

Text classification

Katya Artemova

Computational Pragmatics Lab, HSE

17 сентября 2019 г.

Today

Intro

Baselines

- Deep averaging network

Convolution neural networks

- Convolutional models of sentence pairs

How to improve your classifier?

- Data augmentation

- Distant and weak supervision

- Active learning

Sentiment analysis

1. **Task:** define expressed opinion of a text (negative, positive, neutral)
2. **Levels:**
 - ▶ classify the whole **document** (is a review positive or negative?)
 - ▶ does **a sentence** express negative or positive opinion? Does **a sentence** express an opinion?
 - ▶ identify what people like and dislike, extract specific aspects
3. **Challenges:** domain specific lexicons, sarcasm, negation, emoticons, abbreviations, slang and noisy user generated data

Today

Intro

Baselines

- Deep averaging network

Convolution neural networks

- Convolutional models of sentence pairs

How to improve your classifier?

- Data augmentation

- Distant and weak supervision

- Active learning

Deep averaging network [1]

Neural Bag-of-Words Model

1. **Task**: map an input sequence of tokens X to one of k labels
2. **Composition** function g averages word embeddings:

$$z = g(w \in X) = \frac{1}{|X|} \sum_{w \in X} v_w,$$

where v_w is a word embedding of word w

3. Estimate **probabilities** for each output label:
 $\hat{y} = \text{softmax}(W_s \times z + b)$ and **predict** the label with highest probability
4. **Training**: minimize cross-entropy error: $\sum_{p=1}^k y_p \log \hat{y}_p$

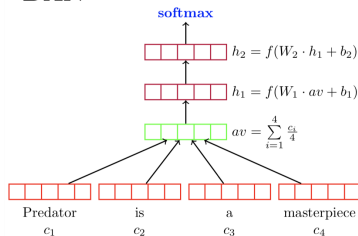
Deep averaging network [1]

The intuition is that each layer learns a more abstract representation of the input than the previous one. Add more layers:

$$z_i = g(z_{i-1}) = f(W_i \times z_{i-1} + b_i)$$

Word dropout: drop word tokens' entire word embeddings from the vector average

DAN



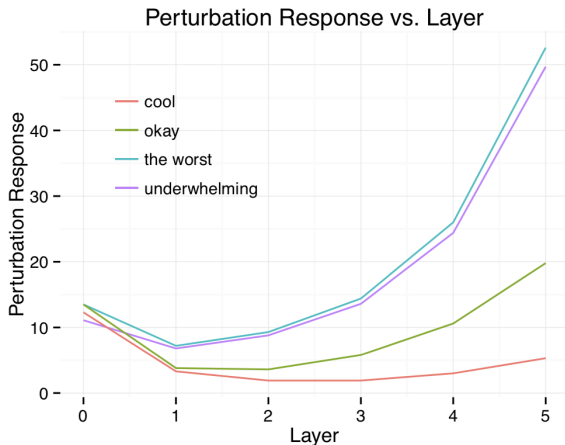
Deep averaging network [1]

Predictions of DAN on real (top) and synthetic (bottom) sentences that contain negations and contrastive conjunctions

Sentence	DAN	DRecNN	Ground Truth
a lousy movie that's not merely unwatchable, but also unlistenable	negative	negative	negative
if you're not a prepubescent girl, you'll be laughing at britney spears' movie-starring debut whenever it does n't have you impatiently squinting at your watch	negative	negative	negative
blessed with immense physical prowess he may well be, but ahola is simply not an actor	positive	neutral	negative
who knows what exactly godard is on about in this film, but his words and images do n't have to add up to mesmerize you.	positive	positive	positive
it's so good that its relentless, polished wit can withstand not only inept school productions, but even oliver parker's movie adaptation	negative	positive	positive
too bad, but thanks to some lovely comedic moments and several fine performances, it's not a total loss	negative	negative	positive
this movie was not good	negative	negative	negative
this movie was good	positive	positive	positive
this movie was bad	negative	negative	negative
the movie was not bad	negative	negative	positive

Deep averaging network [1]

Perturbation analysis: in the template “the film’s performances were awesome” replace the final word with increasingly negative polarity words (cool, okay, underwhelming, the worst)



