Text Classification

CPE 393: Text Analytics

Dr. Sansiri Tarnpradab

Department of Computer Engineering
King Mongkut's University of Technology Thonburi

Intro

Pattern Matching

Text Visualization

Web Scraping

Text Preparation

Text Feature Representation

Text Classification

Text Summarization

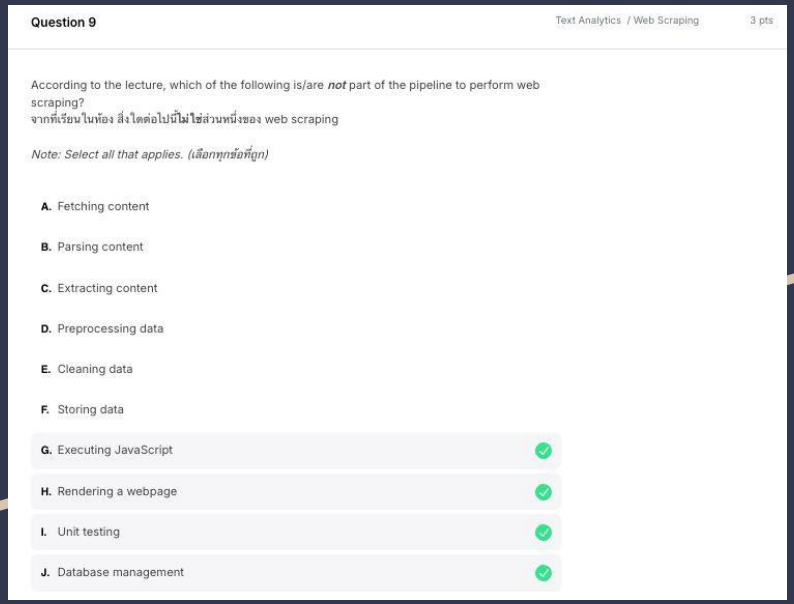
Text Clustering

Topic Modeling

TBA Advanced Topic

???

Announcement



Question 9 — Text Analytics / Web Scraping — 3 pts

According to the lecture, which of the following is/are ***not*** part of the pipeline to perform web scraping?
จากที่เรียนในห้อง สิ่งใดต่อไปนี้ไม่ใช่ส่วนหนึ่งของ web scraping

Note: Select all that applies. (เลือกทุกข้อที่ถูก)

- **A.** Fetching content
- **B.** Parsing content
- **C.** Extracting content
- **D.** Preprocessing data
- **E.** Cleaning data
- **F.** Storing data
- **G.** Executing JavaScript ✓
- **H.** Rendering a webpage ✓
- **I.** Unit testing ✓
- **J.** Database management ✓

Project Proposal

- Proposal: 1-page
- Group: 3-4 members per group
- Options: Application or Research
- Examples
- Due 9/10/2024

Midterm Exam

Tentatively by next week

Quiz 1

Finalized the grading criteria

Outline

- Introducing Text Classification
- Application
- Formulation
- Types of Classification
- Levels of Classification
- Contributing Factors
- Evaluation Metrics
- Methods
  - Traditional
  - Modern

Classification

Text Classification

Text Classifier

What *is*

Classification?

➔ The operation of separating various entities into several classes.[1]

➔ The problem of identifying which of a set of categories an observation belongs to.[2]

1 Handbook of Statistical Analysis and Data Mining Applications (Second Edition)
2 Wikipedia, "Statistical Classification"

What *is*

Text Classification?

- A task of assigning a label or class to a given text[3]
- Involves classifying text by performing text analysis techniques on your text-based documents[4]
- **Goal**: to categorize or predict a class of unseen text documents, often with the help of supervised machine learning[5]

3 Hugging Face, "Text Classification"
4 MonkeyLearn, "Document Classification"
5 datacamp, "Understanding Text Classification in Python"

What *is*

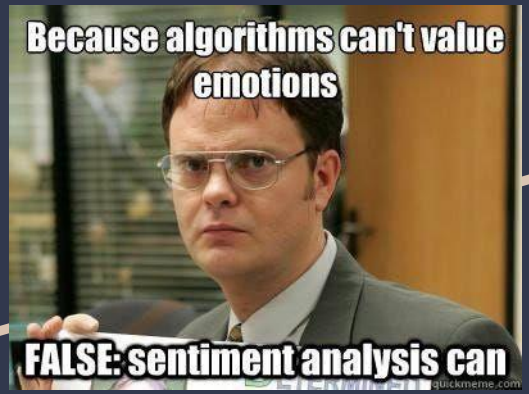
Text Classifier?

- ➔ A machine learning model that has been trained to recognize patterns in natural language text[6]
- ➔ Is trained by being showned lots of examples of text already labeled[6]

6 Apple, "Creating a text classifier model"

Applications *of*

Text Classification



Ref: https://medium.com/analytics-vidhya/working-on-nlp-with-textblob-56d0344bb648

Sentiment classification (Sentiment analysis)

- Movie reviews, Restaurant reviews, etc. ☺️😀☹️🤢
- Social media monitoring

Spam classification

- Detect unsolicited and unwanted emails
- Improve user experience

Language Detection

- Detect language of customer feedback
- Often a step before text is analyzed

Topic Labeling

- Assignment of “tags” based on the topic or theme
- Help in summarizing and distinguishing a piece of text based on its theme

Fake news detection

- Clickbait detection
- Sarcasm detection
- Bipartisan Press’s Political Bias detection

Text summarization

- Extractive text summarization
- Abstractive text summarization

Formulate

Text Classification

[1,0,0,1,1,0,0,1,0,0,0,1,0,0,0]

