Sentiment Polarity Classification for Khmer

Sokheng Khim[1], **Ye Kyaw Thu**[1,2], Sethserey Sam[1]

[1] Institute of Digital Research and Innovation (IDRI),
Cambodia Academy of Digital Technology (CADT), Cambodia

[2] Language and Semantic Technology Research Team (LST),
National Electronics and Computer Technology Center (NECTEC), Thailand

November 27, 2023

1. Introduction
2. Related Work
3. Corpus Building
4. Methodology
5. Evaluation Metric
6. Conclusion

Introduction

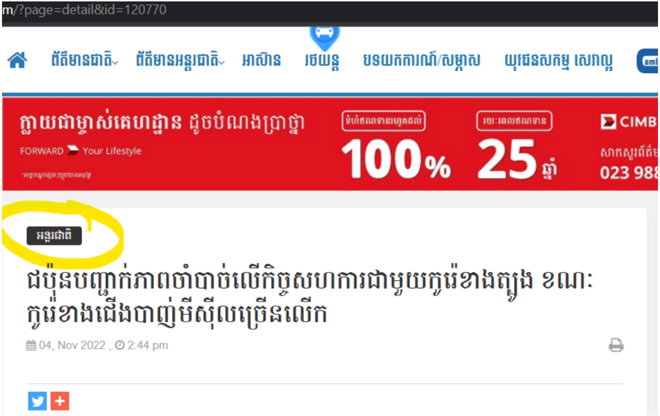
- Sentiment analysis, often referred to as opinion mining, stands as a cornerstone task in both natural language processing and computational linguistics
- Pivotal for comprehending user-generated content, such as social network posts or product reviews
- Significant attention from both the industrial and academic communities
- In this paper, we evaluate the efficacy of both traditional machine learning techniques and the FastText model
- positive, negative, neutral

Related Work

- Rina Buoy, Nguonly Taing and Sovisal Chenda (2021), Khmer Text Classification Using Word Embedding and Neural Networks
- Gather data from Wikipedia texts and two main local news website (Thmey Thmey and Sabay)
- Dataset is roughly 1 million sentences, around 30 million words
- Each news article can have more than one labels
- 13,902 articles in total have 4,687 articles with single label
- Data split into two smaller datasets multi-class (single label) and multi-label (one or more labels) classification

Related Work: Khmer Language Data

- Class names are in Khmer language, not mentioned how many classes in the paper



m/?page=detail&id=120770

អន្តរជាតិ

ជប៉ុនបញ្ជាក់ភាពចាំបាច់លើកិច្ចសហការជាមួយកូរ៉េខាងត្បូង ខណៈកូរ៉េខាងជើងបាញ់មីស៊ីលច្រើនលើក

04, Nov 2022 , 2:44 pm

Figure: An example of the class name in the Thmey news website

Related Work: Model Setup

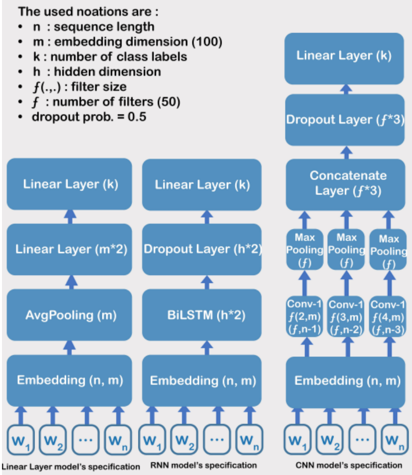


Figure: Architectures of Linear, RNN and CNN (figure adapted from poster, Rina Buoy et al., 2021)

