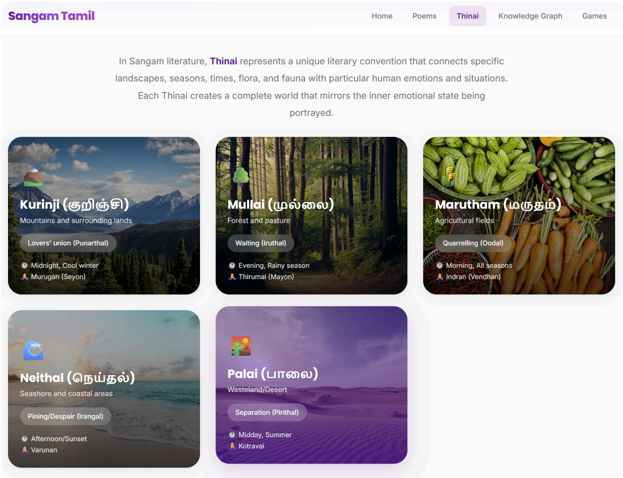

Figure 13 – Thinai pages describe landscape, season, time of day and deity

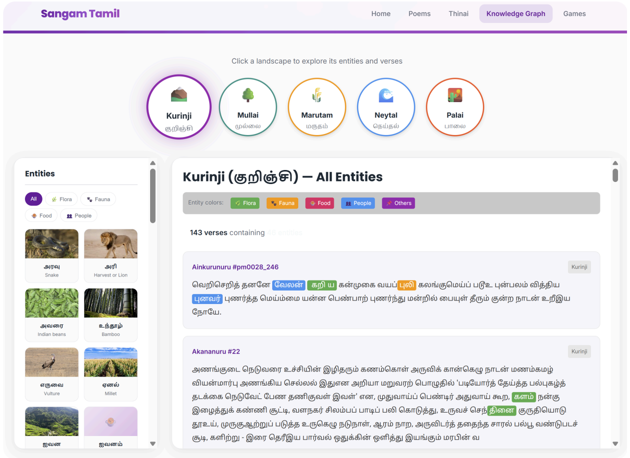

Figure 14 – Knowledge-graph explorer: pick a Thinai, filter entities by category, see verses with entities highlighted

The explorer needs no query language: users click a Thinai to learn about it, explore all categories of entities of that landscape, or find every verse that matches an entity or set of entities. Each entity carries an English gloss and, where available, an image, which makes the site usable by non-Tamil speakers.

Thinai Quest presents a Tamil verse and asks the player to guess its Thinai. Hints — the meaning, the entities present, or the source text — can be revealed at a points cost, and a correct answer scores 100. The game turns the classifier's own reasoning into a learning exercise.

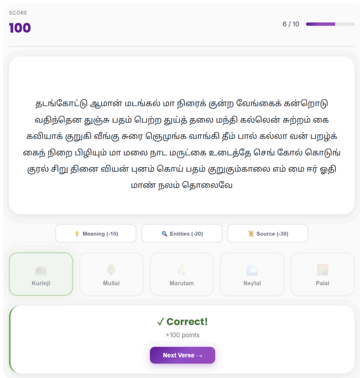

Figure 15 – Thinai Quest

9 Project journey and lessons learned

When	Milestone	What happened
Aug 2025	Proposal	Dissertation outline and proposal on Tamil cultural heritage; topic narrowed from a broad multimodal Tamil-heritage model to Thinai and knowledge graphs
Oct 2025	Kick-off	Corpus scraped from Project Madurai; ground truth assembled; TF-IDF SVM baseline (macro F1 0.52); failed ‘remove Marutham’ experiment
Nov 2025	Entity insight	Switch to entity-only features; 176 citation-verified seed entities; first SVM-weight discovery; IndicBERT experiments begun
Dec 2025	Mid-semester	Iterative human-in-the-loop expansion (v1 → v16); OWL ontology; knowledge-graph phase 1; mid-semester report and viva
Jan 2026	Final report	417-entity set and F1 0.82; KG of 2,313 nodes; website and Thinai Quest built; report iterated through seven versions
Feb 2026	Final viva	Final presentation and evaluation; ThinaiKG public at sangamtamil.com

