

PyThaiNLP Workshop

4th NIDA BADs
1 November 2019



(Louis) Chakri Lowphansirikul

Research Assistant
@**VISTEC-depa Thailand AI Research Institute**

www.aires.in.th

Outline

Talking session (~40 minutes)

- Natural Language Processing
 - Basic tasks
 - Downstreamed tasks
- NLP Libraries
- Thai NLP Library
 - Features of PyThaiNLP

Hand-On Workshop PyThaiNLP (~2 hours)

- Getting started with PyThaiNLP
- Text Classification on Truevoice dataset
- Text Clustering, Document similarity

What is Natural Language Processing?

Automatically process or
understand natural
language.

Basic Tasks

Word segmentation

“ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมօ เจียนบทและกำกับโดย นวพล ว่องรัตนฤทธิ์”



ฟรีแลนซ์ .. ห้าม ป่วย ห้าม พัก ห้าม รัก หมօ เจียน บท
และ กำกับ โดย นวพล ว่องรัตนฤทธิ์

Named Entity Recognition

“ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมօ เขียนบทและกำกับโดย นวพล ว่องไวทันฤทธิ์”

Movie: ฟรีแลนซ์ .. ห้าม ป่วย ห้าม พัก ห้าม รัก หมօ

เขียน บท และ กำกับ โดย Person: นวพล ว่องไวทันฤทธิ์

Part of Speech Tagging

“ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมօ เขียนบทและกำกับโดย นวพล ว่องไว้”

Movie: ฟรีแลนซ์ .. ห้าม ป่วย ห้าม พัก ห้าม รัก หมօ

N PUNCT V V V V V N

เขียน บท และ กำกับ โดย Person: นวพล ว่องไว้

V N CCONJ V ADP PROPN PROPN

N = Noun,

PUNCT = Punctuation,

CCONJ = Coordinating Conjunction

V = Verb,

ADP = Adposition,

PROPN = Proper Noun,

Subword segmentation

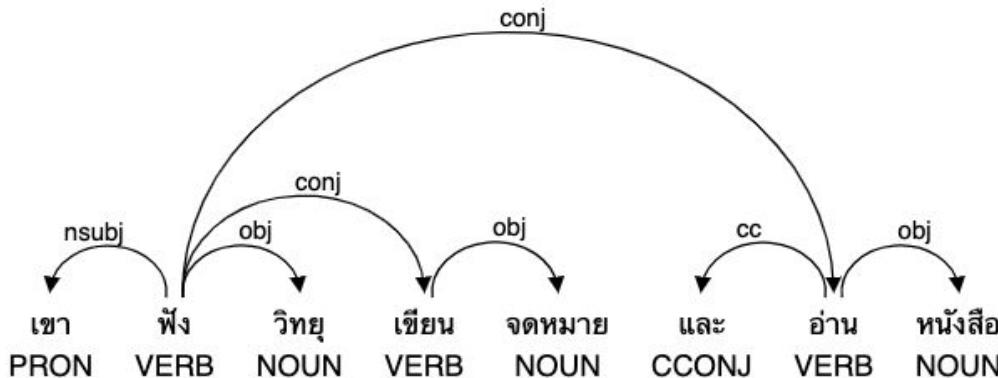
- Handle out of Vocabulary problem (person name, places, etc.)
- Applying Byte Pair Encoding (BPE) algorithm to group most frequency pair of adjacent characters.

“ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมօ เขียนบทและกำกับโดย นวพล ชั่รังรัตนฤทธิ์”

ฟรี แลน ซ์ .. ห้าม ป่วย _ ห้าม พัก _ ห้าม รัก หมօ[์]
_ เขียนบทและ กำกับโดย _ นว พล _ ชั่รัง รัตน ฤทธิ์

Dependency Parsing

- Extracting relationship between words that represent the grammatical structure of sentence



nsubj = Nominal subject,
conj = Conjunction,

obj = Object,
cc = Coordinating conjunction

Downstreamed Task

Information Retrieval

- Basis tasks like Word Segmentation and NER are essential for Information Retrieval system.
- Without proper segmentation can lead to Zero Search Results.



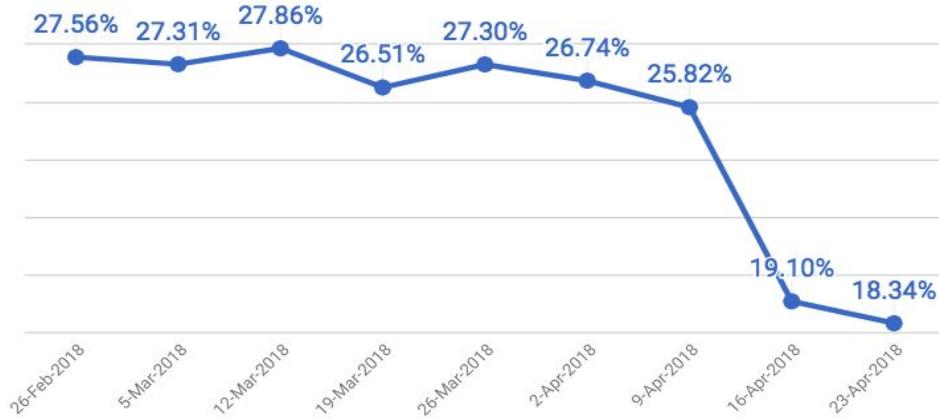
Without Word Segmentation, system can only search from only one keyword “รามงทองหล่อ”

With Word Segmentation, system can search from two keywords “รามง” and “ทองหล่อ”

Information Retrieval (Cont.)

“ จากการพัฒนาระบบ Search อย่างต่อเนื่องด้วยเทคนิค
หลักหลายรูปแบบ เช่น การใช้ตัวตัดคำ และ Spell
correction

เราสามารถลดจำนวน Weekly *zero search result rate*
ได้ถึงหนึ่งในสาม ซึ่งถือว่าเป็นจุดเริ่มต้นที่ดีในการพัฒนา
 เพราะเรายังมีอีกหลายเทคนิคที่จะมาแก้ไขปัญหานี้ ”



Text Classification

Quora Insincere Questions Classification

- Identify questions that are insincere (intended to make a statement rather than look for helpful answers)
- Based on false information, sexual content (incest, beastality, pedophilia), disparaging or inflammatory

Category	Example questions
Normal Question	What is the reason why we really need Bitcoin? What is the procedure to invest in mutual fund? How do I get a job in UX?
Insincere Question	Why do American firms steal Chinese technology? Why does NASA fake all their pictures? Why are liberals so stupid and dumb?

