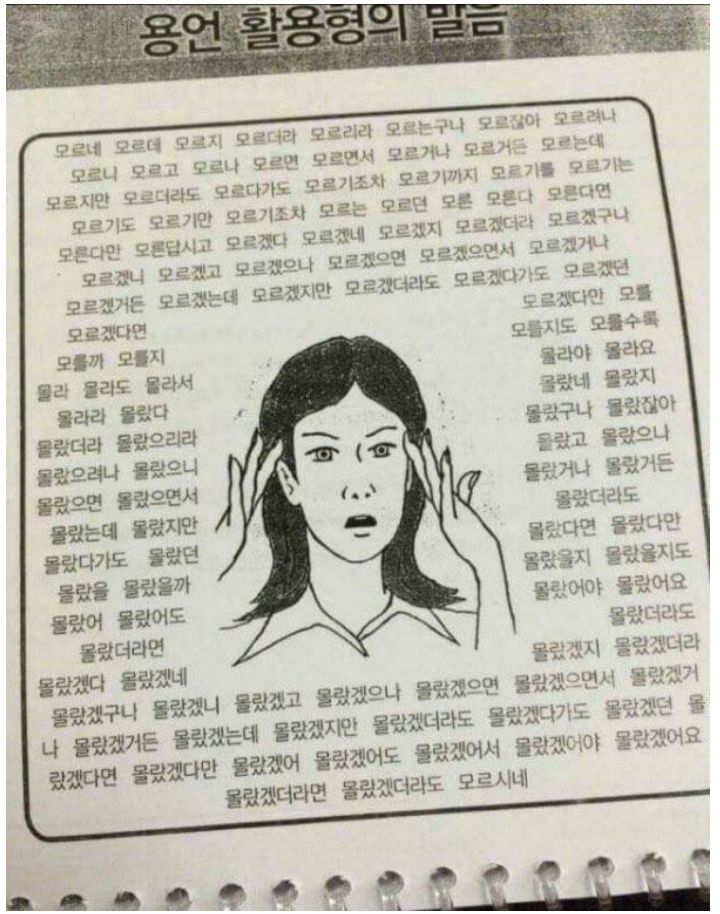
Character-level Convolutional Networks for Text Classification

dukim@dongguk.edu
Daeung Kim

Korean language is…

Morphemes + Suffix

모르	네
모르	데
모르	지
모르	더라
모르	리라
모르	는구나
모르	잖아
모르	라나
모르	니
모르	고
모르	나
모르	면 ….



http://news.chosun.com/site/data/html_dir/2017/06/25/2017062500228.html

Korean language is...

Journal of the Korean Institute of Industrial Engineers
Published Online, pp. 00-00, June 2018.
ISSN 1225-0988 | EISSN 2234-6457

<Original Research Paper>

단어와 자소 기반 합성곱 신경망을 이용한 문서 분류

모경현 · 박재선 · 장명준 · 강필성†

고려대학교 산업경영공학부

Text Classification based on Convolutional Neural Network with Word and Character Level

Kyounghyun Mo · Jaesun Park · Myeongjun Jang · Pilsung Kang

School of Industrial Management Engineering, Korea University

Index

Introduction

- In Text Classification, we need to select…
  - The best features to represent the documents.
    - Discrete Embedding
      - Statistics of some ordered words combinations (ex. n-grams)
      - TFIDF
    - Distributed Embedding
      - Word2vec
  - The best possible Machine Learning classifiers
    - Traditional Methods : ex) Multinomial Logistic regression
    - Deep learning Methods : ex) CNN, LSTM

Introduction

- In this paper, they selected…
  - The best features to represent the documents.
    - Discrete Embedding (One hot encoding)
      - Statistics of characters
  - The best possible Machine Learning classifiers
    - Deep learning Methods : CNN

Introduction

- Related works
  - Character-level n-grams with linear classifiers
    - I. Kanaris, et al, 2007
  - Incorporating character-level features to CNN
    - Use words as a basis
    - Character-level feature extracted at word or word n-gram
      - C. D. Santos and B. Zadrozny, 2014
      - Y. Shen, et al. 2014