- Supervised Learning
- Training data → Text items
- Corresponding label
- Model will learn a relationship between data and labels

Example: Text summarization task

Sentences: **s** = $[s_1, s_2, \dots, s_N]$

Labels: **t** = $[t_1, t_2, \dots, t_N]$

Binary labels

1 → part-of-summary sentence
0 → Otherwise

Task: Find the most probable tag sequence, given sentences in the document

$$\arg\max_{\boldsymbol{t} \in \mathcal{T}} p(\boldsymbol{t}|\boldsymbol{s})$$

Types *of*

Text Classification

Binary Classification

Each item belongs to *one of the two* classes
Examples: Spam (spam/ham), Fake News (True/Fake), Sentiment (Positive/Negative)

Multiclass Classification

Single-label:

Each item belongs to *one of the multiple* classes
Examples: Label news articles under one of these categories {business, entertainment, politics, sport or tech}

Multi-label:

Each item belongs to *multiple* classes
Examples: Label an article to classes of topics (mathematics, biology, chemistry, ecology, medical, etc.)

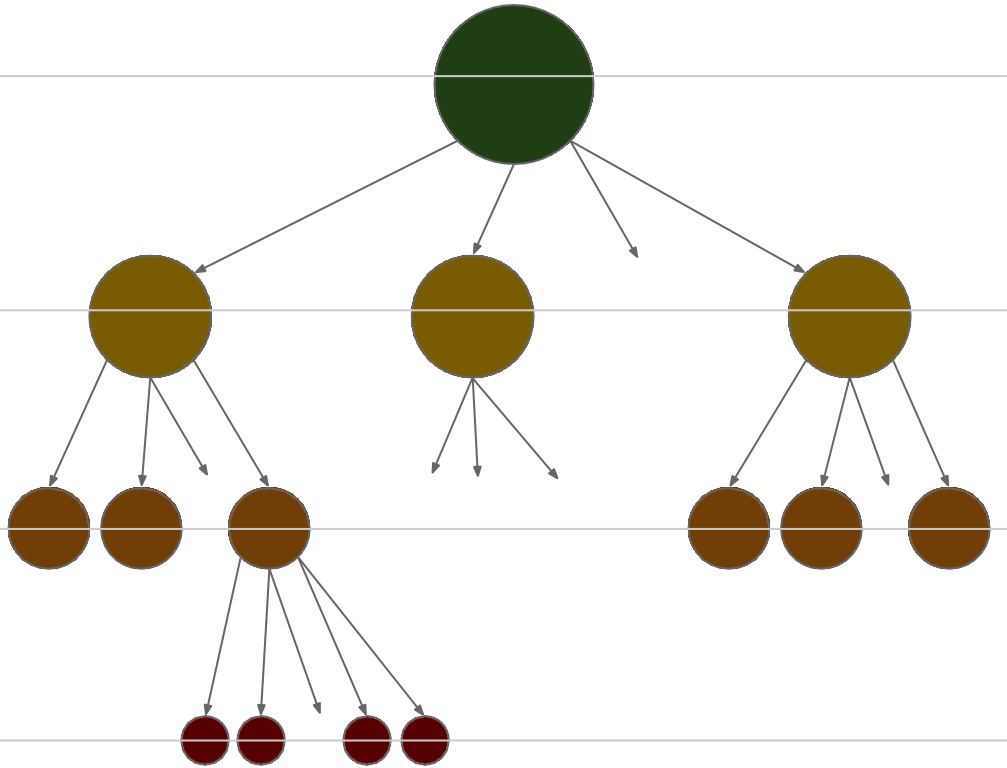
Different **Levels**

Document-level

Paragraph-level

Sentence-level

Sub-sentence-level



Factors

affecting the quality of

Text Classifier

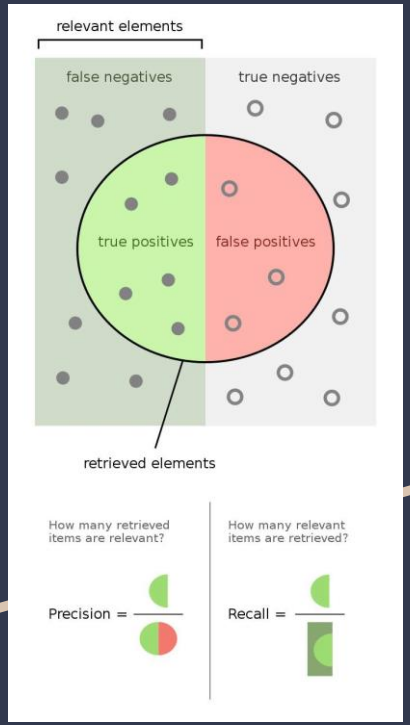
Data

- Quantity
- Quality

Algorithms

Feature Engineering

Evaluation *Metrics*



Ref: https://en.wikipedia.org/wiki/Precision_and_recall

Accuracy is the ratio of correctly predicted events to the total events (events ≡ examples)

$$Acc = \frac{No.\ correctly\ predicted\ examples}{Total\ no.\ of\ examples}$$

Recall is the ratio of a number of events you can correctly recall to a number of all correct events

Precision is the ratio of a number of events you can correctly recall to a number all events you recall (mix of correct and wrong recalls). In other words, it is how precise of your recall.

