VIETNAMESE SENTIMENT ANALYSIS: THEORY IN ACTION

Duy V. Huynh, Anh D. Phan, Huy C. Khuong
Faculty of Information Technology, Ton Duc Thang University
{51703006, 51702058, 51702112}@student.tdtu.edu.vn

SENTIMENT ANALYSIS

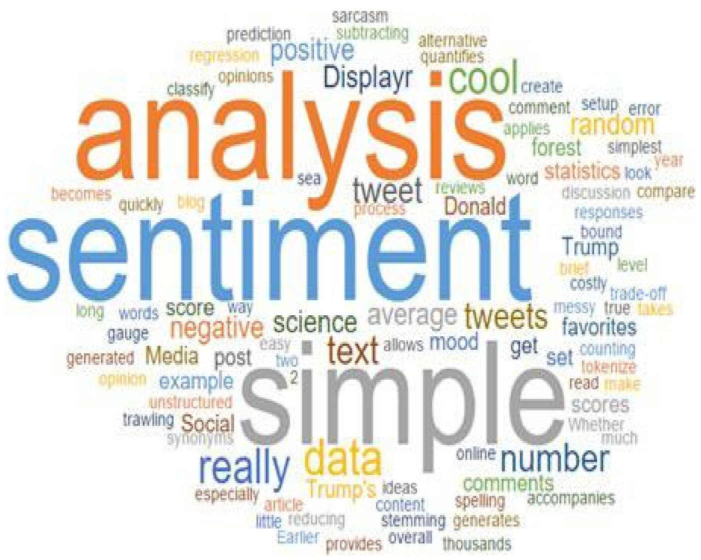
- Predict **polarity** of a given text:

  “Sản phẩm này rất tuyệt vời” → Positive

  “Sản phẩm này rất tệ” → Negative

  “Sản phẩm này bình thường” → Neutral

⇒ Text Classification



SENTIMENT ANALYSIS: USE CASES

RELATED WORK: BENCHMARKS

- English:
  - **Yelp-2:** 60,000 training samples and 38,000 test samples for negative and positive classes.
  - **Yelp-5:** 650,000 training samples and 50,000 test samples for each class of fine-grained sentiment labels.
  - **IDMb:** 25,000 movie reviews for each training and test sets.
- Vietnamese:
  - **VLSP-2016:** 5100 training samples and 1050 test sample with uniform class distribution (⅓ for each class).
  - **UIT-VSFC:** over 16,000 sentences for sentiment analysis and topic classification.
  - **AIVIVN:** 16087 sentences for training and 10981 sentences for testing which is divided into 2 classes (positive and negative).

RELATED WORK: VLSP-2016

- Several proposed methods include traditional **machine learning** and modern **deep learning** approaches.
  - SVM/MLNN/LSTM, Ensemble, Multi-channel LSTM,..
- **Pham et al., 2016** proposed a **lexicon-based** classifier and got state-of-the-art result.

Model	F1
Perceptron/SVM/Maxent	80.05
SVM/MLNN/LSTM	71.44
Ensemble: Random forest, SVM, Naive Bayes	71.22
Ensemble: SVM, LR, LSTM, CNN	69.71

Quynh-Trang Thi Pham, Xuan-Truong Nguyen, Van-Hien Tran, Thi-Cham Nguyen, Mai-Vu Tran (2016). DSKTLAB: Vietnamese Sentiment Analysis for Product Reviews. VLSP 2016 Workshop. In conjunction with 12th IEEE - RIVF.

RELATED WORK: LEXICON-BASED

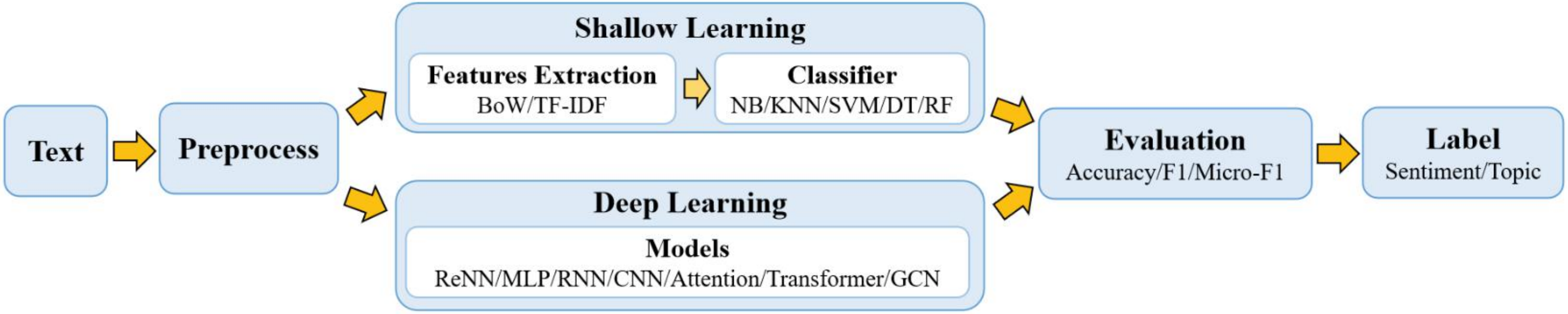
- Lexicon-based methods require a huge dictionary with pre-defined keywords and its polarity weights.
- Example:
  - “tốt”: positive
  - “xấu”: negative
  - “bình thường”: neutral

It cost too much, require expertise level and ambiguity problem.

