Deep Pyramid Convolution Neural Networks for Text Categorization

Rie Johnson & Tong Zhang

Presented By: Balwant Singh Bisht

Introduction

Text categorization is the task of assigning predefined categories to free-text documents.

- Spam Detection
- Sentiment Classification
- Topic Classification

Various Approches

- Pre- 2014: Bag of Words + Linear Classifiers (SVM)
- After-2014: Move to Deep Learning

Why DPCNN?

Deep Character level CNNs

Pros: Don’t have to deal with millions of distinct words

Cons: Less accurate and very slow than word-level CNNs

Shallow CNN

Short range associations

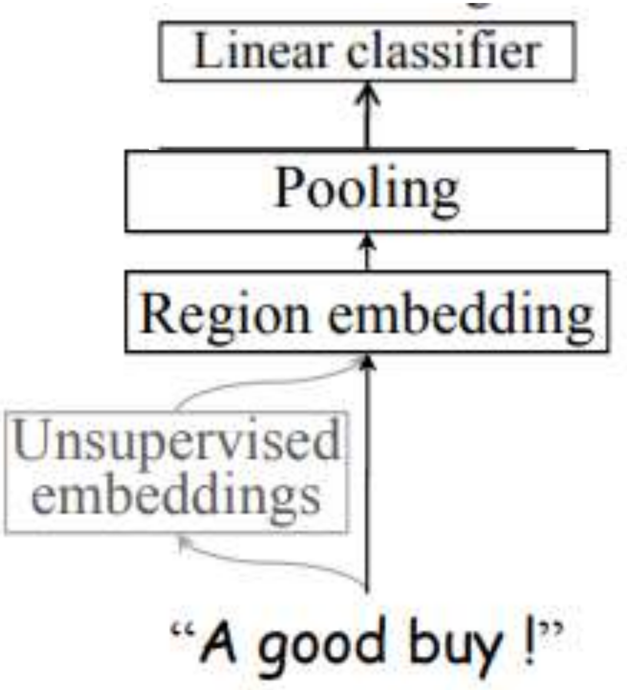
LSTM

Long range associations but slow

DPCNN

Long range associations and fast

Shallow CNN



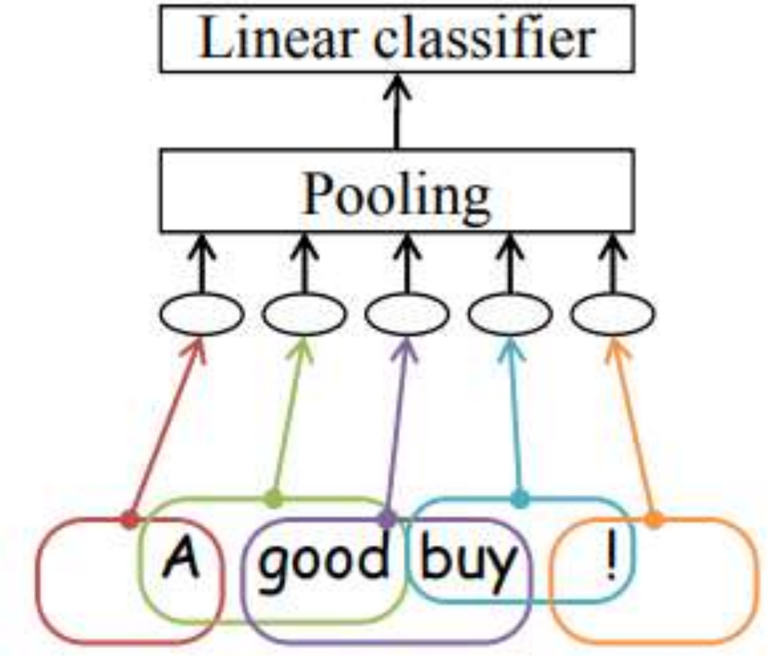
