Text classification

Computational Pragmatics Lab, HSE

February 12, 2020

Today

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

Intro

1. Always use fasttext as a strong baseline!
2. Know your data: explore the data, look for domain-specific features
3. Pretrain your models on noisy datasets
4. Augment labelled datasets if the classes are inbalanced
5. Use active learning if you want to annotated data

Today

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:
   $\hat{y} = \mathrm{softmax}(W_s \times z + b)$ 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

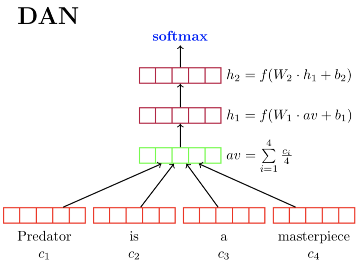


Figure: Deep Averaging Network

Deep averaging network [1]

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

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

Deep averaging network [1]

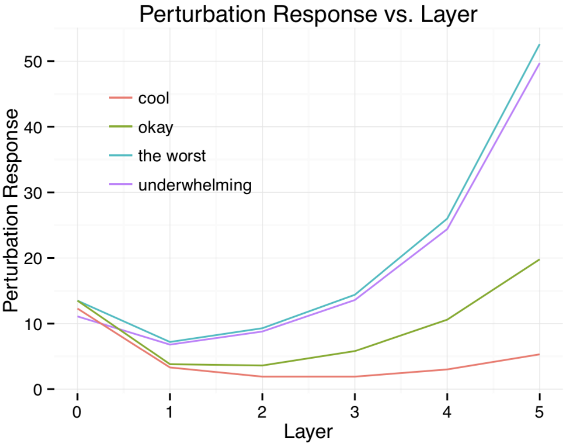


Figure: **Perturbation analysis**: in the template "the film's performances were awesome" replace the final word with increasingly negative polarity words (cool, okay, underwhelming, the worst)

Today

Convolution neural networks for text classification

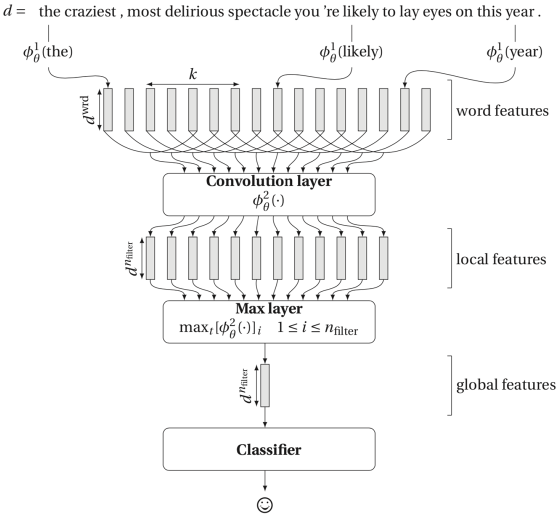


Figure: CNN for text classification

Image source: P.R. Lebret. Word Embeddings for Natural Language Processing.

Convolution neural networks for text classification

1. **Embedding layer**:
   $\phi^1_\theta(w_1, w_2, \dots, , w_T) = (E_{w_1} E_{w_2}, \dots, E_{w_T}) \in \mathbb{R}^{d_{wrd} \times T}$
2. **Convolutional layer**:
   $\phi^2_\theta(w_t, \dots, w_{t+k}) = W_2(W_1 \oplus (E_{w_t}, \dots, E_{w_{t+k}}) + b_1) + b_2),$
   $\phi^2_\theta \in \mathbb{R}^{n_{filter}}$ – filter, k — kernel (window) size,
   $W_1 \in \mathbb{R}^{n_h \times (kd_{wrd})}$, $W_2 \in \mathbb{R}^{n_{filter} \times n_h}$
3. **max pooling**: feature-wise reduction
   $[\phi^3_\theta]_i = \max_t [\phi^2_\theta(\cdot)]_i$

CNN for sentence classification [2]

