Sentiment Strength Prediction Using Auxiliary Features

Huijun Chen[1], Xin Li[1], Yanghui Rao[1*],
Haoran Xie[2], Fu Lee Wang[3], Tak-Lam Wong[2]

[1] School of Data and Computer Science, Sun Yat-sen University
[2] The Education University of Hong Kong
[3] Caritas Institute of Higher Education
* The Corresponding Author

Outline

- What is sentiment strength?
- Why predict sentiment strength?
- How to predict?
- Experiment

What is Sentiment Strength?

- The sentimental intensity of documents

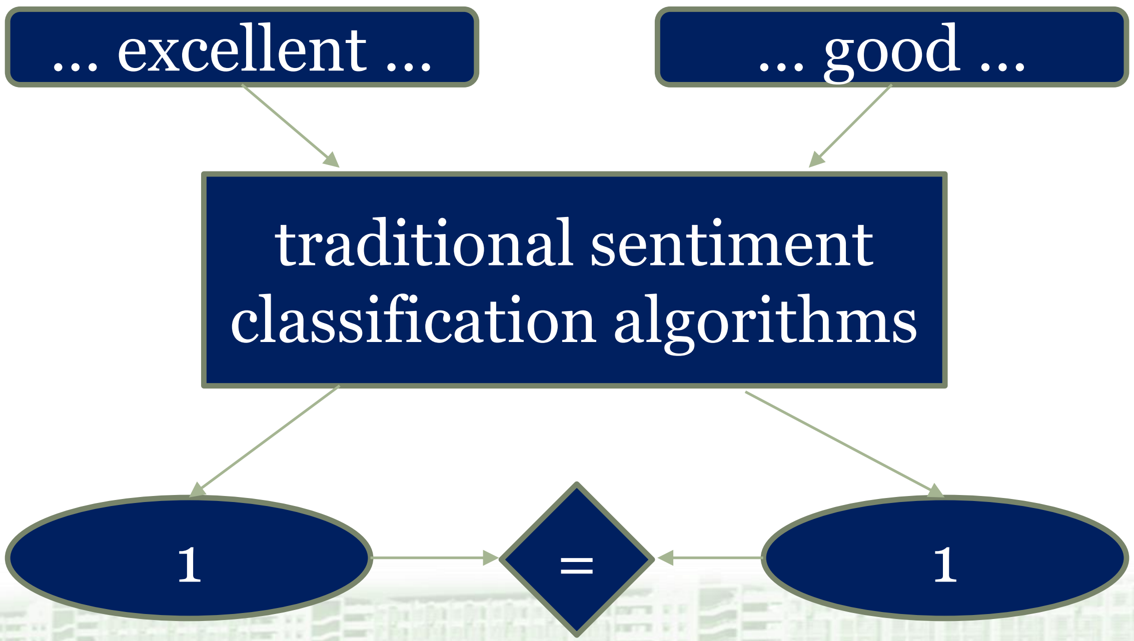
  -- fine-grained sentiment analysis
- Not only the categories of documents (e.g. positive, negative, neutral)

  -- coarse-grained sentiment classification

Why Predict Sentiment Strength?

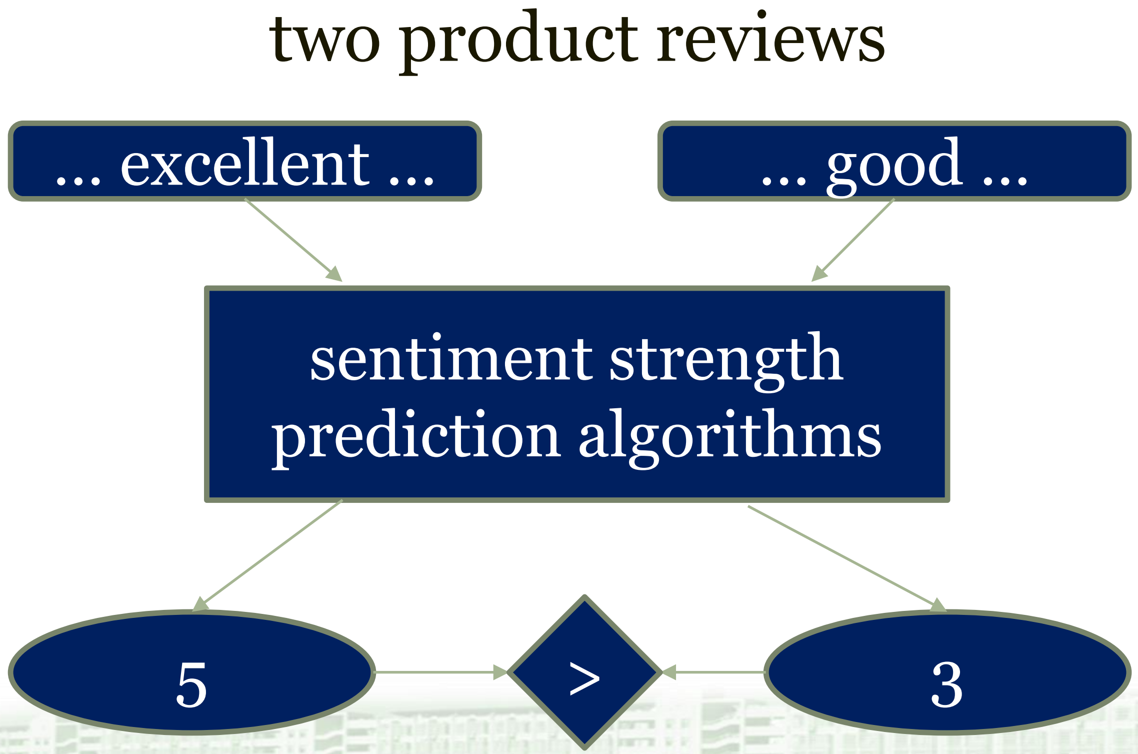
- An example

two product reviews



Why Predict Sentiment Strength?

- An example



Why Predict Sentiment Strength?

- Sentimental intensity is an important feature for cognitive computing and decision-making of potential customers.
- Coarse-grained sentiment classification

  -- captures only the most dominant sentiment

  -- ignores cognition indicators (e.g., the confidence of each sentiment)

Why Predict Sentiment Strength?

- Early works on sentiment strength prediction

  -- SentiStrength

  Problem:

  - Have focused mainly on exploiting lexical features
  - Heavily dependent on certain key words
  - perform limited since sentiments of words are sensitive to the topic domain or even aspect

How to Predict?

- Our solution

  Hybrid CNN (HCNN) model

  -- a neural network-based framework

  -- exploit the auxiliary features of sentiments from the corpus

  -- not rely on well-established lexicons

How to Predict?

- Definition

-- treat overall document sentiments as a list of real values ranging from 0 to 1

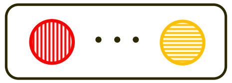
-- each value denotes the intensity of the corresponding sentiment

- Our goal

-- predict a strength vector that can reflect multiple sentimental orientations of a document

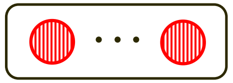
How to Predict?