Max(avg) Pooling

Region embedding = Convolution

Unsupervised embeddings

One hot encoding

Text Region Embedding: Basic



- In each oval, computation in the following form takes place

  $$\boldsymbol{\sigma}(\mathbf{W}(\mathbf{x}) + \mathbf{b}$$

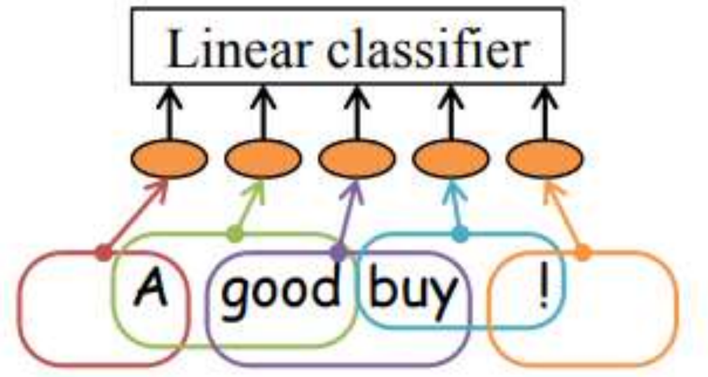
  σ is component wise non-linearity

  $$\boldsymbol{\sigma}(x) = \max(0, x)$$

- Input **x** is one-hot representation of each text-region(e.g '**good buy**')

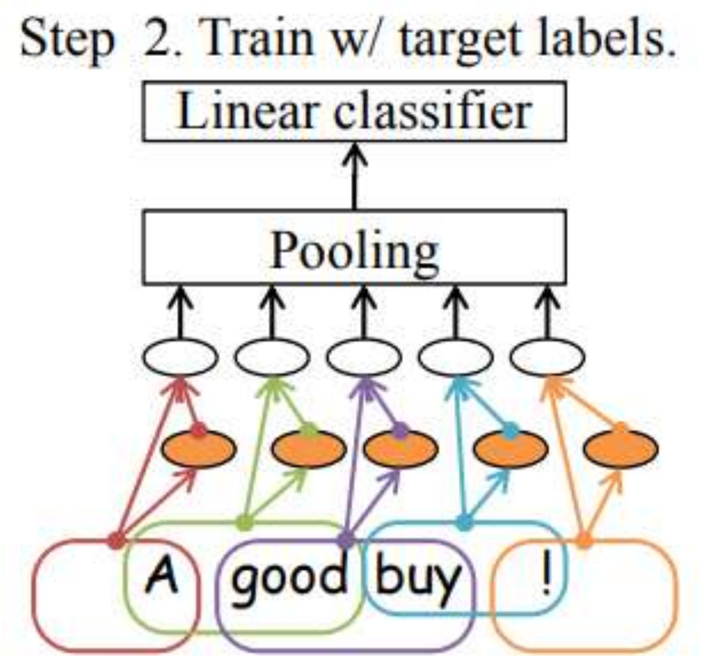
Text Region Embedding: Enhanced with unsupervised embeddings

Step 1. Train tv-embedding with two-view embedding learning objectives.



- Unsupervised embeddings are obtained by tv embeddings
- tv stands for two views
- Predict adjacent text regions (one view) based on a text region (the other view)