F-measure (F1-score) is the balance (harmonic mean) between precision and recall

$$F1 = 2\frac{P \times R}{P + R}$$

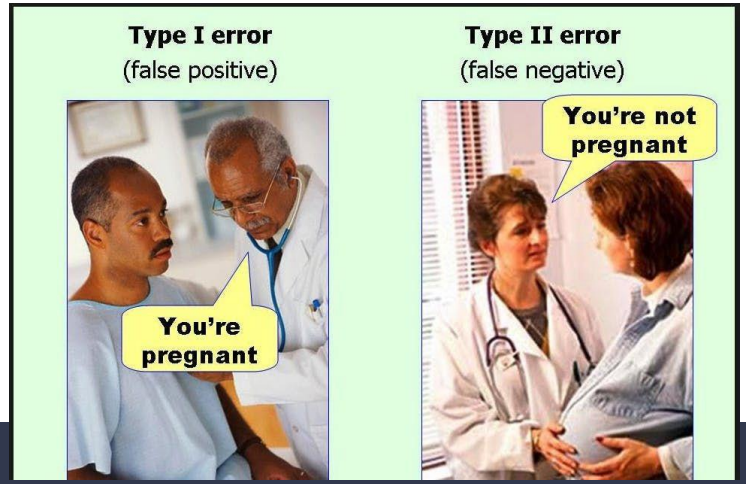
		Predicted Class	
		Yes	**No**
Actual Class	**Yes**	TP	FN (Type II error)
	No	FP (Type I error)	TN

$$Acc = \frac{TP+TN}{TP+TN+FP+FN}$$

$$R = \frac{TP}{TP+FN}$$

$$P = \frac{TP}{TP+FP}$$

$$F1 = 2\frac{P \times R}{P+R}$$



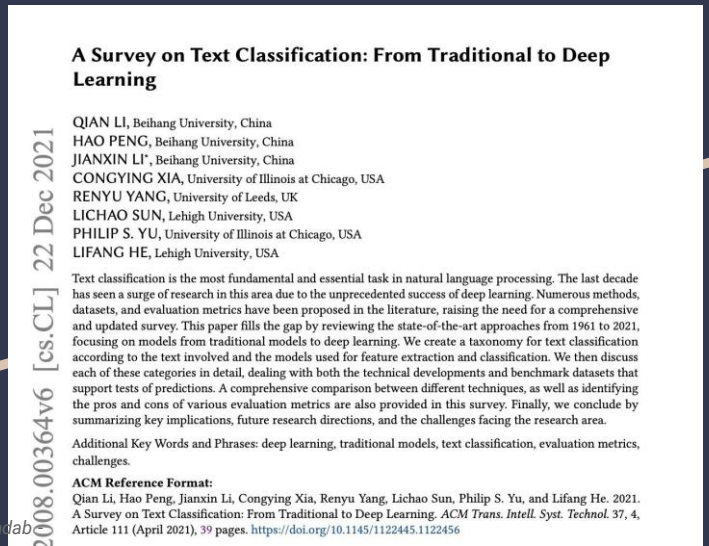
Confusion Matrix

Ref: https://hackernoon.com/idiots-guide-to-precision-recall-and-confusion-matrix-b32d36463556

Methods *for*

Text Classification

A link to checkout this survey paper



A Survey on Text Classification: From Traditional to Deep Learning

QIAN LI, Beihang University, China
HAO PENG, Beihang University, China
JIANXIN LI*, Beihang University, China
CONGYING XIA, University of Illinois at Chicago, USA
RENYU YANG, University of Leeds, UK
LICHAO SUN, Lehigh University, USA
PHILIP S. YU, University of Illinois at Chicago, USA
LIFANG HE, Lehigh University, USA

Text classification is the most fundamental and essential task in natural language processing. The last decade has seen a surge of research in this area due to the unprecedented success of deep learning. Numerous methods, datasets, and evaluation metrics have been proposed in the literature, raising the need for a comprehensive and updated survey. This paper fills the gap by reviewing the state-of-the-art approaches from 1961 to 2021, focusing on models from traditional models to deep learning. We create a taxonomy for text classification according to the text involved and the models used for feature extraction and classification. We then discuss each of these categories in detail, dealing with both the technical developments and benchmark datasets that support tests of predictions. A comprehensive comparison between different techniques, as well as identifying the pros and cons of various evaluation metrics are also provided in this survey. Finally, we conclude by summarizing key implications, future research directions, and the challenges facing the research area.

Additional Key Words and Phrases: deep learning, traditional models, text classification, evaluation metrics, challenges.

ACM Reference Format:
Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2021. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Trans. Intell. Syst. Technol.* 37, 4, Article 111 (April 2021), 39 pages. https://doi.org/10.1145/1122445.1122456

Traditional

- Naïve Bayes
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Decision Trees (DT)
- Integration (Ensemble)

Modern (Deep learning)

- Convolutional Neural Network (CNN)
- Recurrent Neural Network (RNN)
- Gated Recurrent Unit (GRU)
- Long Short-term Memory (LSTM)
- Transformer
- Others …
  - Transfer learning

Traditional Method

Naïve Bayes

Naïve Bayes: *Concept*

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

Posteriori = (Likelihood X Prior)/Evidence

- Simplest Machine Learning model
- Based on Bayes' Theorem
  - Use Bayesian inference from probability theory to infer the label, given the text input

Basics

- **X**: Data sample (aka "evidence")
- **H**: Hypothesis that X belongs to class C
- **P(H)**: Prior probability
- **P(X)**: Probability that sample data is observed
- **P(X|H)** (Likelihood): the probability of observing the sample X, given that the hypothesis holds

Classification task

Determine **P(H|X)**, the probability that the hypothesis holds given the observed data sample X

P(H|X): Posteriori probability of a hypothesis H

Naïve Bayes: *Text Classification (1)*

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

Posteriori = (Likelihood X Prior)/Evidence

P(c|d) → Determine a class **c** given the document **d**

- c ∈ C where C denotes all classes
- **Goal**: the classifier returns the class ĉ which has the maximum posterior probability given the document.