Related Work: Model & Result

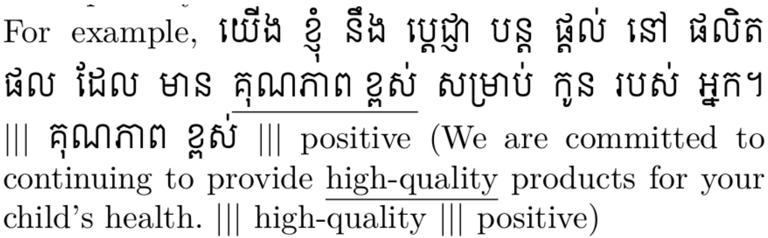
- Word Embedding Model: trained by using SVM and FastText
- Classifier Models Setup: trained by linear layer, RNN, and CNN
- For the training; Multi-class, Multi-label
- Multi-class : used cross-entropy loss function
- Multi-label : use binary cross-entropy loss function
- Results: RNN is outperform than other model for both multi-class and multi-label

Corpus Building

- Our Khmer Polarity Corpus is developed by collecting manually sentence that are commonly found on the actual data sources such as local news (SBM News, The Phnom Penh Post, Thmey Thmey, Fresh news), Social media (Facebook, Youtube), Wikipedia, and several website like food, health , sport, tourism (e.g. Healthy Cambodia, Sabay News).
- This valuable data was gathered around one month to reach 10K sentences and developing the polarity corpus.
- We analyzed the sentiment words from each sentence and classified these words into three categories (positive, negative and neutral).

Corpus Building

- An example of one sentence gives one keyword with specific sentiment polarity:

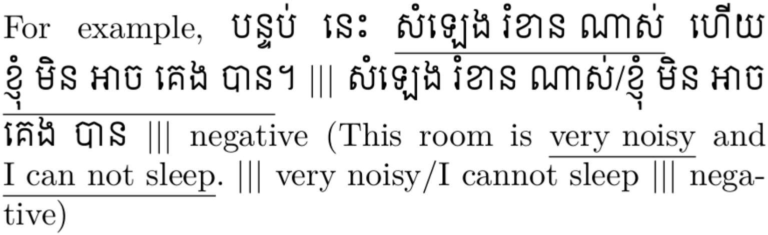


For example, យើង ខ្ញុំ នឹង ប្តេជ្ញា បន្ត ផ្តល់ នៅ ផលិត ផល ដែល មាន គុណភាព ខ្ពស់ សម្រាប់ កូន របស់ អ្នក។ ||| គុណភាព ខ្ពស់ ||| positive (We are committed to continuing to provide high-quality products for your child's health. ||| high-quality ||| positive)

Figure: Example sentence no. 1

Corpus Building

- One sentence example contains multiple keywords and their sentiment polarity is the same:

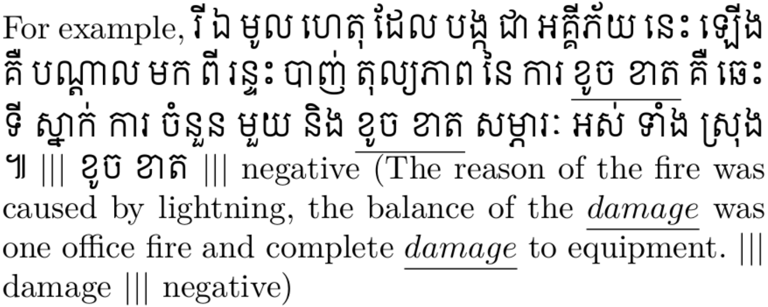


For example, បន្ទប់ នេះ សំឡេង រំខាន ណាស់ ហើយ ខ្ញុំ មិន អាច គេង បាន។ ||| សំឡេង រំខាន ណាស់/ខ្ញុំ មិន អាច គេង បាន ||| negative (This room is very noisy and I can not sleep. ||| very noisy/I cannot sleep ||| negative)

Figure: Example sentence no. 2

Corpus Building

- Keyword duplication can occur within a single sentence.