Text Region Embedding: Enhanced with unsupervised embeddings

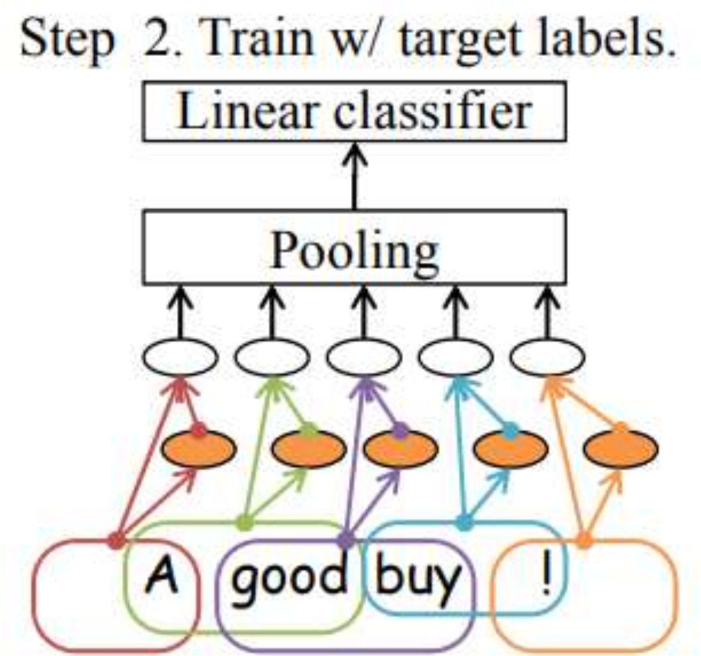


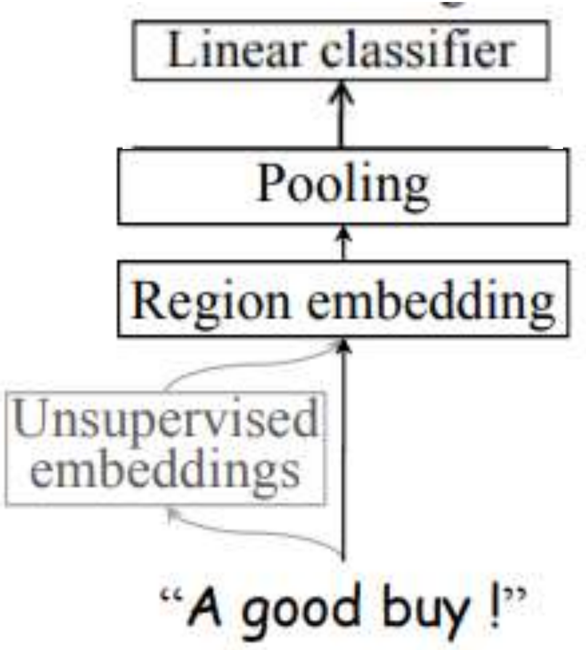
tv-embedding is used to produce additional input to the base model and train it with labeled data

Region embedding funtion for multiple tv-embeddings

$$g(\mathbf{x}, \{\mathbf{x}^{(i)}\}_i) = \sigma\left(\mathbf{W}\mathbf{x} + \sum_i \mathbf{W}^{(i)}\mathbf{x}^{(i)} + \mathbf{b}\right)$$

Shallow CNN


Step 2. Train w/ target labels.
Linear classifier
Pooling
A
good
buy
!

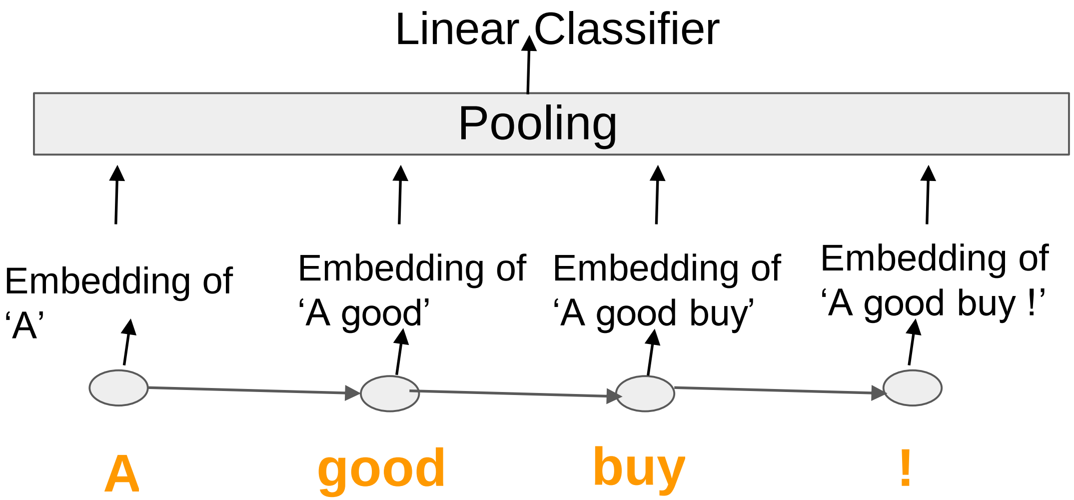

Linear classifier
Pooling
Region embedding
Unsupervised embeddings
"A good buy !"

