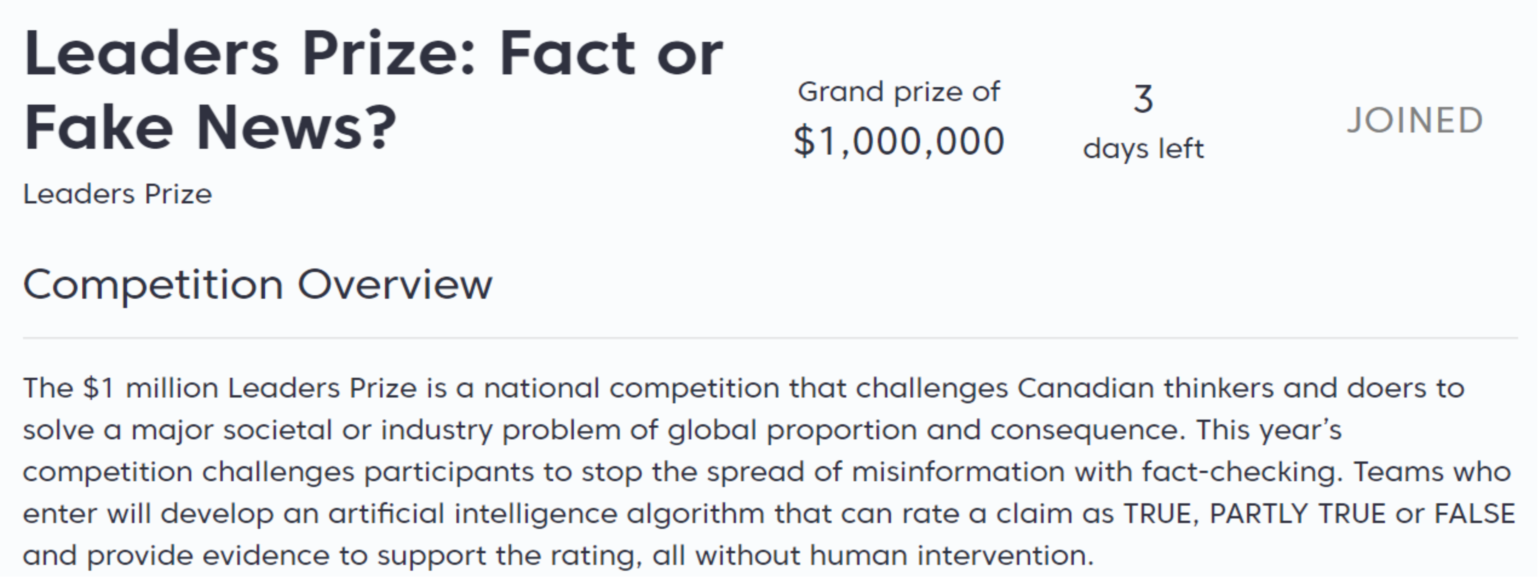
MIE1624 Course Project

Fake News Detection

Group 2

- Tinglin Duan
- Tianyi Xie
- Junxi Xu
- Zhennan Ying
- Qianyue Zhang

Introduction



Leaders Prize: Fact or Fake News?