Sentiment Analysis

- Sentiment Analysis Dataset from Thai Social Media post by Wisesight (Thailand) Co., Ltd
- Classify posts in 4 categories (positive, negative, neutral or question)

Category	Example sentences
Positive	ว้าววว ชีสียืด ยืดดดดดดดดด ๘๘ , พรุ่งนี้หยุดวันพุธพอตี555 , ทีอ้อ..อยากลอง น่ากินอ่ะ
Neutral	ข้อ1แเณม1, ใช่แล้วเด้อ. วันนี้ไม่ว่าง
Negative	อันนี้น้ำจิ้มไม่อร่อย, เห็นคิวแท็กซี่แล้วห้อ, เน็ตเน่าบ่อยมาก สัญญาณเต็ม แต่เน็ตไม่วิ่ง
Question	4Gหรือ2G? , วันพุธมีโปรดัก20เปอเซ็น ได้ถึงกีทุ่ม,1000 สิทธิ์ทั่วประเทศไทยใหมครับ

Sentiment Analysis Dataset from Thai Social Media post by Wisesight:

<https://github.com/PyThaiNLP/wisesight-sentiment>

Sentiment Analysis (Cont.)

- Word Segmentation + Deep Learning Model (e.g. ULMFit)

Segment text into tokens

↓

Out[8]:	category	texts	processed	wc	uwc
0	pos	เด็กอยากกินเอมเคคค	เด็ก อยาก กิน เอม เค xxrep	6	6
1	neu	ถ้าเป็นมาสต้า....ต้องมาสต้าเชียงใหม่95	ถ้า เป็น มาส ต้า. xxrep ต้อง มาส ต้า เชียง ใหม่ 95	10	9
2	neu	มาเที่ยวเวียดนาม ในทริปมีพนุชด้วยอะ ที่รู้เพ...	มา เที่ยว เวียดนาม ใน ทริป มี พนุช ด้วย อะ ...	49	42
3	pos	รีวิวรองพื้น Marc Jacobจะหน่อย เมื่อวานไปสอยต...	รีวิว รองพื้น marc jacob จะ หน่อย เมื่อวาน ไป ...	136	104
4	neu	D. รถ Hilux รุ่น Rocco #HiluxRevoThailand	d .รถ hilux รุ่น rocco # hiluxrevothailand	8	8

Sentiment Analysis (Cont.)

The model outputs as class probabilities.

Probability (from 0.0 to 1.0)

Out[229]:

	category	preds	loss	neg	neu	pos	q	hit	texts	processed	wc
0	neu	neu	0.490570	0.093375	0.612277	0.292863	0.001485	True	กระเทียม?	กระเทียม ?	2
1	neu	neu	0.234147	0.006215	0.791245	0.199381	0.003158	True	ได้สีค่ะ รออะไร555	ได้สี ค่ะ รอ อะไร 5 xxrep	7
2	neu	neu	0.009935	0.001864	0.990114	0.007970	0.000052	True	❤️❤️❤️ หนูม กะลา คอนเสิร์ตที่เนคต้าผับ หาดใหญ่ 2...	❤️ หนูม กะลา คอนเสิร์ต ที่ เนค ตา ผับ หาด ...	51
3	neu	neu	0.112569	0.005808	0.893536	0.100359	0.000297	True	555555	5 xxrep	2
4	neu	neu	0.007246	0.000022	0.992780	0.005423	0.001775	True	สามารถสั่งกลับบ้านได้ ตามช่วงเวลาที่ให้บริการ ค...	สามารถ สั่ง กลับบ้าน ได้ ตาม ช่วงเวลา ที่ ให้บร...	10

Question Answering

- Standford Question Answering Dataset (SQuAD)

Predictions by BERT (single model) (Google AI Language)

Article EM: 82.6 F1: 86.1

Black_Death

The Stanford Question Answering Dataset

The Black Death is thought to have originated in the arid plains of Central Asia, where it then travelled along the Silk Road, reaching Crimea by 1343. From there, it was most likely carried by Oriental rat fleas living on the black rats that were regular passengers on merchant ships. Spreading throughout the Mediterranean and Europe, the Black Death is estimated to have killed 30–60% of Europe's total population. In total, the plague reduced the world population from an estimated 450 million down to 350–375 million in the 14th century. The world population as a whole did not recover to pre-plague levels until the 17th century. The plague recurred occasionally in Europe until the 19th century.

Where did the black death originate?

Ground Truth Answers: the arid plains of Central Asia Central Asia Central Asia

Prediction: Central Asia

How did the black death make it to the Mediterranean and Europe?

Ground Truth Answers: merchant ships. merchant ships Silk Road

Prediction: Spreading

Visualization of model prediction: ([link](#))

Thai Question Answering

Thai Question Answering Dataset (4,000 question-answer pairs) published by NECTEC.

```
▼ data:  
  ▼ 0:  
    question_id: 1  
    question: "นายกรัฐมนตรีคนที่ 7 ของประเทศไทยคือใคร"  
    answer: "ปริ๊ต พนมยงค์"  
    answer_begin_position : 178  
    answer_end_position: 191  
    article_id: 25946  
  ▼ 1:  
    question_id: 2  
    ▼question: "กีฬาประจำชาติแห่งแคนาดาที่อยู่ห่างไกลที่สุดคือกีฬาอะไร"  
    answer: "ซูโน่"  
    answer_begin_position : 65  
    answer_end_position: 70  
    article_id: 8324
```

Thai Question Answering (Cont.)

Apply DrQA, an Wikipedia-based English Question Answering System to Thai dataset (As a part of NSC 2019).