What the region embedding layer of shallow CNN Does?

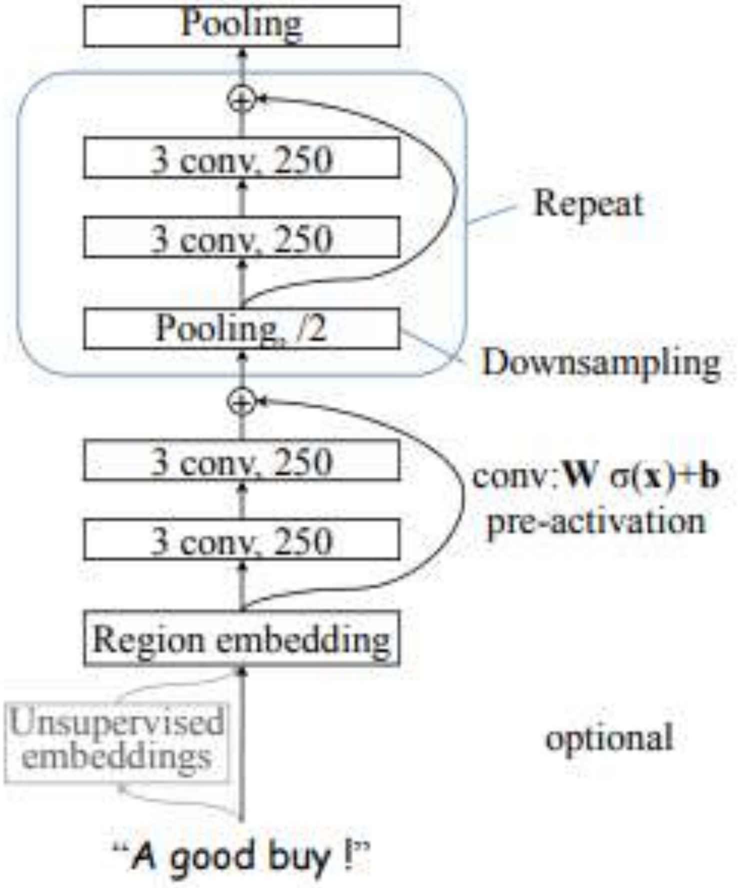
- Captures Short range associations
  - Captures more complex concepts with single layer, word embedding takes multiple layers

- Disadvantage
  - Small text regions cannot capture long range associations
  - LSTM addresses the issue but computationally slow