Typical **rule-based** approach.

Taboada, Maite & Brooke, Julian & Tofiloski, Milan & Voll, Kimberly & Stede, Manfred. (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics. 37. 267-307. 10.1162/COLI_a_00049.

LEXICON-FREE CLASSIFIER



Flowchart of traditional Text Classification **(Qian Li et al., 2020).**

Li, Qian & Peng, Hao & Li, Jianxin & Xia, Congying & Yang, Renyu & Sun, Lichao & Yu, Philip & He, Lifang. (2020). A Text Classification Survey: From Shallow to Deep Learning.

TEXT PRE-PROCESSOR

“Trương Đại   học   Tôn Đức   Thắng.;#^!”

- Remove duplicate spaces: “Trương Đại học Tôn Đức Thắng.;#^!”
- Word segmentation: “Trương Đại_học Tôn_Đức_Thắng.;#^!”
- Lowercase: “trương đại_học tôn_đức_thắng.;#^!”
- Remove redundant punctuation: “trương đại_học tôn_đức_thắng”
- Diacritics restoration: “trường đại_học tôn_đức_thắng”
- And more...

FEATURE EXTRACTION

- **One-hot vector:** A word is represented by a vector that indicate the **look-up index** of this word in vocabulary.

vector "tôi"

1	0	0

vocabulary

tôi	đi	học

- **Bag of Words:** A representation of text that describes the **occurrence of words within a document**. It involves two things: A vocabulary of known words. A measure of the presence of known words.

vector "tôi đi học"

1	1	1

vocabulary

tôi	đi	học

FEATURE EXTRACTION

- **Word Piece:** Given the vocabulary which is initialized with individual characters in the language, then the **most frequent combinations of symbols** in the vocabulary are iteratively added to the vocabulary. A word representation is encoded by their **sub-linguistic-unit** as follows:

    "đường" "đ, uờ, ng"

    "phố" "ph, ố"

FEATURE EXTRACTION

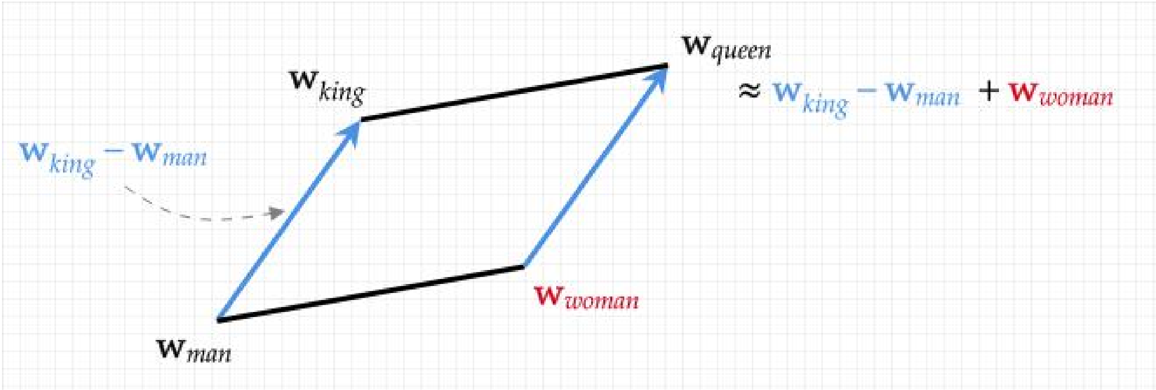
- **Term Frequency - Inverse Document Frequency:** a **numerical statistic** that is intended to reflect how **important a word** is to a document in a collection or corpus **(Wikipedia., 2020)**.

Recommended tf-idf weighting schemes

weighting scheme	document term weight	query term weight
1	$f_{t,d} \cdot \log \frac{N}{n_t}$	$\left(0.5 + 0.5 \frac{f_{t,q}}{\max_t f_{t,q}}\right) \cdot \log \frac{N}{n_t}$
2	$1 + \log f_{t,d}$	$\log\left(1 + \frac{N}{n_t}\right)$
3	$(1 + \log f_{t,d}) \cdot \log \frac{N}{n_t}$	$(1 + \log f_{t,q}) \cdot \log \frac{N}{n_t}$