- Convert words to vectors



-- each word vectors consist of two parts

- one-hot vectors (corpus-specific)

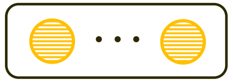


Problem: suffers from the curse of dimensionality

Solution: document frequency thresholding method

Problem: cannot record the sentimental meaning and relevance between different words

- pre-trained vectors (domain-independent)

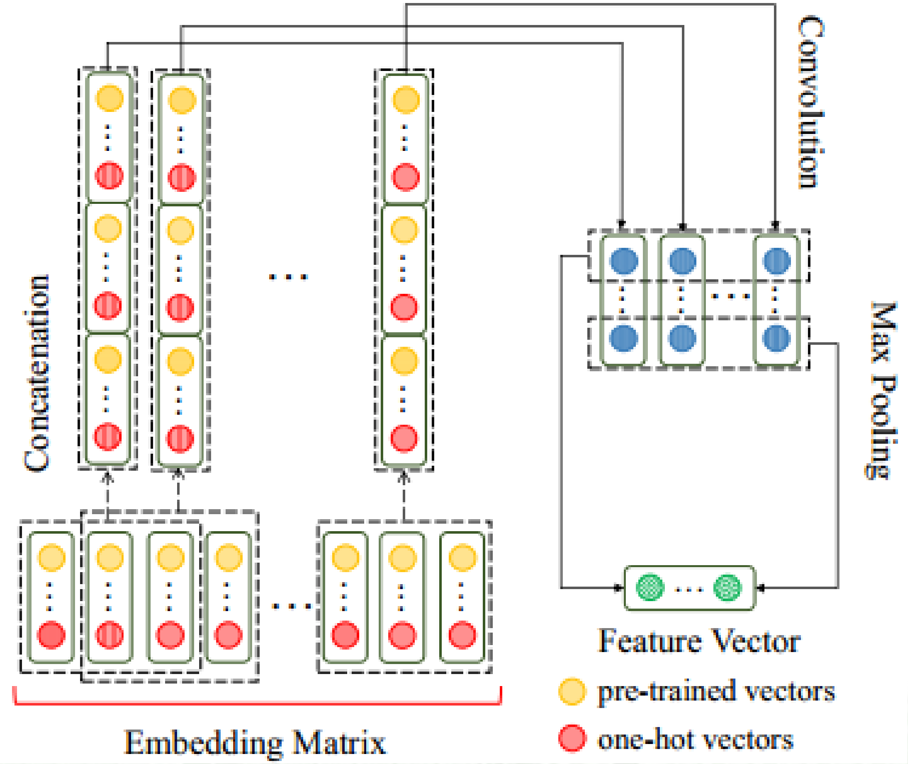


real-valued , trained from large-scale corpora

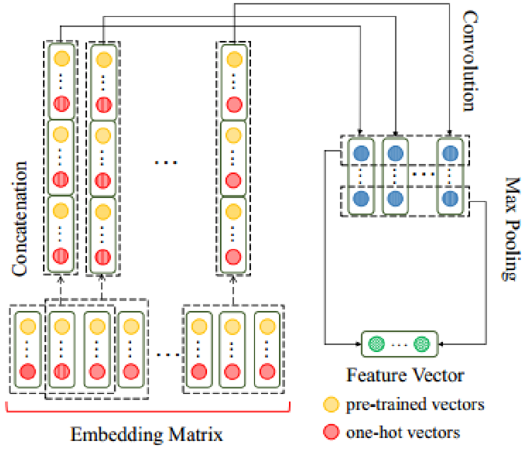
中山大學 SUN YAT-SEN UNIVERSITY

How to Predict?

- Convolution and pooling operation



How to Predict?



- Convolution and pooling operation

Input: embedding matrix, i.e., a sequence of vectors $\{v_1, \ldots, v_n\}$

Initialize: a convolution kernel W with fixed-sized window k and a bias vector b

Step 1: sliding the window and concatenate $(\oplus)$ the sequence of k embeddings, e.g. when the window centralizes in the i-th word: $z_i = v_i \oplus v_{i+1} \oplus \ldots \oplus v_{i+k}$

Step 2: apply the matrix-vector operation $(\odot)$, e.g. the i-th column of feature map C:

$$C[i] = W \odot z_i + b$$

Step 3: apply max-pooling operation, i.e., the j-th element of feature vector f is the maximum of the j-th row in C: $f[j] = max(C[:, j])$

How to Predict?

- Step 1

  -- to learn semantic representation
- Step 2

  -- to learn POS representation of POS-tags
- Step 3

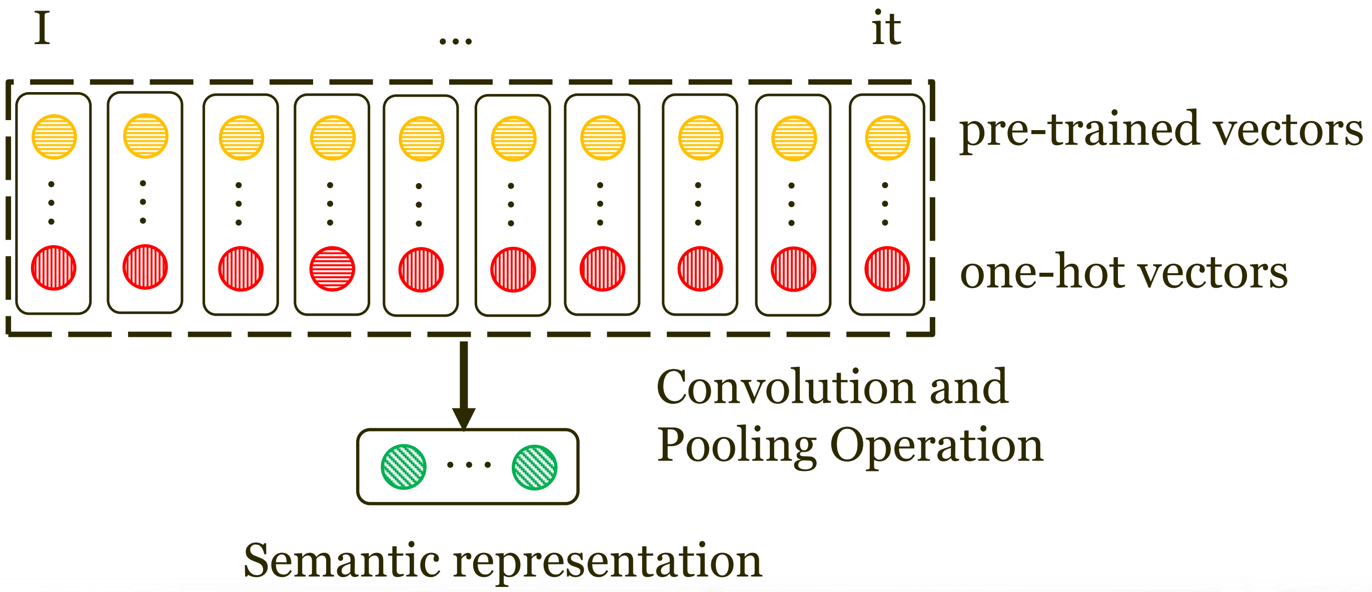
  -- to learn hybrid feature representation
- Step 4

  -- to predict the sentiment strength

How to Predict?

- Step 1: semantic representation

Example: a document “I bought this new English book because I like it”



How to Predict?