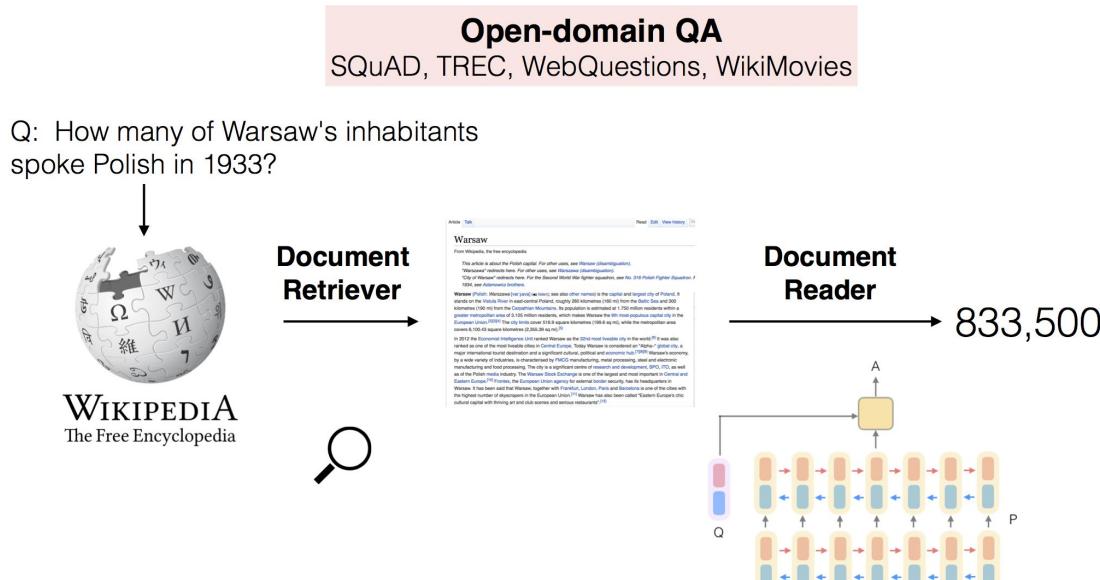


Image from: <https://github.com/facebookresearch/DrQA>

Thai Question Answering (Cont.)

Question:

ใครคือผู้ก่อตั้งวัดร่องข้น

Answer (rank: 1): ເຄີມຂໍຍ ໂພນິຕພັດນໍ (normalized score: 45.39)

วัดร่องขุ่น วัดร่องขุ่น เป็นวัดพุทธ ตั้งอยู่ในอำเภอเมืองเชียงราย จังหวัดเชียงราย ออกรอบแบบและก่อสร้าง โดย เฉลิมชัย ໂມບັດພິພັນ ตั้งแต่ พ.ศ. 2540 จนถึงปัจจุบัน โดยเฉลิมชัยคาดว่างานก่อสร้างวัดร่องขุ่นจะไม่เสร็จลงภายในช่วงชีวิตของตน วัดร่องขุ่นถือเป็นมาจากการก่อสร้างเมื่อ จังหวัดน่าน เมื่อวันที่ 5 พฤษภาคม พ.ศ. 2557 เวลา 18.05 น. ก็เดินผ่านดินไหวขนาด 6.3 มีศูนย์กลางอยู่ที่อำเภอแม่ล่า จังหวัดเชียงราย และแผ่นดินไหวตามหลาຍครึ้ง สิ่งความเสียหายให้กับวัดร่องขุ่นเป็นอย่างมาก เช่น ผนังโน斯ทีบูนกระเทาหัก กระเบื้องหัก ยอดพระธาตุหัก ภาพเขียนเสียหายหมด ทำให้ต้องปิดวัดเพื่อซ่อมแซม ตั้งแต่วันที่ 6 พฤษภาคม ปีเดียวganประวัติ ประวัติ อาจารย์ เฉลิมชัย ໂມບັດພິພັນได้รับแรงบันดาลใจในการสร้างวัดมาจาก 3 สิ่งต่อไปนี้คือ 1. ชาติ : ด้วยความรักบ้านเมือง รักงานศิลป์ จึงห่วงสร้างงานศิลปะที่ยั่งใหญ่ไว้เป็นสมบัติของแผ่นดิน 2. ศาสนา : ธรรมะได้เปลี่ยนชีวิตของอาจารย์เฉลิมชัยจากใจที่ร้อนกลายเป็นเย็น จึงขออุทิศตนให้แก่พระพุทธศาสนา 3. พระมหาภัตtriy : จากการเข้าเฝ้า ถวายงานพระบาทสมเด็จพระเจ้าอยู่หัวภูมิพลอดุลยเดชหลาຍครึ้ง ทำให้เฉลิมชัยรักพระองค์ท่านมาก จากการพบเห็นพระอัจฉริยะภาพทางศิลปะและพระเมตตาของพระองค์ท่าน จนบังเกิดความตื้นตันและสำนึกในพระมหากรุณาธิคุณ จึงประถนาที่จะสร้างงานพุทธศิลป์ถวายเป็นงานศิลปะประจำชาติประเทศของประเทศไทยของอุบลสักความหมายของอุบลส. - สีขาว : พระบูรีสุทธิคุณของพระพุทธเจ้า - สะพาน : การเดินข้ามจากวัวสังสารสู่พุทธภูมิ - เชือยว หรือ ปากพญาumar : กิเลสในใจ - สีของสะพาน : มีสุรอมกัน ข้างละ 8 ตัว 2 ข้าง รวมกันแนนอนกิเลส 16 - กิ่งกลางของสะพาน : เข้าพระสุเมรุ - ดอกบัวทิพย์ : มี 4 ดอก ใหญ่ตระหง่านชั้นด้านข้างอุบลสักแทนชั้มพระอริยเจ้า 4 พระองค์ คือ พระโสดาบัน พระสกิทาคามี พระอนาคตี และพระอรหันต์ - บันไดทางขึ้น : มี 3 ขั้นแทน อนิจจัง ทุกขัง อนัตตา

Thai Question Answering (Cont.)

Question:

ใครคือผู้กำกับภาพยนตร์เรื่อง "ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมอย"

Answer (rank: 1): นวพล ชั่รังษ์ตันถทธ. (normalized score: 69.27)

ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมอย "ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมอย" () เป็นภาพยนตร์ไทยแนวโรแมนติก ที่ออกฉายในปี พ.ศ. 2558 เขียนบท และกำกับโดย นวพล ชั่รังษ์ตันถทธ. นำแสดงโดย ชนนี้ สุวรรณเมธานนท์ และดาวิกา โฮร์น กำหนดออกฉายในวันที่ 3 กันยายน พ.ศ. 2558 นักแสดงนำ แสดง. - ชนนี้ สุวรรณเมธานนท์ รับบท ยุ่น (อัศนัย ครีซิริ) - ดาวิกา โฮร์น รับบท อิม (แพทท์หญิงชนนิการ์ กระจั่งศรัณย์) - วิโอเลต วงศ์อุ่น (วิโอเลต วงศ์อุ่น) - ต่อ พงศ์ จันทร์บุบพา รับบท พี่เป้ง - ณฐพล บุญประกอบ รับบท ไก่ - บรรจง ปิลัญธนะกุล รับบท เพื่อนหม้ออิม - อดิสรา ศรีสิริเกشم รับบท แฟนเจ - ชลสิทธิ อุปนิสิต รับบท เจิดการออก嫁และรายได้ การออก嫁และรายได้. จีทีเอชจัดฉาย "ฟรีแลนซ์..ห้ามป่วย ห้ามพัก ห้ามรักหมอย" รอบสื่อมวลชน เมื่อวันที่ 1 กันยายน 2558 ณ พารากอนชีนีเพล็กซ์ ศูนย์การค้าสยามพารากอน และเข้าฉายรอบปกติในอีกสองวันถัดมา โดยทำรายได้ในวันเปิดตัว 11.60 ล้านบาท ระหว่างวันที่ 24-30 กันยายน ค.ศ. 2015 ภาพยนตร์เรื่องนี้ทำรายได้ 86.07 ล้านบาทเพลงประจำรอบ. - Vacation Time (Thai Version) - วิโอเลต วงศ์อุ่น และ อภิวัชร์ เอื้อถาวรสุข (ต้นฉบับโดย Part Time Musicians) รางวัล

