Text Categorization Survey - From the Vector Space Model to Deep Learning

Tim vor der Brück

Lucerne University of Applied Sciences and Arts | School of Information Technology

Swisstext 2016
Winterthur, Switzerland
June 2016

Introduction

Text categorization (TC) denotes the problem to automatically distribute texts into several classes, usually by a supervised statistical machine learning method. Its applications are manifold and include:

- Discern between spam and ham emails
- Distribute support emails in companies to the correct person in charge
- Assess the polarities (positive or negative) of sentences or paragraphs

Classical Vector Space Model

For a long time, text categorization methods were predominantly based on the vector space model

- Idea: Represent document as bag of words (BoW, possibly use certain word n-grams in addition)
- Each word is assigned a unique id
- Document vector component (also called feature) at position i is given as weighted occurrence of word with id i in this document
- Popular weight measures:
  - $TF \times IDF$: a word is strongly weighted if it appears often in the considered document but rarely in the entire corpus
  - GSS (Sebastiani 2002, normally used for binary weights)
  - Odds-Ratio
- Documents are usually categorized by applying a Support Vector Machine (SVM) or a Nearest Neighbor approach on the feature maps (Sebastiani 2002)
- Drawback of the vector space / bag of words model: word sequence is disregarded, Example from sentiment analysis (Socher 2015)
  - White blood cells destroying an infection → positive
  - An infection destroying white blood cells → negative

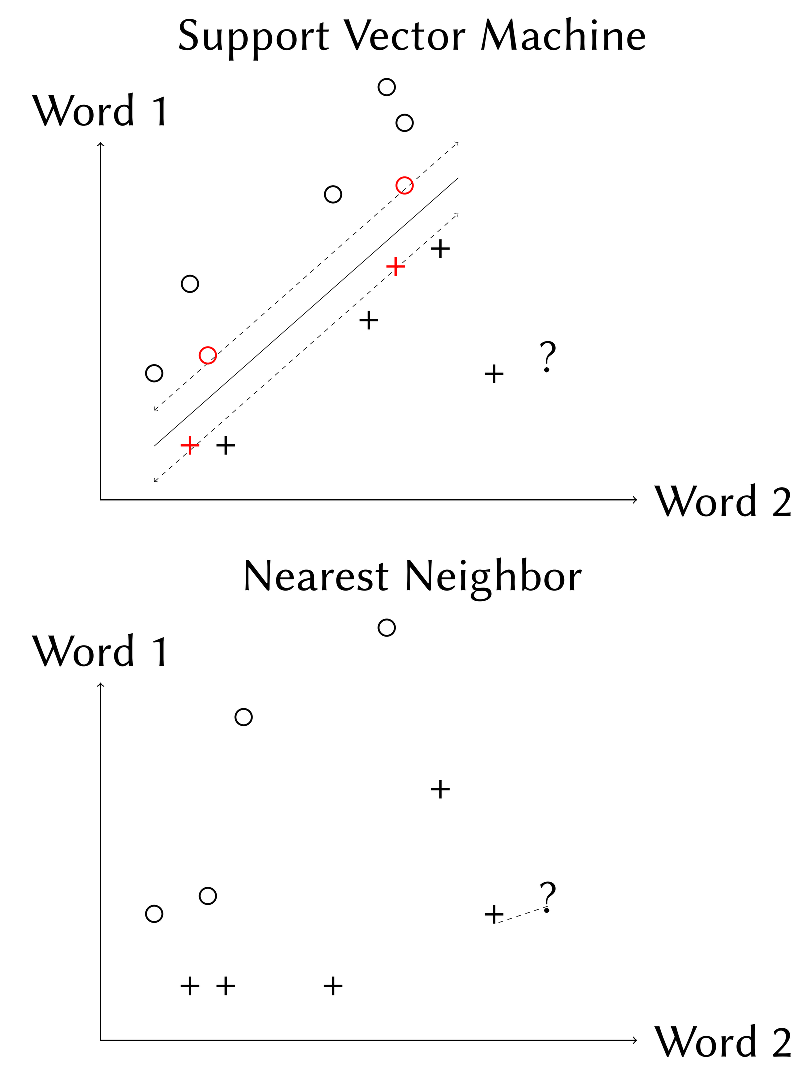

Fig. 1: Support Vector Machine and Nearest Neigbor based categorization of a prevIf aren't changing: previsouly unseen document (indicated by a question mark)

