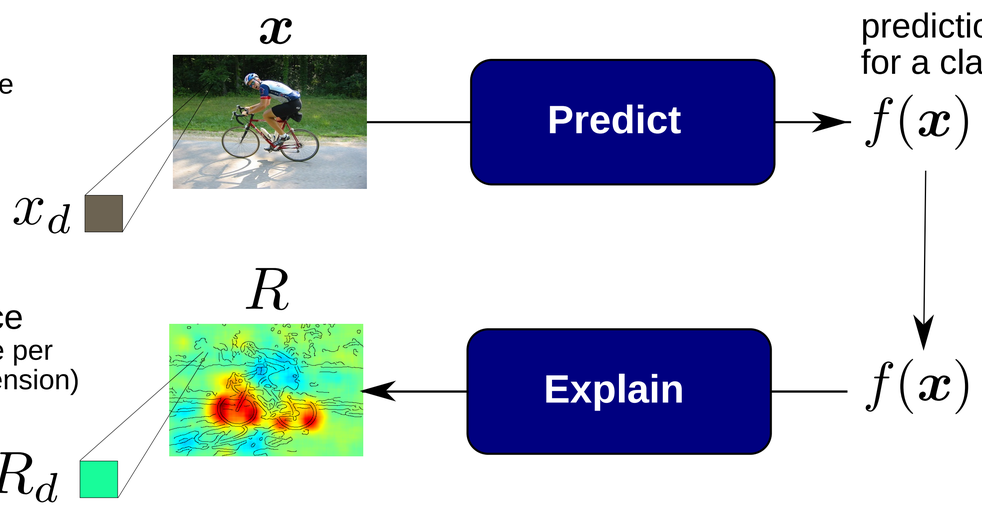
Explaining Predictions of Non-Linear Classifiers in NLP

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek

Fraunhofer Heinrich Hertz Institute — Technische Universität Berlin

Explaining Classifier Predictions

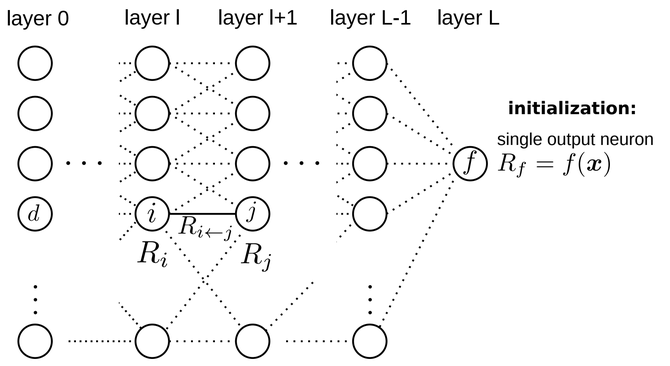


Explaining Neural Network Predictions

LRP

Layer-wise Relevance Propagation

[recently proposed by Bach et al. 2015]



initialization:
single output neuron
$R_f = f(\boldsymbol{x})$

Linear layer $z_{ij} = x_i w_{ij} + \frac{b_j}{n}$ n : receptive field of neuron j

$z_j = \sum_i z_{ij}$

$s(z_j) = \epsilon \cdot (1_{z_j \geq 0} - 1_{z_j < 0})$ stabilizing term

message proportional to neuron contribution in forward pass

$R_{i\leftarrow j} = \frac{z_{ij} + \frac{s(z_j)}{n}}{\sum_i z_{ij} + s(z_j)} R_j$ message

$R_i = \sum_j R_{i\leftarrow j}$ sum of incoming messages

Activation layer $R_i = R_j$

non-linear activation is taken into account by next linear layer

Max-pooling layer $R_i = \begin{cases} R_j & i = \arg\max_{i'} x_{i'} \\ 0 & \text{else} \end{cases}$

Properties:

(1) layer-wise conservation of score:

$\sum_d R_d = ... = \sum_i R_i^{(l)} = \sum_j R_j^{(l+1)} = ... = f(\boldsymbol{x})$

(2) signed relevance: indicate input regions that support or inhibit a specific classification decision

(3) relevance indicates contribution of input dim to actual classification decision (static)

SA

Sensitivity Analysis
or Gradient-Magnitude

[e.g. Dimopoulos et al. 1995, Denil et al. 2014]

$R_d = \left(\frac{\partial f}{\partial x_d}\right)^2$ Backpropagation

Relevance aggregation: for both LRP and SA by summation

Word-level Relevance w.r.t. our CNN Model:

$R(\boldsymbol{w_t}) = \sum_{i=1}^{D} R_{i,t}$

$\boldsymbol{w_t}$ t^{th} input word

D word-embedding dimension

Properties:

(1) conservation of squared gradient norm:

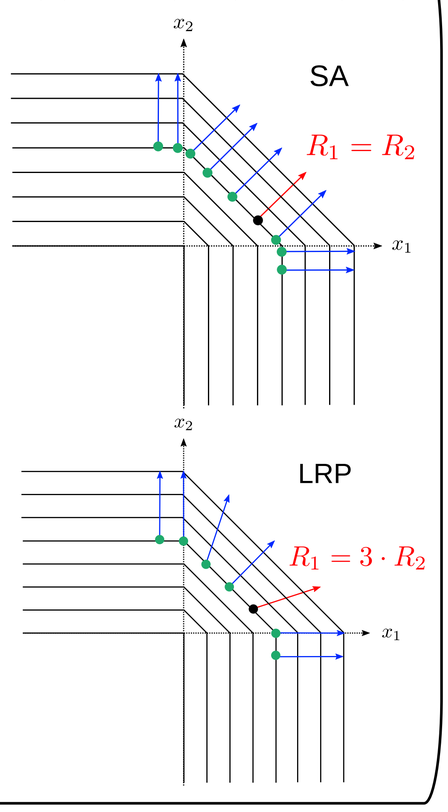
$\sum_d R_d = \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x})\|_2^2$

(2) positive relevance: does not discriminate between input regions that support or inhibit a specific classification decision

(3) relevance indicates sensitivity of classification decision to changes in the input dimensions (dynamic)

Intuition

$f(\boldsymbol{x}) = \max(0, x_1) + \max(0, x_2)$



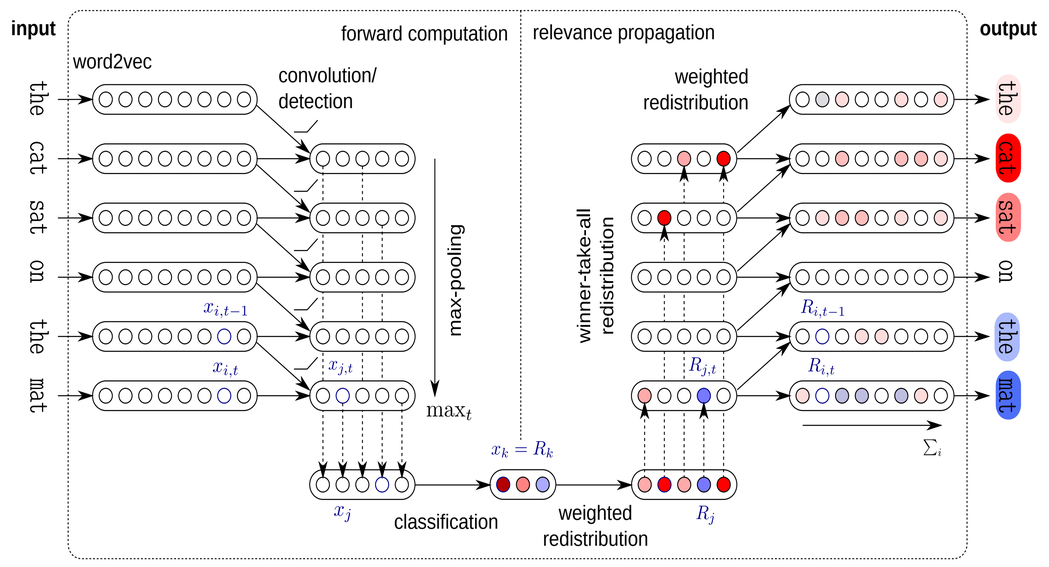
Experimental Setup

Model: word-based CNN [similar to Kim 2014]

`Conv` → `ReLU` → `1-max-Pool` → `FC`

horizontal concat. of D-dim pre-trained word-embeddings $(\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_N})$ as input