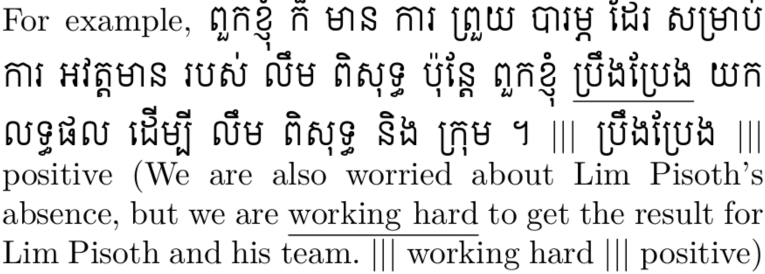


For example, រី ឯ មូល ហេតុ ដែល បង្ក ជា អគ្គីភ័យ នេះ ឡើង គឺ បណ្ដាល មក ពី រន្ទះ បាញ់ តុល្យភាព នៃ ការ ខូច ខាត គឺ ឆេះ ទី ស្នាក់ ការ ចំនួន មួយ និង ខូច ខាត សម្ភារៈ អស់ ទាំង ស្រុង ។ ||| ខូច ខាត ||| negative (The reason of the fire was caused by lightning, the balance of the *damage* was one office fire and complete *damage* to equipment. ||| damage ||| negative)

Figure: Example sentence no. 3

Corpus Building

- Positive and negative keywords can coexist in one sentence. In this case, we choose to keep the keywords based on the overall attitude of the statement.



For example, ពួកខ្ញុំ ក៏ មាន ការ ព្រួយ បារម្ភ ដែរ សម្រាប់ ការ អវត្តមាន របស់ លឹម ពិសុទ្ធ ប៉ុន្តែ ពួកខ្ញុំ ប្រឹងប្រែង យក លទ្ធផល ដើម្បី លឹម ពិសុទ្ធ និង ក្រុម ។ ||| ប្រឹងប្រែង ||| positive (We are also worried about Lim Pisoth's absence, but we are working hard to get the result for Lim Pisoth and his team. ||| working hard ||| positive)

Figure: Example sentence no. 4

Corpus Building: Normalization for Khmer

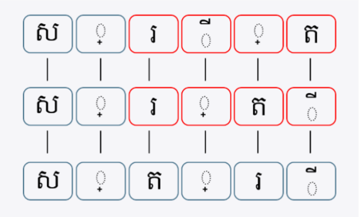


Figure: Character order normalization for Khmer

- Generally, we have to work with UTF-8 (i.e. Unicode encoding)
- Normalization step is necessary for Khmer language

Corpus Building: Statistics

Table: Statistics of the Khmer Polarity Corpus used in the experiments (Note: including header line)

Khmer Corpus Information	Training	Testing
Number of Sentence	9,015	1,001
Number of Word	698,068	75,893
Frequency of positive	5,251	583
Frequency of negative	2,933	325
Frequency of neutral	830	92

Methodology: Machine Learning

- Comparison between five machine learning models and shallow neural network and we used five ML methods
    1. KNN or K-NN: K-Nearest Neighbor
    2. Decision Tree
    3. Random Forest
    4. SVM: Support Vector Machine
    5. SGD: Stochastic Gradient Descent

Methodology: K-Nearest Neighbor

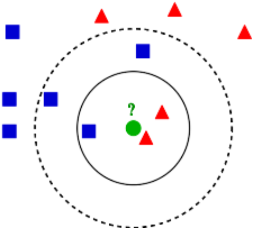


Figure: KNN algorithm: if k=3, predicted as red triangle and if k=5, predicted as blue squre (Courtesy Wikipedia)

- A non-parametric model and does not require any training
- Simple ML modeling technique and few parameters to tune
- K should be wisely selected
- For features to be treated fairly, appropriate scaling should be offered
- Due to the need to track all training data and locate neighbor nodes, slow in real time

Methodology: Decision Tree

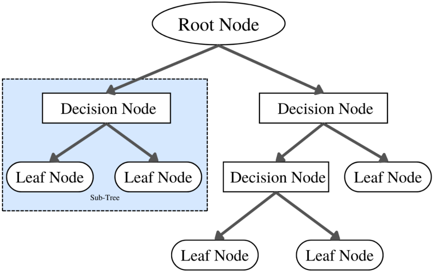


Figure: Decision Tree algorithm

- Non-parametric model and used to solve regression and classification problems
- Algorithm to select conditions: for CART(classification and regression trees), we use gini index as the classification metric
- $gindex = 1 - \sum P_t^2$
- Supports automatic feature interaction wheres KNN can't.