FEATURE EXTRACTION

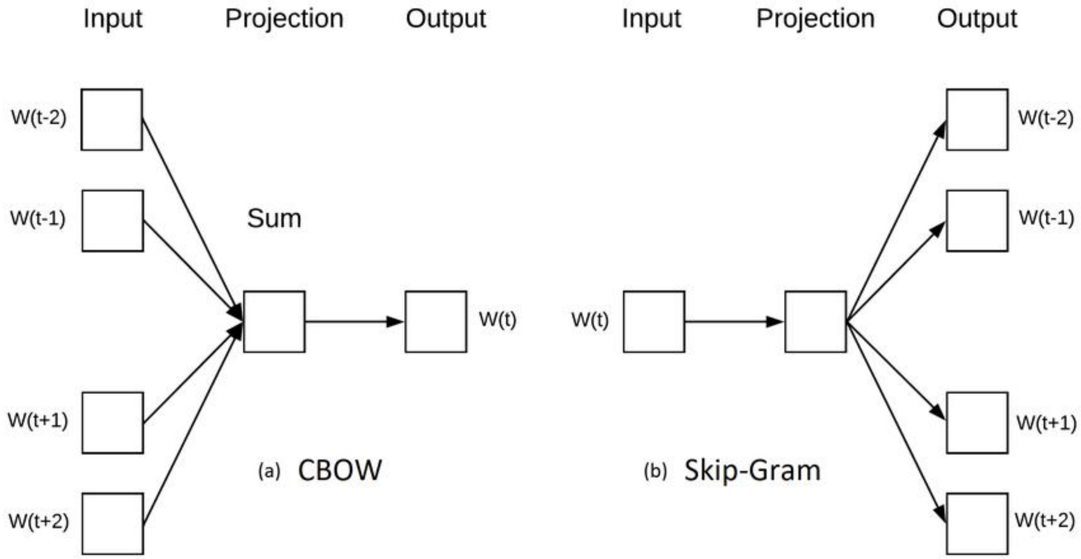
- **Word Embedding:** A word embedding is a **learned representation** for text where words that have the **same meaning** have a **similar representation** **(Mikolov et al., 2013).**



Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th NIPS - Volume 2* (*NIPS'13*).

FEATURE EXTRACTION

- Word embedding is learned using **CBOW** (Continuous Bag of Words) and **Skip-gram (Mikolov et al., 2013)**.



Mikolov, Tomas & Corrado, G.s & Chen, Kai & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space.

CONTEXTUALIZED WORD EMBEDDING

- Drawbacks of static word embedding **(Word2Vec, Glove):**
  - The first is that it is static and cannot solve the **polysemous problem** when a word has different meanings in different contexts.
  - The second is the **out-of-vocabulary** problem. If a word did not appear in the training corpus, it has no pre-trained vector.

Take **context information** into account and learn representations for word embeddings. Nowadays, these language models are standard baseline for most NLP-tasks.

MACHINE LEARNING CLASSIFIER

- Most of traditional classifiers base on **supervised learning** approach: Naive Bayes, Decision Tree, Gaussian Process, Support Vector Machine,..
- Objective of supervised learning is to learn an approximation **f** of optimal function **f*** under the given training dataset **D** $\sim \mathcal{D}$.
- To be successful, **D** = {(x_1,y_1), (x_2,y_2), .. (x_N,y_N)} are assumed to be **identical**, **independently**, **distributed** **(i.i.d.)** samples from $\mathcal{D}$.

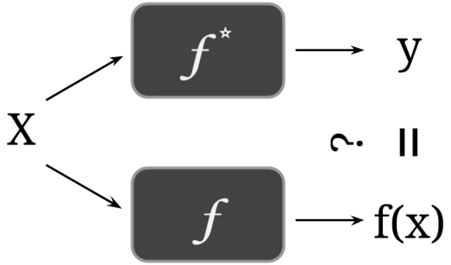


Image Credit: https://khanh-personal.gitbook.io/ml-book-vn/chapter1/hai-goc-nhin-ve-supervised-learning

DEEP LEARNING CLASSIFIER

- Beyond statistical models, many deep neural networks models have been proposed and achieved big success.

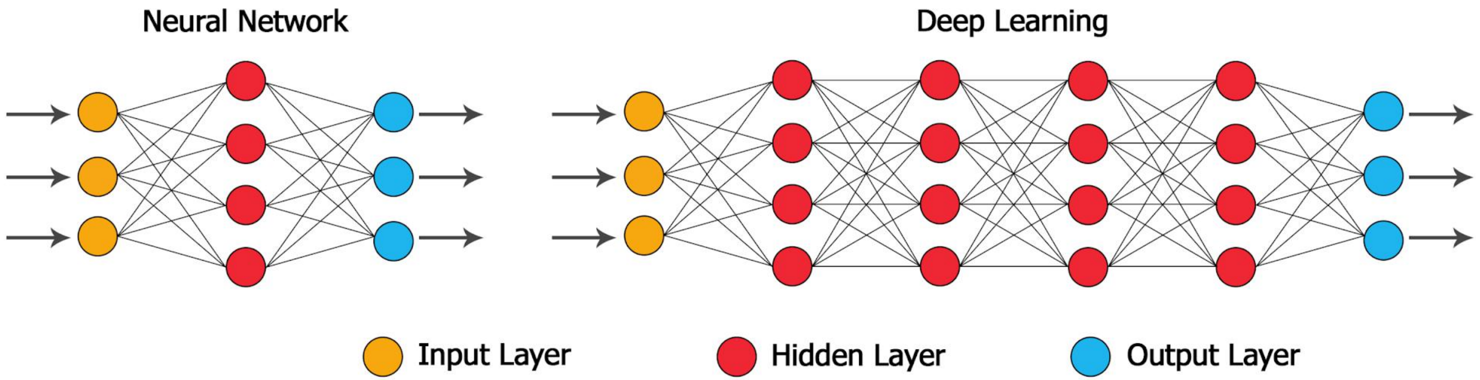


Image Credit: https://alphabold.com/neural-networks-and-deep-learning-an-overview/

LONG SHORT-TERM MEMORY

- **LSTM** is a variant of **RNN-family** models. It is autoregressive, trainable, and capable of capturing hidden pattern in long sequence data by dealing with **vanishing gradient problem**.