Today

Intro

Baselines

Deep averaging network

Convolution neural networks

Convolutional models of sentence pairs

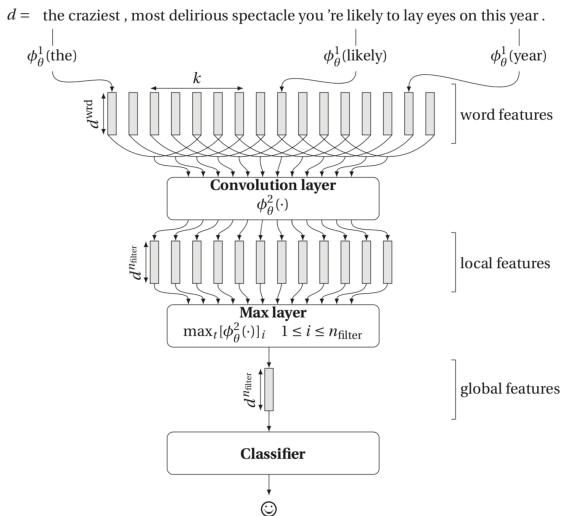
How to improve your classifier?

Data augmentation

Distant and weak supervision

Active learning

Convolution neural networks for text classification



Convolution neural networks for text classification

1. Embedding layer:

$$\phi_{\theta}^1(w_1, w_2, \dots, w_T) = (E_{w_1} E_{w_2}, \dots, E_{w_T}) \in \mathbb{R}^{d_{\text{wrd}} \times T}$$

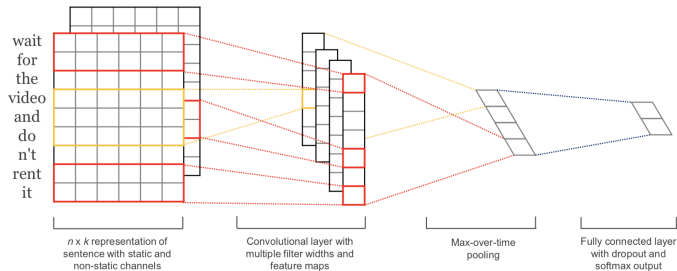
2. Convolutional layer:

$$\begin{aligned} \phi_{\theta}^2(w_t, \dots, w_{t+k}) &= W_2(W_1 \oplus (E_{w_t}, \dots, E_{w_{t+k}}) + b_1) + b_2, \\ \phi_{\theta}^2 &\in \mathbb{R}^{n_{\text{filter}}} - \text{filter, } k - \text{kernel (window) size,} \\ W_1 &\in \mathbb{R}^{n_h \times (kd_{\text{wrd}})}, W_2 \in \mathbb{R}^{n_{\text{filter}} \times n_h} \end{aligned}$$

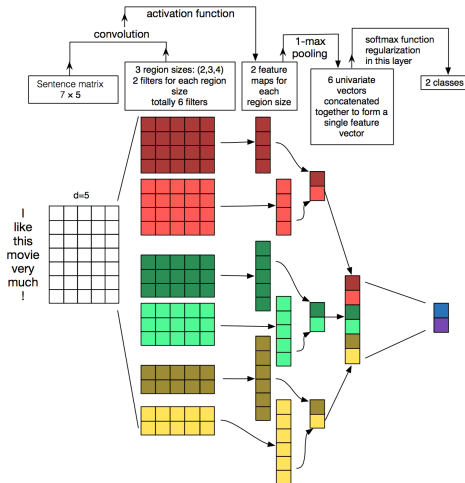
3. max pooling: feature-wise reduction

$$[\phi_{\theta}^3]_i = \max_t [\phi_{\theta}^2(\cdot)]_i$$

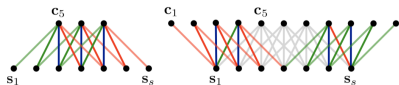
CNN for sentence classification [2]



CNN for sentence classification [3]



Narrow VS wide convolution



- ▶ $m \in \mathbb{R}^m$ - weights, $s \in \mathbb{R}^s$ - input sequence
- ▶ convolution:
$$c_j = m^T s_{j-m+1:j}$$
- ▶ narrow convolution: $s \geq m$,
 $c \in \mathbb{R}^{s-m+1}$, $j \in [m, s]$
- ▶ wide convolution:
 $c \in \mathbb{R}^{s+m-1}$,
 $j \in [1, s+m-1]$
- ▶ $s_i = 0, i < 1, i > s$

Convolutional models of sentence pairs

Why measure similarity between sentences?