Methodology: Random Forest

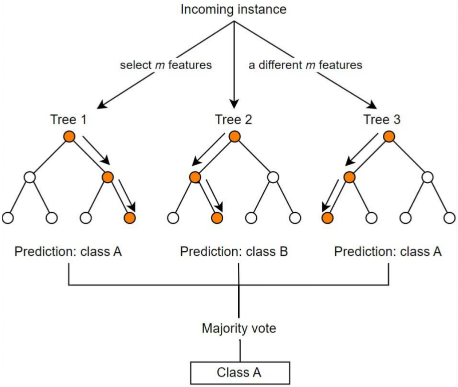


Figure: Random Forests algorithm

- Ensemble model and multiple decision trees are combined to get a stronger model
- In theory: applied bagging method, more robust and handles overfitting efficiently
- Supports implicit feature selection and derives feature importance
- computationally complex and slower when forest becomes large

Methodology: Support Vector Machine

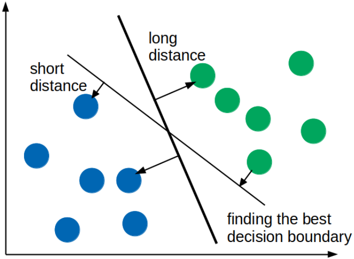


Figure: SVM algorithm

- Supervised learning, extensively used in text classification
- In Linear SV: maximizes the classification margin
- In non linear SVM: a kernel function is used to derive a new hyperplane for all the training data. Afterward, a linear curve will classify the labels in the hyperplane.
- Gaussian kernel, polynomial kernel, Sigmoid kernel, Laplace RBF kernel etc.

Methodology: Support Vector Machine

- SVM uses kernel trick to solve complex solutions
- Hinge loss provides higher accuracy
- Outliers can be well handled using soft margin constant
- Hyper parameters and kernels are to be carefully tuned for sufficient accuracy
- Longer training time for larger datasets
- SVM can perform better than neural networks when there are limited training data and many features
- Multi class classification requires multiple models for SVM

Methodology: Stochastic Gradient Descent

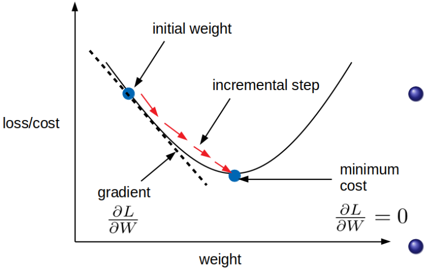
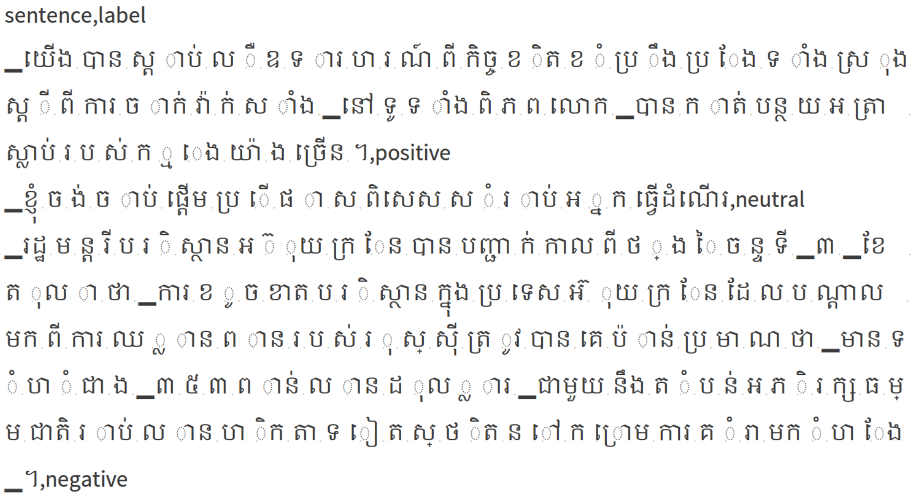