1-dim convolutional operation covering the entire word-embedding space



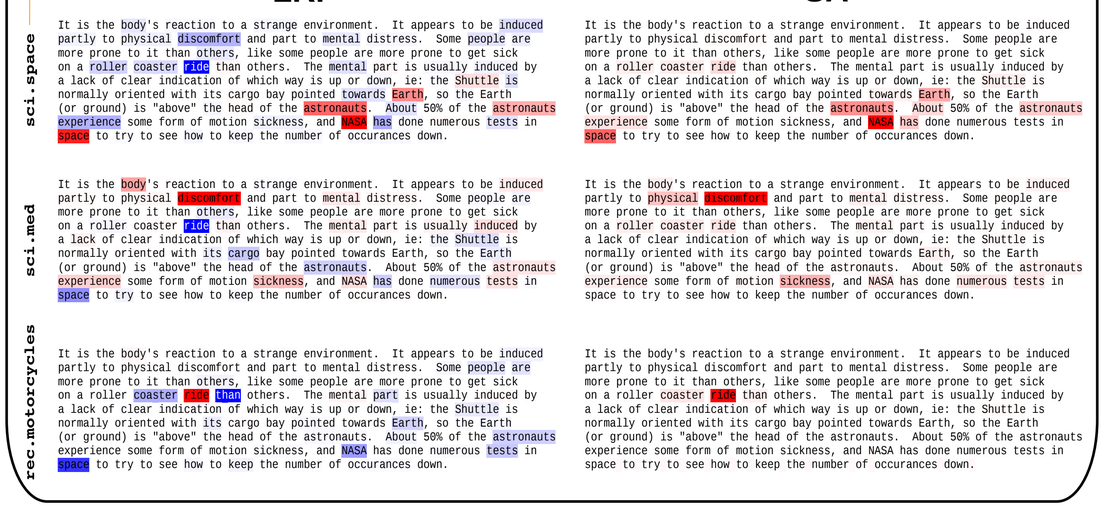
Task: Document classification

20Newsgroups dataset (11314/7532 train/test documents)
newsposts subdivided into 20 fine-grained categories

Document Highlighting

target class for the LRP/SA relevance estimation

visualize word-level relevance:
map positive relevance to red, negative to blue
normalize opacity to extremal relevance per document



Quantitative Eval.: Word Deleting

e.g. word deleting by decreasing LRP relevance:

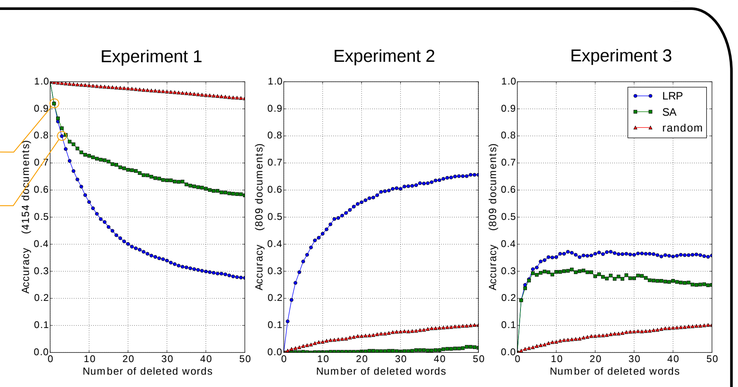
original document the cat sat on the mat

