PyThaiNLP: Thai Natural Language Processing in Python

Wannaphong Phatthiyaphaibun[♦], Korakot Chaovavanich[†], Charin Polpanumas[†], Arthit Suriyawongkul[‡], Lalita Lowphansirikul[♦], Pattarawat Chormai[§¶], Peerat Limkonchotiwat[♦], Thanathip Suntorntip[♣], Can Udomcharoenchaikit[♦]

[♦]VISTEC, [†]PyThaiNLP, [‡]Trinity College Dublin,
[§]Technische Universität Berlin, [¶]Max Planck School of Cognition, [♣]Wisesight
wannaphong.p_s21@vistec.ac.th

Abstract

We present PyThaiNLP, a free and open-source natural language processing (NLP) library for Thai language implemented in Python. It provides a wide range of software, models, and datasets for Thai language. We first provide a brief historical context of tools for Thai language prior to the development of PyThaiNLP. We then outline the functionalities it provided as well as datasets and pre-trained language models. We later summarize its development milestones and discuss our experience during its development. We conclude by demonstrating how industrial and research communities utilize PyThaiNLP in their work. The library is freely available at https://github.com/pythainlp/pythainlp.

1 Introduction

In recent years, the field of natural language processing has witnessed remarkable advancements, catalyzing breakthroughs for various applications. However, Thai has remained comparatively underserved due to the challenges posed by limited language resources (Arreerard et al., 2022).

Thai is the de facto national language of Thailand. It belongs to Tai linguistic group within the KraDai language family. According to Ethnologue (Eberhard et al., 2023), there are 60.2 million users of Central Thai, of which 20.8 million are native (2000). If including the Northern (6 million, 2004), Northeastern (15 million, 1983), and Southern (4.5 million, 2006) variants, there are estimated 85.7 million users of Thais speakers around the world.

Thai is a scriptio continua or has neither spaces nor other marks between the words or sentences in its most common writing style. (Sornlertlamvanich et al., 2000). The lack of clear word and sentence boundaries leads to ambiguity that cannot be disambiguated using merely just grammatical knowledge (Supnithi et al., 2004).

Although many closed-source open APIs for NLP have an ability to process Thai language[1], we believe that an open-source toolbox is essential for both researchers and practitioners to not only access the NLP capabilities but also gain full transparency and trust on both training data and algorithms.[2] This allows the community to adapt and further develop the functionalities as needed, making a crucial step towards democratizing NLP.

This paper introduces PyThaiNLP, an open-source Thai natural language processing library written in Python programming language. Its features span from a simple dictionary-based word tokenizer, to a statistical named-entity recognition, and an instruction-following large language model. The library was released in 2016 under an Open Source Initiative-approved Apache License 2.0 that allows free use and modification of software, including commercial use.

2 Open-source Thai NLP before PyThaiNLP

Before PyThaiNLP started in 2016, some free and open-source software do exist for different Thai NLP tasks, but there were no unified open-source toolkits that unified multiple tools or tasks in a single library, and the number of available Thai NLP datasets was low compared to high-resource languages like Chinese, English, or German.

Natural Language Toolkit (NLTK) (Bird and Loper, 2004), one of the most comprehensive and most popular NLP libraries in Python at the time, did not support Thai. OpenNLP, another popular free and open-source NLP toolkit written in Java,

[1] Such as those provided by commercial cloud service providers and "AI for Thai", the government-funded Thai AI service platform at https://aiforthai.in.th/.

[2] For a discussion about concentrated power and the political economy of 'open' AI, see Widder et al. (2023).

started having Thai models in version 1.4 (2008)[3] but in version 1.5 (2010) Thai was no longer listed in its supported languages[4].

Open Thai language resources, like annotated corpora, were also limited in size and number. "Publicly available" datasets tend to have restricted access, either through restrictive licenses[5] or the registration requirement, or both.

Because there is a few toolkits available, limited in documentation and performance, short of rigorous benchmarking, and/or lack of maintenance, Thai NLP reseachers had to spend their limited time and resources building basic components and/or collecting a dataset before they could proceed further for more advanced problems. The limited availability of source codes and datasets also affects reproducibility.

Examples of Thai NLP tools and datasets before PyThaiNLP:

- **Word tokenization:** ICU BreakIterator (IBM Corporation et al., 1999) [Unicode License] based on Gillam (1999), LibThai (Thai Linux Working Group, 2001) [LGPL], KU Wordcut (Sudprasert and Kawtrakul, 2003) [GPL], SWATH (Charoenpornsawat, 2003) [GPL] based on Meknavin et al. (1997), LexTo (National Electronics and Computer Technology Center, 2006) [LGPL], OpenNLP (Bierner et al., 2008) [LGPL], TLex (Haruechaiyasak and Kongyoung, 2009) [Freeware], and wordcutpy (Satayamas, 2015) [LGPL]. Haruechaiyasak et al. (2008) provided a comparative study of some of these tools.

- **Part-of-speech (POS) tagging:** OpenNLP and RDRPOSTagger (Nguyen et al., 2014) [GPL] support Thai POS tagging. There are corpora such as ORCHID (Sornlertlamvanich et al., 1999) and NAiST (Kawtrakul et al., 2002) which provide not only POS but also word boundaries.

- **Named-entity recognition (NER):** Polyglot (Al-Rfou, 2015) [GPL], a multilingual NLP software, supports Thai NER based on Al-Rfou et al. (2015). For datasets, BEST-2009 corpus (Kosawat et al., 2009) is available but cannot be used commercially, as its license is Creative Commons Attribution-NonCommercial-ShareAlike Public License.