- Step 2: POS representation of POS-tags

  POS cluster

Group	POS tags
J	Adjectives, Adverbs
N	Nouns
V	Verbs
O	Other POS tags

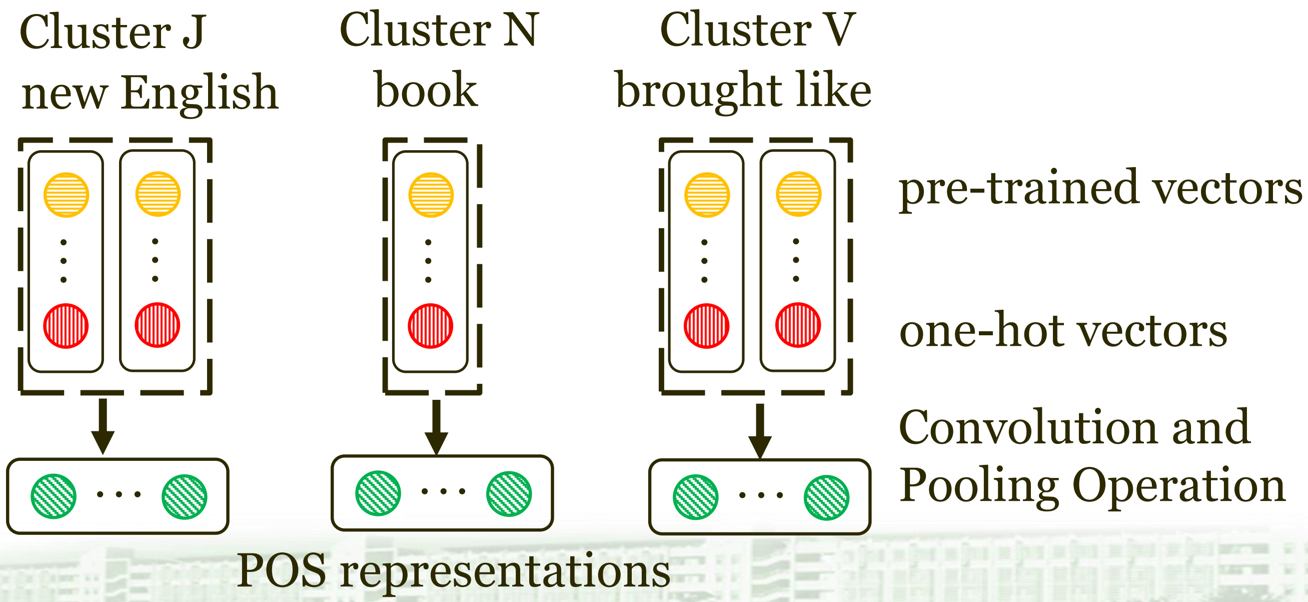
-- preserve only clusters “J”, “N”, and “V”

-- cluster “O” : background words, non-discriminative information, noise

How to Predict?

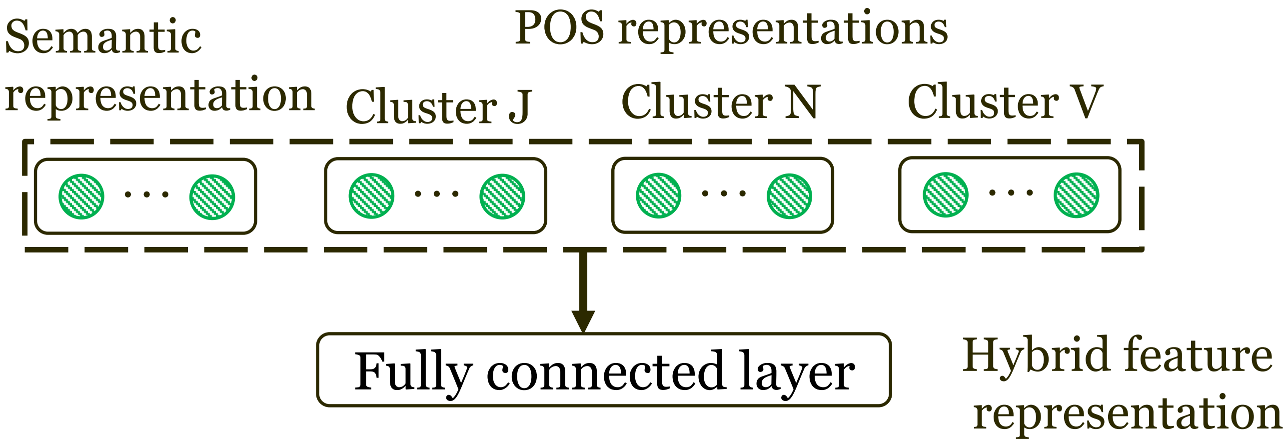
- Step 2: POS representation of POS-tags

Example: a document “I bought this new English book because I like it” with POS tag sequence {O,V, O, J, J, N, O, O, V, O}



How to Predict?