Translation

Source language

I'm going to talk today about energy and climate. And that might seem a bit surprising, because my full-time work at the foundation is mostly about vaccines and seeds, about the things that we need to invent and deliver to help the poorest two billion live better lives. But energy and climate are extremely important to these people; in fact, more important than to anyone else on the planet. The climate getting worse means that many years, their crops won't grow: there will be too much rain, not enough rain; things will change in ways their fragile environment simply can't support. And that leads to starvation, it leads to uncertainty, it leads to unrest. So, the climate changes will be terrible for them.

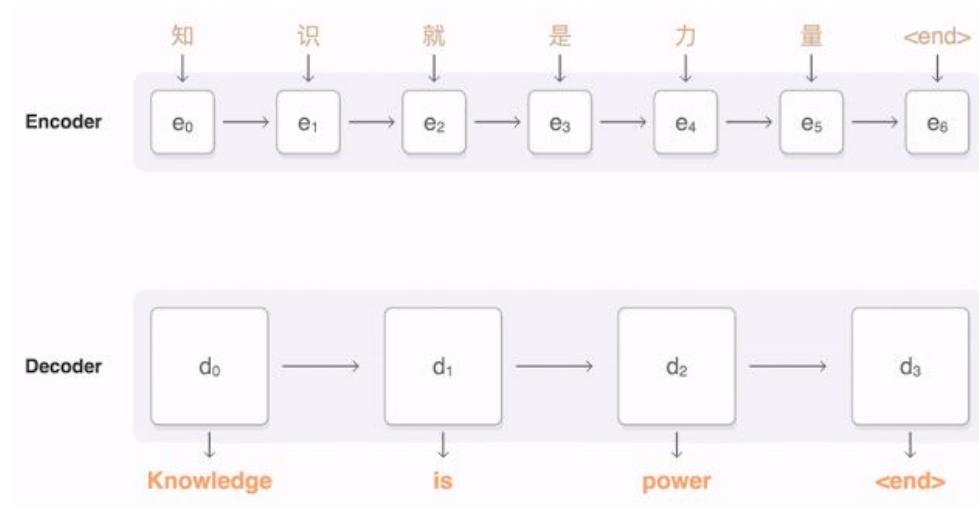
Target language

วันนี้ผมจะพูดถึงเรื่องของพลังงานและสภาพภูมิอากาศ อาจจะแปลกล้อคู่สักหน่อย เพราะว่า งานหลักของผมที่มุ่งเน้นไว้จะเกี่ยวกับพัฒนาชีวินทรีในกีเมล็ดพันธุ์พืช และก็ออกไปในทางประดิษฐ์คิดค้นและเผยแพร่ว่าไปใช้ เพื่อช่วย ให้กลุ่มคนที่จนที่สุด ในโลกสองพันล้านคนมีคุณภาพชีวิตที่ดีขึ้น แต่ว่าพลังงานและสภาพภูมิอากาศมีความสำคัญต่อคนกลุ่มนี้เป็นอย่างยิ่งครับ จริงๆแล้ว มันสำคัญต่อพวกรเขามากกว่าต่อใครในโลกก็ว่าได้ การที่สภาพภูมิอากาศแย่ลงย่อมหมายถึงการสูญเสียผลผลิตทางการเกษตรไปหลายปี ผ่านมากตมากาก่อนไป หรือไม่ก็ตไม่เพียงพอ อะไรก็เปลี่ยนไปในทางที่ สภาพแวดล้อมที่แสนจะเป็นรากจะรับมือดีไปอีกไม่ไหว ซึ่งทำให้เกิดภาวะอดอยาก ความไม่แน่นอน และเกิดสถานการณ์ที่ไม่สงบ ดังนั้น การเปลี่ยนแปลงสภาพภูมิอากาศจะทำให้พวกรเขาย่ำแย่ลง

Transcription of the TED 2010 Talk “Innovating to zero!” - Bill Gates

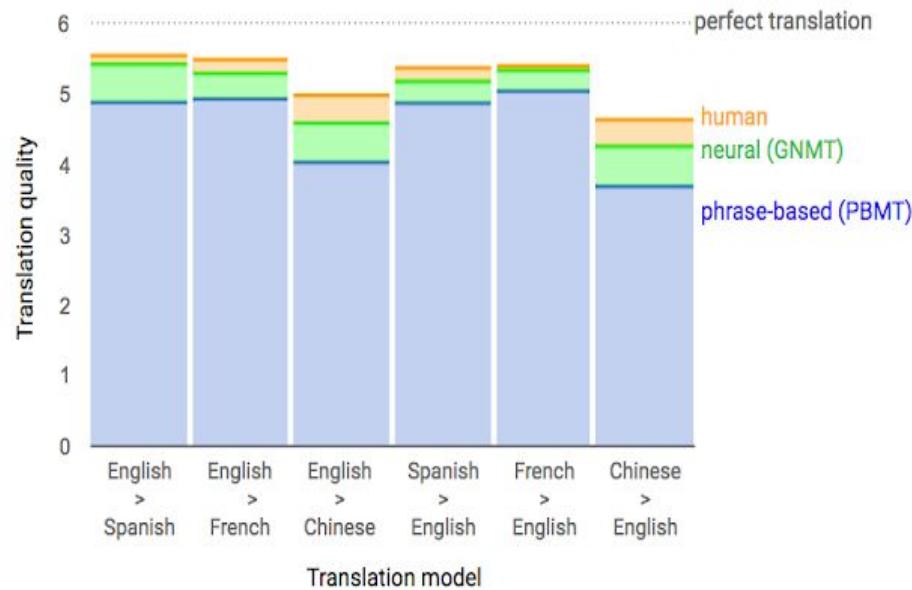
Machine Translation

Google's Neural Machine Translation with Encoder-Decoder Architecture (GNMT).



[Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](#)

Machine Translation (Cont.)



[Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](#)

Thai Machine Translation

Example Thai Machine Translation trained on [Opensubtitles 2018](#) dataset

	Good translation	Bad translation
English → Thai	<p>Edward , you ' re missing the point . → เอ็ดเวิร์ด คุณ ไม่เข้าใจประเด็น</p> <p>Now that we ' re done with our traditional greeting , can I tell you what I want ? → ตอนนี้เราเสร็จสูตรกับการ ต้อนรับอย่างเป็นทางการแล้ว ผม ขอโอกาสได้ใหม่ว่า ผู้ต้องการ อะไร</p>	<p>I would want to , you know , replace him . → ฉันอยากจะไป คุณก็รู้ แทนที่เขา</p> <p>All right , just out of curiosity and that's all ... → สิทธิทั้งหมด เพียงแค่ออกจากการ อยากรู้อยากเห็น เพียงแต่ ทั้งหมด ...</p>

Thai Machine Translation

Example Thai Machine Translation trained on [Opensubtitles 2018](#) dataset

	Good translation	Bad translation
Thai → English	<p>คุณมาทำอะไรที่นี่ → What are you doing here ?</p> <p>เหยื่อสองรายแรกเป็นโสเภณี ถูกฆ่าในสถานที่ เดียวกันกับที่ ที่ เธอทำงาน → The first two victims were prostitutes , killed in the same place where they worked .</p>	<p>อย่างที่คุณเห็น ไม่มีที่ว่างสำหรับฉันที่จะ แทรกเท้าไป → As you can see , it's empty for me to go in</p> <p>ผู้ชนะแต่ละการแข่งขัน จะได้รับ ลูกบอล ยังสีแดง → The winner ' s gonna get the ball back .</p>

Word/Sentence/Document Vector

Change a word into “secret code”

		How many legs?	Run fast?	Mammal?	How Fast?
	“Dog”	4	Yes	Yes	37 mph
	“Cat”	4	Yes	Yes	30 mph
	“Sloth”	4	No	No	0.15 mph

Word/Sentence/Document Vector

Change a word into “secret code” and encode into numbers.



How many legs? Run fast? Mammal? How Fast?

“Dog” [4 , 1 , 1 , 37]



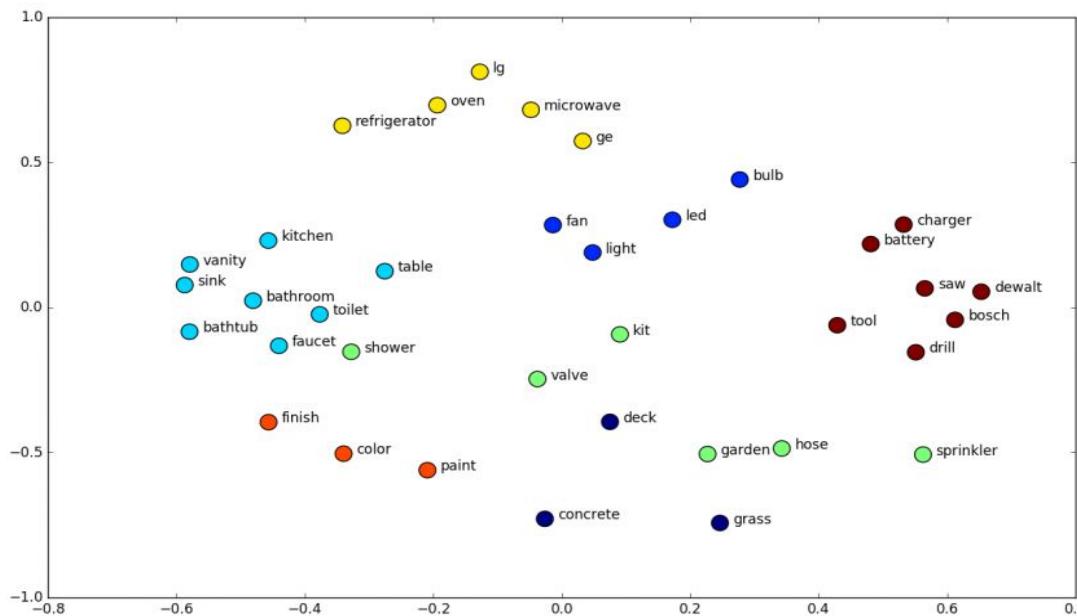
“Cat” [4 , 1 , 1 , 30]



“Sloth” [4 , 1 , 1 , 0.15]