- **Automatic speech recognition (ASR):** Thai Language Audio Resource Center (Thai ARC) corpus (Hoonchamlong et al., 1997) provides audio recordings of dialects and speech styles, with transcripts; it is not designed specifically for ASR. NECTEC-ATR (Kasuriya et al., 2003a), LOTUS (Kasuriya et al., 2003b), LOTUS-BN (Chotimongkol et al., 2009), LOTUS-Cell (Chotimongkol et al., 2010), CU-MFEC (Kertkeidkachorn et al., 2012) and TSync-2 are ASR corpora for different domains and tasks; their licenses are not fully open. See Charoenporn et al. (2004), Wutiwiwatchai and Furui (2007), and Kertkeidkachorn et al. (2012) for reviews.

Apart from the ones listed above, more open-source Thai word tokenizers were released after 2009 as a result of BEST (Benchmark for Enhancing the Standard of Thai language processing) evaluation for Thai word segmentation organized by the National Electronics and Computer Technology Center (NECTEC) in 2009 (Kosawat, 2009), and 2010[6]. Unfortunately, these tokenizers are no longer maintained and are not accessible at the time of writing. The most impactful contribution from BEST, however, is the BEST-2010 word segmentation dataset that was publicly released. This dataset provides a basis for a lot of modern Thai open-source word segmentation software.

We should also mention the Thai Language Toolkit (TLTK) (Aroonmanakun and Thamrongrattanarit, 2018). While releasing its source code a few years after, it is richer in features than PyThaiNLP at the time. Its first release on Python Package Index (version 0.3.4, February 2018) includes statistical syllable and word segmentation (Aroonmanakun, 2002), POS tagging, and spelling suggestion. Its latest version, as

[3] https://opennlp.sourceforge.net/models-1.4. Its README from December 2008 also mentioned Thai components: https://web.archive.org/web/20081219153426/http://opennlp.sourceforge.net/README.html

[4] https://opennlp.sourceforge.net/models-1.5. Arreerard et al. (2022), however, reports that Apache OpenNLP supports these basic Thai NLP tasks: word tokenization, part-of-speech tagging, and sentence detection.

[5] Even today, this practice continues: take, for instance, the LST20 corpus from NECTEC, which has multiple layers of linguistic annotation. However, the free version can only be used for non-commercial purposes. See https://opend-portal.nectec.or.th/en/dataset/lst20-corpus.

[6] https://thailang.nectec.or.th/archive/indexa290.html

of writing, features discourse unit segmentation, NER, grapheme-to-phoneme conversion, IPA transcription, romanization, and more. To date, TLTK and PyThaiNLP are the only two comprehensive Thai NLP libraries for Python. However, TLTK's documentation is still quite limited.

3 PyThaiNLP and Its Ecosystem

Our primary objective is to ensure the user-friendliness and simplicity of the library. Drawing inspiration from NLTK, we follow numerous established interfaces. For example, `word_tokenize` and `pos_tag`. In addition, we also create datasets and pre-trained models for the Thai language. Figure 1 illustrates the overview of PyThaiNLP's functionalities and its ecosystem. Table 1 displays the development milestones of PyThaiNLP.

We will discuss here only popular features and major datasets/models.

3.1 Features

3.1.1 Word and Sentence Tokenization

PyThaiNLP supports many word tokenization algorithms.[7] The default algorithm is NewMM which is dictionary-based maximum matching (Sornlertlamvanich, 1993) and utilizes Thai character cluster (Theeramunkong et al., 2000). The pure-Python tokenizer performs reasonably well on public benchmarks. Chormai et al. (2020) demonstrated that it is the fastest word tokenizer on the BEST 2010 benchmark, with 71.18 % accuracy (compared to state-of-the-art at 95.60 %). Thanathip Suntorntip ported NewMM to Rust programming language[8], resulting in an even faster word tokenizer in our toolbox.

For sentence tokenization, we trained a conditional random field (CRF) model, using python-crfsuite (Peng and Korobov, 2014), on translated TED transcripts and Thai sentence boundaries are assumed to be denoted by English sentence boundaries (Lowphansirikul et al., 2021b).

3.1.2 Spell Checking

For spell checking, we have many engines; the Norvig (2007) one uses a spelling dictionary from Thai National Corpus (Aroonmanakun et al., 2009), symspellpy (mmb L, 2018) that is a Python port of SymSpell v6.7.1, and phunspell (Wright, 2021) that is a port of Hunspell.

3.1.3 Phonetic Algorithm and Transliteration

PyThaiNLP supports a couple of grapheme-to-phoneme (g2p) conversion engines. We trained Thai-g2p model with data from Wiktionary[9], a free online dictionary.

PyThaiNLP implemented many Thai Soundex algorithms. For example, Lorchirachoonkul (1982), Udompanich (1983), Thai-English cross-language Soundex (Suwanvisat and Prasitjutrakul, 1998), and MetaSound (Metaphone-Soundex combination) (Snae and Brückner, 2009).

PyThaiNLP supports the following transliteration implementations: Thai romanization using the Royal Thai General System of Transcription (RTGS), transliteration of romanized Japanese/Korean/Mandarin/Vietnamese texts to Thai using Wunsen library (cakimpei, 2022)[10], and Thai word pronunciation.

3.1.4 Sequence Tagging (NER and POS)

We create a named-entity recognition model called Thai NER (Phatthiyaphaibun, 2022) by finetuning the WangchanBERTa model (Lowphansirikul et al., 2021a) and CRF model.

For part-of-speech tagging, we trained a CRF tagger, a perceptron tagger (Honnibal, 2013), a unigram tagger, and finetuned the WangchanBERTa model. The POS training sets are derived from ORCHID corpus (Sornlertlamvanich et al., 1999), Blackboard Treebank annotated based on the LST20 Annotation Guideline (Boonkwan et al., 2020), and Parallel Universal Dependencies (PUD) treebanks (Smith et al., 2018).