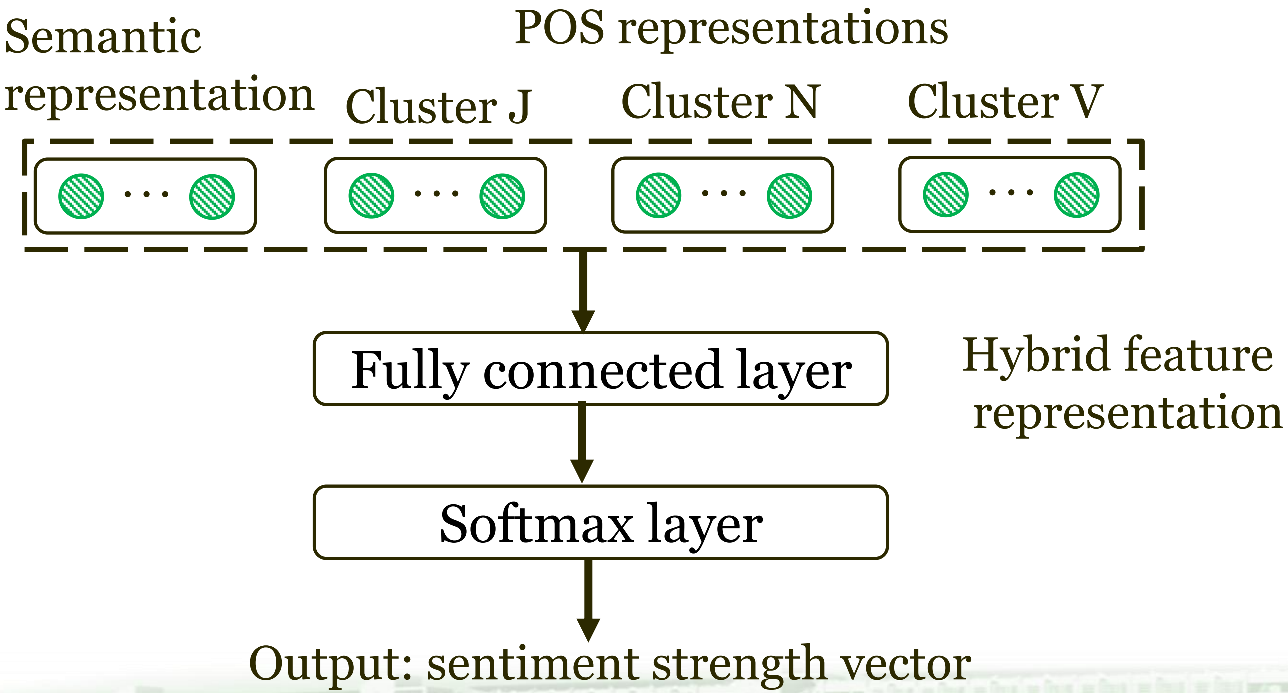
- Step 3: hybrid feature representation

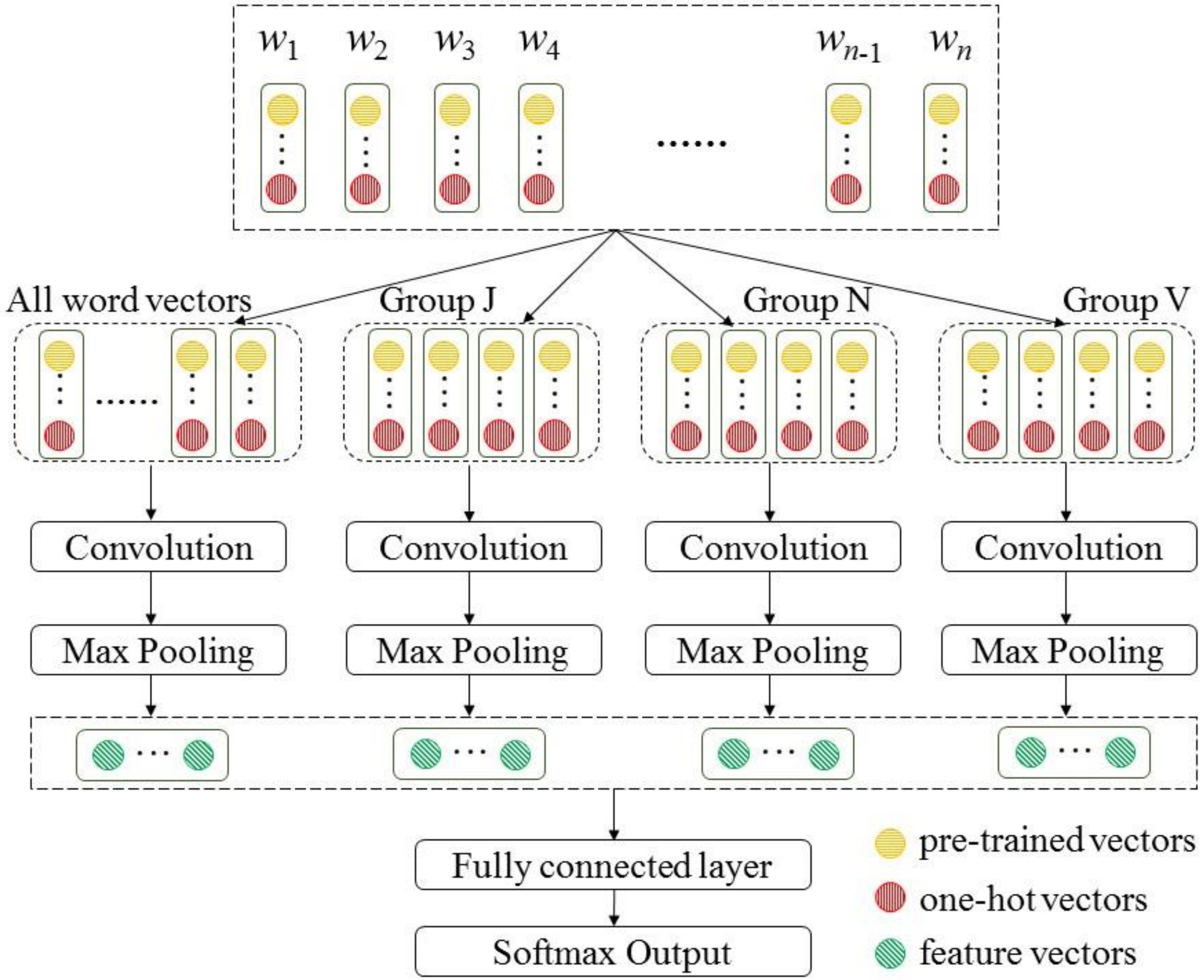


-- activation function: ReLU
-- regularization: dropout

How to Predict?

- Step 4: predict the sentiment strength





w_1 w_2 w_3 w_4 …… w_{n-1} w_n
All word vectors
Group J
Group N
Group V
Convolution
Max Pooling
Fully connected layer
Softmax Output
pre-trained vectors
one-hot vectors
feature vectors

How to Predict?

- Training

  -- Objective function

  Kullback-Leibler divergence (KL divergence)

  -- back-propagation

  -- stochastic gradient descent

  -- Adadelta optimizer

中山大學 SUN YAT-SEN UNIVERSITY

Experiment

- Dataset

  -- a real world short corpus, including six subsets which represented different types of social environments

Dataset	Type	# of documents	Mean Words
BBC	News	1000	64.76
Digg	Stories	1077	33.63
MySpace	Friends	1041	19.76
Runners World	Interest	1046	64.25
Twitter	Microblog	4242	16.81
YouTube	Video	3407	17.38

Experiment

- Baselines

-- Character to Sentence Convolutional Neural Network (CharSCNN)

Two convolutional layers are employed to extract features from character to sentence. The result of the second convolutional layer is then passed to two fully-connected layers to compute the sentiment score for each label.

-- Convolutional Neural Network (CNN)

A straightforward convolutional architecture that employs one convolutional layer with multiple kernels to learn sentence representation and add dropout to prevent over-fitting. Sentiment strength was the output distribution of the softmax layer.

Experiment

- Baselines

-- Long Short-Term Memory (LSTM)

LSTM takes the whole corpus as a single sequence, and the mean of the whole hidden states of all words is used as the feature for prediction. The output of the softmax layer is treated as the sentiment strength.

-- Convolutional Gated Recurrent Neural Network (ConvGRNN)

The model learns semantic representation hierarchically. Firstly, the model obtains sentence vectors by convolving pre-trained word embeddings. Secondly, the generated sentence vectors are fed into Gated RNN to produce the representation vector of each document.

Experiment

- Baselines

  -- Supervised SentiStrength (Ssth)

  Ssth is a method specifically designed for sentiment strength detection over our employed corpus. It is a lexicon-based classifier that uses linguistic information and rules to predict sentiment strengths in short informal English text, and the supervised version tends to be more accurate than unsupervised SentiStrength and many other machine learning methods.

Experiment

- Parameters

  -- Word vectors

  GloVe, random

  -- POS annotation

  openNLP POS tagger

  -- Initialize

  uniform distribution $(-\sqrt{6/(r+c)}, \sqrt{6/(r+c)})$, r and c are numbers of rows and columns in the parameter matrix

Table 4: Parameter setting of HCNN.

Parameter	Value
dim_{pre}	200
$(win_{K_w}, win_{K_{pos}})$	(1, 1)
$(dim_{K_w}, dim_{K_{pos}})$	(80, 2)
dim_{hyb}	100

Experiment

- Evaluation Metrics

  -- fine-grained metrics

  root mean square error (RMSE)

  Pearson’s correlation coefficient (Corr)

  -- coarse grained metric

  Accuracy (Acc)

- Ablation Experiment

  HCNN-POS, HCNN-one-hot, HCNN-GloVe

Experiment

- Experiment Modes

  -- single source

  randomly selected 60% as training samples, 20% as validation samples, and the remaining 20% for testing

  -- cross sources

  A subset as training set

  B subset 20% for validation, 80% for testing

Experiment

- Single sources testing

Model	BBC			Digg			MySpace		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1260**	**0.4462**	**0.9150**	**0.1297**	**0.6000**	0.8519	**0.1024**	**0.6585**	**0.9278**
HCNN - POS	0.1287	0.3925	**0.9150**	0.1410	0.4928	0.8472	0.1141	0.5532	0.9183
HCNN - one-hot	0.1340	0.3121	**0.9150**	0.1370	0.5200	**0.8560**	0.1141	0.5185	0.8942
HCNN - *GloVe*	0.1342	0.2834	**0.9150**	0.1572	0.2798	0.7963	0.1203	0.4443	0.8942
CharSCNN	0.1335	0.2860	0.9100	0.1505	0.3329	0.8279	0.1233	0.4291	0.8990
CNN	0.2061	0.3172	0.8900	0.2139	0.4533	0.8380	0.3098	0.5375	0.9038
LSTM	0.2943	0.2989	0.8800	0.2743	0.4501	0.8519	0.2993	0.5015	0.9038
ConvGRNN	0.2002	0.1788	0.9100	0.1548	0.4260	0.8287	0.2745	0.5514	0.9187
Ssth	0.1538	0.4430	0.7600	0.1641	0.4174	0.7070	0.1319	0.4762	0.8990