Leaders Prize

Grand prize of $1,000,000

3 days left

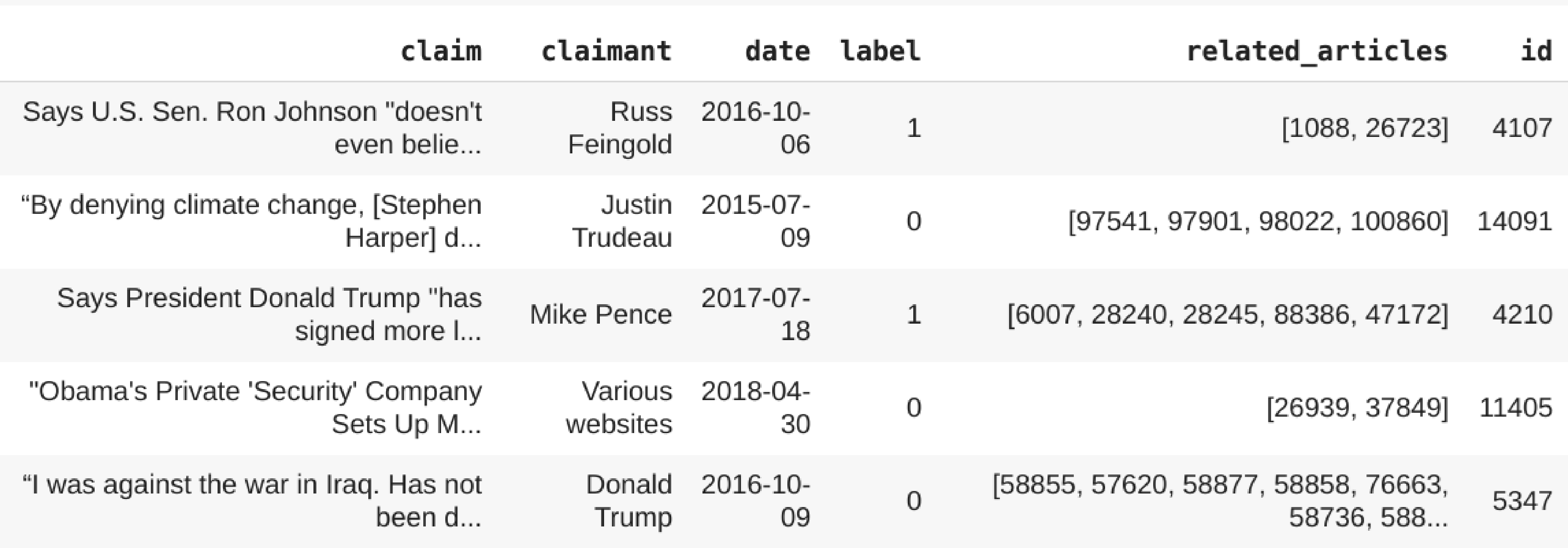
JOINED

Competition Overview

The $1 million Leaders Prize is a national competition that challenges Canadian thinkers and doers to solve a major societal or industry problem of global proportion and consequence. This year’s competition challenges participants to stop the spread of misinformation with fact-checking. Teams who enter will develop an artificial intelligence algorithm that can rate a claim as TRUE, PARTLY TRUE or FALSE and provide evidence to support the rating, all without human intervention.

Input	Metadata file with claims and their detailed information
Output	Predicted truth labels without any human intervention on real dataset
Models	Naïve Bayes, LSTM, CNN

Data Description

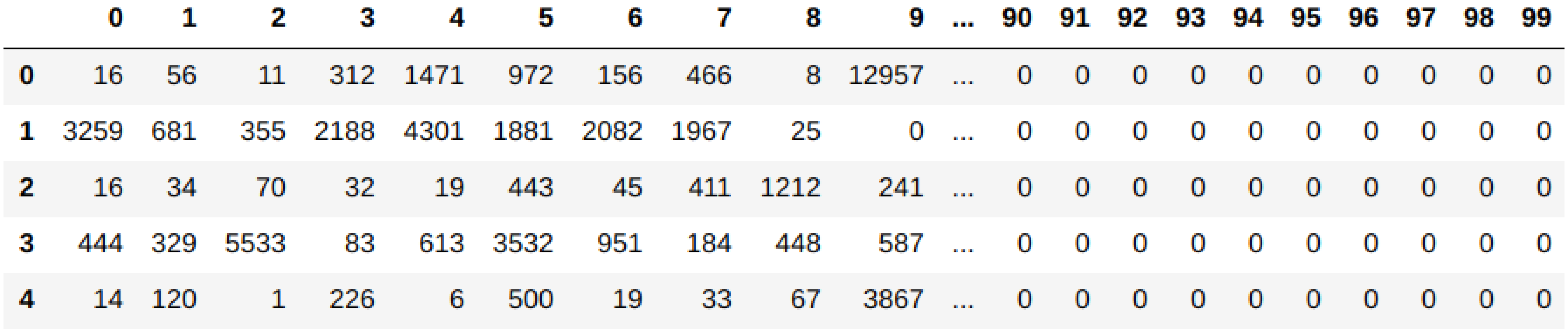


claim	claimant	date	label	related_articles	id
Says U.S. Sen. Ron Johnson "doesn't even belie...	Russ Feingold	2016-10-06	1	[1088, 26723]	4107
“By denying climate change, [Stephen Harper] d...	Justin Trudeau	2015-07-09	0	[97541, 97901, 98022, 100860]	14091
Says President Donald Trump "has signed more l...	Mike Pence	2017-07-18	1	[6007, 28240, 28245, 88386, 47172]	4210
"Obama's Private 'Security' Company Sets Up M...	Various websites	2018-04-30	0	[26939, 37849]	11405
“I was against the war in Iraq. Has not been d...	Donald Trump	2016-10-09	0	[58855, 57620, 58877, 58858, 76663, 58736, 588...	5347