initial weight
incremental step
loss/cost
minimum cost
gradient
$\frac{\partial L}{\partial W}$
$\frac{\partial L}{\partial W} = 0$
weight

Figure: SGD algorithm

- Stochastic means "random", Gradient means "slope" or "slant" of a surface, Modification of GD algorithm
- Randomly picks one data point from the whole training data at each iteration to reduce the computations enormously
- online SGD, Batch SGD
- Approximate and iterative method for mathematical optimization

Methodology: Machine Learning



sentence,label

_យើង បាន ស្ដ ាប់ ល ឺ ឧ ទ ារ ហ រ ណ៍ ពី កិច្ច ខ ិត ខ ំ ប្រ ឹង ប្រ ែង ទ ាំង ស្រ ុង ស្ត ី ពី ការ ច ាក់ វ៉ា ក់ ស ាំង _នៅ ទូ ទ ាំង ពិ ភ ព លោក _បាន ក ាត់ បន្ថ យ អ ត្រា ស្លាប់ រ ប ស់ ក ្ន ែង យ៉ា ង ច្រើន ។,positive

_ខ្ញុំ ច ង់ ច ាប់ ផ្តើម ប្រ ើ ផ ា ស ពិសេស ស ំ រ ាប់ អ ្ន ក ធ្វើដំណើរ,neutral

_រដ្ឋ ម ន្ត្រី ប រ ិ ស្ថាន អ ៊ ុយ ក្រ ែន បាន បញ្ជា ក់ កាល ពី ថ ្ង ៃ ច ន្ទ ទី _៣ _ខែ ត ុល ា ថា _ការ ខ ូ ច ខាត ប រ ិ ស្ថាន ក្នុង ប្រ ទេស អ៊ ុយ ក្រ ែន ដែ ល ប ណ្តាល មក ពី ការ ឈ ្ល ាន ព ាន រ ប ស់ រ ុ ស្ ស៊ី ត្រ ូវ បាន គេ ប៉ ាន់ ប្រ មា ណ ថា _មាន ទ ំ ហ ំ ជា ង _៣ ៥ ៣ ព ាន់ ល ាន ដ ុល ្ល ារ _ជាមួយ នឹង ត ំ ប ន់ អ ភ ិ រ ក្ស ធ ម្ ម ជាតិ រ ាប់ ល ាន ហ ិក តា ទ ៀ ត ស្ ថ ិត ន ៅ ក ្រោម ការ គ ំ រា មក ំ ហ ែង _។,negative

Figure: Training data format for machine learning.

Methodology: FastText

Table: Comparison of popular word-embedding methods (adapted from Transfer Learning for NLP, Paul Azunre, 2021

Embedding	Strengths	Weaknesses
SkipGram: word2vec	Works well with a small training dataset and rare words	Slower training, plus lower accuracy for frequent words
CBOW: word2vec	Several times faster in training and better accuracy for frequent words	Doesn't work as well with little training data and rare words
GloVe	Vectors have more interpretability than other methods	Higher memory requirement during training to store co-occurrences of words
FastText	Can handle out-of-vocabulary words	Higher computing cost; larger and more complex model