Lessons learned

- **Domain theory is the best feature engineer.** A two-thousand-year-old grammar (Tolkappiyam's Karupporul) outperformed 10,000 statistical n-grams. When a corpus is symbolic, encode the symbols.
- **Negative results steer.** Removing a class, cleaning too aggressively (v5–v9 dropped to ~78%), and the BERT runs each closed a door and pointed to the right one.
- **Quality over quantity.** A 237-entity filtered set matched a 1,000-term set within 1%; the signal lives in a small, high-quality core of concrete entities, with a handful of high-value abstract terms bridging the final gap.
- **Keep the human in the loop — and version everything.** Sixteen lexicon versions, per-version metadata and a changelog made every gain reproducible and every regression explainable.
- **Interpretability pays twice.** The SVM's weights were both the classifier and the entity-discovery engine; a black-box model would have delivered neither the knowledge graph nor the 83% acceptance rate.
- **Ship it.** Publishing the site and the game turned a dissertation into a resource that students, scholars and non-Tamil speakers can actually use.

10 Conclusions, limitations and future work

10.1 Conclusions

ThinaiKG addressed the fundamental challenge of interpreting Sangam poetry, where meaning is conveyed implicitly through references to landscape, ecology and cultural practice. By integrating NLP, machine learning, ontology engineering and knowledge-graph construction, it showed that Thinai classification and cultural entity extraction can be achieved reliably even in a low-resource, culturally complex domain. The entity-guided one-vs-rest Linear SVM achieved a macro F1 of 0.82, far above conventional text-classification baselines; the semi-supervised, human-in-the-loop extraction strategy discovered culturally meaningful entities with an 83% acceptance rate; and the resulting knowledge graph transforms implicit poetic knowledge into an explicit, machine-interpretable, richly connected representation that supports semantic querying and exploration. Knowledge-centric, ontology-guided machine learning is well suited to computational humanities — particularly for symbolic, metaphorical, low-resource texts — and ThinaiKG is both a technical contribution and a step towards the long-term digital preservation of Tamil cultural heritage.

10.2 Limitations

- Scope is restricted to the core Sangam Akam corpus; Puram poetry and later Tamil traditions follow different thematic and symbolic conventions.
- Curated entity dictionaries and linear decision boundaries may miss subtler semantic interactions in complex poetic structures.
- Human-in-the-loop validation ensures reliability but introduces dependency on expert knowledge, limiting scalability.
- The graph currently supports semantic traversal; richer reasoning and inference remain to be added.

10.3 Future work

- **Advanced semantic models** — hybrid architectures pairing ontology-guided reasoning with language models trained on classical Tamil, once a larger annotated corpus exists; the BERT line of experiments can be resumed on that basis.
- **Broader coverage** — extend the framework to Puram poetry and post-Sangam literature for comparative analysis across genres and periods.
- **Graph reasoning** — rule-based or probabilistic inference to derive Thinai associations and surface latent cultural patterns.
- **Multilingual access** — link ThinaiKG to multilingual descriptions and translations for non-Tamil audiences and cross-cultural research.
- **Education and crowdsourcing** — integrate the graph into learning platforms and digital exhibits, extend the game to other educational uses, and use controlled crowdsourcing to scale validation without sacrificing cultural integrity.

In one line: ancient wisdom, made machine-readable — without losing the scholar's eye.

A Appendix

A.1 Thinai ontology schema

Classes. *Thinai* (Kurinji, Mullai, Marutham, Neytal, Palai); *Verse* (an individual Sangam verse, the contextual container for entities); *Entity* (flora, fauna, geographical features, professions, emotions, symbolic objects); *EntityCategory* (Flora & Fauna, Geography, Profession, Emotion, Food, ...).

Relationships. *belongsToThinai* (Verse → Thinai); *containsEntity* / APPEARS_IN (Verse ↔ Entity); *associatedWithThinai* / CHARACTERIZES (Entity → Thinai); HAS_CATEGORY (Entity → EntityCategory). Design principles: minimalism, extensibility, interpretability.