3.1.5 Coreference Resolution and Entity Linking

For coreference resolution, we create Han-Coref, a Thai coreference resolution corpus and model (Phatthiyaphaibun and Limkonchotiwat, 2023).

For entity linking, PyThaiNLP supports it using BELA model (Plekhanov et al., 2023).

3.1.6 Word Embeddings

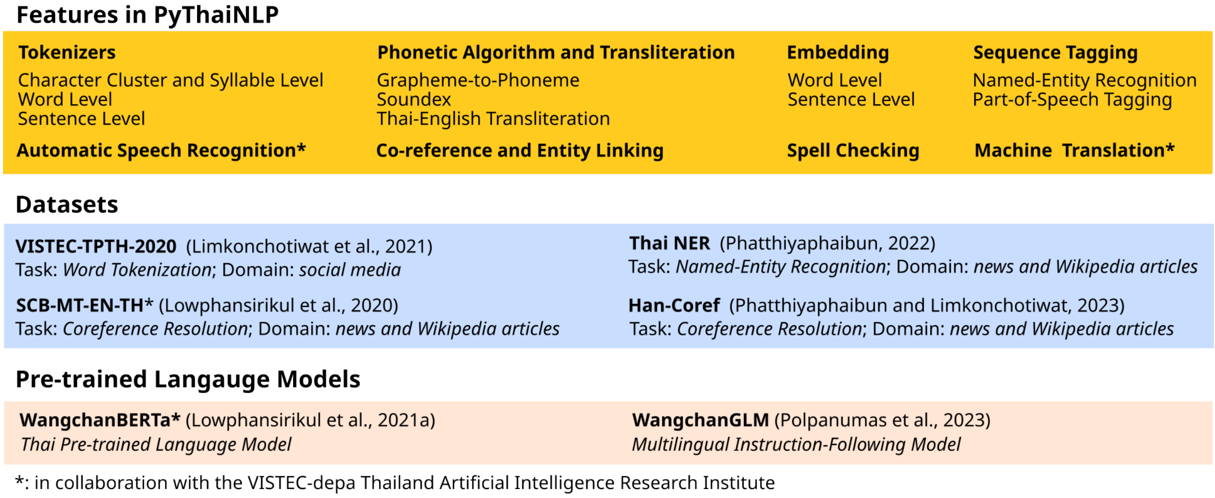
We extract token embeddings from our thai2fit (Polpanumas and Phatthiyaphaibun, 2021), a

[7] For the ease of experimenting with different word tokenization algorithms, Pattarawat Chormai has created a Thai word tokenizers collection as a Docker container image: `https://github.com/PyThaiNLP/docker-thai-tokenizers`.

[8] `https://github.com/pythainlp/nlpo3`

[9] `https://www.wiktionary.org/`

[10] The library implements various transliteration systems that recommended by the Royal Society of Thailand.



Features in PyThaiNLP

Tokenizers
Character Cluster and Syllable Level
Word Level
Sentence Level

Phonetic Algorithm and Transliteration
Grapheme-to-Phoneme
Soundex
Thai-English Transliteration

Embedding
Word Level
Sentence Level

Sequence Tagging
Named-Entity Recognition
Part-of-Speech Tagging

Automatic Speech Recognition*

Co-reference and Entity Linking

Spell Checking

Machine Translation*

Datasets

VISTEC-TPTH-2020 (Limkonchotiwat et al., 2021)
Task: *Word Tokenization*; Domain: *social media*

SCB-MT-EN-TH* (Lowphansirikul et al., 2020)
Task: *Coreference Resolution*; Domain: *news and Wikipedia articles*

Thai NER (Phatthiyaphaibun, 2022)
Task: *Named-Entity Recognition*; Domain: *news and Wikipedia articles*

Han-Coref (Phatthiyaphaibun and Limkonchotiwat, 2023)
Task: *Coreference Resolution*; Domain: *news and Wikipedia articles*

Pre-trained Langauge Models

WangchanBERTa* (Lowphansirikul et al., 2021a)
Thai Pre-trained Language Model

WangchanGLM (Polpanumas et al., 2023)
Multilingual Instruction-Following Model

*: in collaboration with the VISTEC-depa Thailand Artificial Intelligence Research Institute

Figure 1: Functionalities, datasets, and pre-trained language models available in PyThaiNLP's ecosystem.

word-level ULMFiT language model (Howard and Ruder, 2018) (Howard and Gugger, 2020) trained on Thai Wikipedia, and use them as word embeddings for PyThaiNLP. It was the state-of-the-art pre-trained model in many Thai classification benchmarks (Polpanumas and Suwansri, 2020) before the multilingual BERT model was released (PyCon Thailand, 2019).

3.1.7 Machine Translation

We collaborated with VISTEC-depa Thailand Artificial Intelligence Research Institute (AIResearch.in.th)[11] to create the English-Thai translation dataset and model. The model outperformed Google Translate on an out-of-sample test set at the time of release (Lowphansirikul et al., 2021b).

3.1.8 Automatic Speech Recognition

In order to develop a dataset for ASR, PyThaiNLP members contribute to the development of Common Voice corpus (Ardila et al., 2020), including Thai sentence cleanup and validation rules for its Sentence Collector[12], an online campaign inviting people to contribute Thai sentences, and offline events for volunteers to contribute their voices and voice validation.

Utilizing Common Voice Corpus 7.0, we created a Thai ASR model in collaboration with AIResearch.in.th and achieved the lowest character error rate in a benchmark (VISTEC-depa AI Research Institute of Thailand, 2023).