Methodology: FastText

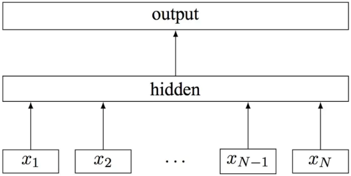
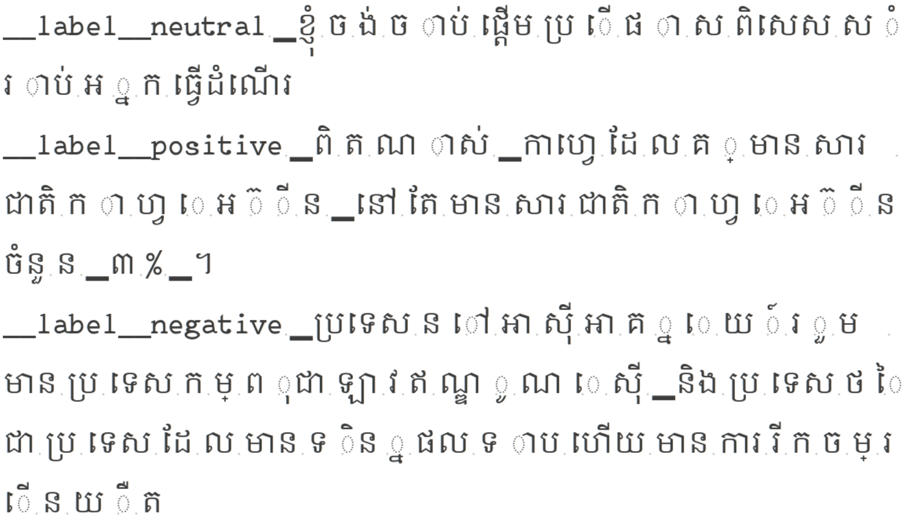


Figure: Model architecture of the FastText *t* for a sentence with N ngram features x_1, ..., x_n

- Learns vectors for the n-grams that are found within each word, as well as each complete word
- We used sub-word level embedding (i.e. sentencepiece)
- Our main approach is the combination of sentencepiece and FastText for Khmer polarity classification
- Baselines are traditional machine learning approaches such as KNN, Decision Tree, Random Forest, SVM, SGD

Methodology: FastText



```
__label__neutral _ខ្ញុំ ចង់ ចាប់ ផ្ដើម ប្រើ ផាស ពិសេស សំរាប់ អ្នក ធ្វើដំណើរ
__label__positive _ពិត ណាស់ _កាហ្វេ ដែល គ្មាន សារ ជាតិ កាហ្វេអ៊ីន _នៅ តែ មាន សារ ជាតិ កាហ្វេអ៊ីន ចំនួន _៣ % _។
__label__negative _ប្រទេស នៅ អាស៊ី អាគ្នេយ៍ រួម មាន ប្រទេស កម្ពុជា ឡាវ ឥណ្ឌូណេស៊ី _និង ប្រទេស ថៃ ជា ប្រទេស ដែល មាន ទិន្នផល ទាប ហើយ មាន ការ រីក ចម្រើន យឺត
```

Figure: Training data format of FastText.

Evaluation Metric: Precision

Precision

Precision is a good measure to determine, when the costs of False Positive is high.

$$\begin{aligned} \mathit{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} && (1) \\ &= \frac{\text{True Positive}}{\text{Total Predicted Positive}} && (2) \end{aligned}$$

Evaluation Metric: Recall

Recall

Recall is a good measure to determine, when the costs of False Negative is high.

$$\mathit{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

$$= \frac{\text{True Positive}}{\text{Total Actual Positive}} \tag{4}$$

Evaluation Metric: F-measure or F1 score

F1 Score

F1 score is a good measure to seek a balance between Precision and Recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

Results and Discussion

Table: Machine learning model results for **bigram counts**

Model	**F1-Score**	**Precision**	**Recall**	**Acc.**
KNN, macro avg:	0.31	0.31	0.31	0.43
weighted avg:	0.44	0.43	0.43	
Decision Tree, macro avg:	0.33	0.33	0.29	0.55
weighted avg:	0.45	0.55	0.46	
Random Forest, macro avg:	0.35	0.34	0.30	0.56
weighted avg:	0.47	0.56	0.47	