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d)$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

Note that the denominator **P(d)** can be dropped because it is constant for all classes.

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

A document **d** can be represented as a set of features **f_1, f_2, … , f_n**

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, \ldots, f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

Naïve Bayes: *Text Classification (2)*

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, ..., f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

Naïve Bayes Assumption → conditional independence assumption that the probabilities **$P(f_i|c)$** are independent given the class **c**

$$P(f_1, f_2, ..., f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot ... \cdot P(f_n|c)$$

The final eq. for the class chosen by naive Bayes classifier is:

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{f \in F} P(f|c)$$

Naïve Bayes: *Text Classification Example*

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Ref: https://web.stanford.edu/~jurafsky/slp3/4.pdf (page 7)

- **Task**: Classifying sentiment of an input document (Positive or Negative)
- Training set = 5
- Test set = 1
- **Goal**: determine a class of the sentence *"predictable with no fun"*
  - Sound positive or negative to you?

TODO:

- ➔ Feature vector
- ➔ Compute prior probability of each class
- ➔ Compute conditional probabilities of each feature
- ➔ Anything else?

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Ref: https://web.stanford.edu/~jurafsky/slp3/4.pdf (page 7)

TODO:

- ❏ Feature vector → Bag of words
- ❏ Compute prior probability of each class
- ❏ Compute conditional probabilities of each feature
- ❏ Anything else?

$$P(f_1, f_2, ..., f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot ... \cdot P(f_n|c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c) \prod_{f \in F} P(f|c)$$

Training	Class	just	plain	boring	entirely	predictable	and	lacks	energy	no	surprises	very	few	laughs	powerful	the	most	fun	film	of	summer
just plain boring	Negative	1	1	1																	
entirely predictable and lacks energy	Negative				1	1	1	1	1												
no surprises and very few laughs	Negative						1			1	1	1	1	1							
very powerful	Positive											1			1						
the most fun film of the summer	Positive															2	1	1	1	1	1
	Add one Smoothing	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Conditional probability	Prior probablity	just	plain	boring	entirely	predictable	and	lacks	energy	no	surprises	very	few	laughs	powerful	the	most	fun	film	of	summer
Negative	0.6000	0.0588	0.0588	0.0588	0.0588	0.0588	0.0882	0.0588	0.0588	0.0588	0.0588	0.0588	0.0588	0.0588	0.0294	0.0294	0.0294	0.0294	0.0294	0.0294	0.0294
Positive	0.4000	0.0345	0.0345	0.0345	0.0345	0.0345	0.0345	0.0345	0.0345	0.0345	0.0345	0.0690	0.0345	0.0345	0.0690	0.1034	0.0690	0.0690	0.0690	0.0690	0.0690

Inference		Likelihood																			
	Prior probablity	just	plain	boring	entirely	predictable	and	lacks	energy	no	surprises	very	few	laughs	powerful	the	most	fun	film	of	summer
predictable with no fun (Negative)	0.600					0.0588				0.0588								0.0294			
predictable with no fun (Positive)	0.400					0.0345				0.0345								0.0690			

0.0000611
0.0000328

The model predicts the class ***negative*** for the given test sentence

Modern Methods

Some **Background**

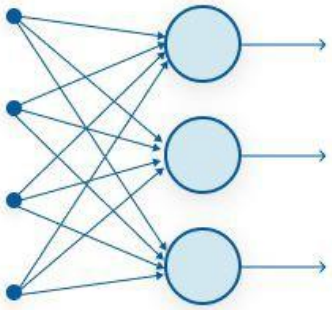
2003 Feedforward neural language model

2016 RNN-based

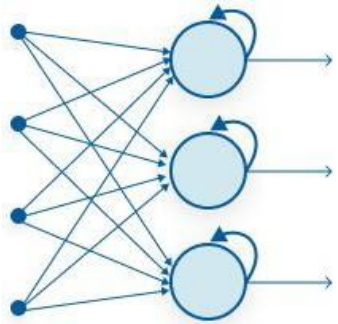
2017 Transformer-based

2019 Large transformer-based (GPT-2)

2022 Massive transformer-based for chatbot (chatGPT)



Feed-Forward Neural Network

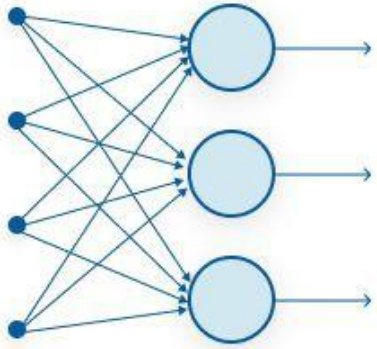


Recurrent Neural Network

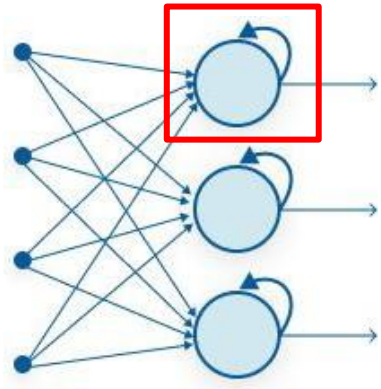
Ref: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/

ANN **vs** *RNN*

- The cyclic structure enables RNNs to maintain a memory of past inputs
- The cyclic structure makes RNNs suitable for sequential data processing tasks.



