Text Representations for Classification Tasks

For Machine Learning Lovers at Gdańsk (19.09.2016)

Karol Draszawka
Department of Computer Systems Architecture

Agenda

- Text classification tasks
- Quick survey on text representations for text classification tasks
- Trends, extrapolations, strong claims

Text classification tasks

Text classification

- Examples:
  - sentence classification
    - sentiment analysis
    - question classification
    - event detection (binary classification: positive if sentence contains an event)
  - dialog act classification
  - recognizing textual entailment (RTE)
  - spam detection
  - **topic categorization**

Text classification

cont.

- Formally: (finding a function that maps *x* (representing a text document) into a subset of a predefined set of categories)
  - if subsets are restricted to contain only one element, then it is *single-label* classification, else *multi-label*
- The task is:
  - to find such a function that do its job optimally
    - machine learning techniques
    - build such function manually based on linguistic filters and rules (still frequent approach, especially when small amount of data)
- open/close world classification (to read: *Breaking the Closed World Assumption in Text Classification*)

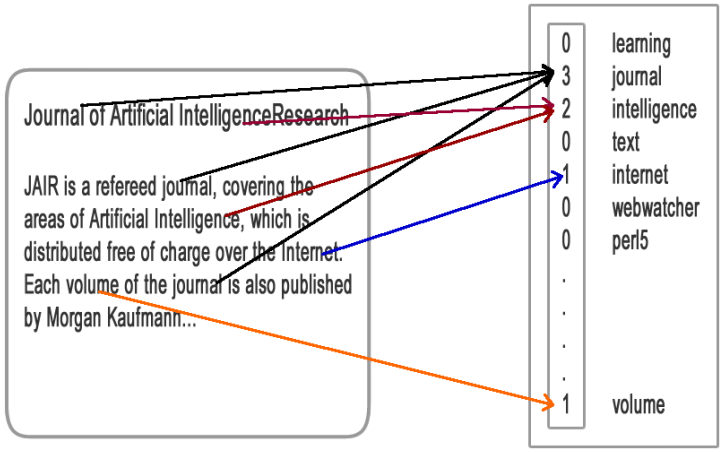
Specific challenges

- Wikipedia LSHTC Challenges
  - 4 editions, 1-3 as Pascal Challenges. 4th on Kaggle
  - unfortunately on Kaggle data only in a preprocessed format (BOW representation, LibSVM format)
  - my problem with this problem:
    - highly inconsistent manually labelled data

Quick survey on text representations for text classification tasks

Bag-of-Words (BOW)

- idea:
  - text represented as a vector of attributes (word occurrences, word counts)
- popular preprocessing:
  - to lower case
  - stemming, lemmatization
  - stop-words removal, count-based filtering
  - POS-Tagging



Source: www.openeco.eu Emotions in Text.

- popular postprocessing:
  - optionally weighting schemes (TF-IDF, BM25, Confidence Weights) applied
  - dimensionality reduction (LSA, ESA etc.)

Extensions to BOW

- N-grams (and bag of them)
  - bag of short sequences of words (up to 5 at most - Google), typically up to 3
- bag-of-links
  - available only for web pages
-