Results and Discussion

Table: Machine learning model results for **bigram counts**

Model	**F1-Score**	**Precision**	**Recall**	**Acc.**
SVM, macro avg:	0.29	0.33	0.25	0.58
weighted avg:	0.43	0.58	0.43	
SGD, macro avg:	0.48	0.35	0.31	0.57
weighted avg:	0.50	0.57	0.47	

Results and Discussion

Table: Machine learning model results for **bigram Tf-Idf**

Model	**F1-Score**	**Precision**	**Recall**	**Acc.**
KNN, macro avg:	0.36	0.35	0.35	0.48
weighted avg:	0.47	0.48	0.47	
Decision Tree, macro avg:	0.40	0.38	0.38	0.54
weighted avg:	0.50	0.54	0.51	
Random Forest, macro avg:	0.44	0.38	0.35	0.60
weighted avg:	0.55	0.60	0.53	
SVM, macro avg:	0.70	0.37	0.34	0.59
weighted avg:	0.61	0.59	0.51	

Results and Discussion

Table: Machine learning model results for **bigram Tf-Idf** SGM and SGM tuning

Model	**F1-Score**	**Precision**	**Recall**	**Acc.**
SGD, macro avg: weighted avg:	0.41 0.52	0.37 0.58	0.35 0.52	0.58
SGD Tuning, macro avg: weighted avg:	0.55 0.57	0.38 0.60	0.35 0.53	0.60

Results and Discussion

Table: Classification results with the FastText model

-	F1-Score	Precision	Recall
Positive	0.81	0.78	0.84
Negative	0.68	0.67	0.69
Neutral	0.33	0.53	0.23
P@1	0.74		
R@1	0.74		

Conclusion

- We explored several text features, and they are unigram-counts, unigram TF-IDF, bigram-counts, bigram-TF-IDF, and FastText
- Compared between five machine learning models and shallow neural network FastText
- FastText achieved the best classification result and the training/testing speed also very fast
- Keep updating the Khmer polarity corpus
- This work is related to language understanding
- Many potential applications such as hate-speech detection

Thank you! Any questions?

Reference

1. Kudo, Taku, and John Richardson. "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing." arXiv preprint arXiv:1808.06226 (2018).
2. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. "Bag of Tricks for Efficient Text Classification." In Proc. of the 15th Conf. of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427-431, Valencia, Spain.
3. Rina Buoy, Nguonly Taing, Sovisal Chenda, Khmer Text Classification Using Word Embedding and Neural Networks, 2021, URL: https://arxiv.org/abs/2112.06748

Reference

4. sentencepiece GitHub https://github.com/google/sentencepiece
5. FastText Tutorial: https://fasttext.cc/docs/en/supervised-tutorial.html
6. ML experiment log for Khmer polarity classification: https://github.com/ye-kyaw-thu/error-overflow/blob/master/kh-polarity-exp2.md
7. FastText training log for Khmer polarity classification: https://github.com/ye-kyaw-thu/error-overflow/blob/master/testing-tabpfn.md

Reference

8. Pedro Domingos. 2012. A few things to know about machine learning. Commun. ACM 55, 10 (Oct. 2012), 78-87. https://doi.org/10.1145/2347736.2347755
9. https://www.user.tu-berlin.de/mtoussai/teaching/Lecture-MachineLearning.pdf
10. https://en.wikipedia.org/wiki/Stochastic_gradient_descent
11. https://stanford.edu/ shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks
12. https://www.kaggle.com/getting-started/27261
13. https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222

Reference

14. https://en.wikipedia.org/wiki/Gini_coefficient
15. https://en.wikipedia.org/wiki/Support_vector_machine
16. http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf
17. https://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf
18. https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
19. https://realpython.com/gradient-descent-algorithm-python/
20. https://towardsdatascience.com/implementing-sgd-from-scratch-d425db18a72c

Reference

21. https://en.wikipedia.org/wiki/F-score
22. https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f
23. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9
24. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45, pp. 427-437.