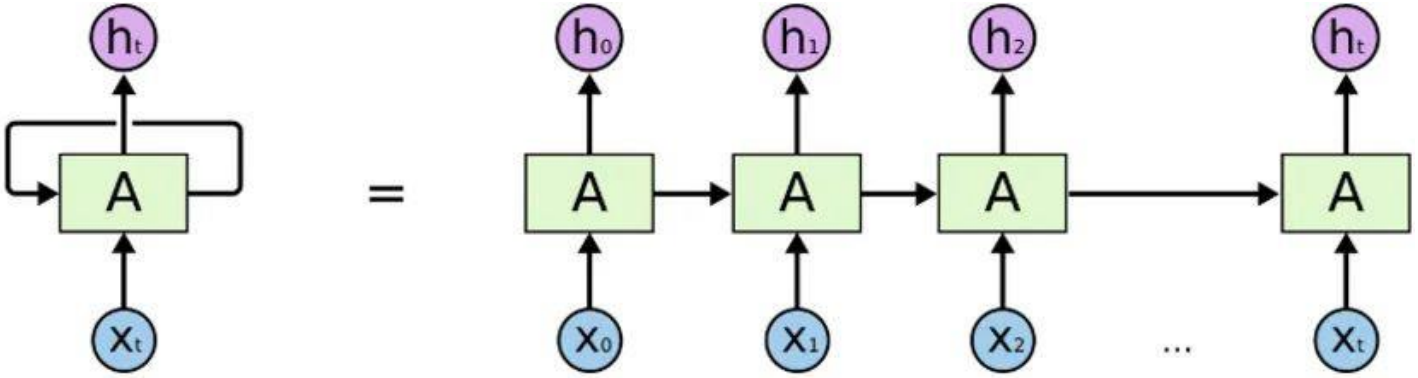
Feed-Forward Neural Network



Recurrent Neural Network

RNN: *Recurrent Neural Networks*

$$h_t = \sigma(W^{(hh)} h_{t-1} + W^{(hx)} x_t)$$



Weaknesses:

- Vanishing gradient problem → during backpropagation through time (BPTT)
- Capturing long-term dependencies
- Lack of Parallelization → computationally inefficient, longer training time

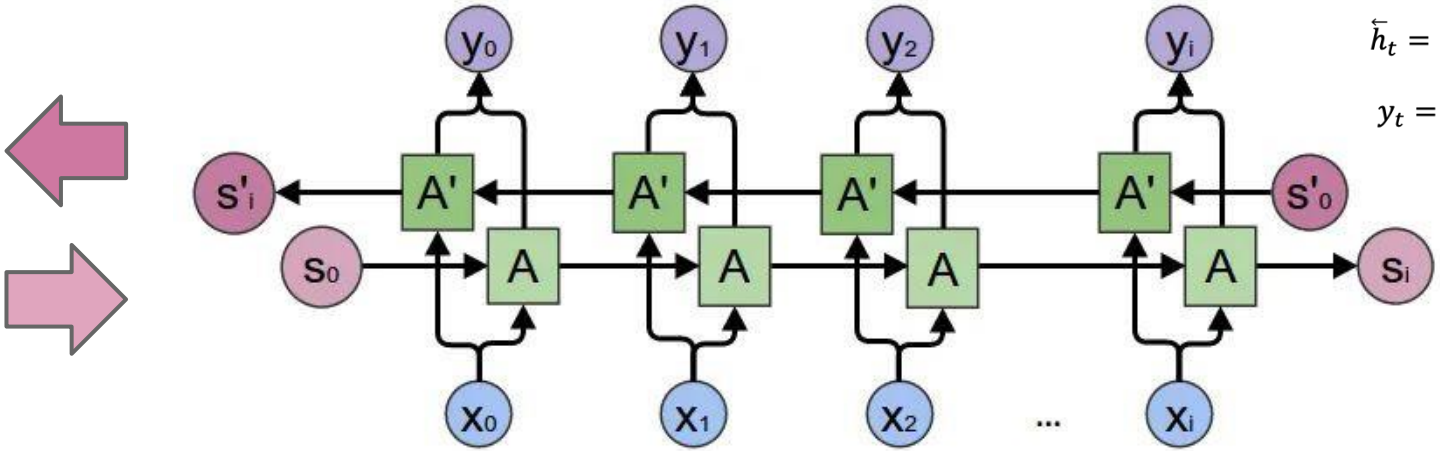
Ref img: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Bidirectional *RNN*

$$\vec{h}_t = \sigma(\vec{W}^{(hh)}\vec{h}_{t-1} + \vec{W}^{(hx)}x_t)$$

$$\overleftarrow{h}_t = \sigma(\overleftarrow{W}^{(hh)}\overleftarrow{h}_{t+1} + \overleftarrow{W}^{(hx)}x_t)$$

