Character-level Convolutional Networks for Text Classification

Xiang Zhang, Junbo Zhao, Yann LeCun

15/11/13

野中 尚輝

Paper information

- 発表学会：- (ArXiv updated 2015/9/10)
- New York 大学
  - LeCunさんの研究グループ
    - CNNを提案した人
    - Facebook AI ResearchのDirector
- テキスト分類について調べていたので選択

Introduction

- Convolutional networks (以下ConvNets)をテキスト分類に利用
- アルファベット+記号 (計60文字)をベクトルとして表現し、ConvNetsの入力とする
- 文章分類・感情分析の精度をbag-of-words、n-gramなどの手法と比較
- 純粋に文字を特徴量としてConvNetsを適用した研究は初

Related works

- ConvNetsの言語表現への直接の適用
  - 従来の手法と同程度の精度
    - （Syntactic or semantic knowledgeなしで）
- 文字を特徴量とした言語処理
  - N-gram + 線形分類器
  - 文字レベルの特徴 + ConvNets
    - 単語またはn-gramに対してConvNetsを適用し、得られた分散表現により、単語を表現する

Model

- Key module
  - Temporal convolutional model
    - 1-D convolutionを計算

$$h(y) = \sum_{x=1}^{k} f(x) \cdot g(y \cdot d - x + c),$$

  - Temporal max-pooling
    - 1-D max-pooling

$$h(y) = \max_{x=1}^{k} g(y \cdot d - x + c),$$

Model

- Key module
  - Non-linearity
    - Rectifier or thresholding $h(x) = \max\{0, x\},$
    - (Convolution layerはReLUに似る)
  - SGDで学習
  - Minibatch サイズは128
  - モメンタム 0.9
  - 初期のstep sizeは0.01で3epochごとに半減

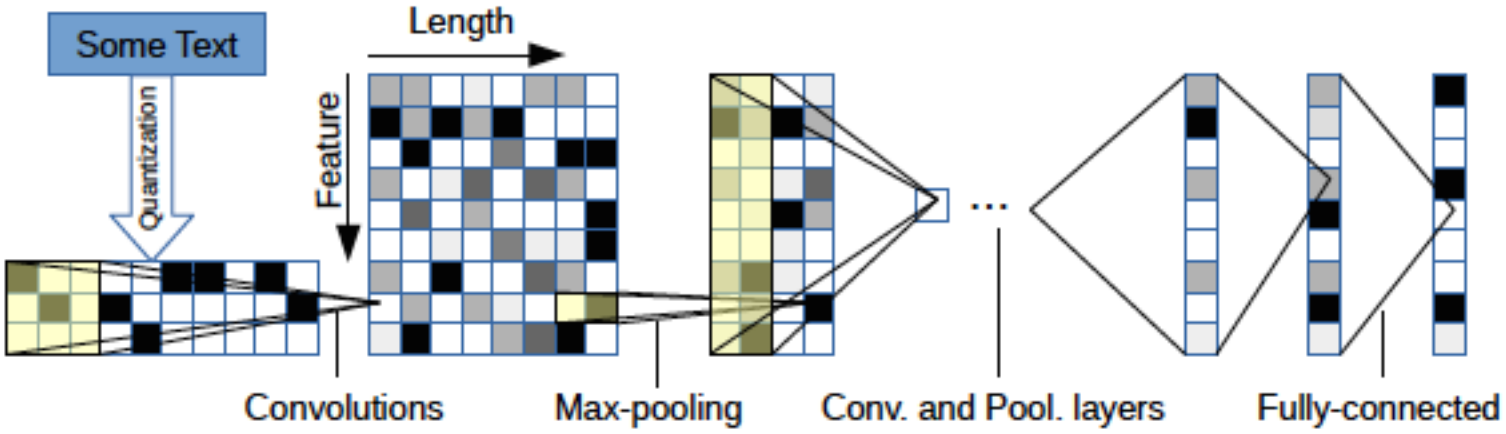
Model

- Character quantization
  - m次元のベクトルを設定
    - 1-of-m encodingで文字を量子化
  - 文字の連続（単語、文章）を量子化
    - 長さはl_0 に固定
    - l_0以上のものは無視
    - 空白や対象外の文字はall-zero vector
  - 後ろから順に量子化していく
  - 大文字と小文字の区別はなし
    - （実験では区別する条件もある）

Model

- Model design
  - 6層Convolution、3層full connect
    - 大小2種のConvNetsを用意
  - Feature数70、input length 1014
    - 対象の文字が70種で、1014字を解析する
  - Dropoutを各full connect層間に挿入
  - 重みの初期化はガウス分布に従う

Model



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	

Model

- Data augmentation
  - Data augmentationの技術は、汎化性能を向上させるのに重要
    - テキストデータではsignal transformationなどはできない
    - 同義語を置換してdata augmentation
      - 文章内の置換可能な語を確率的に置換

Comparison models

- Traditional methods
  - Bag-of-words
    - 最頻出50,000単語を選択（各データセット）
      - 出現回数をそのまま or TFIDF値
  - Bag-of-ngrams
    - 最頻出500,000 n-gramを選択（各データセット）
      - 出現回数をそのまま or TFIDF値
  - Bag-of-means
    - Word2vecを適用（各データセット）
      - 300次元、5回以上出現する語が対象