Character-level CNN for text classification

- Character Quantization
- Model Design
- Data Augmentation

Character-level CNN for text classification

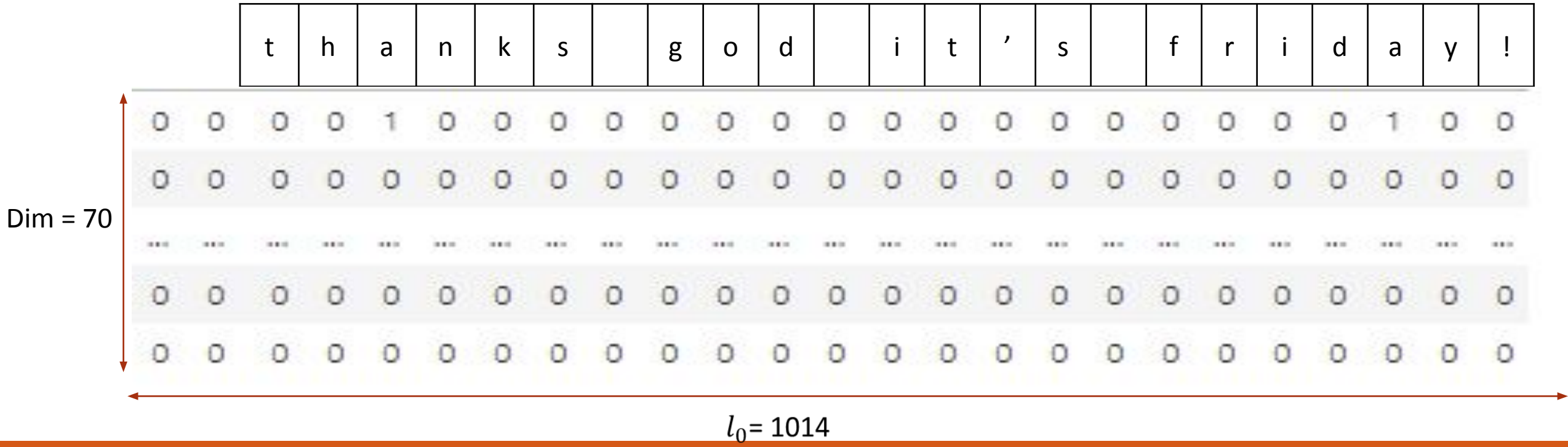
- Character Quantization
  - Embedding dim = 70, quantize the each character using one-hot encoding.
  - Any characters that are not in the alphabet including blank characters are quantized as all-zero vectors.

```
abcdefghijklmnopqrstuvwxyz0123456789
-,;.!?:'''/\|_@#$%ˆ&*˜‘+-=<>()[]{}
```

  - Fixed length : l_0 = 1014
  - The character quantization order is backward.

Character-level CNN for text classification

- Example : Thanks God It’s Friday!
  - Split it to characters and turn it into list:



Character-level CNN for text classification

- Model Design

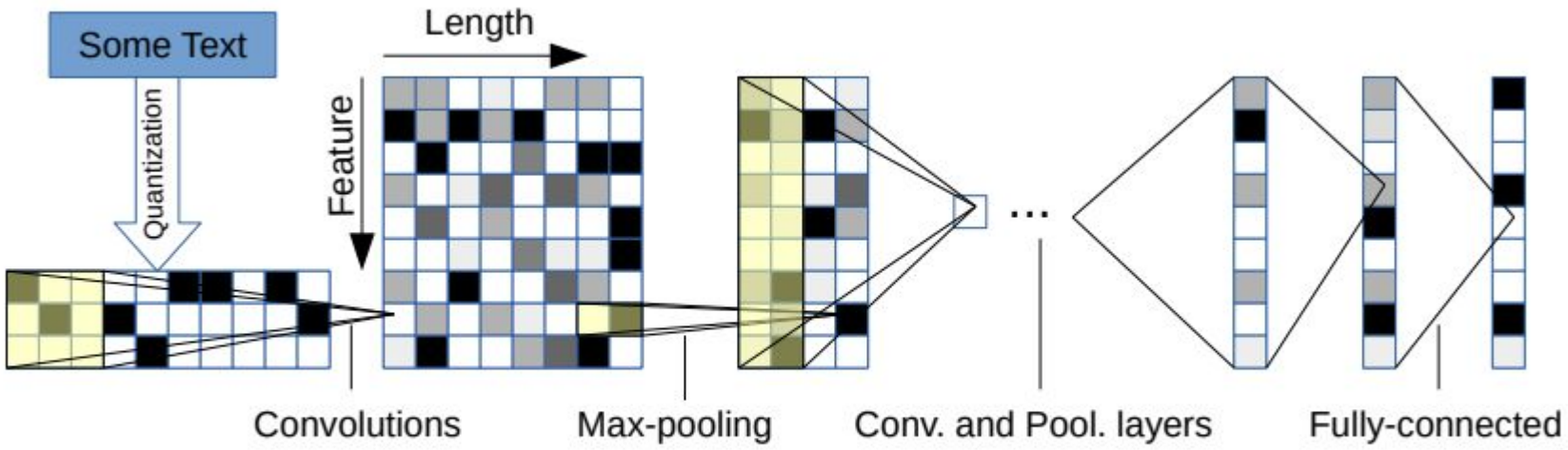


Figure 1: Illustration of our model

Character-level CNN for text classification

- Model Design
  - Two CNN(Large and Small)
  - 6 CNN layers and 3 fully-connected layers
  - Dropout (after 7, 8 layer)
  - Weight initialize using Gaussian Dist.
    - The large model : N(0, 0.02)
    - The small model : N(0, 0.05)

Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

Layer	Output Units Large	Output Units Small
7	2048	1024
8	2048	1024
9	Depends on the problem	

Character-level CNN for text classification

- Data Augmentation
  - Augmentation using Thesaurus
  - Augmentation using Synonyms obtained from thesaurus

- Large-scale Datasets and Results

Dataset	Classes	Train Samples	Test Samples	Epoch Size
AG's News	4	120,000	7,600	5,000
Sogou News	5	450,000	60,000	5,000
DBPedia	14	560,000	70,000	5,000
Yelp Review Polarity	2	560,000	38,000	5,000
Yelp Review Full	5	650,000	50,000	5,000
Yahoo! Answers	10	1,400,000	60,000	10,000
Amazon Review Full	5	3,000,000	650,000	30,000
Amazon Review Polarity	2	3,600,000	400,000	30,000

Comparison Models

- Traditional Methods :
  - Bag-of-words & its TFIDF
  - Bag-of-n-grams & its TFIDF
  - Bag-of-means on word embedding(300dim)
- Deep learning Methods :
  - Word-Based CNN (Freeze, Fine-tune)
  - Word-Based LSTM(Freeze, Fine-tune)
- Choice of Alphabet
  - Whether to distinguish between upper-case and lower-case letters.

Experiments & Results

Model	AG	Sogou	DBP.	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.
BoW	11.19	7.15	3.39	7.76	42.01	31.11	45.36	9.60
BoW TFIDF	10.36	6.55	2.63	6.34	40.14	28.96	44.74	9.00
ngrams	7.96	2.92	1.37	**4.36**	43.74	31.53	45.73	7.98
ngrams TFIDF	**7.64**	**2.81**	**1.31**	4.56	45.20	31.49	47.56	8.46
Bag-of-means	**16.91**	**10.79**	**9.55**	**12.67**	**47.46**	**39.45**	**55.87**	**18.39**
LSTM	13.94	4.82	1.45	5.26	41.83	29.16	40.57	6.10
Lg. w2v Conv.	9.92	4.39	1.42	4.60	40.16	31.97	44.40	5.88
Sm. w2v Conv.	11.35	4.54	1.71	5.56	42.13	31.50	42.59	6.00
Lg. w2v Conv. Th.	9.91	-	1.37	4.63	39.58	31.23	43.75	5.80
Sm. w2v Conv. Th.	10.88	-	1.53	5.36	41.09	29.86	42.50	5.63
Lg. Lk. Conv.	8.55	4.95	1.72	4.89	40.52	29.06	45.95	5.84
Sm. Lk. Conv.	10.87	4.93	1.85	5.54	41.41	30.02	43.66	5.85
Lg. Lk. Conv. Th.	8.93	-	1.58	5.03	40.52	28.84	42.39	5.52
Sm. Lk. Conv. Th.	9.12	-	1.77	5.37	41.17	28.92	43.19	5.51
Lg. Full Conv.	9.85	8.80	1.66	5.25	38.40	29.90	40.89	5.78
Sm. Full Conv.	11.59	8.95	1.89	5.67	38.82	30.01	40.88	5.78
Lg. Full Conv. Th.	9.51	-	1.55	4.88	38.04	29.58	40.54	5.51
Sm. Full Conv. Th.	10.89	-	1.69	5.42	**37.95**	29.90	40.53	5.66
Lg. Conv.	12.82	4.88	1.73	5.89	39.62	29.55	41.31	5.51
Sm. Conv.	15.65	8.65	1.98	6.53	40.84	29.84	40.53	5.50
Lg. Conv. Th.	13.39	-	1.60	5.82	39.30	**28.80**	40.45	**4.93**
Sm. Conv. Th.	14.80	-	1.85	6.49	40.16	29.84	**40.43**	5.67