Long Range Associations by LSTM

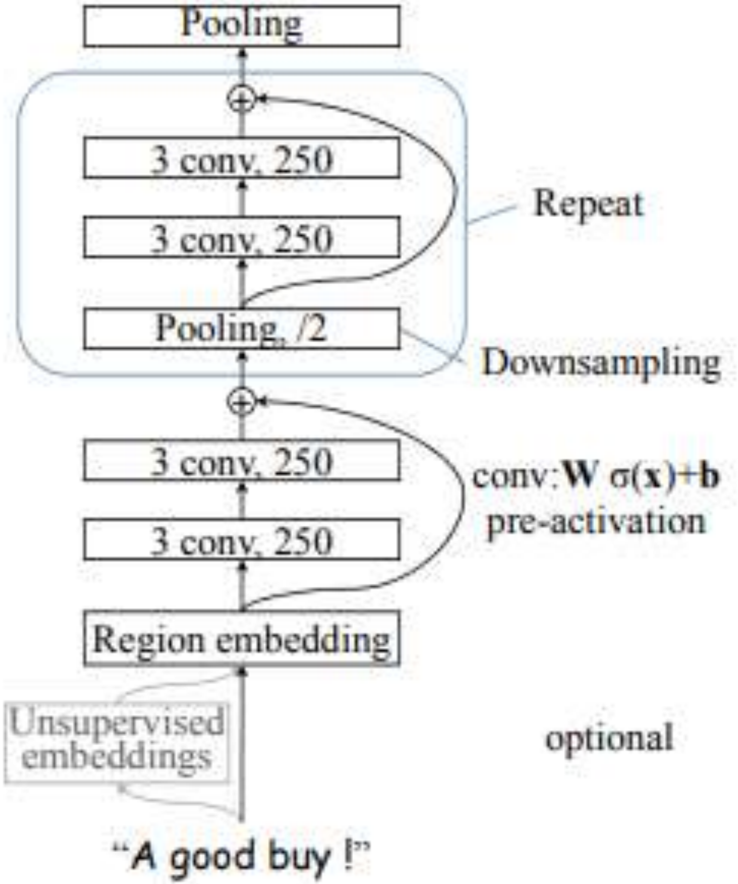


Ouput at each word: **embedding of text seen so far**
Large region size but sequential computation (thus slow)



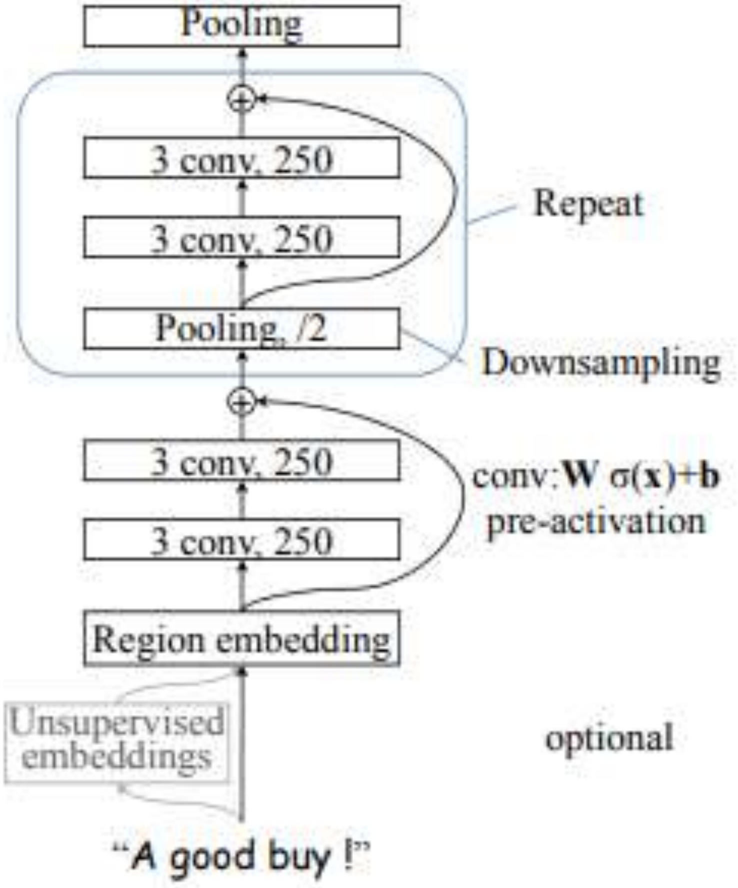
DPCNN Structure

- Downsampling with the number of feature maps fixed
  - Reduces per block computation
  - Total computation bounded by constant
- Shortcut connections with pre activation and identity mapping
  - Enables training
- Text region embedding enhanced with unsupervised embeddings
  - Improves accuracy



Downsampling with number of feature maps fixed

- Max-pooling with size 3 and stride 2
  - Component wise maximum over 3 contiguous internal vectors
  - Reduces size of internal representation by half
  - Computation time for each block is halved