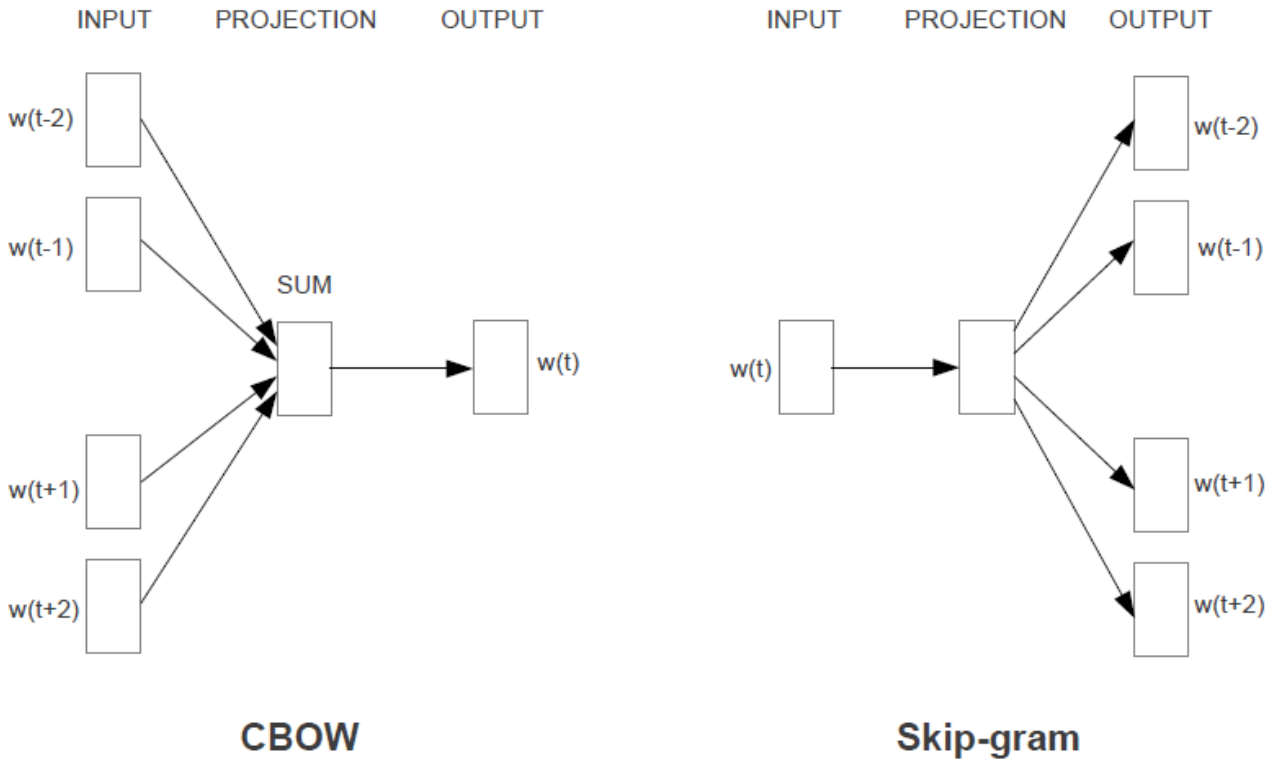
Wikipedia 4th LSHTC winning solution (Kaggle)

Distributed word representation

- word2vec (CBOW & Skip-gram models)
- GloVe

CBOW & Skip-gram models

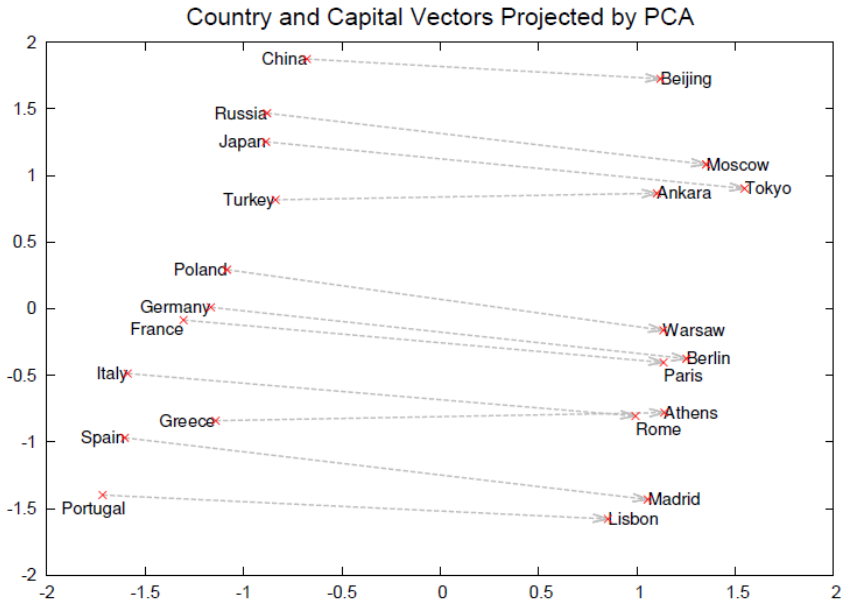
Architectures



Source: Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

CBOW & Skip-gram models

- word vector representations can be validated by examining preservation of semantic and syntactic relationship between words in a continues vector space
- on the right, a picture of a 2D PCA projection of 1000-dimensional word vector representations learned using Skip-gram model
- e.g. check which word representation vector is the closest to the result of $v('\text{scientist}') - v('\text{Einstein}') + v('\text{Picasso}')$



Source: Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. "Distributed representations of words and phrases and their compositionality", In Proc. Advances in Neural Information Processing Systems, 2013.