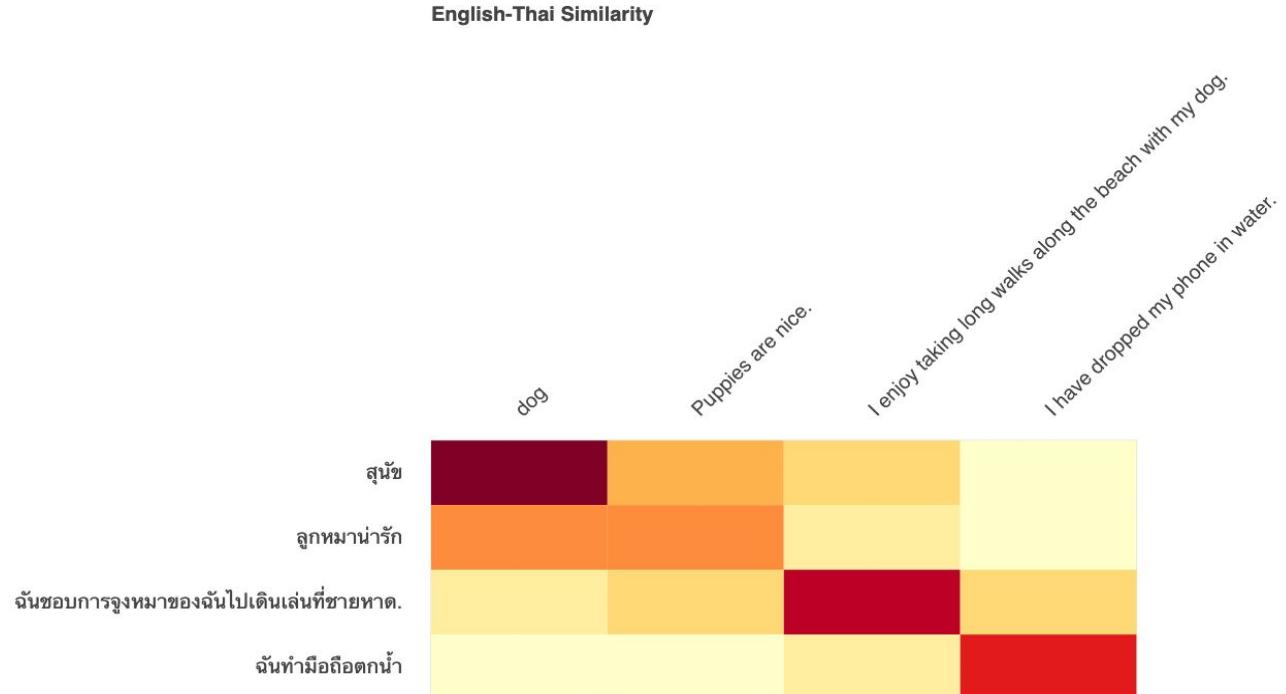
Word/Sentence/Document Vector

- Words with similar meaning are closer (cosine distance) than words with different meaning
- Represents words as vectors is more compact than one hot coding.



Word/Sentence/Document Vector (Cont.)

Google's Multilingual Sentence Encoder ([paper](#), [notebook](#))



NLP libraries

NLP Libraries

For Basic usage:

1. spaCy: Industrial-Strength Natural Language Processing (<https://spacy.io/>)
2. NLTK: Natural Language Toolkit (<https://www.nltk.org>)
3. Standford CoreNLP (<https://stanfordnlp.github.io/CoreNLP/>)

For NLP Research:

1. AllenNLP: Deep Learning for NLP (<https://allennlp.org/>)

spaCy

NLTK



AllenNLP

spaCy's Features

Tokenization	Segmenting text into words, punctuation marks etc.
Part-of-speech (POS) Tagging	Assigning word types to tokens, like verb or noun.
Dependency Parsing	Assigning syntactic dependency labels, describing the relations between individual tokens, like subject or object.
Lemmatization	Assigning the base forms of words. For example, the lemma of "was" is "be", and the lemma of "rats" is "rat".
Sentence Boundary Detection (SBD)	Finding and segmenting individual sentences.
Named Entity Recognition (NER)	Labelling named "real-world" objects, like persons, companies or locations.
Entity Linking (EL)	Disambiguating textual entities to unique identifiers in a Knowledge Base.
Similarity	Comparing words, text spans and documents and how similar they are to each other.
Text Classification	Assigning categories or labels to a whole document, or parts of a document.
Rule-based Matching	Finding sequences of tokens based on their texts and linguistic annotations, similar to regular expressions.

Spacy's Example 1: NER, POS

```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_lg")

doc = nlp("""Donald John Trump (born June 14, 1946) is the 45th and current president of
the United States. Before entering politics, he was a businessman and television personality.""")

spacy.displacy.render(doc, style='ent')
```

Donald John Trump PERSON (born June 14, 1946 DATE) is the 45th ORDINAL and current president of
the United States GPE . Before entering politics, he was a businessman and television personality.

Spacy's Example 2: Sentence Similarity

```
import spacy
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

nlp = spacy.load("en_core_web_lg")

docs = ["I am a Software Engineer at VISTEC.",
        "I work in this company as a front-end developer.",
        "\"Fat Man\" was the codename for the nuclear bomb."]

def plot_similarity(texts, rotation=90):

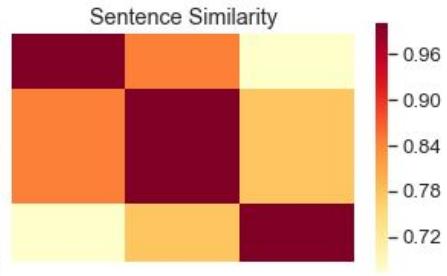
    features = [nlp(text) for text in texts]
    corr = [[doc_1.similarity(doc_2) for doc_2 in features] for doc_1 in features]
    min_corr = np.amin(corr)
    plt.figure(figsize=(5, 3))
    sns.set(font_scale=1.2)
    g = sns.heatmap(
        corr,
        xticklabels=texts,
        yticklabels=texts,
        vmin=min_corr,
        vmax=1,
        cmap="YlOrRd", square=True)
    g.set_xticklabels(texts, rotation=rotation)
    g.set_title("Sentence Similarity")

plot_similarity(docs)
```

I am a Software Engineer at VISTEC.

I work in this company as a front-end developer.

"Fat Man" was the codename for the nuclear bomb.



I am a Software Engineer at VISTEC.

I work in this company as a front-end developer.

"Fat Man" was the codename for the nuclear bomb.

Languages Support

Spacy

English en *
German de
Greek el
Spanish es
French fr
Italian it
Lithuanian lt
Norwegian Bokmål nb
Dutch nl
Portuguese pt

CoreNLP

English en *
German de
Greek zh
Spanish es
French fr
Arabic ar

NLTK

English en *

* All the libraries focus mainly on English language, most of the tasks (e.g. Tokenization, POS, NER, Word vectors, Sentence Segmentation) are available for English languages.

References: <https://spacy.io/usage/models>, <https://stanfordnlp.github.io/CoreNLP/human-languages.html>

Thai NLP library

PyThaiNLP

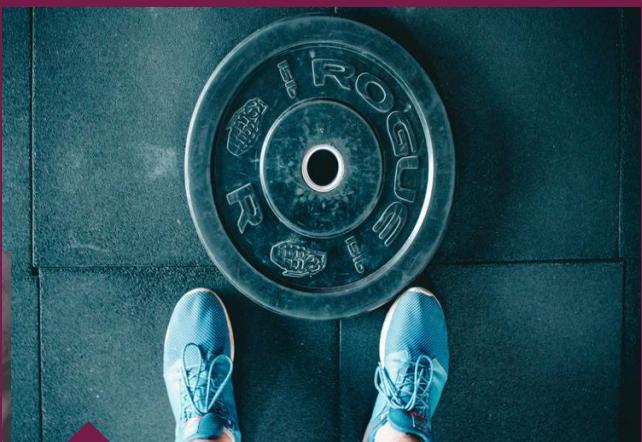


Tokenization Problem

Also known as the ຕາ|ກລມ vs ຕາກ|ລມ dilemma. We were not satisfied with our models but we did not know why.

Serious Ambiguity

What is the officially correct way to break up words or sentences? Wait, do we even have sentences?



Few Training Sets

Thai Wikipedia dump is 228MB as of today. We only had one small dataset to benchmark tokenization.