$$y_t = f(\,[\vec{h}_t; \overleftarrow{h}_t\,]\,)$$

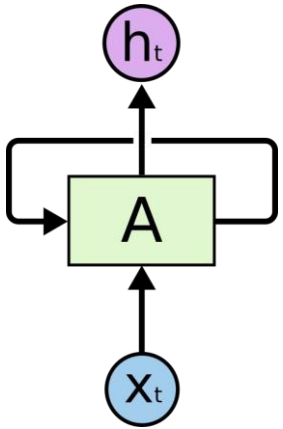


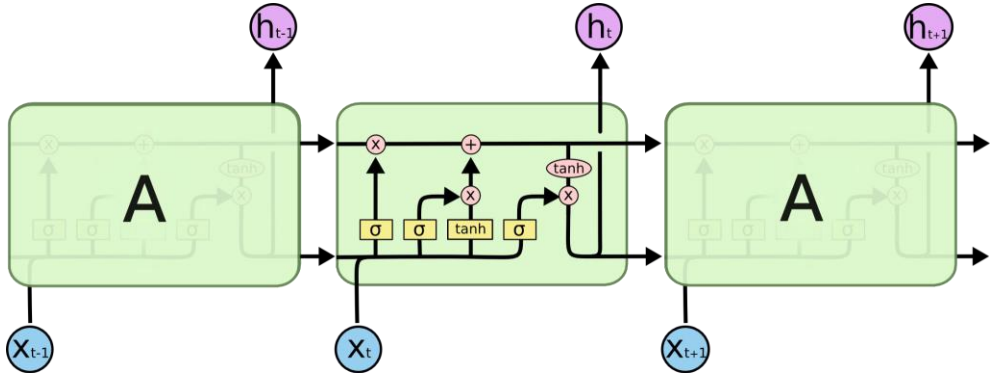
- Two RNNs stacked on top of each other
- Incorporate both left and right context
- Output is computed based on the hidden state of both RNNs

Ref img: http://colah.github.io/posts/2015-09-NN-Types-FP/

LSTM: *Long-Short Term Memory Networks*

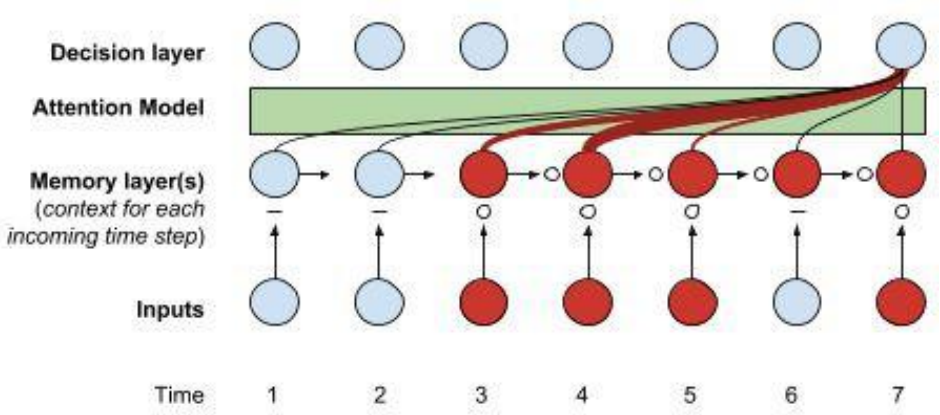
- One of the variants of RNN
- Better than traditional RNN for its memory
- Gates: a forget gate, an input gate, and an output gate.





Ref img: https://www.naukri.com/learning/articles/rnn-vs-gru-vs-lstm/

Attention *Mechanism*

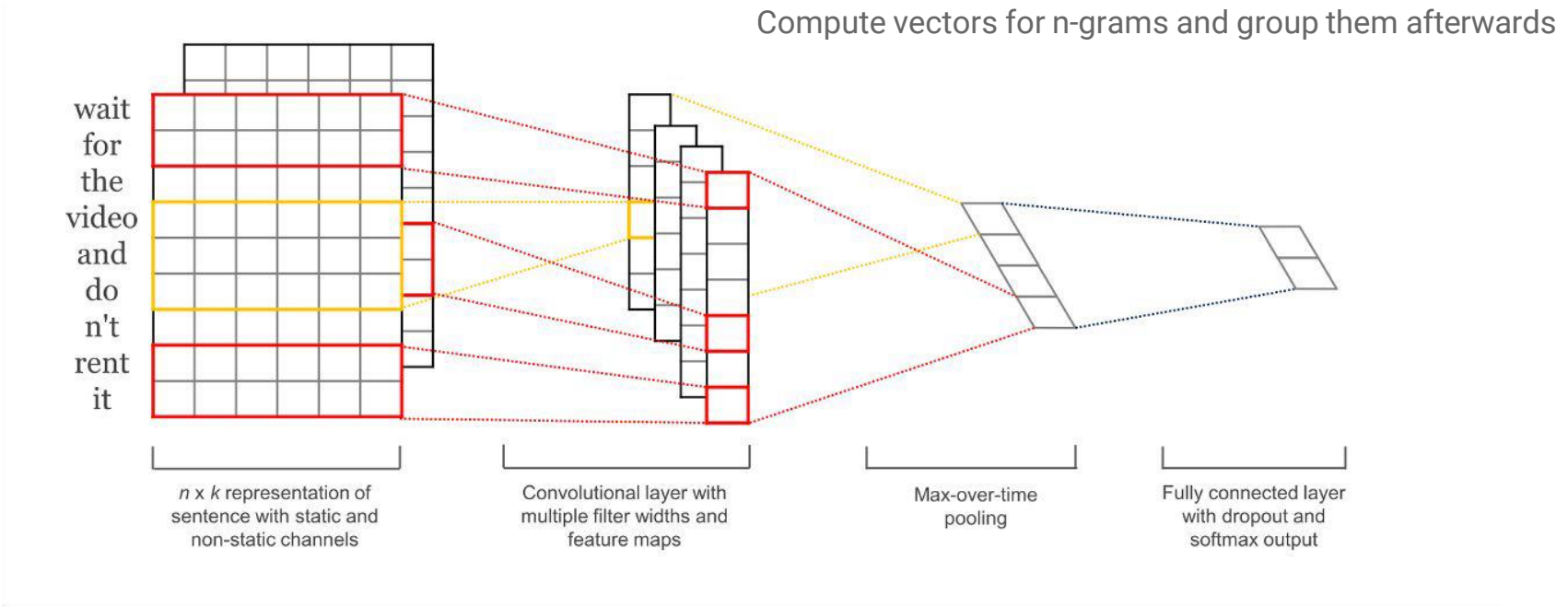


- Enable the model to focus on relevant parts of the input sequence (aka a highlighter)
- The model selectively attends to different parts of the input sequence
- Attention weights are computed using a softmax function [0,1]