CBOW & Skip-gram models

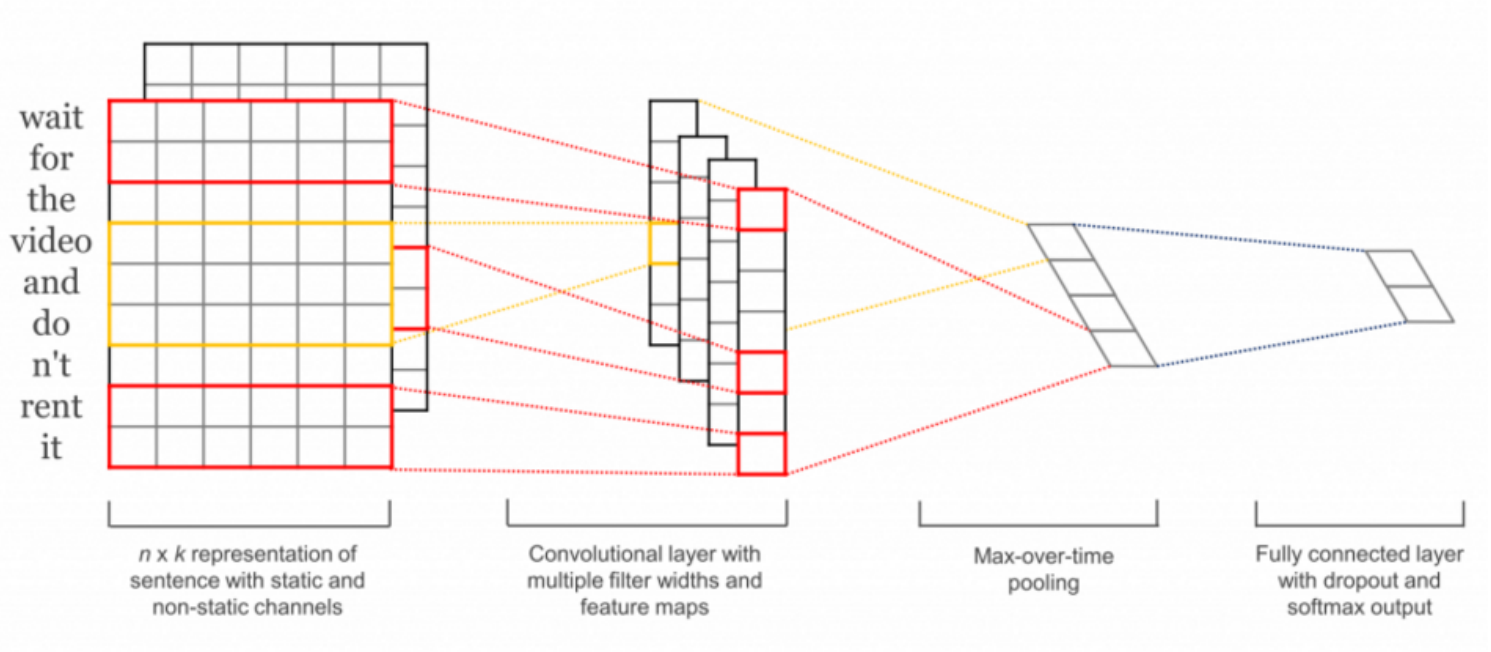
Results and training time

Table 6: *Comparison of models trained using the DistBelief distributed framework. Note that training of NNLM with 1000-dimensional vectors would take too long to complete.*

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days x CPU cores]
			Semantic	Syntactic	Total	
NNLM	100	6B	34.2	64.5	50.8	14 x 180
CBOW	1000	6B	57.3	68.9	63.7	2 x 140
Skip-gram	1000	6B	66.1	65.1	65.6	2.5 x 125