Shortcut connections with pre-activations

Eases training of deep networks

Written as z + f(z) where f represents skipped layers (two convolution layers with pre-activation)

Pre-activation
Activation is done before weighting
$W\boldsymbol{\sigma}(\mathbf{x}) + b$

$$\boldsymbol{\sigma}(x) = \max(x, 0)$$

Linearity eases training (empirically)

No need for dimension matching

Experiments: Dataset

	AG	Sogou	Dbpedia	Yelp.p	Yelp.f	Yahoo	Ama.f	Ama.p
# of training documents	120K	450K	560K	560K	650K	1.4M	3M	3.6M
# of test documents	7.6K	60K	70K	38K	50K	60K	650K	400K
# of classes	4	5	14	2	5	10	5	2
Average #words	45	578	55	153	155	112	93	91

- AG and Sogou are news
- Dbpedia is an ontology
- Yahoo consists of questions and answers
- Yelp and Amazon are reviews: .p represents binary labels, .f represents stars
- Vocabulary size: 30K words

Experiments: Training

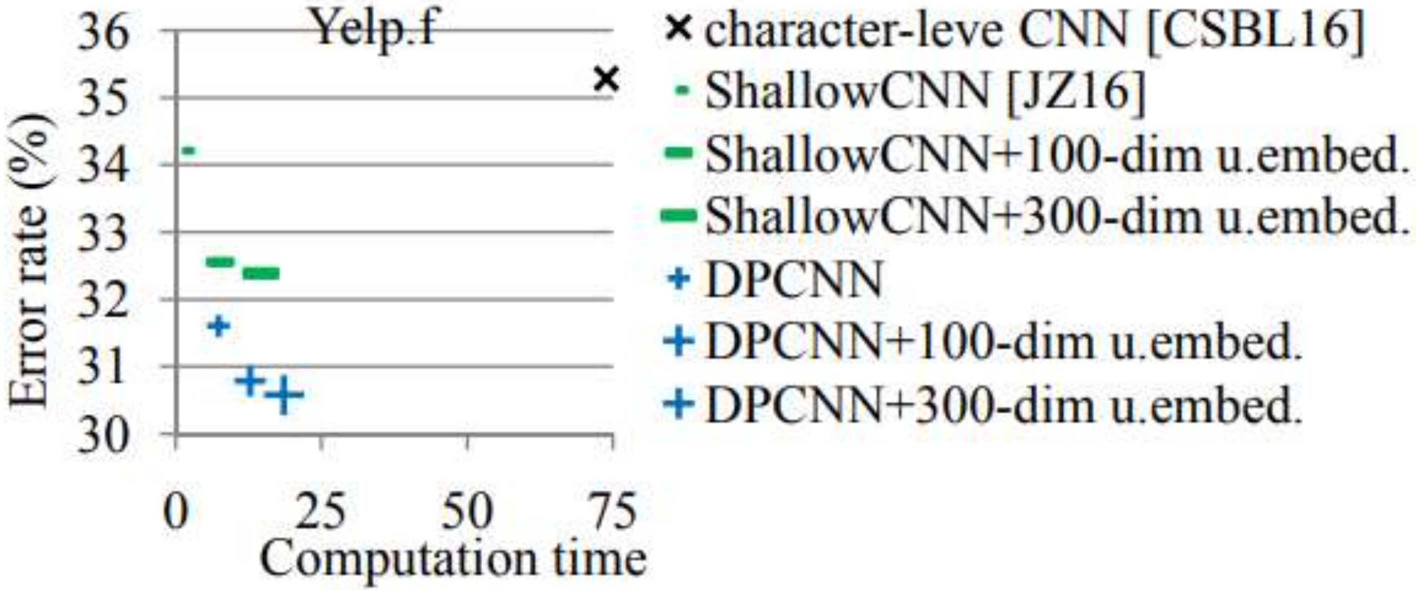
- Minimize log loss with softmax with mini-batch SGD
- Weights were initilized by Gaussian distribution with zero mean and standard deviation 0.01
- Input to region embedding layer was fixed to BOW input
- Output dimensionaly: 250