Model	Runners World			Twitter			YouTube		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1133**	**0.4942**	**0.8857**	**0.1177**	**0.5625**	**0.9058**	**0.1360**	**0.7264**	**0.8942**
HCNN - POS	0.1210	0.4331	0.8667	0.1189	0.5523	**0.9058**	0.1433	0.7105	0.8750
HCNN - one-hot	0.1195	0.3824	0.8476	0.1198	0.5348	0.8939	0.1370	0.7235	**0.8942**
HCNN - *GloVe*	0.1285	0.2630	0.8286	0.1310	0.4039	0.8610	0.1370	0.5562	0.8061
CharSCNN	0.1263	0.1433	0.8278	0.1309	0.4037	0.8632	0.1586	0.6280	0.8649
CNN	0.2967	0.4325	**0.8857**	0.3438	0.3948	0.8763	0.2432	0.6704	0.8899
LSTM	0.3458	0.1955	0.8000	0.3112	0.3887	0.8716	0.2273	0.6866	0.8767
ConvGRNN	0.1955	0.0178	0.8333	0.2385	0.5202	0.8928	0.2582	0.6204	0.8856
Ssth	0.1543	0.2956	0.8373	0.1454	0.4422	0.8868	0.1613	0.6024	0.8473

Experiment

Model	BBC			Digg			MySpace		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1260**	**0.4462**	**0.9150**	**0.1297**	**0.6000**	0.8519	**0.1024**	**0.6585**	**0.9278**
HCNN - POS	0.1287	0.3925	**0.9150**	0.1410	0.4928	0.8472	0.1141	0.5532	0.9183
HCNN - one-hot	0.1340	0.3121	**0.9150**	0.1370	0.5200	**0.8560**	0.1141	0.5185	0.8942
HCNN - *GloVe*	0.1342	0.2834	**0.9150**	0.1572	0.2798	0.7963	0.1203	0.4443	0.8942

Model	Runners World			Twitter			YouTube		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1133**	**0.4942**	**0.8857**	**0.1177**	**0.5625**	**0.9058**	**0.1360**	**0.7264**	**0.8942**
HCNN - POS	0.1210	0.4331	0.8667	0.1189	0.5523	**0.9058**	0.1433	0.7105	0.8750
HCNN - one-hot	0.1195	0.3824	0.8476	0.1198	0.5348	0.8939	0.1370	0.7235	**0.8942**
HCNN - *GloVe*	0.1285	0.2630	0.8286	0.1310	0.4039	0.8610	0.1370	0.5562	0.8061

1) all useful to boost performance, especially for the fine-grained metrics Corr and RMSE.
2) did not affect Acc much
-- the basic convolutional architecture was effective to capture enough features for coarse-grained polarity detection
3) POS-level features: limited when each document's words are extremely sparse (e.g., Twitter).
-- very short text cannot provide enough POS information

Experiment

Model	BBC			Digg			MySpace		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1260**	**0.4462**	**0.9150**	**0.1297**	**0.6000**	0.8519	**0.1024**	**0.6585**	**0.9278**
CharSCNN	0.1335	0.2860	0.9100	0.1505	0.3329	0.8279	0.1233	0.4291	0.8990
CNN	0.2061	0.3172	0.8900	0.2139	0.4533	0.8380	0.3098	0.5375	0.9038
LSTM	0.2943	0.2989	0.8800	0.2743	0.4501	0.8519	0.2993	0.5015	0.9038
ConvGRNN	0.2002	0.1788	0.9100	0.1548	0.4260	0.8287	0.2745	0.5514	0.9187
Ssth	0.1538	0.4430	0.7600	0.1641	0.4174	0.7070	0.1319	0.4762	0.8990

Model	Runners World			Twitter			YouTube		
	RMSE	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*	*RMSE*	*Corr*	*Acc*
HCNN	**0.1133**	**0.4942**	**0.8857**	**0.1177**	**0.5625**	**0.9058**	**0.1360**	**0.7264**	**0.8942**
CharSCNN	0.1263	0.1433	0.8278	0.1309	0.4037	0.8632	0.1586	0.6280	0.8649
CNN	0.2967	0.4325	**0.8857**	0.3438	0.3948	0.8763	0.2432	0.6704	0.8899
LSTM	0.3458	0.1955	0.8000	0.3112	0.3887	0.8716	0.2273	0.6866	0.8767
ConvGRNN	0.1955	0.0178	0.8333	0.2385	0.5202	0.8928	0.2582	0.6204	0.8856
Ssth	0.1543	0.2956	0.8373	0.1454	0.4422	0.8868	0.1613	0.6024	0.8473

1) RMSE and Corr: outperformed by a large margin

-- The regression-oriented objective function (i.e., KL divergence) was better than classification-oriented objective functions in sentiment strength prediction.

2) Acc: indistinctive

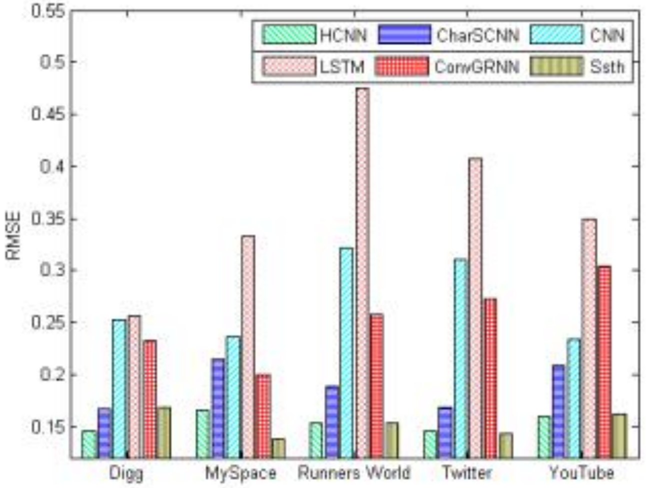
-- Because Acc did not take sentimental distributions into account.