Preprocessing

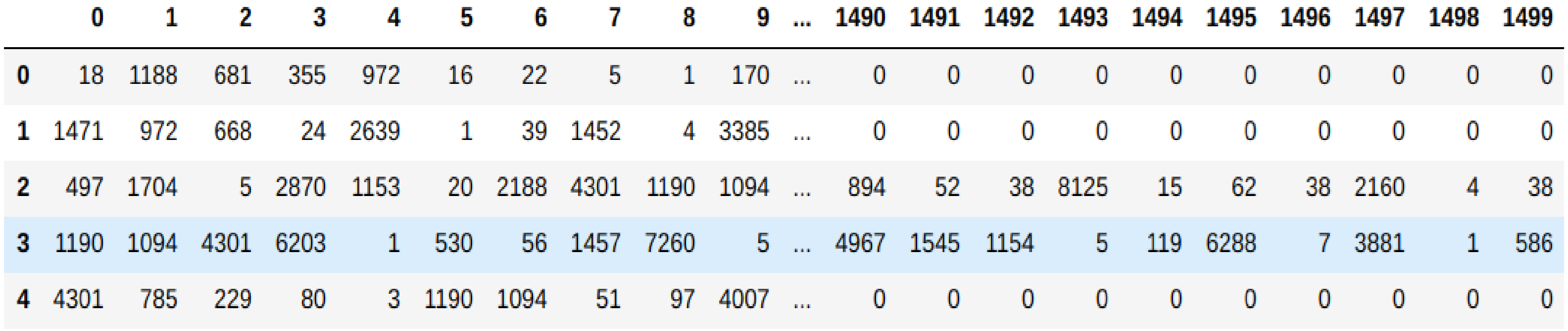
Tokenize and Padding

Claim Tokens:



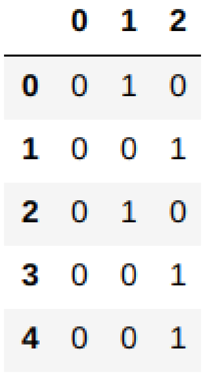
	0	1	2	3	4	5	6	7	8	9	...	90	91	92	93	94	95	96	97	98	99
0	16	56	11	312	1471	972	156	466	8	12957	...	0	0	0	0	0	0	0	0	0	0
1	3259	681	355	2188	4301	1881	2082	1967	25	0	...	0	0	0	0	0	0	0	0	0	0
2	16	34	70	32	19	443	45	411	1212	241	...	0	0	0	0	0	0	0	0	0	0
3	444	329	5533	83	613	3532	951	184	448	587	...	0	0	0	0	0	0	0	0	0	0
4	14	120	1	226	6	500	19	33	67	3867	...	0	0	0	0	0	0	0	0	0	0

Related Articles Tokens:



	0	1	2	3	4	5	6	7	8	9	...	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499
0	18	1188	681	355	972	16	22	5	1	170	...	0	0	0	0	0	0	0	0	0	0
1	1471	972	668	24	2639	1	39	1452	4	3385	...	0	0	0	0	0	0	0	0	0	0
2	497	1704	5	2870	1153	20	2188	4301	1190	1094	...	894	52	38	8125	15	62	38	2160	4	38
3	1190	1094	4301	6203	1	530	56	1457	7260	5	...	4967	1545	1154	5	119	6288	7	3881	1	586
4	4301	785	229	80	3	1190	1094	51	97	4007	...	0	0	0	0	0	0	0	0	0	0

Targets:



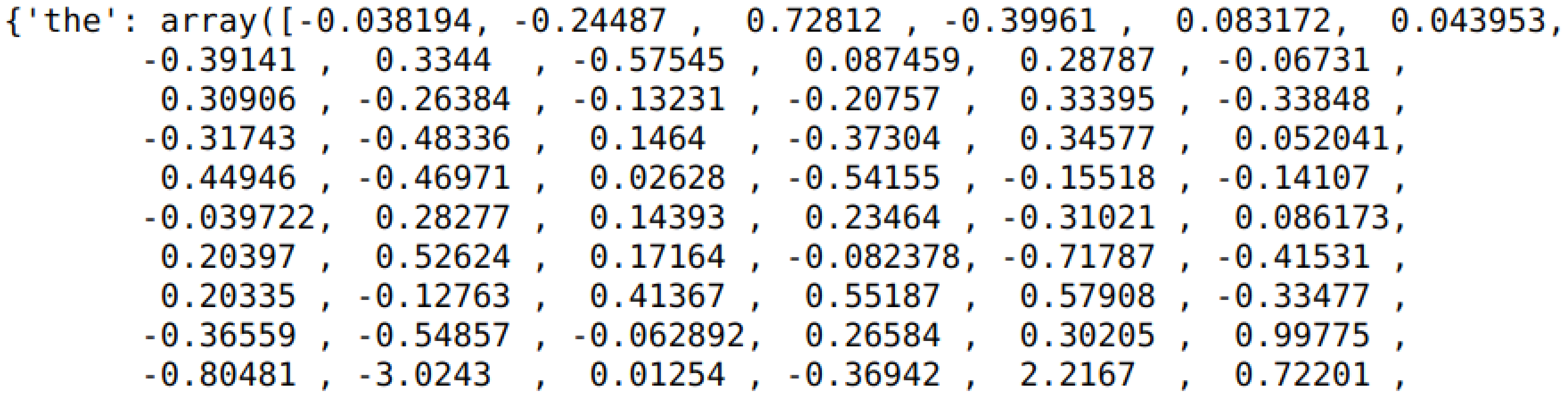
	0	1	2
0	0	1	0
1	0	0	1
2	0	1	0
3	0	0	1
4	0	0	1

Preprocessing

Word Embedding

Word Embedding:

glove.6B.100d.txt has words with their trained vectors in plain text. Transfer plain text to dictionary with ‘key’ as word and ‘value’ as vector. Map vectors to words in claims and get word embedding (weight) matrix as the input of the neural network



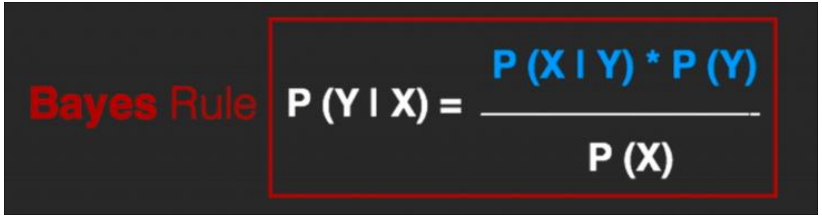
```
{'the': array([-0.038194, -0.24487 ,  0.72812 , -0.39961 ,  0.083172,  0.043953,
       -0.39141 ,  0.3344  , -0.57545 ,  0.087459,  0.28787 , -0.06731 ,
        0.30906 , -0.26384 , -0.13231 , -0.20757 ,  0.33395 , -0.33848 ,
       -0.31743 , -0.48336 ,  0.1464  , -0.37304 ,  0.34577 ,  0.052041,
        0.44946 , -0.46971 ,  0.02628 , -0.54155 , -0.15518 , -0.14107 ,
       -0.039722,  0.28277 ,  0.14393 ,  0.23464 , -0.31021 ,  0.086173,
        0.20397 ,  0.52624 ,  0.17164 , -0.082378, -0.71787 , -0.41531 ,
        0.20335 , -0.12763 ,  0.41367 ,  0.55187 ,  0.57908 , -0.33477 ,
       -0.36559 , -0.54857 , -0.062892,  0.26584 ,  0.30205 ,  0.99775 ,
       -0.80481 , -3.0243  ,  0.01254 , -0.36942 ,  2.2167  ,  0.72201 ,
```