3.2 Datasets

3.2.1 VISTEC-TPTH-2020: Word Tokenization, Spell Checking and Correction

VISTEC-TPTH-2020 is a Thai word tokenization and spell checking dataset in the social media domain, the largest one to date (Limkonchotiwat et al., 2021). We collected 50,000 sentences from top trending posts on Twitter in 2020 and selected only posts with substantial character counts. This dataset is a multi-task dataset, including mention detection, spell checking, and spell correction.

3.2.2 Thai NER: Named Entity Recognition

Thai NER is a Thai named-entity recognition dataset. We curated text from various domains including news, Wikipedia articles, government documents, as well as text from other Thai NER datasets. The data is manually re-labeled for consistency (Phatthiyaphaibun, 2022).

3.2.3 Han-Coref: Coreference Resolution

Han-Coref is a coreference resolution dataset containing 1,339 documents in news and Wikipedia domains (Phatthiyaphaibun and Limkonchotiwat, 2023).

3.2.4 scb-mt-en-th-2020: English-Thai Machine Translation

scb-mt-en-th-2020 is an English-Thai sentence pair dataset consisting of 1,001,752 text pairs

[11] AIResearch.in.th is an initiative co-funded by a research university and a government agency, namely Vidyasirimedhi Institute of Science and Technology (VISTEC) in Wang Chan, Rayong, and the Digital Economy Promotion Agency (depa) under the Ministry of Digital Economy and Society, to create AI infrastructure for Thailand.

[12] https://github.com/common-voice/sentence-collector

(Lowphansirikul et al., 2021b). It is a collaborative work with AIResearch.in.th.

3.3 Pre-trained Language Models

WangchanBERTa is an encoder-only pre-trained Thai language model. Based on public benchmarks, it is the current state-of-the-art (Lowphansirikul et al., 2021a). It is also a collaborative work with AIResearch.in.th.

WangChanGLM (Polpanumas et al., 2023) is a multilingual instruction-following model fine-tuned from XGLM (Lin et al., 2022).

4 Community and Project Milestones

4.1 Foundation Years (2016-2019)

Wannaphong Phatthiyaphaibun, a high school student at the time, created PyThaiNLP in 2016 as a hobby project. He wanted to create a simple Thai chatbot in Python. He used PyICU as a word tokenizer and soon found out that Thai language did not have a comprehensive NLP toolkit in Python like NLTK (Bird and Loper, 2004). He decided to create PyThaiNLP and hosted the project on GitHub[13].

After the first few official releases, following Korakot Chaovavanich's suggestion, a "Thai Natural Language Processing" group has been created as a public Facebook group[14]. This serves as a main venue to showcase PyThaiNLP's capabilities and a hub for Thai NLP researchers and practitioners to discuss the field. Today, the group has over 16,000 members and is Thailand's largest NLP interest group. This communication channel also performs a recruiting function for us. The first offline meetup of the group occurred in 24 May 2018 as a bird-of-a-feather session after a Data Science BKK meetup[15].

Many of our main contributors, such as Charin Polpanumas and Arthit Suriyawongkul organically joined the project from the community. At this stage, we created foundational capabilities such as word tokenization, part-of-speech tagging, subword tokenization, named-entity recognition, and word vectors. A lot of code cleaning, reorganization, and documentation also happened around 2018-2019. This included the adoption of PEP 484 type hints[16] and other Python best practices to make the code even more readable and facilitate off-line type checkers. The adoption of PyThaiNLP can be reflected by the number of stars on GitHub the project received over the years (Figure 2).

4.2 Gaining Resources for Large Language Models (2019-present)

The growing activity of PyThaiNLP development can be seen from the number of code commits to the Git repository, which reached its peak in Q4 2019[17]. In 2020, the project began a collaboration with AIResearch.in.th. Their main focus was to create and distribute open-source models and datasets. This collaboration has provided PyThaiNLP with computational resources we need to scale up our operations as well as additional developers for maintaining the project, such as Lalita Lowphansirikul.

Under the collaboration, we have built an English-Thai sentence pair dataset and the state-of-the-art English-Thai translation model (Lowphansirikul et al., 2021b), the RoBERTa-based monolingual language model WangchanBERTa (Lowphansirikul et al., 2021a), and most recently the multilingual instruction-following model WangChanGLM (Polpanumas et al., 2023).

Due to limited computational and human resources, we prioritize features with the highest impact-to-effort ratio. For example, during 2019-2020, there were two types of dominant transformer-based language models: encoder-only BERT family and decoder-only GPT family. We opted to pursue the encoder-only models and trained WangchanBERTa because, at the time, it required relatively fewer resources to train and had better performance across impactful tasks such as text classification, sequence tagging, and extractive question answering. It was not until decoder-only models proved to create more value-added in 2022 that we started to train such models as WangChanGLM.

4.3 Community and Infrastructure for Software Quality

It is important to be noted that the community not only made contributions in the form of feature improvements but also in the areas of documenta-

[13] https://github.com/pythainlp/pythainlp

[14] https://www.facebook.com/groups/thainlp

[15] https://www.facebook.com/groups/thainlp/permalink/564348637279964/

[16] https://peps.python.org/pep-0484/

[17] https://github.com/PyThaiNLP/pythainlp/graphs/contributors

Years	Notable Features
2016	Word tokenization, part-of-speech tagging
2017	Soundex, spell checking, WordNet support
2018	Text classification language model, NER corpus/model, date and time parsing/formatting
2019	Syllable tokenization, date and time spell out
2020	ASR model, machine translation dataset/model, grapheme-to-phoneme conversion
2021	Autoencoding language model, word-to-phoneme conversion
2022	Dependency parsing, nested NER, text augmentation
2023	Coreference resolution dataset/model, generative language model

Table 1: Notable features introduced to PyThaiNLP over the years.

tion, including computational documentation (e.g., Jupyter notebooks), improving code quality and test suite, and streamlining software testing and delivery. Some of which may not be visible to the users but are crucial for the development of the project.