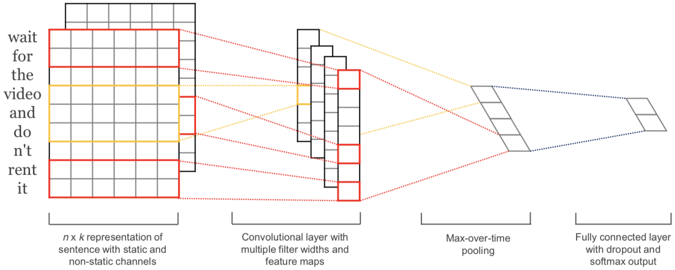


Figure: CNN with two channels for text classification

CNN for sentence classification [3]

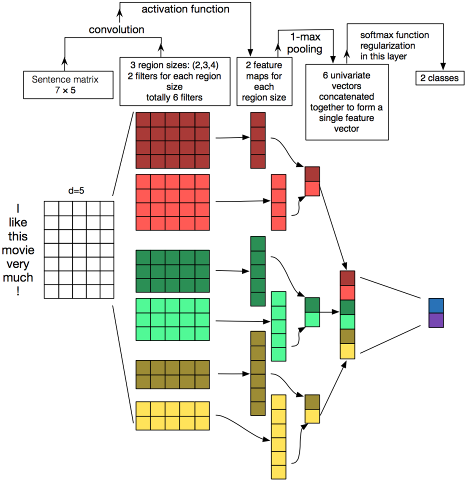


Figure: CNN with multiple channels for text classification

Narrow VS wide convolution

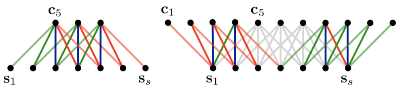


Figure: Narrow VS wide convolution

- $m \in \mathbb{R}^m$ - 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: $c \in \mathbb{R}^{s+m-1}$, $j \in [1, s+m-1]$
- $s_i = 0, i < 1, i > s$

Dynamic Convolutional Neural Network [4]

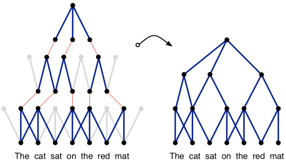


Figure: Dynamic k-max pooling

Dynamic k-max pooling:

1. k-max pooling over a linear sequence of values returns the subsequence of k maximum values in the sequence
2. The pooling parameter k can be dynamically chosen

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

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 [4]

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

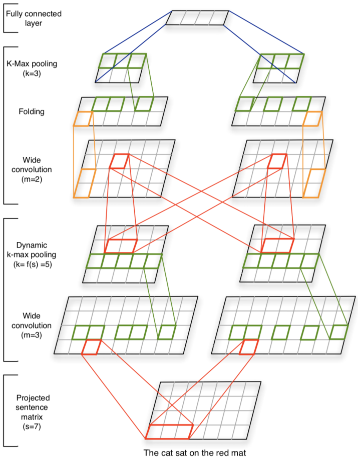


Figure: CNN with dynamic k-max pooling and folding

Convolutional models of sentence pairs

Why measure similarity between sentences?

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

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

Convolutional matching model [5]

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

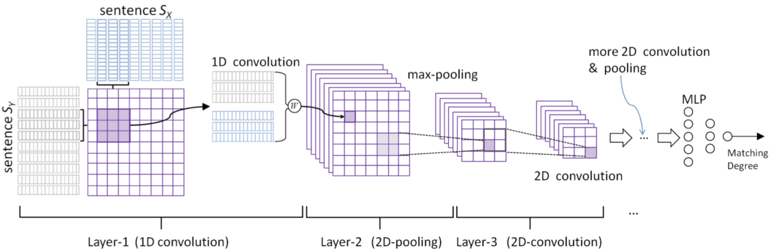


Figure: CNN for sentence matching

Bi-CNN-MI [6]

A1: Detroit manufacturers have raised vehicle prices by ten percent
A2: GM, Ford and Chrysler have raised car prices by five percent