Ref img: https://www.analyticsvidhya.com/blog/2023/06/learn-attention-models-from-scratch/

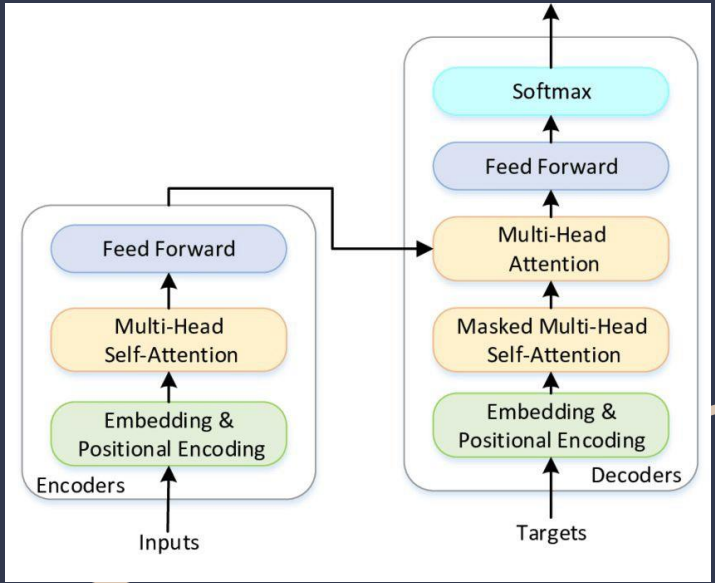
CNN: *Convolutional Neural Networks*

Compute vectors for n-grams and group them afterwards



Ref img: Kim, Y. "Convolutional Neural Networks for Sentence Classification", EMNLP (2014)

Transformer *Model*



Ref img:
Anwar, Azmat, et al. "Constructing Uyghur Named Entity Recognition System Using Neural Machine Translation Tag Projection." Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30–November 1, 2020, Proceedings 19. Springer International Publishing, 2020.

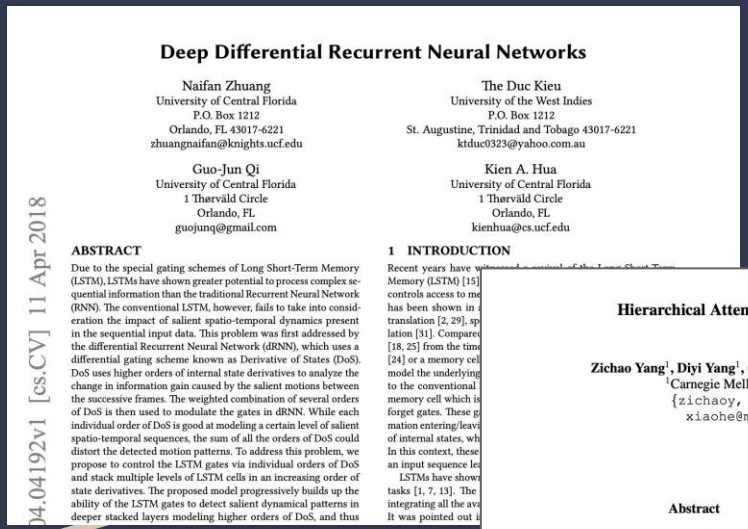
Important Elements

- Input Representation
  - Token embedding
  - Positional encoding
- Encoder
  - Multi-Head Self-Attention Mechanism
  - Feed-Forward Neural Network
- Decoder
  - Multi-Head Self-Attention Mechanism
  - Encoder-Decoder Attention Mechanism
  - Feed-Forward Neural Network (FFN)
- Output Layer

More Examples

- GRU
- Bidirectional RNN/LSTM
- Differential RNN
- CNN
- HAN (Hierarchical Attention Network)
- BERT

Differential RNN



Deep Differential Recurrent Neural Networks

Naifan Zhuang
University of Central Florida
P.O. Box 1212
Orlando, FL 43017-6221
zhuangnaifan@knights.ucf.edu

The Duc Kieu
University of the West Indies
P.O. Box 1212
St. Augustine, Trinidad and Tobago 43017-6221
ktduc0323@yahoo.com.au

Guo-Jun Qi
University of Central Florida
1 Thørväld Circle
Orlando, FL
guojunq@gmail.com

Kien A. Hua
University of Central Florida
1 Thørväld Circle
Orlando, FL
kienhua@cs.ucf.edu

ABSTRACT

Due to the special gating schemes of Long Short-Term Memory (LSTM), LSTMs have shown greater potential to process complex sequential information than the traditional Recurrent Neural Network (RNN). The conventional LSTM, however, fails to take into consideration the impact of salient spatio-temporal dynamics present in the sequential input data. This problem was first addressed by the differential Recurrent Neural Network (dRNN), which uses a differential gating scheme known as Derivative of States (DoS). DoS uses higher orders of internal state derivatives to analyze the change in information gain caused by the salient motions between the successive frames. The weighted combination of several orders of DoS is then used to modulate the gates in dRNN. While each individual order of DoS is good at modeling a certain level of salient spatio-temporal sequences, the sum of all the orders of DoS could distort the detected motion patterns. To address this problem, we propose to control the LSTM gates via individual orders of DoS and stack multiple levels of LSTM cells in an increasing order of state derivatives. The proposed model progressively builds up the ability of the LSTM gates to detect salient dynamical patterns in deeper stacked layers modeling higher orders of DoS, and thus