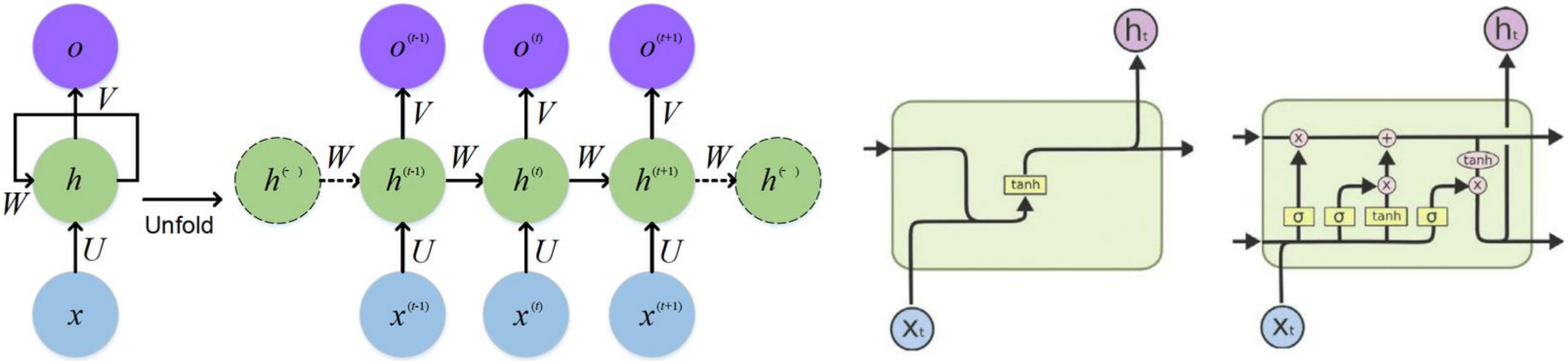
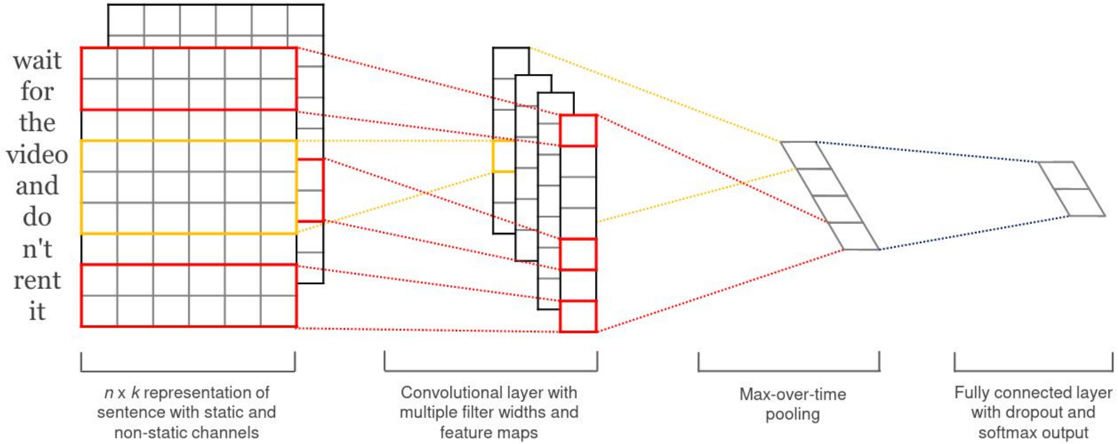


Image Credit: Mamandipoor, Behrooz & Majd, Mahshid & Sheikhalishahi, Mostafa & Modena, Claudio & Osmani, Venet. (2020). Monitoring and detecting faults in wastewater treatment plants using deep learning. Environmental Monitoring and Assessment.

CONVOLUTIONAL NEURAL NETWORK

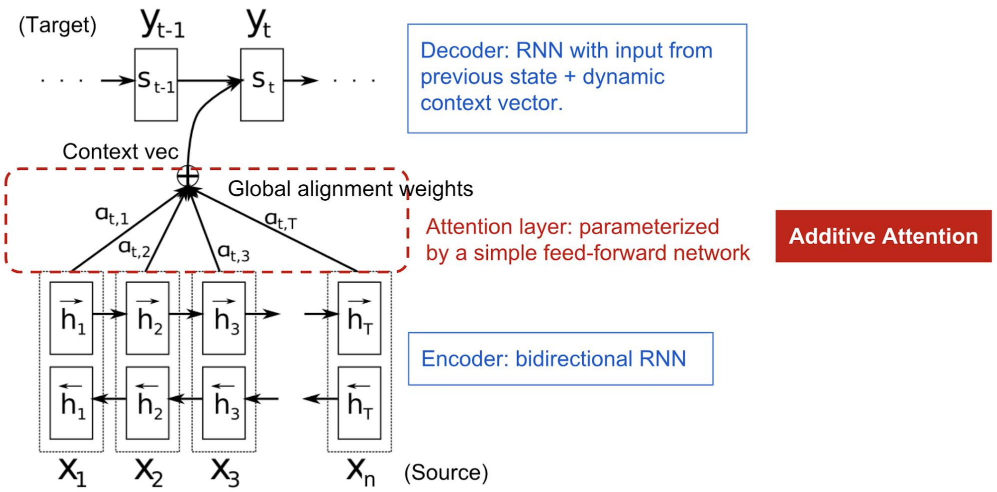
- **CNN** is the world-famous network architecture that disrupts the field of computer vision.
- **Kim Yoon., (2014)** proposed the first version of CNN that can be applied to solve NLP-tasks.