On the infrastructure side, test automation and continuous integration (CI) helps us systematically reinforce code style, detect code security vulnerabilities, maintain code coverage, and test the library in different computer configurations.

We were since 2017 rely on free Travis CI[18] and AppVeyor[19] for continuous integration workflow and later in June 2020 completely migrated to GitHub Actions[20]. Every GitHub pull requests will go through Black[21] for code formatting and Flake8[22] for PEP 8 code style[23] and cyclomatic complexity checks (McCabe, 1976). pip installation package will be built and tested against the test suite in Linux, macOS, and Windows[24]. The package then can be automatically publish to the Python Package Index directly from the CI, once it passed all the tests in every platform.

PyThaiNLP code coverage reached 80 % towards the end of 2018, compare to under 60 % in 2017. Code coverage is a metric that can help assess the quality of the test suite, and it therefore reflects how well the functionalities are thoroughly tested. The coverage went over 90 % in August 2019 and kept stable at this level until 2022[25].

From early 2022, we experienced a gradual drop of the code coverage to 80 %. The main reason is a growing number of features that require a large language model that cannot fit inside our standard GitHub-hosted runners. We have to remove some of the tests for those features. Before 2022, we also tested our library against versions of CPython and PyPy, but now it has been reduced to only CPython 3.8 due to the lack of support for other Python versions in some of our machine learning dependencies.

Some of the common code improvements we made after analyzing code coverage and other tests were the removal of unused code, fixing inconsistent behavior in different operating systems, better handling of a very long string, empty string, empty list, null, and/or negative values, and better handling of exceptions in control flow, resulting a code that is smaller and more robust.

5 PyThaiNLP in the Wild

5.1 PyThaiNLP and Its Research Impact

Researchers worldwide use PyThaiNLP to work with Thai language. For instance, for word tokenization in cross-lingual language model pretraining (Lample and Conneau, 2019), universal dependency parsing (Smith et al., 2018), and cross-lingual representation learning (Conneau et al., 2020). In addition, research and industry-grade tools namely SEACoreNLP[26], an open-source initiative by NLPHub of AI Singapore, and spaCy (Honnibal et al., 2020) include PyThaiNLP as part of their toolkit.

[18] https://www.travis-ci.com/

[19] https://www.appveyor.com/

[20] https://github.com/features/actions

[21] https://github.com/psf/black

[22] https://flake8.pycqa.org

[23] https://peps.python.org/pep-0008/

[24] Easy installation and consistent behavior across platforms are what we aim for. This is one of the reasons why we developed a pure-Python NewMM. The previous implementation of our default word tokenizer requires marisa-trie, a trie data structure library in C++. Unfortunately, marisa-trie does not officially support mingw32 compiler on Windows.

[25] Our code coverage is measured by coverage.py which is included in our continuous integration workflow. The coverage stats are made available online by Coveralls at: https://coveralls.io/github/PyThaiNLP/pythainlp

[26] https://seacorenlp.aisingapore.net/docs/

5.2 PyThaiNLP and Its Industry Impact

PyThaiNLP is used in many real-world business use cases in firms of all sizes both domestic and international. User feedback generally highlights how the library has sped up their product development cycles involving Thai NLP as well as its effectiveness in terms of business outcomes. The most frequently used functionalities are tokenization and text normalization. We introduce here selected use cases from national and multinational firms in banking, telecommunication, insurance, retail, and software development.

Siam Commercial Bank (BKK:SCB; USD 10B market cap) is one of Thailand's largest banks. The bank operates a chatbot to automatically answer customer queries. Their data analytics team finetuned WangchanBERTa for intent classification to enhance its question-answering capabilities as well as to detect personal information in customers' inputs in order to exclude them from their internal training sets. Moreover, the team relies on basic text processing functions such as tokenization and normalization to speed up their development process. They have also found the published performance benchmarks to be useful when selecting models for their tasks.

True Corporation (BKK:TRUE; 6B) is one of the two providers in Thailand's duopoly telecommunication market. Its subsidiary, True Digital Group, uses PyThaiNLP both for digital media analysis and for recommendation engine on production. They featurized their Thai-text contents using thai2fit word vectors and saw a noticeable uplift in user engagement and subsequent business outcomes. They also combined our word vectors with Top2Vec (Angelov, 2020) to perform topic modeling and improve customer experience.

Central Retail Digital (BKK:CRC; 6B) is a digital transformation unit serving Central Retail, Thailand's largest department store. Their data science team used PyThaiNLP mainly to enhance search and recommendation offerings across five business units and other six million customers. Word tokenization and text normalization were used to preprocess product information and search queries as input for the product search system. Since most search systems are built for languages with white spaces as word delimiters, this preprocessing step has allowed their product search to outperform out-of-the-box solutions which are not compatible with Thai. For content-based recommendations, the team featurized production information to create a model that recommends similar products to customers.

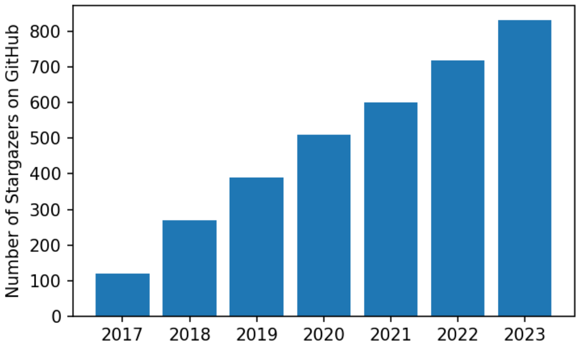


Figure 2: Number of stars PyThaiNLP has received from GitHub users over the years.