example - “the” and its vector in glove.6B.100d.txt

Model Implementation

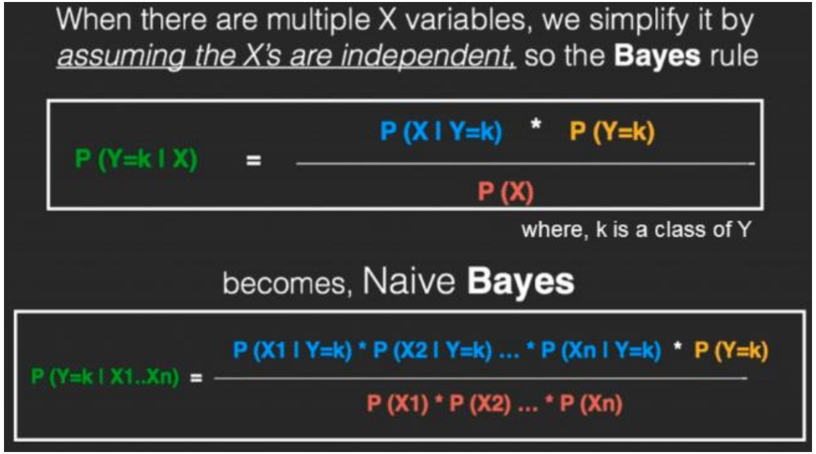
Naïve Bayes

Naive Bayes



Bayes Rule

$$P\,(Y \mid X) = \frac{P\,(X \mid Y) * P\,(Y)}{P\,(X)}$$



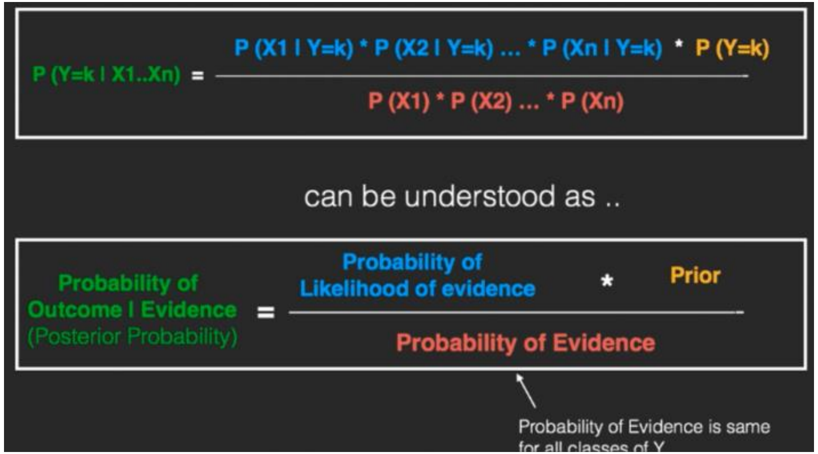
When there are multiple X variables, we simplify it by *assuming the X's are independent*, so the **Bayes** rule

$$P\,(Y{=}k \mid X) = \frac{P\,(X \mid Y{=}k) \;*\; P\,(Y{=}k)}{P\,(X)}$$

where, k is a class of Y

becomes, Naive **Bayes**

$$P\,(Y{=}k \mid X1..Xn) = \frac{P\,(X1 \mid Y{=}k) * P\,(X2 \mid Y{=}k) \ldots * P\,(Xn \mid Y{=}k) \;*\; P\,(Y{=}k)}{P\,(X1) * P\,(X2) \ldots * P\,(Xn)}$$



$$P\,(Y{=}k \mid X1..Xn) = \frac{P\,(X1 \mid Y{=}k) * P\,(X2 \mid Y{=}k) \ldots * P\,(Xn \mid Y{=}k) \;*\; P\,(Y{=}k)}{P\,(X1) * P\,(X2) \ldots * P\,(Xn)}$$

can be understood as ..

$$\begin{array}{c}\text{Probability of}\\ \text{Outcome | Evidence}\\ \text{(Posterior Probability)}\end{array} = \frac{\begin{array}{c}\text{Probability of}\\ \text{Likelihood of evidence}\end{array} \;*\; \text{Prior}}{\text{Probability of Evidence}}$$

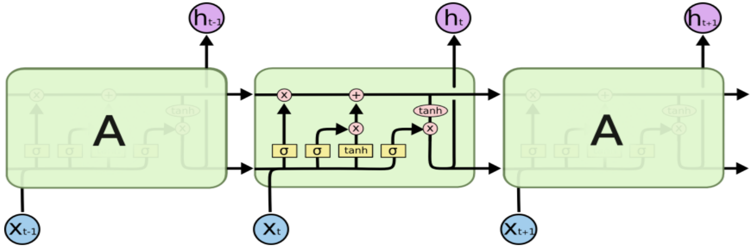
Probability of Evidence is same for all classes of Y

Model Implementation

LSTM

LSTM (Long short-term memory)

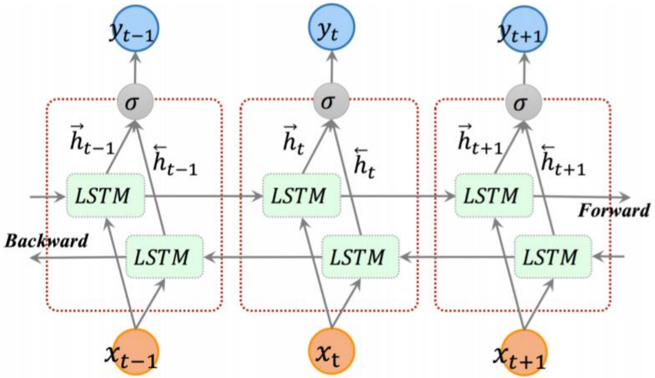
preserve information from **past**



The repeating module in an LSTM contains four interacting layers.