Kim, Yoon. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.

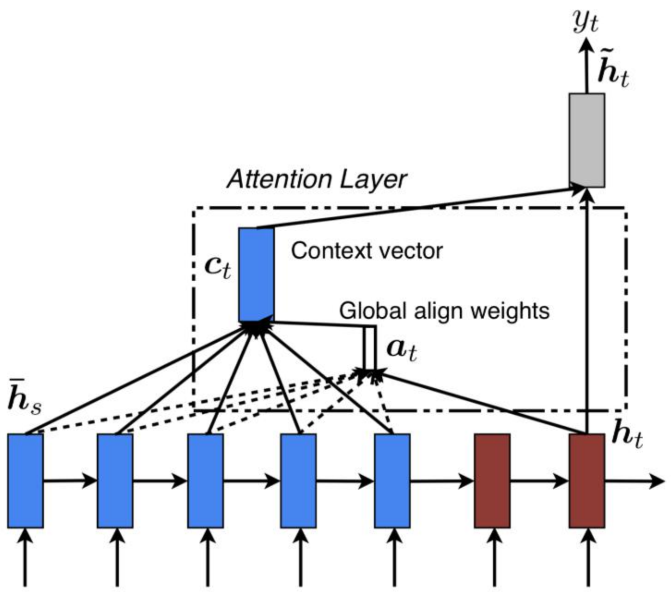
ATTENTION MECHANISM

- Attention is, to some extent, motivated by how we pay **visual attention** to different regions of an image or **correlate words** in one sentence **(Weng Lilian., 2018).**
- **Bahdanau et al., (2014)** proposed his **addictive attention** to improve machine translation model.

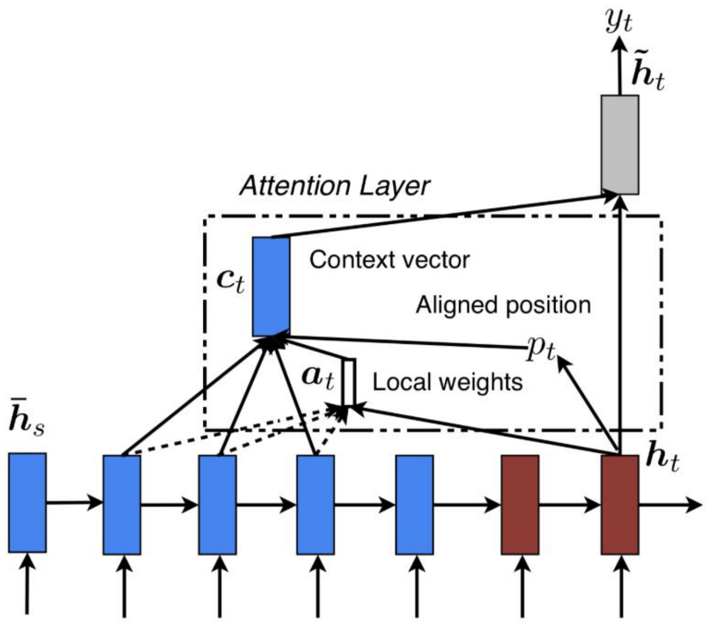


Weng, Lilian (2018). Attention? Attention! https://lilianweng.github.io/lil-log
Bahdanau, Dzmitry & Cho, Kyunghyun & Bengio, Y.. (2014). Neural Machine Translation by Jointly Learning to Align and Translate.

GLOBAL/LOCAL ATTENTION



Global Attention Model

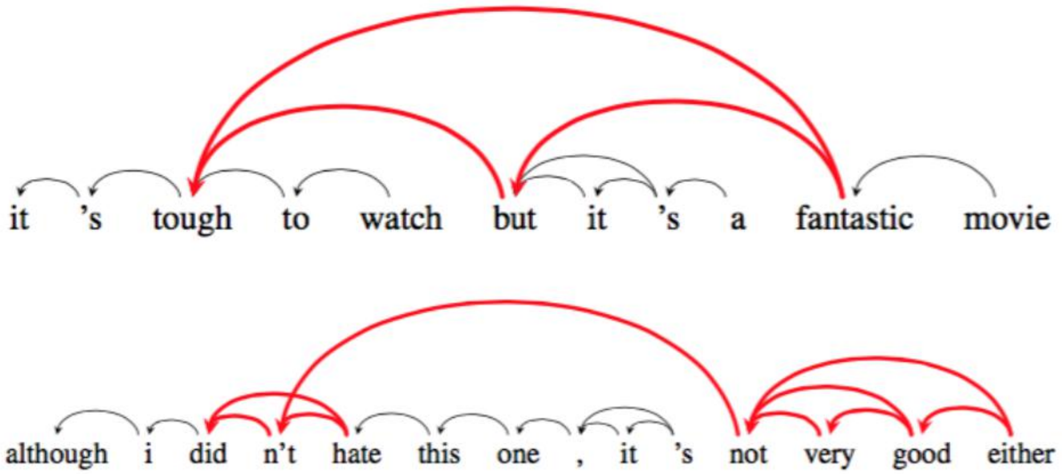


Local Attention Model

Luong, Minh-Thang & Pham, Hieu & Manning, Christopher. (2015). Effective Approaches to Attention-based Neural Machine Translation.