AIA Thailand (HKG:1299; 109B global) is the Thai headquarter of the global insurance firm American Insurance Association. Their data science team employs PyThaiNLP in analyzing their inbound and outbound call logs using word tokenization, text normalization, stop word handling, and local-time-format string handling functionalities. For the inbound calls, they normalize and tokenize the logs to perform topic modeling and identify critical topics of conversation to emphasize both automated voice bot and human staff training and allocation. This resulted in improved percentage of calls that the voice bot fulfilled successfully and reduced call waiting time. For the outbound calls, they perform keyword identification from the logs processed by PyThaiNLP to gain insights to improve customer retention.

VISAI is a VISTEC university spin-off that provides machine learning tools and consulting services. It has finetuned WangchanBERTa to perform text classification, named entity recognition, and relation extraction on unstructured data of their clients to create a queryable knowledge graph. They also use tokenization and text normalization functionalities to facilitate text processing for all their NLP-based products.

6 Conclusion and Future Works

This paper introduces the PyThaiNLP library, explains its features and datasets (as illustrated in Figure 1), and discusses the community and the engineering project supporting the library.

By 2023, we will have implemented the open-source version of most general NLP capabilities available in English for Thai[27]. We see the following items as the next major milestones:

- **Domain-specific datasets/models** Some capabilities are not performing well on specific use cases; for instances, named-entity recognition in financial reports, medical terms translation, and legal documents question answering. We believe more domain-specific datasets and models will help close this gap.

- **Robust benchmark for Thai NLP tasks** As NLP has garnered more attention, more models and datasets, both open- and closed-source, will be available. It will, therefore, be imperative to have a robust benchmark in comparing the models' performance and the datasets' quality.

- **Correctness and consistency** Search key generation (such as Soundex), sorting, and tokenization[28] have to be deterministic and strictly follow a specification, or an application may behave in an unexpected fashion. More test cases and verification might be needed for these features.

- **Efficient mechanism to load and manage datasets/models** To reduce the size of the library and to carter the use in a system with a restricted network connection[29].

- **Seamless integration with language-agnostic tools** The ultimate goal is for developers to no longer need PyThaiNLP as Thai language is supported by standard NLP libraries such as spaCy and Hugging Face (Wolf et al., 2020). We have begun this work with integrating our text processing functions and models to spaCy.

Acknowledgements

First and foremost, we appreciate the contributions from all PyThaiNLP contributors[30]. We would like to thank: 1) VISTEC-depa Thailand AI Research Institute and its director Sarana Nutanong for research collaboration and support in terms of academic guidance, computational resources, and personnel; 2) the companies featured in the industry impact section and respective interviewees Chrisada Sookdhis, Jayakorn Vongkulbhisal, Kowin Kulruchakorn, Phasathorn Suwansri, and Pongtachchai Panachaiboonpipop; 3) Ekapol Chuangsuwanich for academic guidance and contribution to models and datasets; and 4) MacStadium for infrastructure support. We are much obliged to free and open-source software community for software building blocks and best practices, including but not limited to NumFOCUS, fast.ai, Hugging Face, and Thai Linux Working Group. Moreover, we thank organizations who care enough to develop multilingual resources to accommodate low-resource languages, most notably Meta AI. Lastly, we cannot thank enough volunteers of various open-content communities, including Wikipedia, Common Voice, TED Translators, and similar local initiatives; modern NLP will not be possible without their accumulated effort.

Limitations

In our current CI workflow, every code commit to the repository triggers an automated test suit for all supported platforms. The process can be challenging if our package depends on large language models (LLMs) because a single LLM can exhaust the memory of our free-tier CI infrastructure. Some of the components can be cached to reduce build time, but they have to be loaded to the memory in any case. This forced us to drop some LLM-related tests and scarified the code coverage of the library as discussed in Section 4.3.

Even we have a resource to do such tests with the current design, it is neither economical nor sustainable. An improved test utilizing a stub, mock, or spy (proxy) test pattern that provides an off-line "fake inference" can help this. These techniques have been proven useful in other software testing involving expensive database/API queries or network connections. Lyra (2019) and Microsoft (2020) provide such examples, using the Python Standard Library's `unittest.mock`. This can reduce a number of times an LLM is actually being loaded/called. The required inference could be handled either by a non-free tier CI plan from the same or different provider (which should be more affordable now due to reduced number of calls) or by a computer outside the cloud.

[27] https://nlpforthai.com/

[28] Some phonetic algorithm and transliteration rely on syllable tokenization

[29] https://github.com/PyThaiNLP/pythainlp/issues/298

[30] https://github.com/PyThaiNLP/pythainlp/graphs/contributors

References

Rami Al-Rfou. 2015. Polyglot. Available at https://pypi.org/project/polyglot/.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. POLYGLOT-NER: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Dimo Angelov. 2020. Top2Vec: Distributed representations of topics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Wirote Aroonmanakun. 2002. Collocation and Thai word segmentation. In *Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop*, pages 68–75, Pathumthani, Thailand. Sirindhorn International Institute of Technology.

Wirote Aroonmanakun, Kachen Tansiri, and Pairit Nittayanuparp. 2009. Thai National Corpus: A progress report. In *Proceedings of the 7th Workshop on Asian Language Resources*, ALR7, page 153158, USA. Association for Computational Linguistics.

Wirote Aroonmanakun and Attapol Thamrongrattanarit. 2018. Thai Language Toolkit. Available at https://pypi.org/project/tltk/.

Ratchakrit Arreerard, Stephen Mander, and Scott Piao. 2022. Survey on Thai NLP language resources and tools. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6495–6505, Marseille, France. European Language Resources Association.