B1: Mary gave birth to a son in 2000
B2: He is 18 years old and his mother is Mary

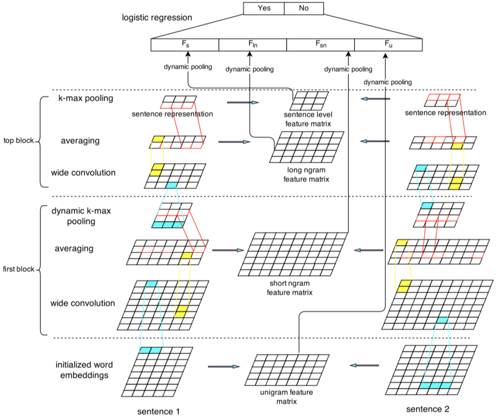


Figure: Siamise CNN for sentence matching

Bi-CNN-MI [6]

1. "Bi-CNN" – double CNNs used in Siamese framework,
2. "MI" for multigranular interaction features
3. Bi-CNN-MI has three parts:
   - the sentence analysis network CNN-SM
   - the sentence interaction model CNN-IM
   - a logistic regression on top of the network that performs paraphrase identification

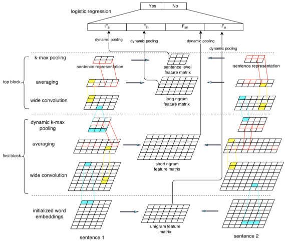


Figure: Siamise CNN for sentence matching

Bi-CNN-MI [6]

Convolution sentence model CNN-SM:

1. wide one-dimensional convolution: $C \in \mathbb{R}^{d \times |S_i| + m - 1)}$, S_i – the number of tokens, m – the filter width, d – the embedding size
2. folding: each odd row and the row behind are averaged
3. dynamic k-max pooling: $k = \max(k_{top}, |S_i|/2 + 1)$

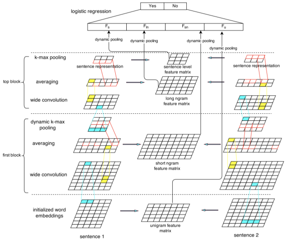


Figure: Siamise CNN for sentence matching

Bi-CNN-MI [6]

Convolution interaction model CNN-IM:

1. $\hat{F}_{l}^{ij} = \exp(\frac{-||S^1_{:,i} - S^2_{:,j}||^2}{2\beta})$
   S denotes the matrix representing sentence

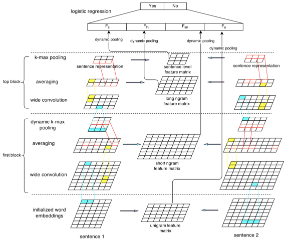


Figure: Siamise CNN for sentence matching

Bi-CNN-MI [6]

Unsupervised pretraining: CNN-LM is used to pretrain convolutional filters

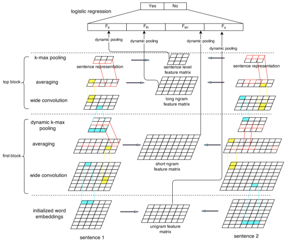


Figure: Siamise CNN for sentence matching

Today

SMOTE: Synthetic Minority Over-sampling Technique

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

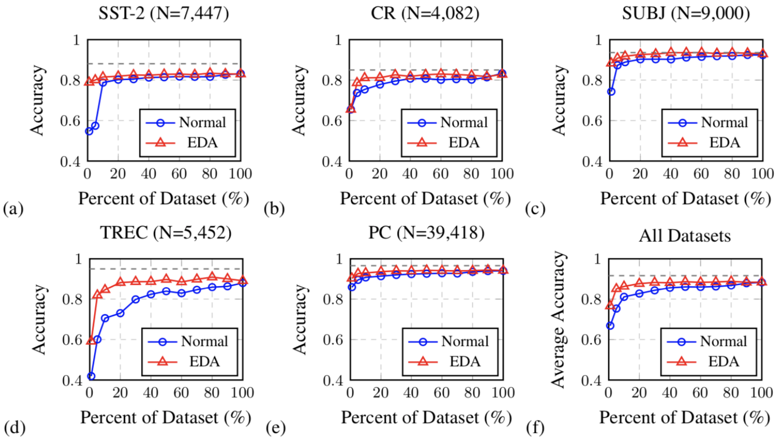
1. Take each minority class sample
2. Choose random samples k minority class nearest neighbors
3. take the difference between the feature vector (sample) under consideration and its nearest neighbor
4. multiply this difference by a random number between 0 and 1, and add it to the feature vector under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