The Commit That Changed Everything

The Founding of PythaiNLP

Commits on Jun 23, 2016

Initial commit



wannaphongcom committed on Jun 23, 2016

"I was supposed to be studying for the entrance exam, but I wanted to make a simple chatbot. I found PyICU for word tokenization, but did not see any comprehensive tool like NLTK, so I figured I would create one."



Tontan / @wannaphongcom
Then-18-year-old high school student

NTLK Clone

@wannaphongcom aggregates pre-made Thai language modules like PyICU

Thai NLP Group

@korakot who frequents Thai ML scene suggested creating a group specifically for Thai NLP

`newmm` and more

@korakot rewrites his own version of maximal matching tokenizer, inspired by @veer66's wordcutpy. Also did text normalization, soundex and many more.

@wannaphongcom

@korakot



Let's Get started!

List of features:

- Word segmentation
- Named Entity Recognition Tagging
- Part of Speech Tagging
- Spell checker
- Thai Transliteration
- Word/Document Vectorization
- Utility functions
- Text cleaning and preprocessing

Word segmentation

- Dictionary-based - Maximal Matching ([newmm](#))
- Learning-based - CNNs model ([attacut](#), [deepcut](#))

```
from pythainlp.tokenize import word_tokenize

text = "ໂອເຄນພວກເຮົາວັກກາຍາບ້ານເກີດ"

word_tokenize(text, engine="newmm")
# output: ['ໂອເຄນ', 'ພວກ', 'ເຮົາ', 'ວັກ', 'ກາຍາ', 'ບ້ານ', 'ເກີດ']

word_tokenize(text, engine="deepcut")
# output: ['ໂອເຄນ', 'ພວກ', 'ເຮົາ', 'ວັກ', 'ກາຍາ', 'ບ້ານ', 'ເກີດ']

word_tokenize(text, engine='attacut')
# output: ['ໂອເຄນ', 'ພວກ', 'ເຮົາ', 'ວັກ', 'ກາຍາ', 'ບ້ານ', 'ເກີດ']
```

Syllable segmentation

- Dictionary-based syllable (พยานค์) Segmenter
- Learning-based syllable segmenter ([SSG](#))

```
from pythainlp.tokenize import syllable_tokenize

text = 'รถไฟสมัยใหม่จะใช้กำลังจากหัวรถจักรดีเซล หรือจากไฟฟ้า'
syllable_tokenize(text)
['รถ', 'ไฟ', 'สมัย', 'ใหม่', 'จะ', 'ใช้', 'กำ', 'ลัง', 'จาก', 'หัว', 'รถ',
 'ดี', 'เซล', ' ', 'หรือ', 'จาก', 'ไฟ', 'ฟ้า']
```

Named-Entity Recognition (NER) Tagging

- LSTMs based model ([thai-ner](#))

```
>>> from pythainlp.tag.named_entity import ThaiNameTagger
>>>
>>> ner = ThaiNameTagger()
>>> ner.get_ner("วันที่ 15 ก.ย. 61 ทดสอบระบบเวลา 14:49 น.")
[('วันที่', 'NOUN', 'O'), (' ', 'PUNCT', 'O'),
 ('15', 'NUM', 'B-DATE'), (' ', 'PUNCT', 'I-DATE'),
 ('ก.ย.', 'NOUN', 'I-DATE'), (' ', 'PUNCT', 'I-DATE'),
 ('61', 'NUM', 'I-DATE'), (' ', 'PUNCT', 'O'),
 ('ทดสอบ', 'VERB', 'O'), ('ระบบ', 'NOUN', 'O'),
 ('เวลา', 'NOUN', 'O'), (' ', 'PUNCT', 'O'),
 ('14', 'NOUN', 'B-TIME'), (':', 'PUNCT', 'I-TIME'),
 ('49', 'NUM', 'I-TIME'), (' ', 'PUNCT', 'I-TIME'),
 ('น.', 'NOUN', 'I-TIME')]
```

Part of Speech (POS) Tagging

- Perceptron/unigram model and artagger ([documentation](#))

```
from pythainlp.tag import pos_tag

words = ['ເກົ້າອື່ນ', 'ມີ', 'ຈໍານວນ', 'ໝາ', ' ', '=', '3']

pos_tag(words, engine='perceptron', corpus='orchid')
# output:
# [(['ເກົ້າອື່ນ', 'NCMN'], ('ມີ', 'VSTA'), ('ຈໍານວນ', 'NCMN'),
#   ('ໝາ', 'NCMN'), (' ', 'PUNC'),
#   ('=', 'PUNC'), ('3', 'NCNM')]

pos_tag(words, engine='unigram', corpus='pud')
# output:
# [(['ເກົ້າອື່ນ', None], ('ມີ', 'VERB'), ('ຈໍານວນ', 'NOUN'), ('ໝາ', None),
#   ('<space>', None), ('<equal>', None), ('3', 'NUM')]

pos_tag(words, engine='artagger', corpus='orchid')
# output:
# [(['ເກົ້າອື່ນ', 'NCMN'], ('ມີ', 'VSTA'), ('ຈໍານວນ', 'NCMN'),
#   ('ໝາ', 'NCMN'), ('<space>', 'PUNC'),
#   ('<equal>', 'PUNC'), ('3', 'NCNM')]
```

Spell checker

- Based on Peter Norvig's spell checker ([documentation](#))

```
from pythainlp.spell import correct

correct("ເສັ້ນຕົວ")
# output: 'ເສັ້ນຕົວ'

correct("ຄວັງ")
# output: 'ຄວັງ'

correct("ສັງເກດ")
# output: 'ສັງເກດ'

correct("ນະໂທ")
# output: 'ນະໂທ'

correct("ເຫດກາຮັດ")
# output: 'ເຫດກາຮັດ'
```

Romanization

- Rule-based: according to the Royal Thai General System of Transcription issued by Royal Institute of Thailand.
- Learning-based LSTM-Encoder-Decoder ([thai romanization](#))

```
from pythainlp.transliterate import romanize

romanize("ສາມາດ", engine="royin")
# output: 'samant'

romanize("ສາມາດ", engine="thai2rom")
# output: 'samat'

romanize("ພາພຍນຕີ", engine="royin")
# output: 'phapn'

romanize("ພາພຍນຕີ", engine="thai2rom")
# output: 'phapphayon'
```

Word/Document Vectorization

- LSTM Language Model based on [ULMFit](#) trained specifically for Thai language ([thai2fit](#))