Gann Bierner, Jason Baldridge, Thomas Morton, and Joern Kottmann. 2008. OpenNLP. Available at https://sourceforge.net/projects/opennlp/.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Prachya Boonkwan, Vorapon Luantangsrisuk, Sitthaa Phaholphinyo, Kanyanat Kriengket, Dhanon Leenoi, Charun Phrombut, Monthika Boriboon, Krit Kosawat, and Thepchai Supnithi. 2020. The annotation guideline of LST20 corpus.

cakimpei. 2022. Wunsen. Available at https://github.com/cakimpei/wunsen.

Thatsanee Charoenporn, Virach Sornlertlamvanich, Sawit Kasuriya, Chatchawarn Hansakunbuntheung, and Hitoshi Isahara. 2004. Open collaborative development of the Thai language resources for natural language processing. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Paisarn Charoenpornsawat. 2003. SWATH: Smart Word Analysis for THai. Available at http://www.cs.cmu.edu/~paisarn/software.html.

Pattarawat Chormai, Ponrawee Prasertsom, Jin Cheevaprawatdomrong, and Attapol Rutherford. 2020. Syllable-based neural Thai word segmentation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4619–4637, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ananlada Chotimongkol, Kwanchiva Saykhum, Patcharika Chootrakool, Nattanun Thatphithakkul, and Chai Wutiwiwatchai. 2009. LOTUS-BN: A Thai broadcast news corpus and its research applications. In *2009 Oriental-COCOSDA International Conference on Speech Database and Assessments*, pages 44–50, Urumqi, China.

Ananlada Chotimongkol, Nattanun Thatphithakkul, Sumonmas Purodakananda, Chai Wutiwiwatchai, Patcharika Chootrakool, Chatchawarn Hansakunbuntheung, Atiwong Suchato, and Panuthat Boonpramuk. 2010. The development of a large Thai telephone speech corpus: LOTUS-Cell 2.0. In *2010 Oriental-COCOSDA International Conference on Speech Database and Assessments*, Kathmandu, Nepal.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

David Eberhard, Gary Simons, and Chuck Fennig. 2023. *Ethnologue: Languages of the World. Twenty-sixth edition*. SIL International.

Richard Gillam. 1999. Text boundary analysis in Java. In *Proceedings of Fifteenth International Unicode Conference*, San Jose, California, USA.

Choochart Haruechaiyasak and Sarawoot Kongyoung. 2009. TLex: Thai lexeme analyser based on the conditional random fields. In *Proceedings of 8th International Symposium on Natural Language Processing*, Bangkok, Thailand.

Choochart Haruechaiyasak, Sarawoot Kongyoung, and Matthew Dailey. 2008. A comparative study on

Thai word segmentation approaches. In *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 1, pages 125–128.

Matthew Honnibal. 2013. A good part-of-speech tagger in about 200 lines of Python.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Yuphaphann Hoonchamlong, Sathaporn Koraksawet, Sarawuth Keawbumrung, and Krissadang Klaijinda. 1997. Thai Language Audio Resource Center.

Jeremy Howard and Sylvain Gugger. 2020. fastai: A Layered API for Deep Learning. *Information*, 11(2):108.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

IBM Corporation et al. 1999. International Components for Unicode. Available at `https://icu.unicode.org`.

Sawit Kasuriya, Virach Sornlertlamvanich, Patcharika Cotsomrong, Takatoshi Jitsuhiro, Genichiro Kikui, and Yoshinori Sagisaka. 2003a. NECTEC-ATR Thai speech corpus. In *2003 Oriental-COCOSDA International Conference on Speech Database and Assessments*, pages 105–111, Singapore.

Sawit Kasuriya, Virach Sornlertlamvanich, Patcharika Cotsomrong, Supphanat Kanokphara, and Nattanun Thatphithakkul. 2003b. Thai speech corpus for speech recognition. In *2003 Oriental-COCOSDA International Conference on Speech Database and Assessments*, pages 54–61, Singapore.

Asanee Kawtrakul, Mukda Suktarachan, Patcharee Varasai, and Hutchatai Chanlekha. 2002. A state of the art of Thai language resources and Thai language behavior analysis and modeling. In *COLING-02: The 3rd Workshop on Asian Language Resources and International Standardization*.

Natthawut Kertkeidkachorn, Supadaech Chanjaradwichai, Teera Suri, Krerksak Likitsupin, Surapol Vorapatratorn, Pawanrat Hirankan, Worasa Limpanadusadee, Supakit Chuetanapinyo, Kitanan Pitakpawatkul, Natnarong Puangsri, Nathacha Tangsirirat, Konlawachara Trakulsuk, Proadpran Punyabukkana, and Atiwong Suchato. 2012. The CU-MFEC corpus for Thai and English spelling speech recognition. In *Proceedings of International Conference on Speech Database and Assessments*, pages 18–23.

Krit Kosawat. 2009. InterBEST 2009: Thai word segmentation workshop. In *Proceedings of 8th International Symposium on Natural Language Processing*, Bangkok, Thailand.

Krit Kosawat, Monthika Boriboon, Patcharika Chootrakool, Ananlada Chotimongkol, Supon Klaithin, Sarawoot Kongyoung, Kanyanut Kriengket, Sitthaa Phaholphinyo, Sumonmas Purodakananda, Tipraporn Thanakulwarapas, and Chai Wutiwiwatchai. 2009. BEST 2009: Thai word segmentation software contest. In *2009 Eighth International Symposium on Natural Language Processing*, pages 83–88.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Peerat Limkonchotiwat, Wannaphong Phatthiyaphaibun, Raheem Sarwar, Ekapol Chuangsuwanich, and Sarana Nutanong. 2021. Handling cross- and out-of-domain samples in Thai word segmentation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1003–1016, Online. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vichit Lorchirachoonkul. 1982. A Thai soundex system. *Information Processing & Management*, 18(5):243–255.