Experiments & Results

$$\frac{error_{Comparison} - error_{Char-CNN}}{error_{Comparison}}$$

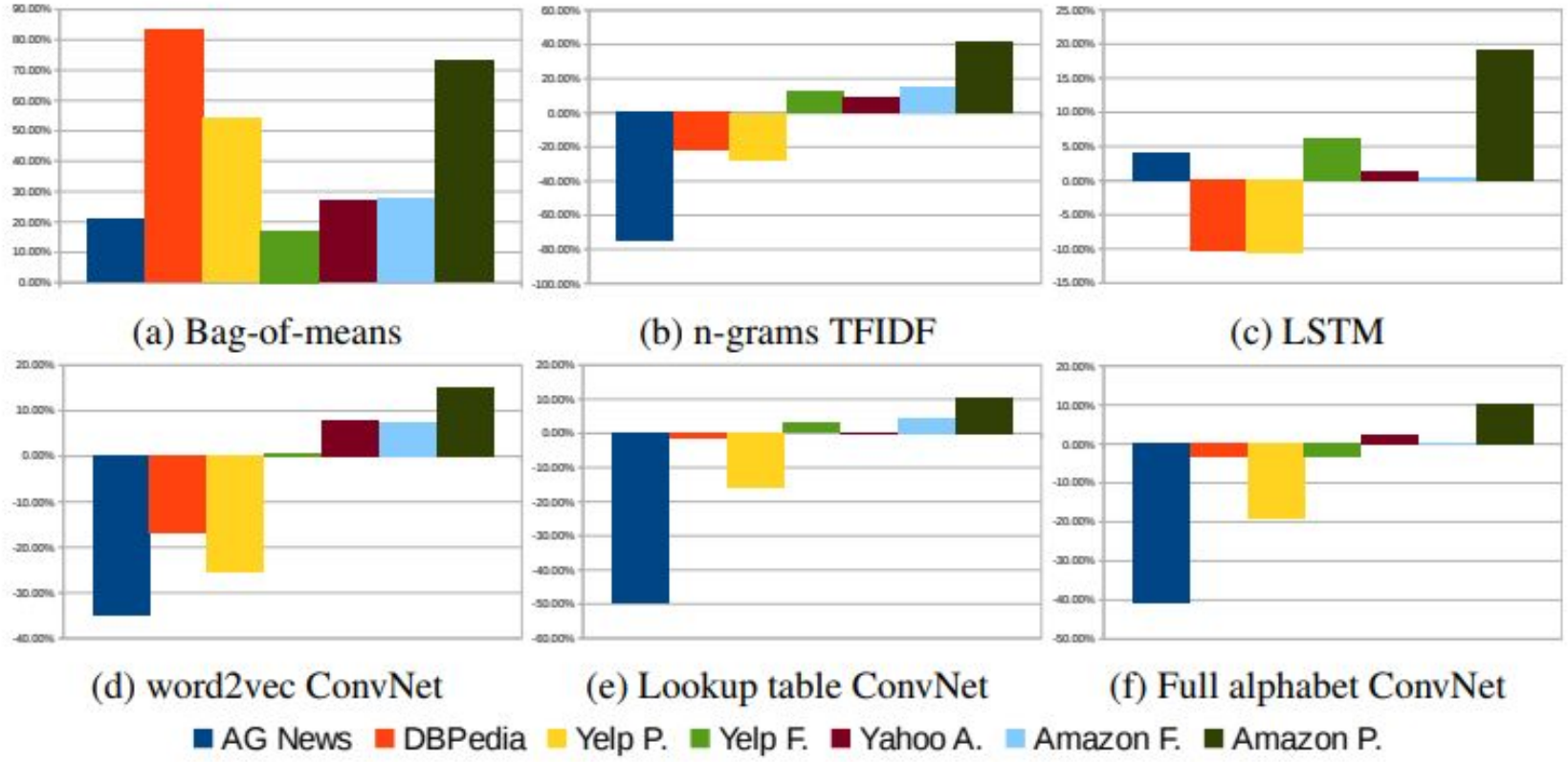


Figure 3: Relative errors with comparison models

Conclusion

- Character-level CNN is an effective method.

- How well the model performs depends on many factors
  - Dataset size
  - Whether the texts are curated
  - Whether the letters are distinguished between upper-case and lower-case

Q & A