Deep Learning

- Learning paradigm based on multi-layered artificial neural networks
- Features are learned automatically by the network ⇒ abandonment of manual feature engineering

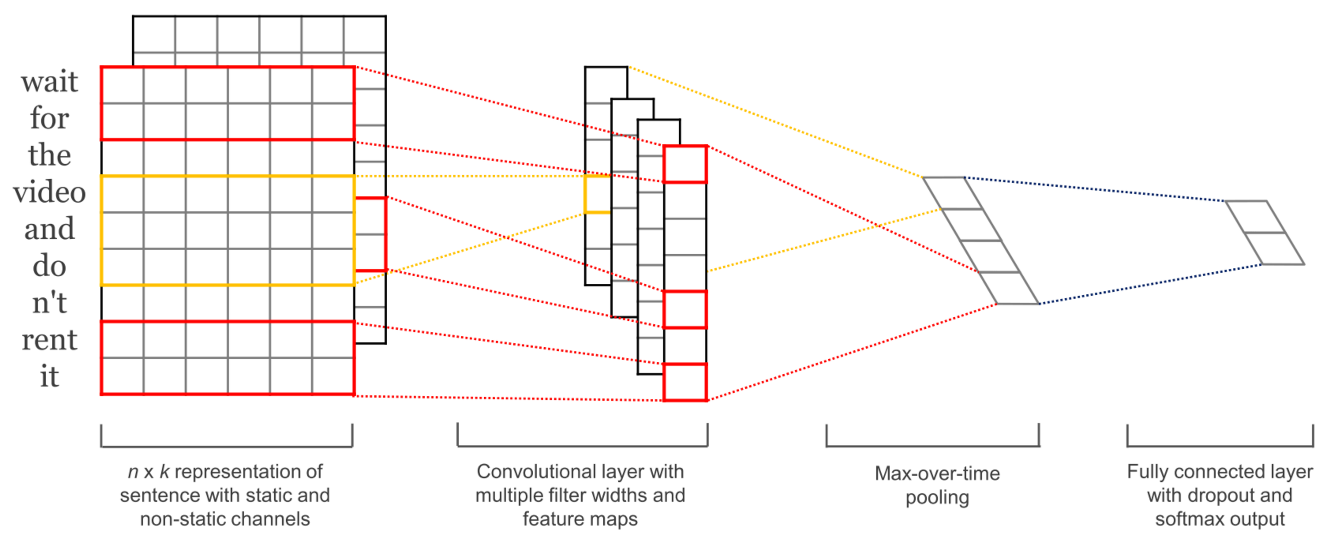

Fig. 2: Architecture of a deep learning TC approach based on Convolutional Neural Networks (from Kim 2014)

- Neural network weights are usually determined by backpropagation with a combination of stochastic gradient descent and momentum (Buduma 2016)

TC with Recursive Neural Networks

- Capture semantics of a sentence via a tree structure (i.e., dependency tree/DAG or constituency tree)
- Drawbacks
  - Construction of such a tree requires a runtime of $\mathcal{O}(m^2)$ (m=text length)
  - Constructed tree can be erroneous or construction can even fail

TC with Convolutional Neural Networks (CNNs)

- Convolution: concept originating primarily from image processing
- Principle: apply the same weight vector iteratively on fixed-size token windows (of size $2N+1$) to obtain filter values for focal tokens
- Convolutional network: network of convolutional layers
- Formally:
  $F(i) := g(b + \sum_{j=-N}^{N} \langle \mathbf{word}(i-j), \mathbf{W}(j+N) \rangle)$
  - $\mathbf{word}(j)$: word vector of size n
  - $\mathbf{W}$: weight vector (in image processing usually a two or three dimensional tensor)
  - b: bias term
  - g: activation function
  - $F(i)$: value of convolutional neuron
- Aggregate the convolution neurons with max-pooling
- Output neurons are determined by soft-max function
- One drawback of Convolutional Neural Networks is their fixed window size which led to the development of Recurrent Convolutional Neural Networks (RCNN)