1. Paraphrase identification
2. Duplicate detection
3. Textual entailment
4. Retrieval
5. Sentence completion
6. Question answering

Binary classification: given S_1 and S_2 , decide whether they mean the same or not

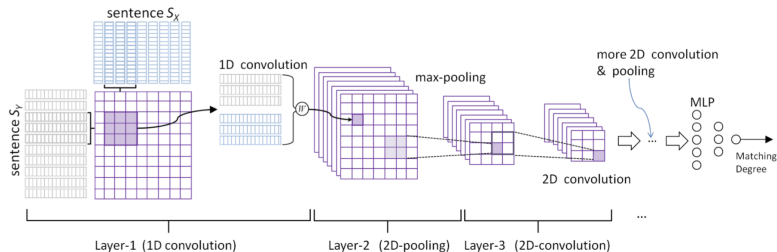
Convolutional matching model [4]

S_X, S_Y – sentences

$$z_{i,j}^{1,f}(x, y) = g(\hat{z}_{i,j}^0) \sigma(w^{l,f} \hat{z}_{i,j}^0 + b^{l,f})$$

$g(v) = 0$ if all the elements in v equals 0, otherwise $g(v) = 1$

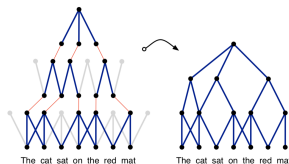
$$\hat{z}^0 = [x_{i:i+k_1-1}^T, y_{j:j+k_1-1}^T]^T$$



Dynamic Convolutional Neural Network [5]

Dynamic k -max pooling:

1. k -max pooling over a linear sequence of values returns the subsequence of k maximum values in the sequence. Secondly
2. the pooling parameter k can be dynamically chosen by making k a function of other aspects of the network or the input.



$$k_l = \max(k_{top}, \frac{L - l}{l} s)$$

l – the number of the current convolutional layer to which the pooling is applied

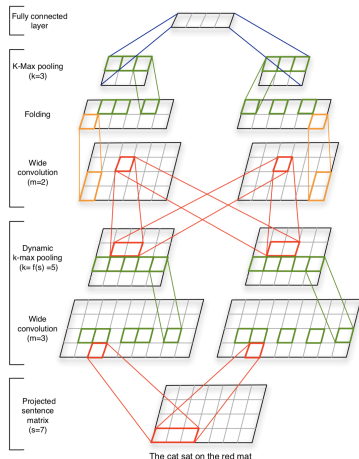
L – the total number of convolutional layers in the network

k_{top} – the fixed pooling parameter for the topmost convolutional layer

Dynamic Convolutional Neural Network [5]

Folding:

After a convolutional layer and before (dynamic) k -max pooling, one just sums every two rows in a feature map component-wise.



Today

Intro

Baselines

Deep averaging network

Convolution neural networks

Convolutional models of sentence pairs

How to improve your classifier?

Data augmentation

Distant and weak supervision

Active learning

SMOTE: Synthetic Minority Over-sampling Technique

The minority class is over-sampled by creating synthetic examples:

1. Take each minority class sample
2. Choose random samples k minority class nearest neighbors
3. take the difference between the feature vector (sample) under consideration and its nearest neighbor
4. multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration

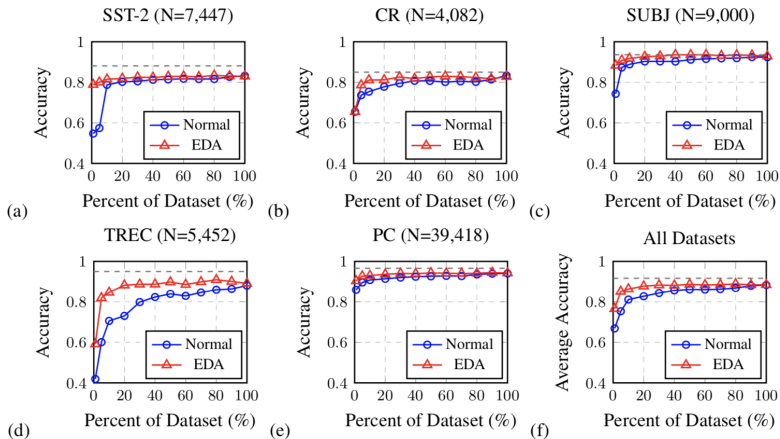
SMOTE is applied only to **feature vectors**, not raw texts!

Python: imbalanced-learn

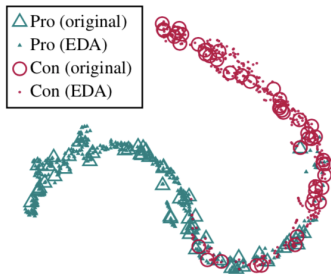
EDA: Easy Data Augmentation Techniques [6]

1. **Synonym Replacement (SR)**: Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.
2. **Random Insertion (RI)**: Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.
3. **Random Swap (RS)**: Randomly choose two words in the sentence and swap their positions. Do this n times.
4. **Random Deletion (RD)**: Randomly remove each word in the sentence with probability p .