SELF-ATTENTION

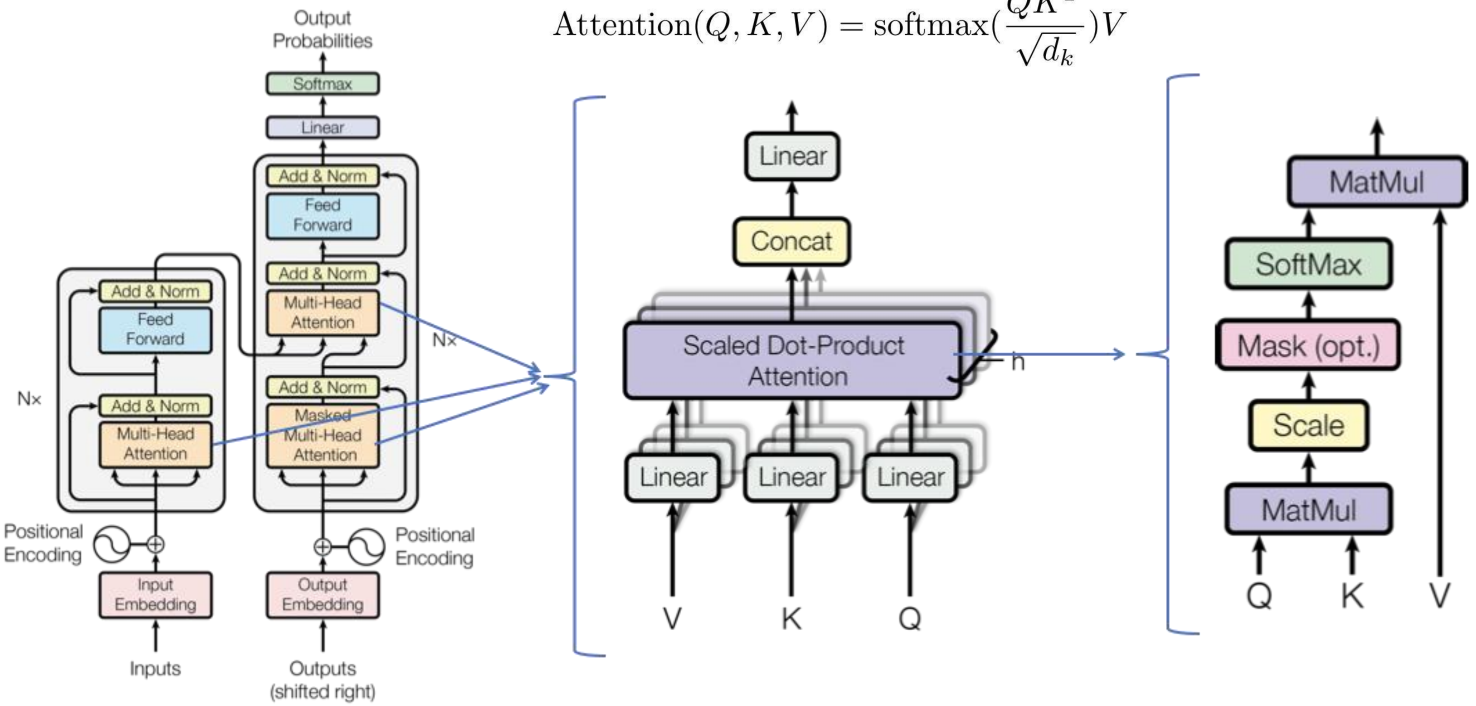
- An attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence **(Cheng et al., 2016).**



Cheng, Jianpeng & Dong, Li & Lapata, Mirella. (2016). Long Short-Term Memory-Networks for Machine Reading.

TRANSFORMER: ATTENTION IS ALL YOU NEED

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia. (2017). Attention Is All You Need.

PRE-TRAINED LANGUAGE MODEL

- **Transformer-based** models have made a significant impact on NLP, transfer learning using these language models is the novel standard of various tasks.

