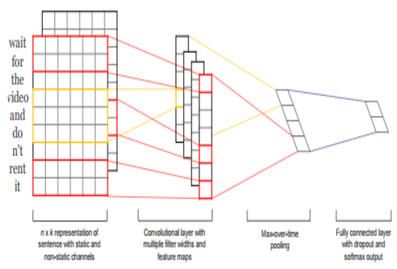
Do Convolutional Networks need to be Deep for Text Classification ?

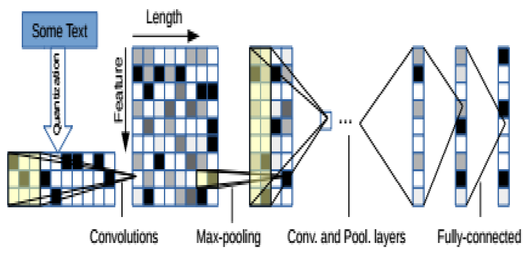
Hoa T. Le, Christophe Cerisara, Alexandre Denis

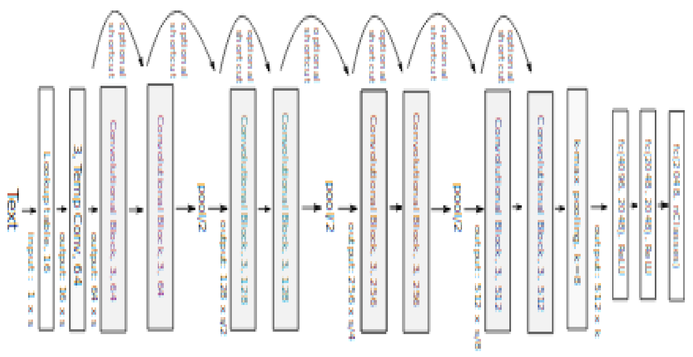
Loria Laboratoire lorrain de recherche en informatique et ses applications

https://arxiv.org/pdf/1707.04108.pdf

Convolution becomes deeper and deeper







Difference between Image and Text

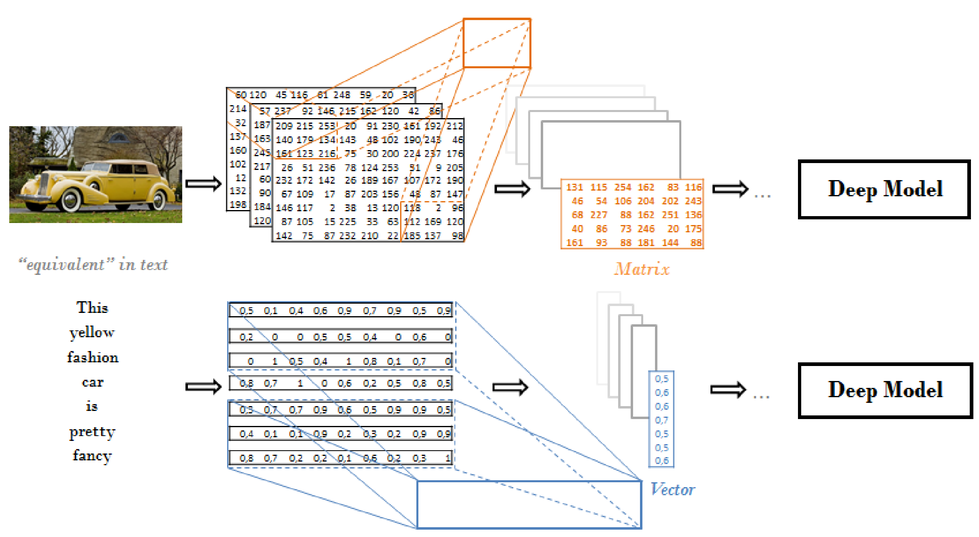
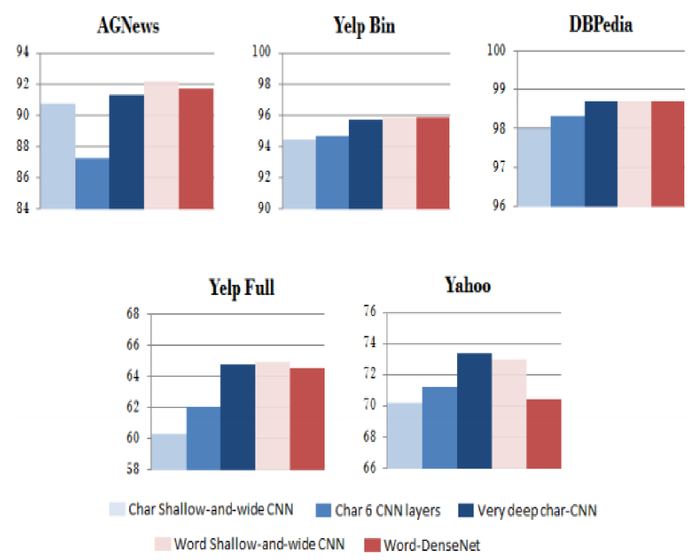