Experiment

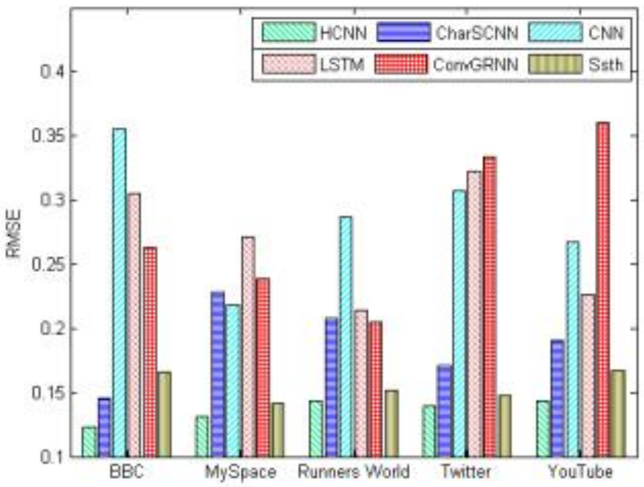
- Cross sources testing

-- RMSE

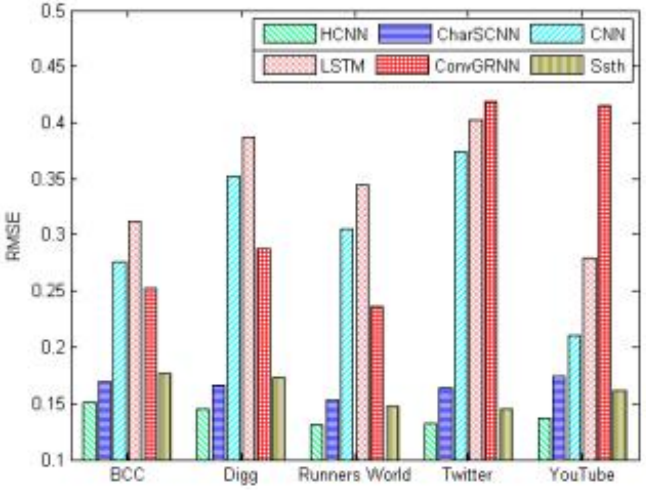
中山大學
SUN YAT-SEN UNIVERSITY



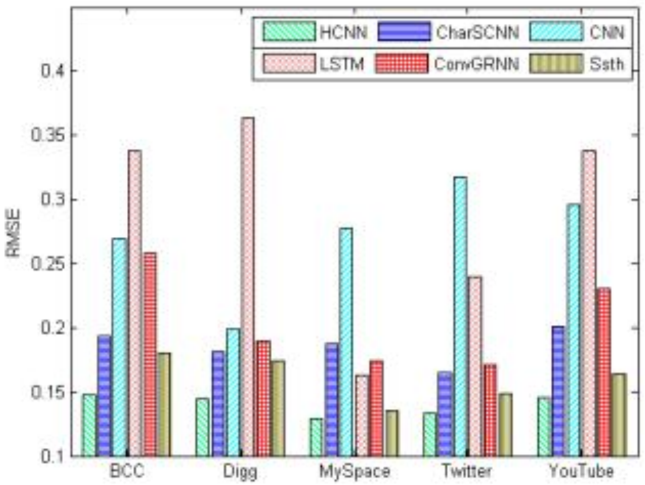
(a) BBC vs Others



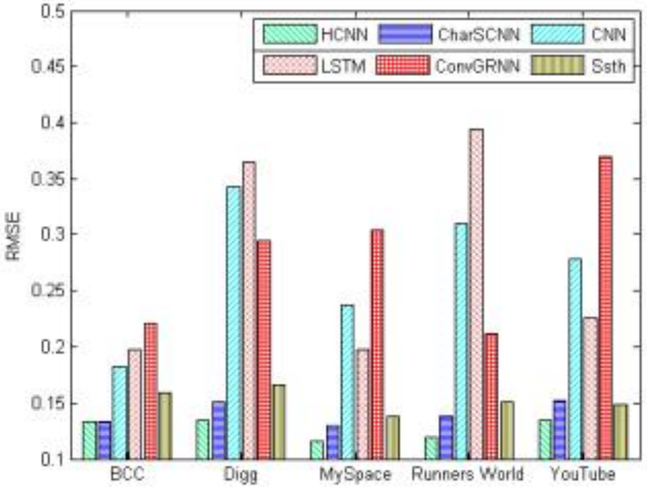
(b) Digg vs Others



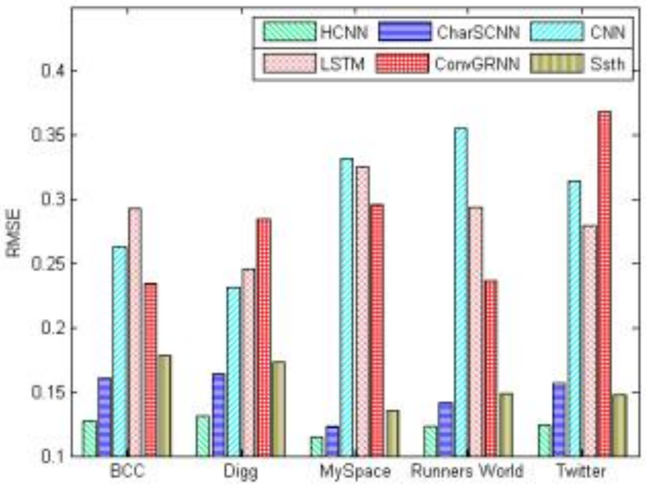
(c) MySpace vs Others



(d) Runners World vs Others



(e) Twitter vs Others



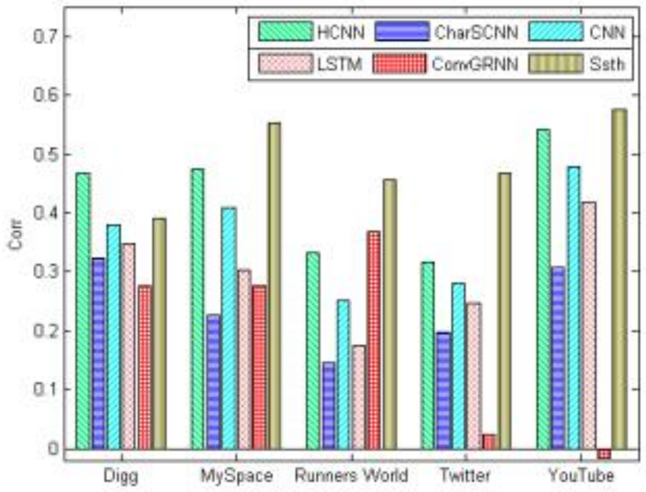
(f) YouTube vs Others

Experiment

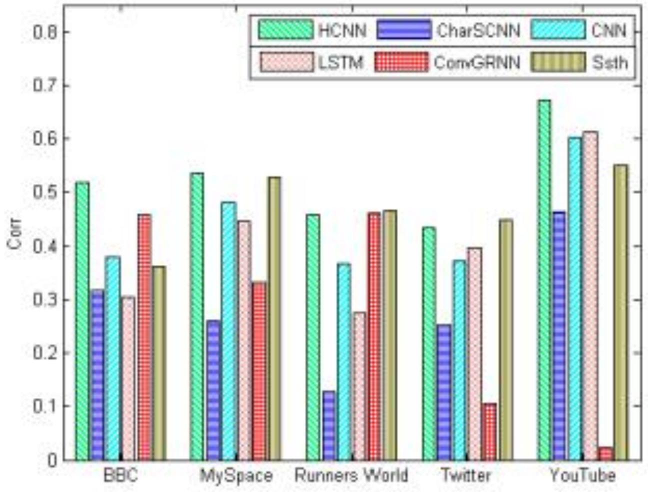
- Cross sources testing

  -- RMSE

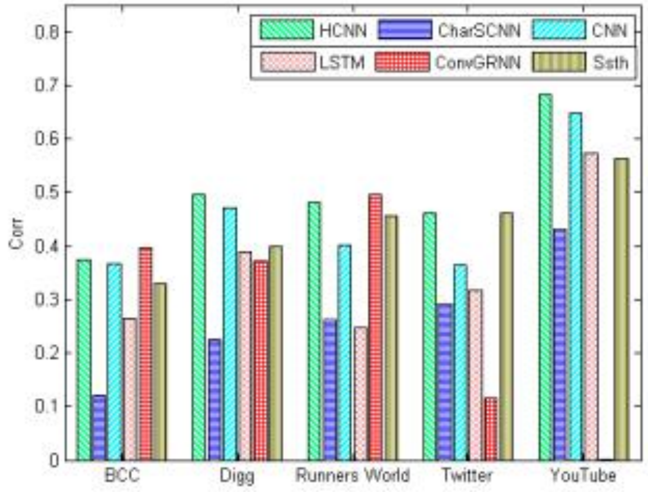
  -- Corr



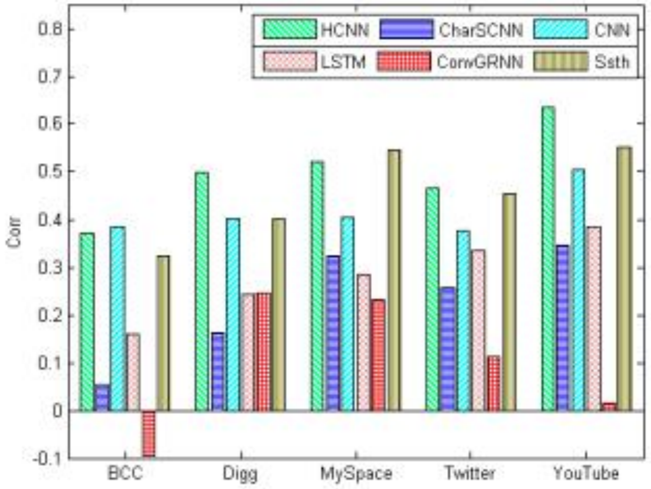
(a) BBC vs Others



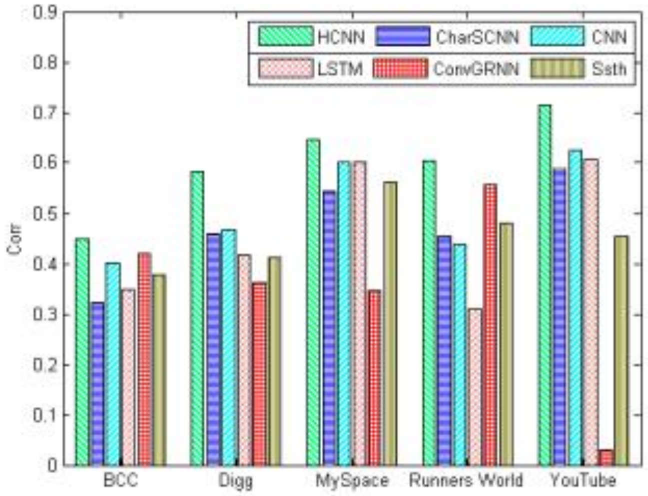
(b) Digg vs Others



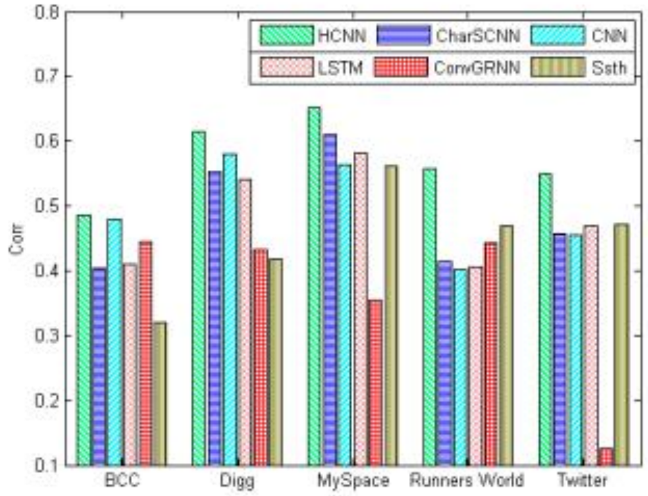
(c) MySpace vs Others



(d) Runners World vs Others



(e) Twitter vs Others



(f) YouTube vs Others

Experiment

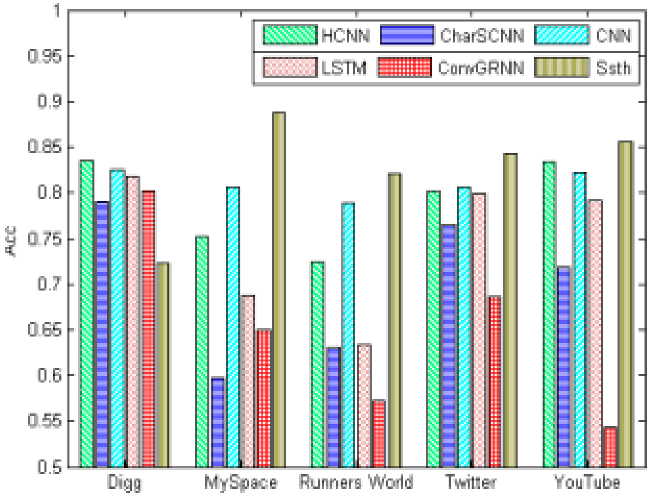
- Cross sources testing

  -- RMSE

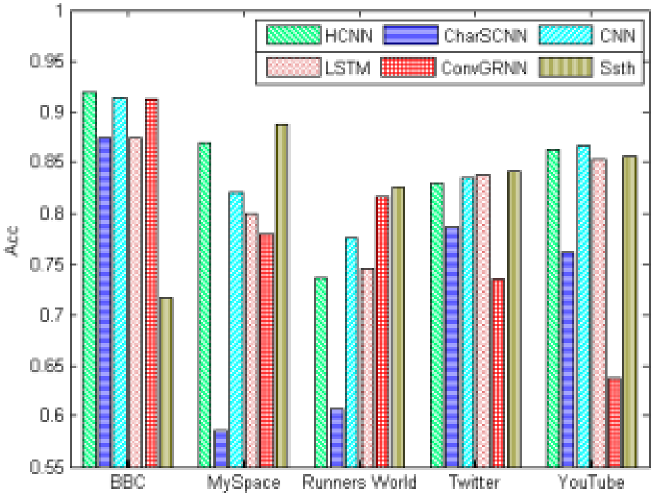
  -- Corr

  -- Acc

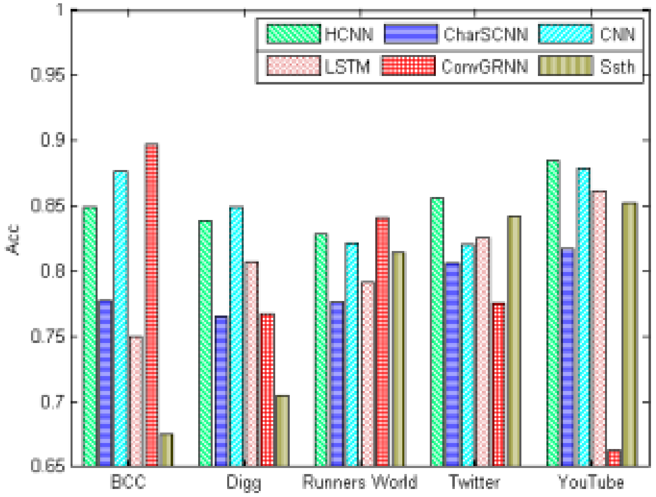
中山大學
SUN YAT-SEN UNIVERSITY



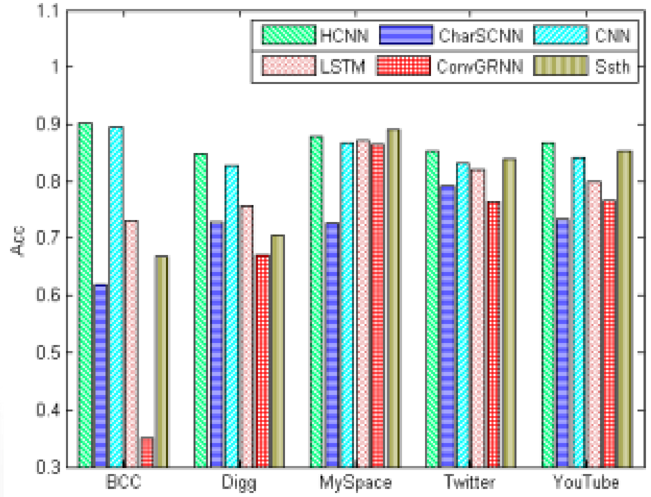
(a) BBC vs Others



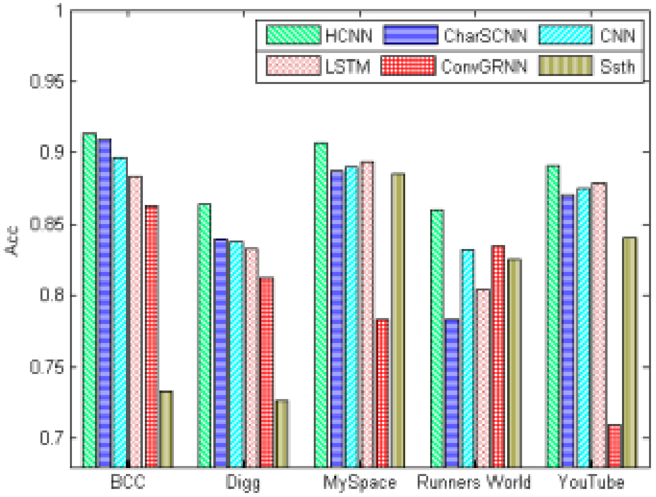
(b) Digg vs Others



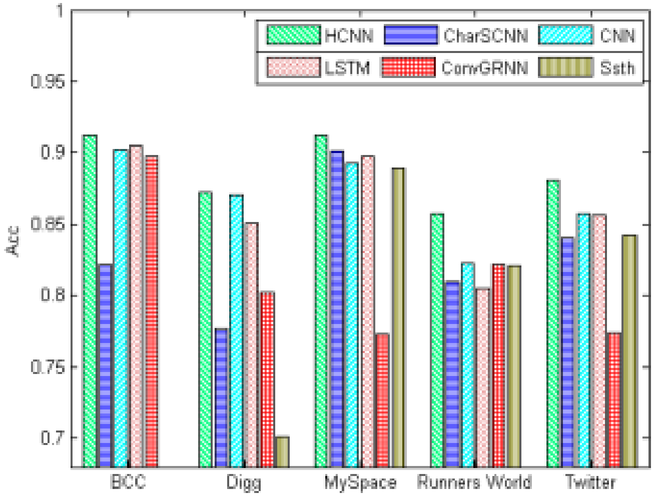
(c) MySpace vs Others



(d) Runners World vs Others



(e) Twitter vs Others



(f) YouTube vs Others

Experiment

- Cross sources testing

  -- RMSE

  -- Corr

  -- Acc

The proposed method perform competitively or better than baselines.

Experiment

- Cross sources testing

-- RMSE

-- Corr

-- Acc

-- ablation experiment

The results indicated that POS-features and pre-trained embeddings had a larger positive impact on HCNN than one-hot vectors.

For example, the prediction error (i.e., RMSE) of HCNN reduced by 15.7%, 2.9%, and 2.7% on average when compared with HCNN - GloVe, HCNN – POS, and HCNN - one-hot, respectively.