Source: Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

Convolutional Neural Networks for Sentence Classification



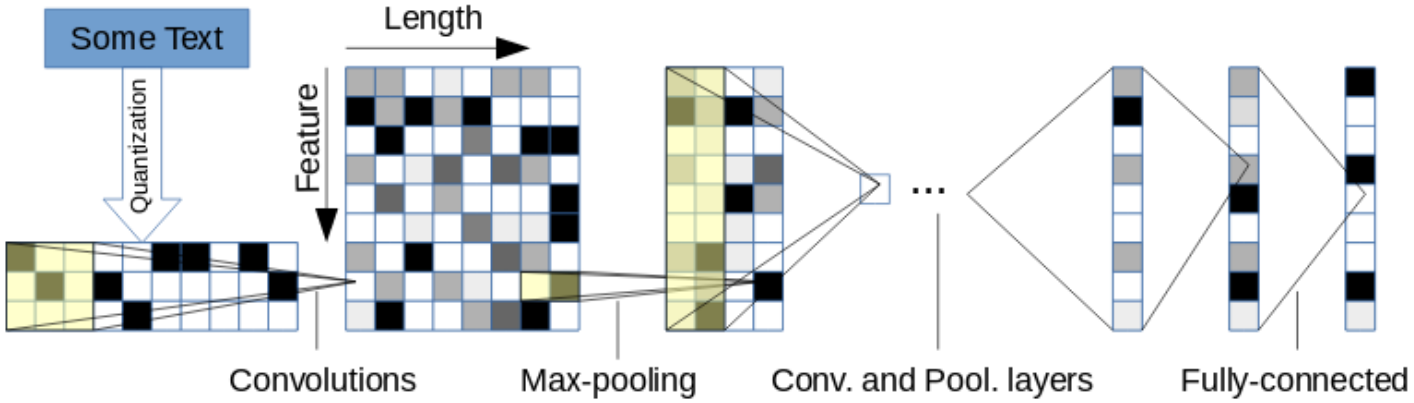
Source: Kim. Y. (2014). Convolutional Neural Networks for Sentence Classification.

Convolutional Neural Networks for Sentence Classification

- convolutional features (kernels) of 3,4,5 lengths, 100 feature maps each
- *word2vec* word embeddings of 300 dimensionality used at the input
  - so, the representation is: **sequence of cBoWs**
- only one convolutional layer, but pretrained word embeddings used
- pooling removes information about position -> turns to be bag-of-features
- experiments for sentence-level classification tasks only

Sequence of characters

Idea



Source:Zhang, X., Zhao, J., & LeCun, Y., Character-level convolutional networks for text classification (2015).

- 9 layers (6 convolutional (some with pooling) + 3 fully connected)
- LeNet adaptation to texts
- data augmentation: synonyms replacements with thesaurus

Sequence of characters

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

Source:Zhang, X., Zhao, J., & LeCun, Y., Character-level convolutional networks for text classification (2015).

- "dataset size forms a dichotomy between traditional and ConvNets models"
- data augmentation helps as usual

Hierarchical Attention Networks

- hierarchical document representation computation
  - sentence representations out of word representations (word2vec initialized)
- bidirectional gated recurrent neural networks as encoders
- two level of *attention* mechanism: word and sentence level
  - each one with a *universal query vector*