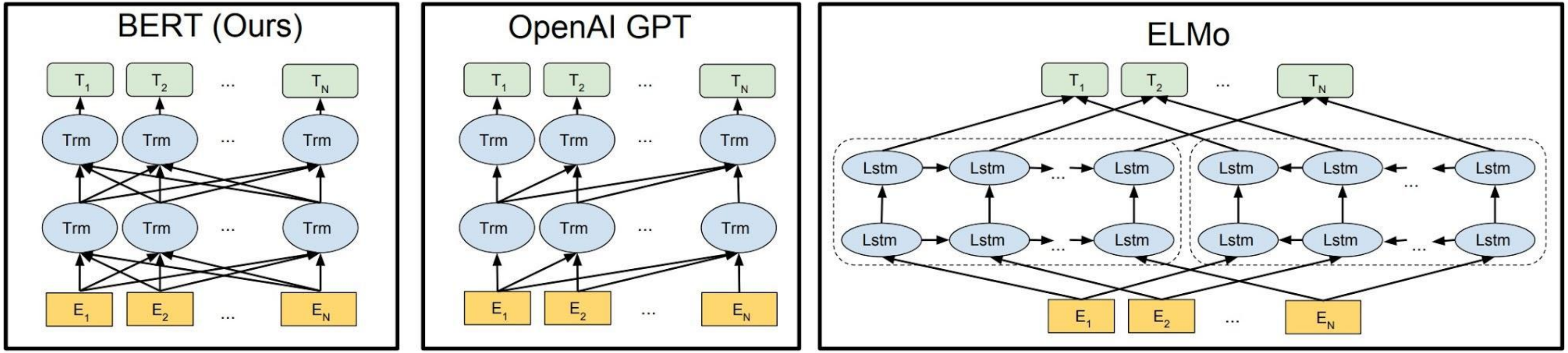


Image Credit: Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

TRANSFER LEARNING PIPELINE

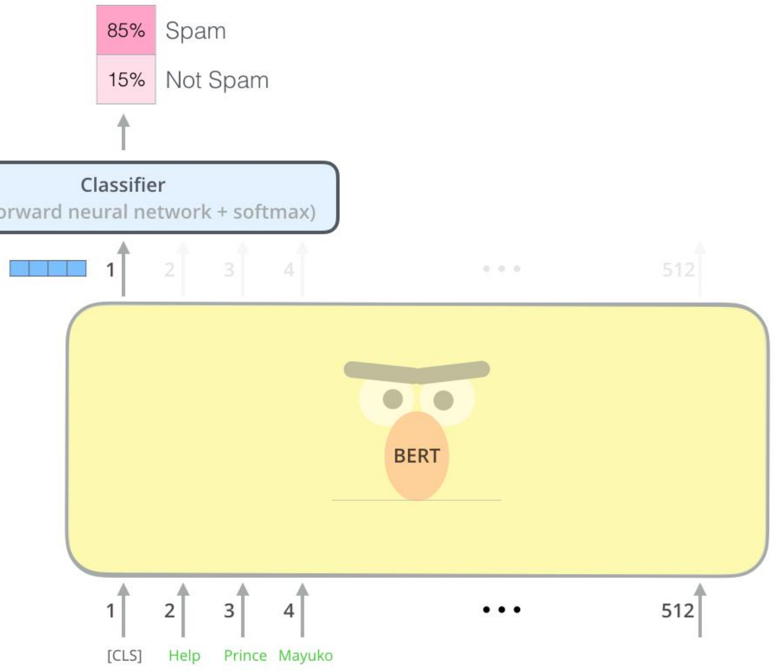


Image Credit: Jay Alammar. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). http://jalammar.github.io/illustrated-bert/

IMPLEMENTATION: VIETNAMESE SENTIMENT ANALYSIS

- **Sentivi** - a simple tool for sentiment analysis which is a wrapper of scikit-learn and PyTorch Transformers models. It is made for easy and faster pipeline to train and evaluate several classification algorithms.
- Source code: https://github.com/vndee/sentivi
- Documentation: https://sentivi.readthedocs.io/
- Install:
  - `pip install sentivi`

SENTIVI

- Text Encoder
  - One-hot
  - Bag of Words
  - Term Frequency - Inverse Document Frequency
  - Word2Vec **(Xuan-Son et al., 2019)**
  - Transformer tokenizer (SentencePiece for Transformer-based classifier)
- Classifier
  - Decision Tree
  - Gaussian Naive Bayes
  - Gaussian Process
  - Nearest Centroid
  - Support Vector Machine
  - Stochastic Gradient Descent
  - Multi-Layer Perceptron
  - Long Short Term Memory
  - Text Convolutional Neural Network
  - Transformer

Xuan-Son Vu, Thanh Vu, Son N. Tran, Lili Jiang. ETNLP: A Visual-Aided Systematic Approach to Select Pre-Trained Embeddings for a Downstream Task.In: Proceedings of the International Conference Recent Advances in Natural Language Processing 2019.

SENTIVI

```python
from sentivi import Pipeline
from sentivi.data import DataLoader, TextEncoder
from sentivi.classifier import SVMClassifier
from sentivi.text_processor import TextProcessor


if __name__ == '__main__':
    text_processor = TextProcessor(methods=['word_segmentation', 'remove_punctuation', 'lower'])

    pipeline = Pipeline(DataLoader(text_processor=text_processor, n_grams=3),
                        TextEncoder(encode_type='one-hot'),
                        SVMClassifier(num_labels=3))

    train_results = pipeline(train='./data/dev.vi', test='./data/dev_test.vi')
    print(train_results)

    pipeline.save('./weights/pipeline.sentivi')
    _pipeline = Pipeline.load('./weights/pipeline.sentivi')

    predict_results = _pipeline.predict(['hàng ok đầu tuýp có một số không vừa ốc siết. chỉ được một số đầu thôi .cần '
                                         'nhất đầu tuýp 14 mà không có. không đạt yêu cầu của mình sử dụng',
                                         'Son đẹpppp, mùi hương vali thơm nhưng hơi nồng, chất son mịn, màu lên chuẩn, '
                                         'đẹppppp'])
    print(predict_results)
    print(f'Decoded results: {_pipeline.decode_polarity(predict_results)}')
```