1 deleted word the ◯ sat on the mat

2 deleted words the ◯ ◯ on the mat

3 deleted words ◯ ◯ ◯ on the mat

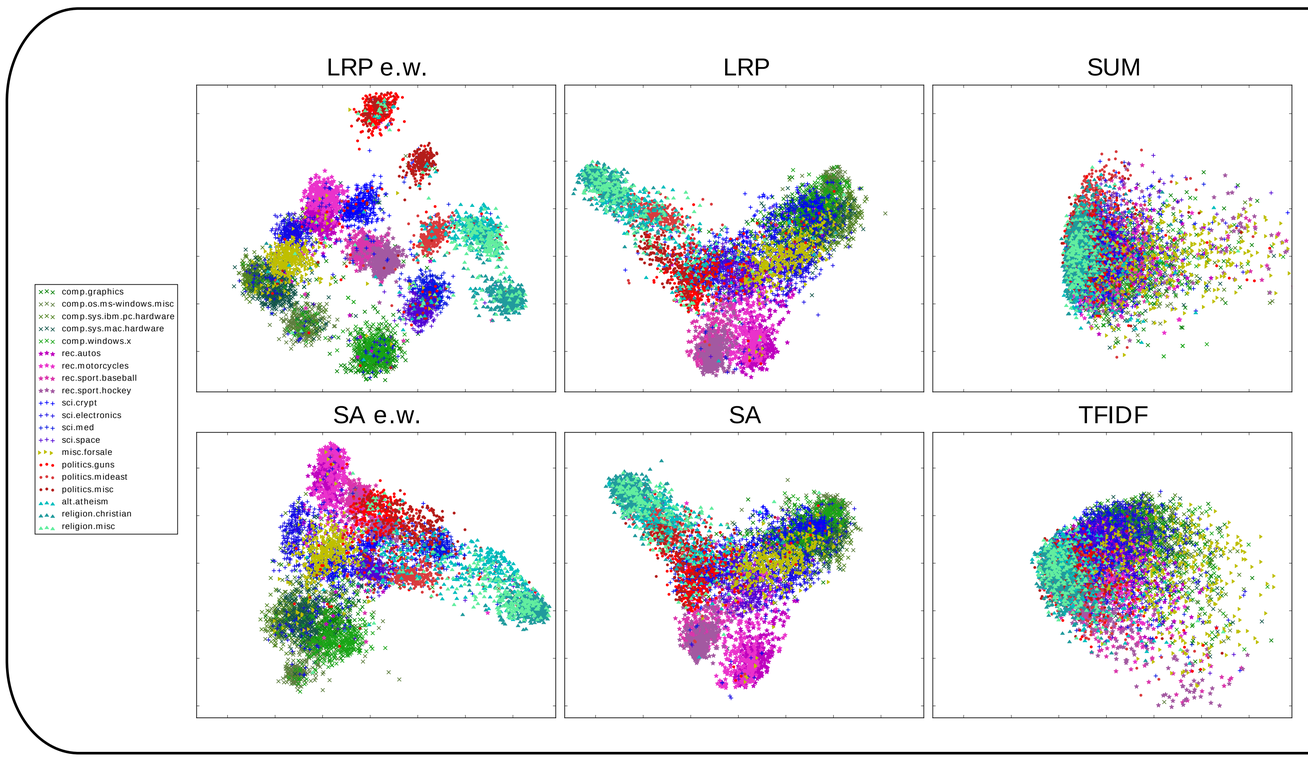
Idea:
delete a sequence of words accordingly to their relevance and track the impact on the classification performance



deletion = word-vector set to zero in the input document	start with documents* initially:	correct classified	false classified	false classified
	LRP/SA target class:	true class	true class	predicted class
	deletion order:	decreasing relevance	increasing relevance	decreasing relevance

* considering only test documents with a length greater or equal to 100 words

From word-vectors to document-vectors

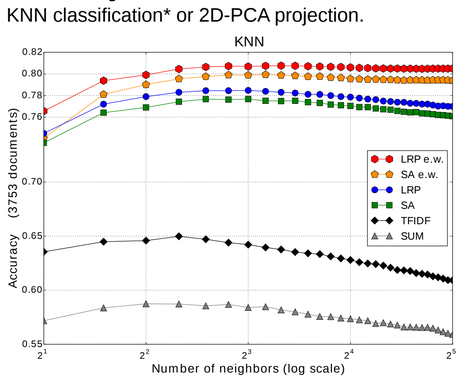


1) Represent documents as a linear comb. of word2vec vectors, using as a weighting scheme LRP/SA relevance, TFIDF scores or uniform weighting (SUM). As target class for the LRP/SA relevance estimation we use the predicted class of the CNN Model.

$\boldsymbol{d} = \sum_{t=1}^{N} R(\boldsymbol{w_t}) \cdot \boldsymbol{w_t}$ weighting based on word-level relevance

$\boldsymbol{d_{\text{e.w.}}} = \sum_{t=1}^{N} \begin{bmatrix} R_{1,t} \\ R_{2,t} \\ \vdots \\ R_{D,t} \end{bmatrix} \odot \boldsymbol{w_t}$ weighting based on single input dimension relevances (e.w. stands for the element-wise operation $\odot$)

2) Normalize the resulting document vectors to unit euclidean norm, and perform KNN classification* or 2D-PCA projection.



* for KNN classification halft of the test documents are used as neighbors, the other half are used for evaluating the KNN performance

More details on LRP www.heatmapping.org

[1] Bach et al., On Pixel-wise Explanations for Non-Linear Classifier Decisions by Layer-wise Relevance Propagation, PLOS ONE, 2015
[2] Samek et al., Evaluating the visualization of what a Deep Neural Network has learned, arXiv:1509.06321, 2015
[3] Montavon et al. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, arXiv:1512.02479 , 2015
[4] Lapuschkin et al., The Layer-wise Relevance Propagation Toolbox for Artifical Neural Networks, JMLR, 2016
[5] Bach et al., Analyzing Classifiers: Fisher Vectors and Deep Neural Networks, CVPR, 2016