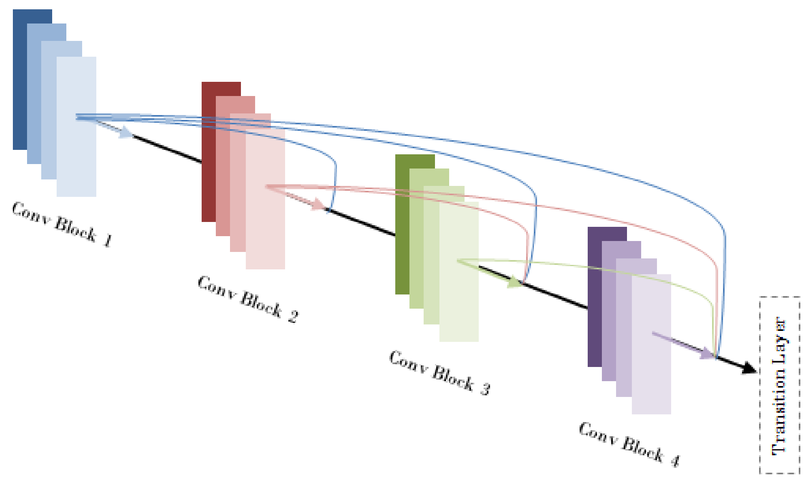
Image has *real-valued* and *dense*. **Text** has *discrete* tokens, *artificial* and sparse values representation. The output of the first convolution layer on image is still a *Matrix* but on text, it is reduced to a *Vector*

Experiments

Models	AGNews	Yelp Bin	Yelp Full	DBPedia	Yahoo
Char shallow-and-wide CNN	90.7	94.4	60.3	98.0	70.2
Char-DenseNet $N_b = (4-4-4-4)$ Global Average-Pooling	90.4	94.2	61.1	97.7	68.8
Char-DenseNet $N_b = (10-10-4-4)$ Global Average-Pooling	90.6	94.9	62.1	98.2	70.5
Char-DenseNet $N_b = (4-4-4-4)$ Local Max-Pooling	90.5	95.0	63.6	98.5	72.9
Char-DenseNet $N_b = (10-10-4-4)$ Local Max-Pooling	92.1	95.0	64.1	98.5	73.4
Word shallow-and-wide CNN	92.2	**95.9**	**64.9**	**98.7**	73.0
Word-DenseNet $N_b = (4-4-4-4)$ Global Average-Pooling	91.7	95.8	64.5	**98.7**	70.4*
Word-DenseNet $N_b = (10-10-4-4)$ Global Average-Pooling	91.4	95.5	63.6	98.6	70.2*
Word-DenseNet $N_b = (4-4-4-4)$ Local Max-Pooling	90.9	95.4	63.0	98.0	67.6*
Word-DenseNet $N_b = (10-10-4-4)$ Local Max-Pooling	88.8	95.0	62.2	97.3	68.4*
bag of words (Zhang et al., 2015)	88.8	92.2	58.0	96.6	68.9
ngrams (Zhang et al., 2015)	92.0	95.6	56.3	98.6	68.5
ngrams TFIDF (Zhang et al., 2015)	92.4	95.4	54.8	**98.7**	68.5
fastText (Joulin et al., 2016)	**92.5**	95.7	63.9	98.6	72.3
char-CNN (Zhang et al., 2015)	87.2	94.7	62.0	98.3	71.2
char-CRNN (Xiao and Cho, 2016)	91.4	94.5	61.8	98.6	71.7
very deep char-CNN (Conneau et al., 2016)	91.3	95.7	64.7	**98.7**	73.4
Naive Bayes (Yogatama et al., 2017)	90.0	86.0	51.4	96.0	68.7
Kneser-Ney Bayes (Yogatama et al., 2017)	89.3	81.8	41.7	95.4	69.3
MLP Naive Bayes (Yogatama et al., 2017)	89.9	73.6	40.4	87.2	60.6
Discriminative LSTM (Yogatama et al., 2017)	92.1	92.6	59.6	**98.7**	**73.7**
Generative LSTM-independent comp. (Yogatama et al., 2017)	90.7	90.0	51.9	94.8	70.5
Generative LSTM-shared comp. (Yogatama et al., 2017)	90.6	88.2	52.7	95.4	69.3



DenseNet for Text



Multiple convolutional filters output 2D matrices, which are all concatenated together before going into another dense block. Following (Conneau et al., 2016), we experiment two most effective configurations for word and character-level: Nb = (4 − 4 − 4 − 4) and Nb = (10 − 10 − 4 − 4), which are the number of convolutional layers in each of the four blocks.

Tasks & Datasets

Dataset	#y	#train	#test	Task
AGNews	4	120k	7.6k	ENC
Yelp Binary	2	560k	38k	SA
Yelp Full	5	650k	38k	SA
DBPedia	14	560k	70k	OC
Yahoo	10	1 400k	60k	TC

#train: number of training tokens, **#test**: number of test tokens, **#y**: number of target class; **ENC**: English News Categorization, **SA**: Sentiment Analysis, **OC**: Ontology Classification, **TC**: Topic Classification

Context

The reason for the recent success of Deep Learning is the ability to learn very complex features when the model goes deeply. This results in a real amelioration of performance in image and speech processing. A lot of works have gone in this direction to evaluate its effectiveness in NLP.

Our study aims to clarify this hypothesis by providing a thorough evaluation of:

- Shallow CNN on char-level
- Deep CNN on char-level
- Shallow CNN on word-level
- Deep CNN on word-level

Besides, we introduce & validate for the first time an adaptation of DenseNet for Text Classification.

Conclusions

- Very deep models *do not* seem to bring a significant advantage over shallow networks for text classification, as opposed to the observed performances in image processing.
- A *global max-pooling* (Collobert and Weston, 2008), which retrieves the *most influential features* could already be good enough for sparse and discrete input text, and gives similar results than a local max-pooling with a deep network.
- Char-level could be a choice but *word-level* is still the *most effective* method. Moreover, in order to use char-level representation, we must use a very deep model, which is less practical because it takes a long time to train.