Text pre-processor

Running pipeline

Load data from text file

Text encoding type

Classifier

Save pipeline

Load saved pipeline

Predict

SENTIVI

```
One Hot Text Encoder: 100%|██████████| 6/6 [00:00<00:00, 11602.50it/s]
One Hot Text Encoder: 100%|██████████| 2/2 [00:00<00:00, 4966.61it/s]
Input features view be flatten into np.ndarray(6, 35328) for scikit-learn classifier.
Training classifier...
Testing classifier...
Training results:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         1
           1       0.75      1.00      0.86         3
           2       1.00      1.00      1.00         2

    accuracy                           0.83         6
   macro avg       0.92      0.67      0.62         6
weighted avg       0.88      0.83      0.76         6

Test results:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1
           2       1.00      1.00      1.00         1

    accuracy                           1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2


Saved model to ./weights/pipeline.sentivi
Loaded model from ./weights/pipeline.sentivi
Input features view be flatten into np.ndarray(2, 35328) for scikit-learn classifier.
[2 1]
Decoded results: ['#NEG', '#POS']
One Hot Text Encoder: 100%|██████████| 2/2 [00:00<00:00, 10796.15it/s]
```

Training results

Test results

Prediction results

SENTIVI: A PRODUCTION READY TOOL

```python
# serving.py
from sentivi import Pipeline, RESTServiceGateway

pipeline = Pipeline.load('./weights/pipeline.sentivi')
server = RESTServiceGateway(pipeline).get_server()
```

Load pre-trained pipeline and initialize your REST API server

```
# pip install uvicorn python-multipart
uvicorn serving:server --host 127.0.0.1 --port 8000
```

Serve API using standard ASGI library

```
curl --location --request POST 'http://127.0.0.1:8000/get_sentiment/' \
     --form 'text=Son đẹpppp, mùi hương vali thơm nhưng hơi nồng'

# response
{ "polarity": 2, "label": "#POS" }
```

Call API using **curl**

DATASET

- Crawl from e-commerce website Lazada.
- Train: 2000 samples.
- Test: 500 samples.
- 3 labels: **negative** (#NEG), **positive** (#POS), **neutral** (#NEU).

“Áo rất đẹp. Giao hàng nhanh” → **#POS**

“shop giao hàng lỗi, vải rách 1 bên tay” → **#NEG**

“đặt màu đỏ giao màu đen mà thôi cũng tạm ổn” → **#NEU**

EXPERIMENTS

- We trained **90 models** on **2 datasets** (crawl data and VLSP-small)**:**
  - Traditional Machine Learning Classifier: Naive Bayes, Decision Tree, Nearest Centroid, SGD, SVM.
  - Deep Learning Classifier: MLP, LSTM, CNN.
  - Transformer: PhoBERT **(Dat et al., 2020)**.
  -
- Full report: https://docs.google.com/spreadsheets/d/1ZYHHGeRAp2xfdJhZT2RnVk0b7sep606H9aGmqyXs3kg/edit?usp=sharing

Dat Quoc Nguyen, Anh Tuan Nguyen. PhoBERT: Pre-trained language models for Vietnamese. Findings of EMNLP 2020.

RESULTS

	Naive Bayes	Decision Tree	SVM	LSTM + Attn	BiLSTM + Attn	TextCNN
One-hot	0.77/0.35/0.78	0.87/0.37/0.83	0.86/0.31/0.83	0.88/0.31/0.94	0.88/0.31/0.93	0.89/0.34/0.94
BOW	0.49/0.30/0.59	0.87/0.44/0.85	0.88/0.31/0.83	0.88/0.37/0.92	0.87/0.39/0.89	0.88/0.31/0.94
TF-IDF	0.49/0.30/0.59	0.86/0.43/0.84	0.87/0.32/0.83	0.88/0.44/0.91	0.87/0.39/0.91	0.88/0.31/0.94
W2V	0.07/0.05/0.01	0.86/0.41/0.84	0.88/0.33/0.83	0.88/0.31/0.94	0.88/0.31/0.94	0.87/0.36/0.91
	Tranformer (PhoBERT)					
Sentence Piece	0.90/0.37/0.94					

Experiment results in crawl dataset

RESULTS

	Naive Bayes	Decision Tree	SVM	LSTM + Attn	BiLSTM + Attn	TextCNN
One-hot	0.45/0.45/0.45	0.34/0.33/0.33	0.38/0.32/0.32	0.44/0.45/0.44	0.39/0.30/0.49	0.39/0.42/0.46
BOW	0.51/0.50/0.50	0.46/0.46/0.46	0.55/0.54/0.54	0.55/0.54/0.54	0.58/0.58/0.58	0.39/0.41/0.47
TF-IDF	0.50/0.49/0.49	0.45/0.45/0.45	0.55/0.39/0.39	0.55/0.54/0.54	0.51/0.51/0.51	0.38/0.40/0.46
W2V	0.34/0.23/0.23	0.34/0.34/0.34	0.40/0.39/0.40	0.40/0.39/0.39	0.53/0.52/0.53	0.48/0.48/0.48
	Tranformer (PhoBERT)					
Sentence Piece	0.57/0.56/0.57					

Experiment results in VLSP-small

Q&A