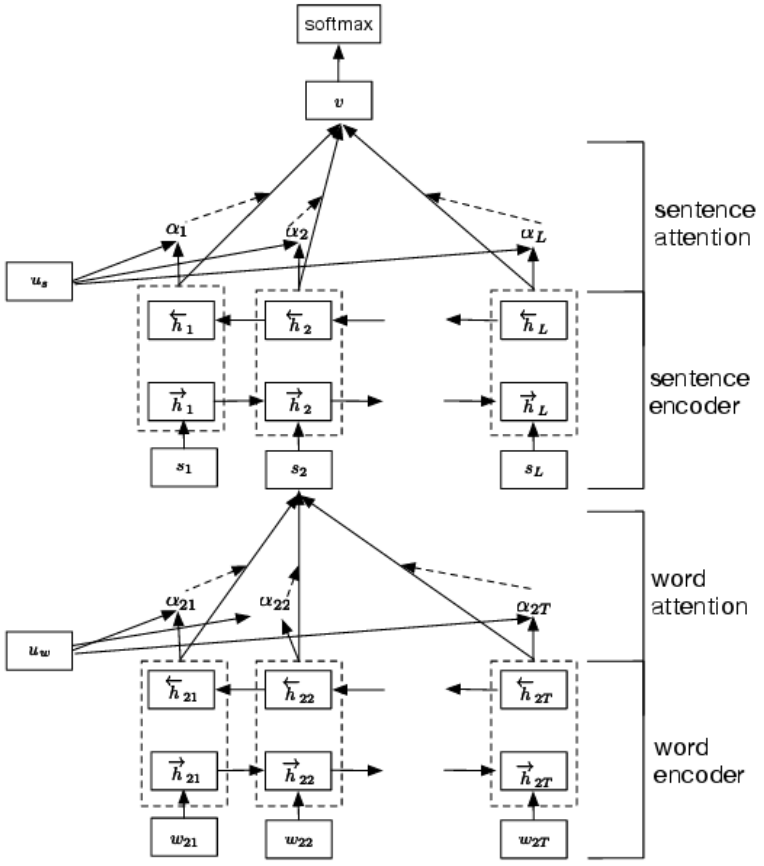


Figure 2: Hierarchical Attention Network.

Source: Yang, Zichao, et al., Hierarchical Attention Networks for Document Classification (2016).

Hierarchical Attention Networks

Results

	Methods	Yelp'13	Yelp'14	Yelp'15	IMDB	Yahoo Answer	Amazon
Zhang et al., 2015	BoW	-	-	58.0	-	68.9	54.4
	BoW TFIDF	-	-	59.9	-	71.0	55.3
	ngrams	-	-	56.3	-	68.5	54.3
	ngrams TFIDF	-	-	54.8	-	68.5	52.4
	Bag-of-means	-	-	52.5	-	60.5	44.1
Tang et al., 2015	Majority	35.6	36.1	36.9	17.9	-	-
	SVM + Unigrams	58.9	60.0	61.1	39.9	-	-
	SVM + Bigrams	57.6	61.6	62.4	40.9	-	-
	SVM + TextFeatures	59.8	61.8	62.4	40.5	-	-
	SVM + AverageSG	54.3	55.7	56.8	31.9	-	-
	SVM + SSWE	53.5	54.3	55.4	26.2	-	-
Zhang et al., 2015	LSTM	-	-	58.2	-	70.8	59.4
	CNN-char	-	-	62.0	-	71.2	59.6
	CNN-word	-	-	60.5	-	71.2	57.6
Tang et al., 2015	Paragraph Vector	57.7	59.2	60.5	34.1	-	-
	CNN-word	59.7	61.0	61.5	37.6	-	-
	Conv-GRNN	63.7	65.5	66.0	42.5	-	-
	LSTM-GRNN	65.1	67.1	67.6	45.3	-	-
This paper	HN-AVE	67.0	69.3	69.9	47.8	75.2	62.9
	HN-MAX	66.9	69.3	70.1	48.2	75.2	62.9
	HN-ATT	**68.2**	**70.5**	**71.0**	**49.4**	**75.8**	**63.6**

Table 2: Document Classification, in percentage

ce: Yang, Zichao, et al., Hierarchical Attention Networks for Document Classification (2016).

Trends, extrapolations, strong claims

Trends

- less symbolic representations, more distributed
- less expert rules, more algebraic computations
- less classical logic, more fuzzy
- less preprocessing, more *from scratch* approaches (papers like: '*Natural Language Processing (Almost) from Scratch*' (2011), '*Text Understanding from Scratch*'(2015), '*QUality Estimation from ScraTCH (QUETCH): Deep Learning for Word-level Translation Quality Estimation*' (2015))
- less linguistic knowledge involved in model prototyping and training, more generic approaches
- less task-specific projects, more multi-task shared hidden knowledge systems
- less *designed-trained-done* projects, more *curriculum learning* systems
- more neural-based approaches more humanesque
- **less artificial**

Towards *Computational Intelligence*

- even nomenclature trend:
  - artificial intelligence does not fit well to complex deep neural language models
  - computational intelligence seems to work nicer here
- AI-approach tries to build intelligence based on our knowledge about a specific domain
- CI tries to build intelligence based on our knowledge about ourselves
- time to abandon artificial intelligence approach for NLP/NLU

Possible objections

- AI works great in other fields... (e.g. games)
- a sledgehammer to crack a nut?
- inferior performance?
- AI approach gives us full understanding and control
- all linguistic knowledge not important?

References

- Puurula, A., Read, J., & Bifet, A. (2014). Kaggle LSHTC4 winning solution. arXiv preprint arXiv:1405.0546.
- Mikolov, Tomas, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013).
- Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.
- Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems (pp. 649-657).
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.