Vectorize the sentence, "อ้วนเสี้ยวเข้ายีดแควันกิจิ่ว ในปี พ.ศ. 735", into one sentence vector with two aggregation meanthods: mean and summation.

```
>>> from pythainlp.word_vector import sentence_vectorizer
>>>
>>> sentence = 'อ้วนเสี้ยวเข้ายีดแควันกิจิ่ว ในปี พ.ศ. 735'
>>> sentence_vectorizer(sentence, use_mean=True)
array([[-0.00421414, -0.08881307,  0.05081136, -0.05632929, -0.06607185,
       0.03059357, -0.113882 , -0.00074836,  0.05035743,  0.02914307,
       ...
       0.02893357,  0.11327957,  0.04562086, -0.05015393,  0.11641257,
       0.32304936, -0.05054322,  0.03639471, -0.06531371,  0.05048079]])
>>>
>>> sentence_vectorizer(sentence, use_mean=False)
array([[-0.05899798, -1.24338295,  0.711359 , -0.78861002, -0.92500597,
       0.42831 , -1.59434797, -0.01047703,  0.705004 ,  0.40800299,
       ...
       0.40506999,  1.58591403,  0.63869202, -0.702155 ,  1.62977601,
       4.52269109, -0.70760502,  0.50952601, -0.914392 ,  0.70673105]])
```

Word/Document Vectorization (Cont.)

Compute cosine similarity between two words: “รถไฟ” and “รถไฟฟ้า” (train and electric train).

```
>>> from pythainlp.word_vector import similarity  
>>> similarity('รถไฟ', 'รถไฟฟ้า')  
0.43387136
```

Compute cosine similarity between two words: “เสือดาว” and “รถไฟฟ้า” (leopard and electric train).

```
>>> from pythainlp.word_vector import similarity  
>>> similarity('เสือดาว', 'รถไฟฟ้า')  
0.04300258
```

Utility functions:

Utility functions:

- Convert Arabic digits (e.g. 1, 3, 10) to Thai digits (e.g. ๑, ๓, ๑๐).
- Converts a number to Thai text and adds a suffix “บาท” (Baht), “สตางค์” (Stang).
- Sorting list of string according to Thai alphabets.
- Count number of Thai character in a sentences
- Date and time formatting , print datetime from Python module: datetime to Thai date and time format.
(วันจันทร์ 10 มิถุนายน 2562, ๑๐ น.ย. 15:59:00 2562)

Text Cleaning:

- Convert HTML string to ASCII character (e.g. from “A&B @.” to “A & B.”)
- Remove redundant spaces, brackets
- Ungroup emoji (“😂🤣😊” => “😂”, “🤣”, “😊”)

Towards an Industry-grade Thai NLP Library

@bact and @artificiala revolutionized the way we code our library

[Documentation](#)

[Tutorials](#)

[Unit Tests](#)

[CI/CD](#)

@heytitle



@wannaphongcom



@korakot



@cstorm125



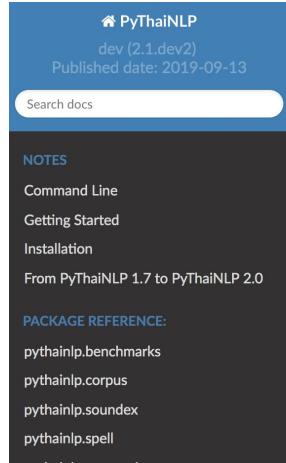
@artificiala



@bact



@lukkiddd



The screenshot shows the PyThaiNLP documentation homepage. At the top, it displays the repository name "PyThaiNLP", the version "dev (2.1.dev2)", and the publication date "Published date: 2019-09-13". Below this is a search bar labeled "Search docs". A sidebar on the left contains links to "NOTES" (Command Line, Getting Started, Installation) and "PACKAGE REFERENCE" (pythainlp.benchmarks, pythainlp.corpus, pythainlp.soundex, pythainlp.spell). The main content area is currently empty.

[Docs](#) » PyThaiNLP documentation

PyThaiNLP documentation

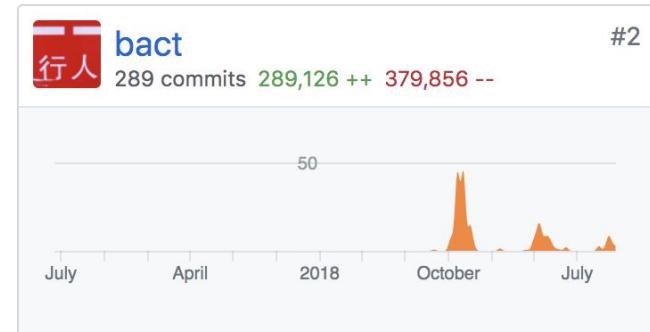
PyThaiNLP is a Python library for natural language processing (NLP)

Notes

- [Command Line](#)
- [Getting Started](#)
- [Installation](#)
- [From PyThaiNLP 1.7 to PyThaiNLP 2.0](#)

Package reference:

- [pythainlp.benchmarks](#)
- [pythainlp.corpus](#)



Open Source

Thai NLP of the people, by the people, for the people
We are always looking for contributors!

License

- PyThaiNLP code uses [Apache Software License 2.0](#)
- Corpus data created by PyThaiNLP project use [Creative Commons Attribution-ShareAlike 4.0 International License](#)
- For other corpus that may included with PyThaiNLP distribution, please refer to [Corpus License](#).

@heytite



@wannaphongcom



@korakot



@cstorm125



@artificiala



@bact



@lukkiddd



Useful Links

Development Version Documentation

<https://www.thainlp.cc/pythainlp/docs/dev/>
<https://www.thainlp.org/pythainlp/docs/dev/>

Stable Version Documentation

<https://www.thainlp.cc/pythainlp/docs/2.0/>
<https://www.thainlp.org/pythainlp/docs/2.0/>

Tutorials

<https://www.thainlp.cc/pythainlp/tutorials>
<https://www.thainlp.org/pythainlp/tutorials>

Github Repository

<https://www.github.com/pythainlp/pythainlp>

Tokenization Benchmarks

<https://github.com/PyThaiNLP/tokenization-benchmark>

Classification Benchmarks

<https://github.com/PyThaiNLP/classification-benchmarks>

Installation Instructions in Thai

<https://gist.github.com/wannaphongcom/6d5503e424246c51b420ef333046768a>

Clone repository for this workshop

```
$ git clone https://github.com/artificiala/pythainlp_workshop_bads.git
```

Github Repository:

https://github.com/artificiala/pythainlp_workshop_bads