EDA: Easy Data Augmentation Techniques [6]



EDA: Easy Data Augmentation Techniques [6]



Latent space visualization of original and augmented sentences in the Pro-Con dataset

Deep model pretraining [7], [8]

CNN for sentiment analysis

Preprocessing: URLs and usernames were substituted by a replacement token, the text was lowercased and finally tokenized

Creation of word embeddings: the word embeddings are learned on an unsupervised corpus containing 300M tweets

Distant-supervised phase: use emoticons to infer noisy labels on tweets in the training set

Supervised phase: the network is trained on the supervised training data.

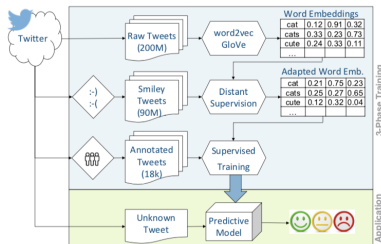
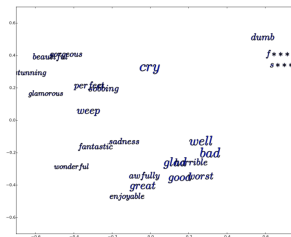


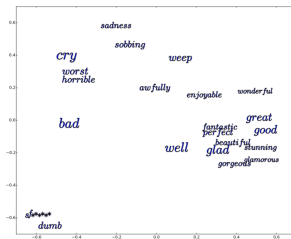
Figure 2: Training Phases Overview.

Deep model pretraining [7], [8]

Word embeddings: before and after



(a)



AL for text classification with CNNs

Pool-based AL scenario:

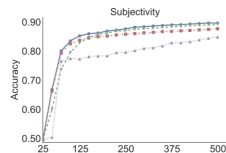
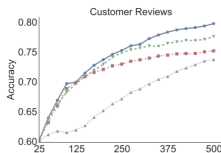
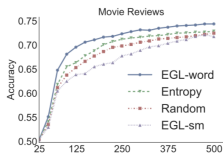
1. L – labelled data, U – unlabeled data, $|L| \ll |U|$
2. Train on L , make queries to U to draw examples to be labeled
3. Query strategy: $x^* = \arg \max_{x_i \in U} \phi(x_i; \theta)$
4. Sampling strategy:
 - ▶ Random sampling
 - ▶ Uncertainty sampling:

$$-\sum_k P(y_i = k | x_i, \theta) \log P(y_i = k | x_i, \theta)$$

- ▶ Expected gradient length:

$$\max_{i \in x_i} P(y_i = k | x_i, \theta) \|\nabla J_{E(U)}(< x_i, y_i = k >; \theta)\|$$

AL for text classification with CNNs







number of labeled examples versus accuracy




Reading

1. Speech and Language Processing. Daniel Jurafsky, James H. Martin, Ch. 4 [\[url\]](#)
2. Natural Language Processing. Jacob Eisenstein, Ch. 2-4, [\[\[GitHub\]](#)
3. Neural Networks for NLP. Jacob Eisenstein, Ch. 15

Reference I

-  M. Iyyer, V. Manjunatha, J. Boyd-Graber и H. Daumé III, “Deep Unordered Composition Rivals Syntactic Methods for Text Classification”, , 2015.
-  Y. Kim, “Convolutional neural networks for sentence classification”, *arXiv preprint arXiv:1408.5882*, 2014.
-  Y. Zhang и B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification”, *arXiv preprint arXiv:1510.03820*, 2015.
-  B. Hu, Z. Lu, H. Li и Q. Chen, “Convolutional neural network architectures for matching natural language sentences”, В *Advances in neural information processing systems*, 2014, с. 2042—2050.

Reference II

-  N. Kalchbrenner, E. Grefenstette и P. Blunsom, “A convolutional neural network for modelling sentences”, *arXiv preprint arXiv:1404.2188*, 2014.
-  J. W. Wei и K. Zou, “Eda: Easy data augmentation techniques for boosting performance on text classification tasks”, *arXiv preprint arXiv:1901.11196*, 2019.
-  A. Severyn и A. Moschitti, “Unitn: Training deep convolutional neural network for twitter sentiment classification”, в *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, с. 464—469.

Reference III



J. Deriu, A. Lucchi, V. De Luca, A. Severyn, S. Müller, M. Cieliebak, T. Hofmann и M. Jaggi, “Leveraging large amounts of weakly supervised data for multi-language sentiment classification”, в *Proceedings of the 26th international conference on world wide web*, International World Wide Web Conferences Steering Committee, 2017, c. 1045—1052.