Conclusion

- NNs clearly outperform traditional approaches based on the Vector Space Models
- Highest F-Score in the experiment was achieved with RCNNs for three out of four data sets

Evaluation (Lai et al. 2015)

Model	20News	Fudan	ACL	SST
BoW+LR	92.81	92.08	46.67	40.86
Bigram+LR	93.12	92.97	47.00	36.24
BoW+SVM	92.43	93.02	45.24	40.70
Bigram+SVM	92.32	93.03	46.14	36.61
Avg. Embedding	89.39	86.89	41.32	32.70
ClassifyLDA-EM	93.60	-	-	-
Labeled-LDA	-	90.80	-	-
CFG	-	-	39.20	-
C and J	-	-	**49.20**	-
RecursiveNN	-	-	-	43.20
RNTN	-	-	-	45.70
Paragraph-Vektor	-	-	-	**48.70**
CNN	94.79	94.04	47.47	46.35
RCNN	**96.49**	**95.20**	**49.19**	47.21

Table 1: Evaluation results given by Macro-averaging over F1-Scores (BoW=Bag of words, RNTN=Recursive Neural Tensor Network, LDA=Latent Dirichlet Allocation)

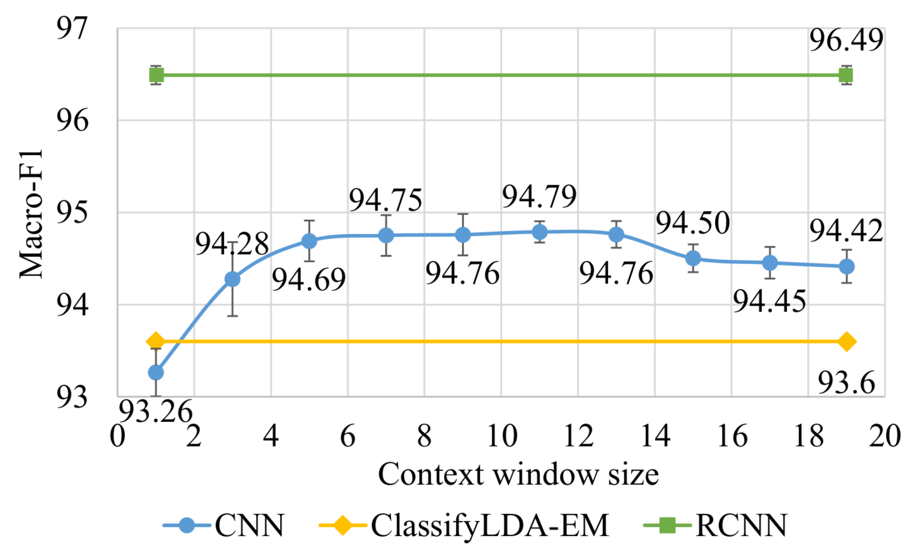

Fig. 3: Macro-F1 depending on different window sizes

References

Buduma, Nikhil (2016). *Early Release - Fundamentals of Deep Learning - Designing next-generation artificial intelligence algorithms.* Boston, USA: O'Reilly.

Kim, Yoon (2014). "Convolutional Neural Networks for Sentence Classification." In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Doha, Qatar.

Lai, Siwei et al. (2015). "Recurrent Convolutional Neural Networks for Text Classification." In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI).* Austin Texas, USA.

Sebastiani, Fabrizio (2002). "Machine Learning in Automated Text Categorization." In: *ACM Computing Surveys* 34.1, pp. 1–47.

Socher, Richard (2015). *Deep Learning for Natural Language Processing.* Talk at Text by the Bay 2015. URL: https://www.youtube.com/watch?v=tdLmf8t4oqM.

Email: tim.vorderbrueck@hslu.ch, Institute homepage: https://www.hslu.ch/de-ch/informatik/