Experiment

- Statistical test

  -- F-test

  to test the underlying assumption of homoscedasticity
  (i.e., the homogeneity of variance)

中山大學
SUN YAT-SEN UNIVERSITY

Experiment

Table 6: The p values of statistical tests on the HCNN and baselines over *RMSE*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.376	0.029	0.000	0.000
CNN	0.002	0.001	0.000	0.000
LSTM	0.011	0.000	0.000	0.000
ConvGRNN	0.006	0.001	0.000	0.000
Ssth	0.469	0.001	0.303	0.000

Table 7: The p values of statistical tests on the HCNN and baselines over *Corr*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.174	0.011	0.034	0.000
CNN	0.358	0.057	0.399	0.003
LSTM	0.150	0.038	0.120	0.000
ConvGRNN	0.047	0.053	0.001	0.000
Ssth	0.461	0.022	0.058	0.007

Table 8: The p values of statistical tests on the HCNN and baselines over *Acc*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.290	0.055	0.001	0.000
CNN	0.369	0.143	0.044	0.279
LSTM	0.270	0.050	0.099	0.005
ConvGRNN	0.219	0.177	0.000	0.000
Ssth	0.020	0.032	0.013	0.001

Experiment

- Statistical test

  -- F-test

  to test the underlying assumption of homoscedasticity
  (i.e., the homogeneity of variance)

  -- t-test

  to test the underlying assumption that the difference in performance between paired models had a mean value of zero.

  (We used the conventional significance level (i.e., p-value) of 0.05.)

Experiment

Table 6: The p values of statistical tests on the HCNN and baselines over *RMSE*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.376	0.029	0.000	0.000
CNN	0.002	0.001	0.000	0.000
LSTM	0.011	0.000	0.000	0.000
ConvGRNN	0.006	0.001	0.000	0.000
Ssth	0.469	0.001	0.303	0.000

Table 7: The p values of statistical tests on the HCNN and baselines over *Corr*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.174	0.011	0.034	0.000
CNN	0.358	0.057	0.399	0.003
LSTM	0.150	0.038	0.120	0.000
ConvGRNN	0.047	0.053	0.001	0.000