Comparison models

- Deep learning methods
  - Word based ConvNets
    - Pre-trained -> word2vecで事前学習
    - End-to-end learned -> look up tableを使用
  - LSTM
    - RNNを使用したモデルも比較対象

Dataset

- 著者らが新たに作成
  - いろいろあるけどデータ量が不十分だった

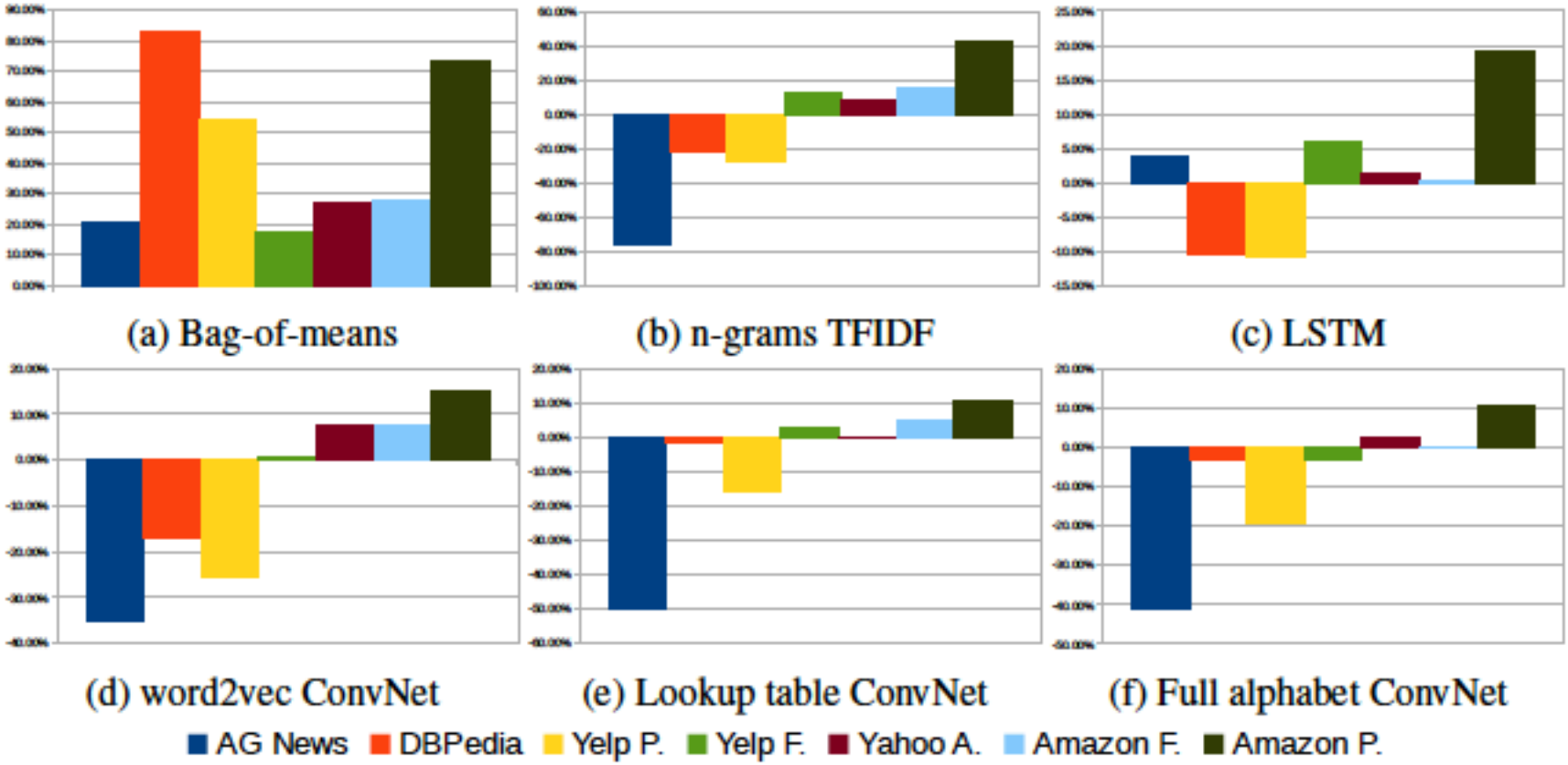
Dataset	Classes	Train Samples	Test Samples	Epoch Size
AG's News	4	120,000	7,600	5,000
Sogou News	5	450,000	60,000	5,000
DBPedia	14	560,000	70,000	5,000
Yelp Review Polarity	2	560,000	38,000	5,000
Yelp Review Full	5	650,000	50,000	5,000
Yahoo! Answers	10	1,400,000	60,000	10,000
Amazon Review Full	5	3,000,000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

Sogou Newsのみ中国語をpinyin化して解析

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

Results



データセットごとに提案手法に対するエラー率を図示する
-> 正に大きくなるほど提案手法より悪く結果を意味する

Results

- 文字ベースのConvNetsは有用な手法
- データサイズによってConvNetsが有効かは決まる
- ConvNetsはユーザ生成データに有用かもしれない
- アルファベットの選択により結果は変わる
- タスクのSemanticは無関係
- Bag-of-meansは正しい使い方でない
- No free lunch

Conclusion

- テキスト分類のための文字に対するConvNetsについて、既存手法と比較した
- データサイズやアルファベットの選択など様々な要因により結果が変わる