Results: Error rates(%) comparison on Large data

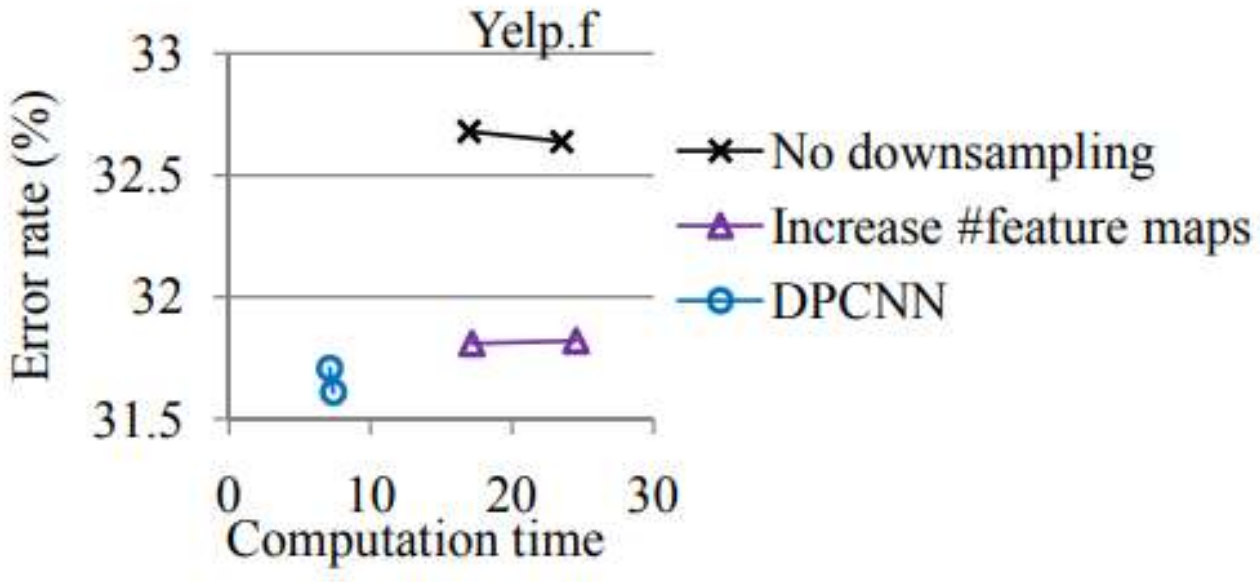
Depth: 15 weight layers plus unsupervised embeddings

	Models	Deep	Unsup. embed.	Yelp.p	Yelp.f	Yahoo	Ama.f	Ama.p
1	DPCNN + unsupervised embed.	✓	tv	**2.64**	**30.58**	**23.90**	**34.81**	**3.32**
2	ShallowCNN + unsup. embed. [JZ16]		tv	*2.90*	*32.39*	24.85	*36.24*	*3.79*
3	Hierarchical attention net [YYDHSH16]	✓	w2v	–	–	*24.2*	36.4	–
4	[CSBL16]'s char-level CNN: *best*	✓		4.28	35.28	26.57	37.00	4.28
5	fastText bigrams (Joulin et al., 2016)			4.3	36.1	27.7	39.8	5.4
6	[ZZL15]'s char-level CNN: *best*	✓		4.88	37.95	28.80	40.43	4.93
7	[ZZL15]'s word-level CNN: *best*	✓	(w2v)	4.60	39.58	28.84	42.39	5.51
8	[ZZL15]'s linear model: *best*			4.36	40.14	28.96	44.74	7.98