Bi-directional LSTM

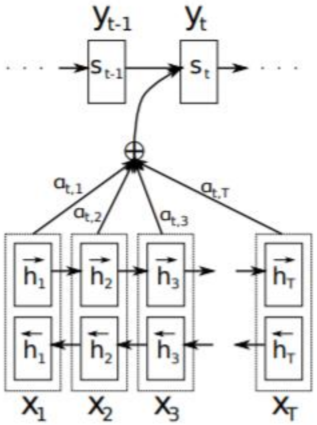
preserve information from **both past and future**



LSTM with Attention

Attention modules let all **intermediate states** be taken into consideration in the decoding process

– deal with **gradient explosion** and **vanishing**



Model Implementation

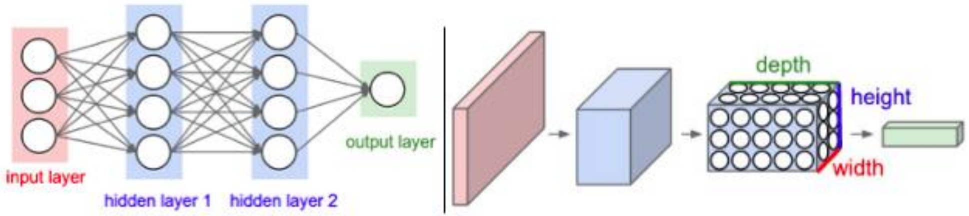
LSTM

Different models of LSTM and their accuracy on test set

LSTM model	Accuracy
General LSTM	58.76%
Bi-directional LSTM	55.87%
LSTM with attention	59.14%
LSTM with attention, supplied with related articles	57.13%

Model Implementation

CNN



Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

CNN-image classification

CNN-sentence classification

(validation accuracy of 60.67% after 5 epochs)

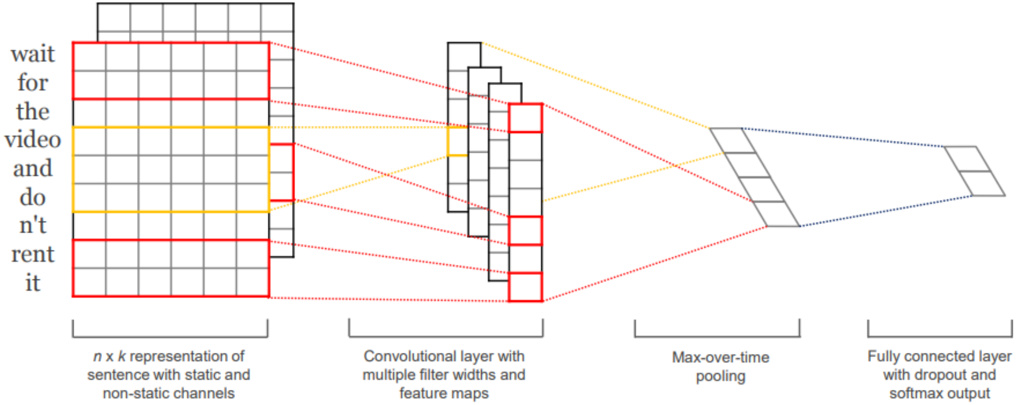

wait
for
the
video
and
do
n't
rent
it
n x *k* representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1: Model architecture with two channels for an example sentence.

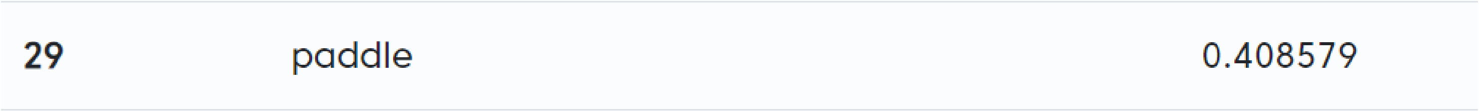
Findings and Analysis

Different models and their accuracy on test set

Model	Accuracy
Naïve Bayes	57.89%
LSTM with attention	59.14%
CNN	60.67%

Note: Our CNN model would not classify any news as "True news". This might be caused by the lack of "True news" samples in the provided dataset, making related information hard to survive when passing the convolutional layer.

After evaluation, we decided to use the LSTM with attention model in our final submission, and get score 0.408579 on the contest platform



29	paddle	0.408579

Thank you!