1 INTRODUCTION

Recent years have witnessed a revival of the Long Short-Term Memory (LSTM) [15] controls access to me has been shown in translation [2, 29], sp lation [31]. Compare [18, 25] from the time [24] or a memory cel model the underlying to the conventional memory cell which is forget gates. These g mation entering/leavi of internal states, wh In this context, these an input sequence le LSTMs have shown tasks [1, 7, 13]. The integrating all the ava It was pointed out i

04.04192v1 [cs.CV] 11 Apr 2018

HAN

Hierarchical Attention Networks for Document Classification

Zichao Yang[1], Diyi Yang[1], Chris Dyer[1], Xiaodong He[2], Alex Smola[1], Eduard Hovy[1]
[1]Carnegie Mellon University, [2]Microsoft Research, Redmond
{zichaoy, diyiy, cdyer, hovy}@cs.cmu.edu
xiaohe@microsoft.com alex@smola.org

Abstract

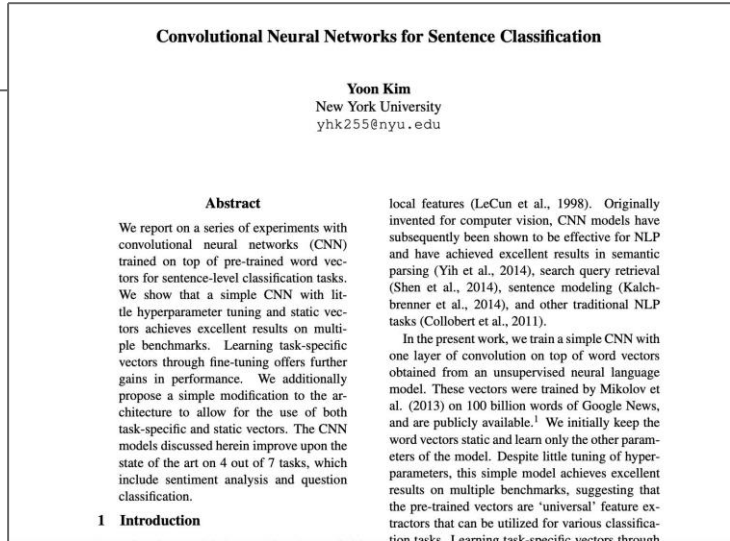
We propose a hierarchical attention network for document classification. Our model has two distinctive characteristics: (i) it has a hierarchical structure that mirrors the hierarchical structure of documents; (ii) it has two levels of attention mechanisms applied at the word- and sentence-level, enabling it to attend differentially to more and less important content when constructing the document representation. Experiments conducted on six large scale text classification tasks demonstrate that the proposed architecture outperform previous methods by a substantial margin. Visualization of the attention layers illustrates that the

pork belly = delicious . || scallops? || I don't even like scallops, and these were a-m-a-z-i-n-g . || fun and tasty cocktails. || next time I in Phoenix, I will go back here. || Highly recommend.

Figure 1: A simple example review from Yelp 2013 that consists of five sentences, delimited by period, question mark. The first and third sentence delivers stronger meaning and inside, the word *delicious, a-m-a-z-i-n-g* contributes the most in defining sentiment of the two sentences.

Although neural-network–based approaches to text classification have been quite effective (Kim, 2014; Zhang et al., 2015; Johnson and Zhang, 2014;

CNN



Convolutional Neural Networks for Sentence Classification

Yoon Kim
New York University
yhk255@nyu.edu

Abstract

We report on a series of experiments with convolutional neural networks (CNN) trained on top of pre-trained word vectors for sentence-level classification tasks. We show that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance. We additionally propose a simple modification to the architecture to allow for the use of both task-specific and static vectors. The CNN models discussed herein improve upon the state of the art on 4 out of 7 tasks, which include sentiment analysis and question classification.

1 Introduction

local features (LeCun et al., 1998). Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results in semantic parsing (Yih et al., 2014), search query retrieval (Shen et al., 2014), sentence modeling (Kalchbrenner et al., 2014), and other traditional NLP tasks (Collobert et al., 2011).

In the present work, we train a simple CNN with one layer of convolution on top of word vectors obtained from an unsupervised neural language model. These vectors were trained by Mikolov et al. (2013) on 100 billion words of Google News, and are publicly available.[1] We initially keep the word vectors static and learn only the other parameters of the model. Despite little tuning of hyperparameters, this simple model achieves excellent results on multiple benchmarks, suggesting that the pre-trained vectors are 'universal' feature extractors that can be utilized for various classifica-

Lab

- **Task**: Classify topic of complaints
- **Library**: Keras
- **Dataset**: Consumer complaint
- **Architecture**:
  - Input
  - Embedding
  - LSTM
  - Dense
  - Softmax
  - Output
- **Preprocessing data**
  - Grouping labels
  - Data cleansing
  - Padding/Truncating
  - Train/Test split
- Download file

Conclusion

- Introducing Text Classification
- Application
- Formulation
- Types of Classification
- Levels of Classification
- Contributing Factors
- Evaluation Metrics
- Methods
  - Traditional
  - Modern

Homework

- Use IMDB Dataset of 50K Movie Review (Download data)
  - Sample only 5K
  - Make sure that your data is balanced after the sampling
- Build a classifier to classify movie review sentiment
- Evaluate the model

Q & A