Results: Error rates vs Computation time



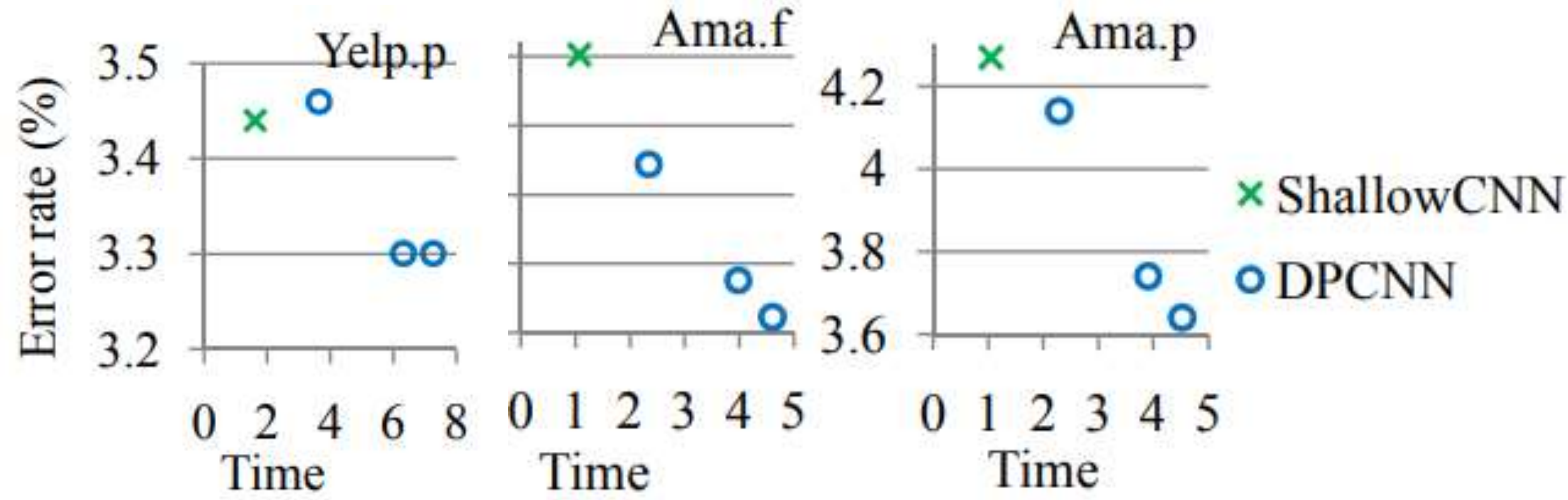
Results: Comparison with non-pyramid models



Results: Error rates(%) comparison on smaller data

	Models	Deep	Unsup. embed.	AG	Sogou	Dbpedia
1	DPCNN + unsupervised embed.	✓	tv	*6.87*	**1.84**	*0.88*
2	ShallowCNN + unsup. embed. [JZ16]		tv	**6.57**	*1.89*	**0.84**
3	[ZZL15]'s linear model: *best*			7.64	2.81	1.31
4	[CSBL16]'s deep char-level CNN: *best*	✓		8.67	3.18	1.29
5	fastText bigrams (Joulin et al., 2016)			7.5	3.2	1.4
6	[ZZL15]'s word-level CNN : *best*	✓	(w2v)	8.55	4.39	1.37
7	[ZZL15]'s deep char-level CNN: *best*	✓		9.51	4.88	1.55

Results: Error rates(%) comparison with depth



Results: Error rates(%) comparison with variations in unsupervised embeddings

	Unsupervised embeddings	Yelp.p	Yelp.f	Yahoo	Ama.f	Ama.p
1	tv-embed. (additional input)	**2.64**	**30.58**	**23.90**	**34.81**	**3.32**
2	tv-embed. (sole input)	2.68	30.66	24.09	35.13	3.45
3	word2vec (pretraining)	2.93	32.08	24.11	35.30	3.65
4	–	3.30	31.61	24.64	35.61	3.64

Conclusion

Problem: To design a high performance deep word-level CNN for text categorization for large training data

Deep pyramid CNN model is proposed with

- Low complexity
- Efficient long range associations in text
- Accuracy better than previous models on six benchmark datasets

Thank You