Ssth	0.461	0.022	0.058	0.007

Table 8: The p values of statistical tests on the HCNN and baselines over *Acc*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.290	0.055	0.001	0.000
CNN	0.369	0.143	0.044	0.279
LSTM	0.270	0.050	0.099	0.005
ConvGRNN	0.219	0.177	0.000	0.000
Ssth	0.020	0.032	0.013	0.001

中山大學
SUN YAT-SEN UNIVERSITY

Experiment

Table 6: The p values of statistical tests on the HCNN and baselines over *RMSE*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.376	0.029	0.000	0.000
CNN	0.002	0.001	0.000	0.000
LSTM	0.011	0.000	0.000	0.000
ConvGRNN	0.006	0.001	0.000	0.000
Ssth	0.469	0.001	0.303	0.000

These results validated the effectiveness of our method on sentiment strength prediction tasks, especially when the training set and the testing set are from different sources.

Table 7: The p values of statistical tests on the HCNN and baselines over *Corr*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.174	0.011	0.034	0.000
CNN	0.358	0.057	0.399	0.003
LSTM	0.150	0.038	0.120	0.000
ConvGRNN	0.047	0.053	0.001	0.000
Ssth	0.461	0.022	0.058	0.007

Table 8: The p values of statistical tests on the HCNN and baselines over *Acc*.

Models	single-source		cross-sources	
	F-test	t-test	F-test	t-test
CharSCNN	0.290	0.055	0.001	0.000
CNN	0.369	0.143	0.044	0.279
LSTM	0.270	0.050	0.099	0.005
ConvGRNN	0.219	0.177	0.000	0.000
Ssth	0.020	0.032	0.013	0.001

Conclusion

- Our model introduced one-hot vectors to capture corpus-specific information and jointly learned hybrid features at semantic and syntactic levels for enhancing model robustness and adaptiveness.
- Experimental results validated the effectiveness of the proposed sentiment strength prediction method.
- Affect indicates positive or negative sentiment, while cognition includes certainty and tentative. Our research can help bridge the cognitive and affective gaps between users and documents.

Thanks

April 6, 2017