SMOTE is applied only to **feature vectors**, not raw texts!
Python: imbalanced-learn

EDA: Easy Data Augmentation Techniques [7]

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 [7]



EDA: Easy Data Augmentation Techniques [7]

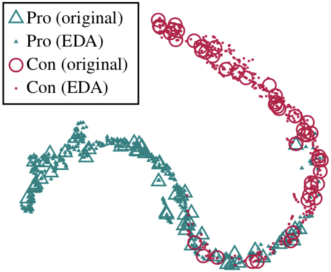


Figure: Latent space visualization of original and augmented sentences in the Pro-Con dataset

Deep model pretraining [8], [9]

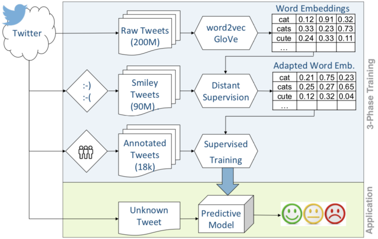
CNN for sentiment analysis

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

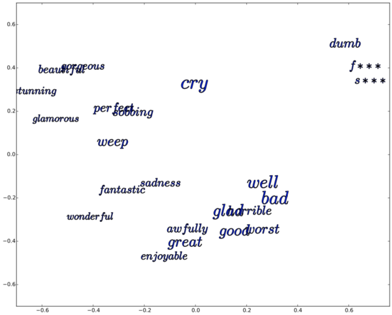
Distant-supervised phase: use emoticons to infer noisy labels on tweets in the training set

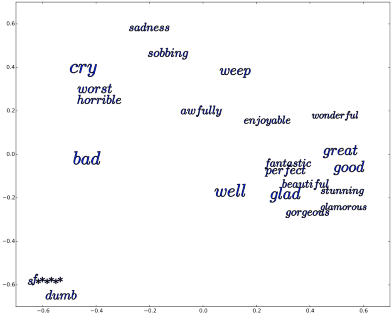
Supervised phase: the network is trained on the supervised training data.



Deep model pretraining [8], [9]

Word embeddings: before and after





AL for text classification with CNNs [10]

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:

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

AL for text classification with CNNs [10]

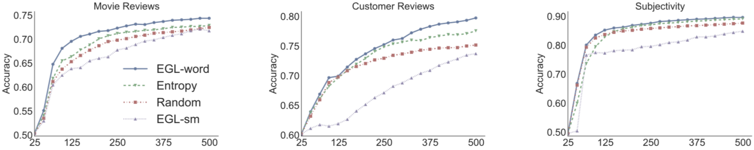


Figure: Number of labeled examples versus accuracy

Conclusions

1. Basic architectures can be improved with pretraining and distant supervision
2. Think of your data and the ways to augment it
3. We really need good representations and metrics for the datasets and tasks

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. Joav Goldberg, Ch. 15

Reference I

M. Iyyer, V. Manjunatha, J. Boyd-Graber, and 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 and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

Reference II

- B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Advances in neural information processing systems*, 2014, pp. 2042–2050.
- W. Yin and H. Schütze, "Convolutional neural network for paraphrase identification," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 901–911.
- J. W. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

Reference III

A. Severyn and A. Moschitti, "Unitn: Training deep convolutional neural network for twitter sentiment classification," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 2015, pp. 464–469.

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

Y. Zhang, M. Lease, and B. C. Wallace, "Active discriminative text representation learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.