A.2 Deliverables in the project folder

Artefact	Location / form
Final dissertation report (v7)	Final Report / Ver 7.0 Project Report.docx; signed PDF 2023AC05196.pdf
Final viva presentation	Final Report / Final Viva Ver 1.0.pptx (33 slides)
Mid-semester report and viva	Midsem Report/, Midsem Viva Ver 2.0.pptx
Source code	VSCODE/ — MidsemDemo pipeline scripts 01–08, scripts/, bert_experiments/, gamification/
Entity lexicons	VSCODE/data/entities/versions v1.0–v16.0 and vv1.0–vv1.2 with CHANGELOG
Knowledge graph	VSCODE/knowledge_graph — thinai_ontology.owl, thinai_knowledge_graph.graphml / .json, kg_statistics.json
Ground truth and corpus	VSCODE/data/ground_truth, VSCODE/data/raw and processed
Website and game	VSCODE/knowledge_graph/website, VSCODE/gamification — live at www.sangamtamil.com
Experiment logs	VSCODE/results, docs/EXPERIMENT_LOG.md, PROJECT_JOURNEY_SUMMARY.md

A.3 Glossary

Akam / Puram

The interior (love, emotion) and exterior (war, public life) divisions of Sangam literature; the Thinai system governs Akam poetry.

Thinai

A landscape-based poetic classification integrating ecology, emotion, social context and symbolism.

Karupporul

The ‘native things’ of a landscape in Tolkappiyam — its flora, fauna, land, occupations and food — used here as the classifier's feature vocabulary.

Project Madurai

Open-access repository of Unicode-encoded classical Tamil texts, the source of the corpus.

One-vs-rest (OvR)

Multi-class strategy training one binary classifier per class against all others.

Human-in-the-loop

Design in which human expertise validates automated outputs — here, every candidate entity.

Knowledge graph

Structured knowledge as nodes (entities) and edges (relationships) enabling semantic querying.

A.4 Selected references

[1] P. Matan and P. Velvizhy, “Systematic review of morphological and semantic analysis in a low-resource language: Tamil,” in *Empowering Low-Resource Languages With NLP Solutions*, Springer, 2024, pp. 86–112.

[2] K. Balasubramaniam and A. Rajendran, “Morphological analysis for Tamil language processing,” *Journal of Computer Linguistics*, vol. 45, no. 2, 2021.

[3] G. Mahesan, “Challenges in computational processing of classical Tamil literature,” *Language Resources and Evaluation*, vol. 54, no. 3, 2020.

[4] S. Sandeep et al., “Word segmentation of ancient Tamil text extracted from Sangam literature and inscriptions,” Proc. ICON, 2025.

[8] S. Bird, “Low-resource languages: Challenges and opportunities,” *Computational Linguistics*, vol. 46, no. 4, 2020.

[9] A. Joshi et al., “State of the art in low-resource NLP,” *ACM Computing Surveys*, vol. 55, no. 2, 2023.

[10] F. Felicetti et al., “Knowledge graphs and artificial intelligence for the intelligent heritage digital twin,” *Heritage*, MDPI, 2025.

[11] E. Hyvönen, “Cultural heritage linked data on the semantic web,” *Semantic Web Journal*, vol. 3, no. 2, 2012.

[13] M. Schimmenti et al., “Knowledge graph generation from cultural heritage texts,” arXiv preprint, 2025.

[15] UNESCO, “Digital preservation of intangible cultural heritage,” UNESCO Report, 2022.

[17] Y. Li et al., “Ontology modeling of traditional ecological knowledge,” *Semantic Web*, vol. 11, no. 5, 2020.

[21] K. Ramachandran, “Thinai system in Sangam literature: A critical survey,” *Tamil Studies Journal*, vol. 48, no. 1, 2018.

[22] J. Wei et al., “Hallucinations in large language models: A survey,” arXiv preprint, 2023.

[23] A. Hogan et al., “Knowledge graphs,” *ACM Computing Surveys*, vol. 54, no. 4, 2021.

The full 24-item reference list is in the dissertation report (Ver 7.0).