Lalita Lowphansirikul, Charin Polpanumas, Nawat Jantrakulchai, and Sarana Nutanong. 2021a. WangchanBERTa: pretraining transformer-based Thai language models.

Lalita Lowphansirikul, Charin Polpanumas, Attapol T. Rutherford, and Sarana Nutanong. 2021b. A large English–Thai parallel corpus from the web and machine-generated text. *Language Resources and Evaluation*, 56(2):477–499.

Matti Lyra. 2019. Effective mocking of unit tests for machine learning.

Thomas J. McCabe. 1976. A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4):308–320.

Surapant Meknavin, Paisarn Charoenpornsawat, and Boonserm Kijsirikul. 1997. Feature-based Thai Word Segmentation. In *Proceedings of the Natural Language Processing Pacific Rim Symposium*, Phuket, Thailand.

Microsoft. 2020. Testing data science and MLOps code.

mmb L. 2018. symspellpy. Available at https://github.com/mammothb/symspellpy.

National Electronics and Computer Technology Center. 2006. Thai Lexeme Tokenizer: LexTo. [online]. Retrieved August 8, 2023, from http://www.sansarn.com/lexto/.

Dat Quoc Nguyen, Dai Quoc Nguyen, Dang Duc Pham, and Son Bao Pham. 2014. RDRPOSTagger: A ripple down rules-based part-of-speech tagger. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 17–20, Gothenburg, Sweden. Association for Computational Linguistics.

Peter Norvig. 2007. How to write a spelling corrector.

Terry Peng and Mikhail Korobov. 2014. python-crfsuite. Available at https://github.com/scrapinghub/python-crfsuite.

Wannaphong Phatthiyaphaibun. 2022. Thai NER 2.0.

Wannaphong Phatthiyaphaibun and Peerat Limkonchotiwat. 2023. Han-Coref: Thai coreference resolution by PyThaiNLP.

Mikhail Plekhanov, Nora Kassner, Kashyap Popat, Louis Martin, Simone Merello, Borislav Kozlovskii, Frédéric A. Dreyer, and Nicola Cancedda. 2023. Multilingual end to end entity linking.

Charin Polpanumas and Wannaphong Phatthiyaphaibun. 2021. thai2fit: Thai language implementation of ULMFiT.

Charin Polpanumas, Wannaphong Phatthiyaphaibun, Patomporn Payoungkhamdee, Peerat Limkonchotiwat, Lalita Lowphansirikul, Can Udomcharoenchaikit, Titipat Achakulwisut, Ekapol Chuangsuwanich, and Sarana Nutanong. 2023. WangChanGLM – the multilingual instruction-following model.

Charin Polpanumas and Phasathorn Suwansri. 2020. Pythainlp/classification-benchmarks: v0.1-alpha.

PyCon Thailand. 2019. How PyThaiNLP's thai2fit outperforms Google's BERT: State-of-the-art Thai text classification and beyond - Charin.

Vee Satayamas. 2015. wordcutpy. Available at https://github.com/veer66/wordcutpy.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal Dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Chakkrit Snae and Michael Brückner. 2009. Novel phonetic name matching algorithm with a statistical ontology for analysing names given in accordance with Thai astrology. *Issues in Informing Science and Information Technology*, 6:497–515.

Virach Sornlertlamvanich. 1993. *Machine Translation*, chapter Word segmentation for Thai in machine translation system. National Electronics and Computer Technology Center.

Virach Sornlertlamvanich, Tanapong Potipiti, Chai Wutiwiwatchai, and Pradit Mittrapiyanuruk. 2000. The state of the art in Thai language processing. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 1–2, Hong Kong. Association for Computational Linguistics.

Virach Sornlertlamvanich, Naoto Takahashi, and Hitoshi Isahara. 1999. Building a Thai part-of-speech tagged corpus (ORCHID). *Journal of the Acoustical Society of Japan (E)*, 20(3):189–198.

Sutee Sudprasert and Asanee Kawtrakul. 2003. Thai word segmentation based on global and local unsupervised learning. In *Proceedings of the 7th National Computer Science and Engineering Conference*, pages 1–8, Chonburi, Thailand.

Thepchai Supnithi, Krit Kosawat, Monthika Boriboon, and Virach Sornlertlamvanich. 2004. Language sense and ambiguity in Thai. In *Proceedings of the 8th Pacific Rim International Conference on Artificial Intelligence*, Auckland, New Zealand.

Prayut Suwanvisat and Somboon Prasitjutrakul. 1998. Thai-English cross-language transliterated word retrieval using soundex technique. In *Proceesings of the National Computer Science and Engineering Conference*, Bangkok, Thailand.

Thai Linux Working Group. 2001. LibThai. Available at https://linux.thai.net/projects/libthai/.

Thanaruk Theeramunkong, Virach Sornlertlamvanich, Thanasan Tanhermhong, and Wirat Chinnan. 2000. Character cluster based Thai information retrieval. In *Proceedings of the Fifth International Workshop on on Information Retrieval with Asian Languages*, IRAL '00, page 7580, New York, NY, USA. Association for Computing Machinery.

Wannee Udompanich. 1983. String searching for Thai alphabet using Soundex compression technique.

VISTEC-depa AI Research Institute of Thailand. 2023. wav2vec2-large-xlsr-53-th (revision 3155938).

David Gray Widder, Sarah West, and Meredith Whittaker. 2023. Open (for business): Big tech, concentrated power, and the political economy of open AI.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

David Wright. 2021. Phunspell. Available at `https://github.com/dvwright/phunspell`.

Chai Wutiwiwatchai and Sadaoki Furui. 2007. Thai speech processing technology: A review. *Speech Communication*, 49(1):8–27.