Natural Language Processing with Deep Learning

Lecture 2 — Evaluation and machine leaning basics

Prof. Dr. Ivan Habernal

October 23, 2024

www.trusthlt.org
Trustworthy Human Language Technologies Group (TrustHLT)
Ruhr University Bochum & Research Center Trustworthy Data Science and Security

This lecture

Basics on supervised machine learning

- Train/dev/test split
- Evaluation
- Loss functions

Learning goals

- Understand ML/DL foundations

Notation

1 Notation

2 Supervised Machine Learning basics

3 Evaluation

Notation

Vectors in linear algebra are columns, for example $\mathbf{x} \in \mathbb{R}^3$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \qquad \text{(bold face, lower case)}$$

We treat them as a row vector by transposing, for example $\mathbf{x}^\intercal = (x_1, x_2, x_3)$ — which is a matrix $\mathbb{R}^{1 \times 3}$

Caveat: 1-D array (a list of numbers) is sometimes considered a vector, so dealing with dimensions might be quite messy

Notation

Matrices are upper-case bold, for example $\mathbf{Z} \in \mathbb{R}^{2 \times 3}$

$$\mathbf{Z} = \begin{pmatrix} z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,1} & z_{2,2} & z_{2,3} \end{pmatrix}$$

Scalars are ordinary lower case letters, for example

$$a, b, c \in \mathbb{R}$$

Notation ambiguity

A dot $\cdot$ means multiple things, depending on context

Simple scalar multiplication, for example $a \cdot b$

$$\cdot : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$$

Dot product $\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i$

$$\cdot : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

Matrix-matrix (matrix-vector/vector-matrix) multiplication, for example $\mathbf{x} \cdot \mathbf{W}$ or $\mathbf{Y} \cdot \mathbf{Z}$

$$\cdot : \mathbb{R}^{m \times n} \times \mathbb{R}^{n \times p} \to \mathbb{R}^{m \times p}$$

Supervised Machine Learning basics

1 Notation

2 Supervised Machine Learning basics

3 Evaluation

Problem setup

We have N labeled **data points** (or examples) as tuples

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

where y_n is the "truth" or "gold label" of $\mathbf{x}_n$.

We have a **model** parametrized by θ that outputs $\hat{y}$

$$\hat{y} = f_\theta(\mathbf{x})$$

We specify a **loss function**, for example

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - f_\theta(\mathbf{x}_i))^2$$

Learning

Our goal is to find such parameters θ that minimize the loss

- In other words, our model “fits” the training data “better”

Overfitting

If our model is sufficiently ”rich” (huge number of parameters), it could minimize the loss by **remembering** our training data perfectly → Not the goal of learning!

Generalization

The actual goal of machine learning is to **generalize** well on previously unseen data

Evaluating generalization?

- Split dataset into training and test
- Models must perform well on test data (hidden during learning)

ML Basics

Three major components of a machine learning system

1. Data
2. Models
3. Learning

Supervised Machine Learning basics

Data

Supervised learning problem: Data

Dataset is a set of input-label tuples (labeled examples)

$$\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), \dots, (\mathbf{x}_N, y_N)\}$$

- Each input $\mathbf{x}_n$ is a D-dimensional vector of real numbers, which are called features, attributes, or covariates
- Label y_n associated with input vector $\mathbf{x}_n$

Independent and identically distributed

Assumption: Our dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ is **Independent and identically distributed (I.I.D)**

- Two data points $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ do not statistically depend on each other

Supervised Machine Learning basics

Model

Models as functions

Predictor: a function from features to output

$$f : \mathbb{R}^D \rightarrow \mathbb{R}$$

In classification we typically predict a probability distribution over categories, e.g.,

$$f : \mathbb{R}^D \rightarrow \mathbb{R}^{|C|}$$

$|C|$ — number of classes and arbitrary mapping, e.g.

$$C = \begin{cases} 0 & \text{Sport} \\ 1 & \text{Politics} \\ 2 & \text{Business} \end{cases}$$

Models as functions

For example

$$C = \begin{cases} 0 & \text{Sport} \\ 1 & \text{Politics} \\ 2 & \text{Business} \end{cases}$$

$$f(\mathbf{x}) \to \underbrace{(0.01, 0.82, 0.17)}_{\sum = 1.0}$$

Learning is finding ‘the best’ parameters

The goal of learning is to

- find a model and its corresponding parameters
- the resulting predictor should perform well on unseen data

Conceptually three distinct phases

1. Prediction or inference
2. Training or parameter estimation
3. Hyperparameter tuning or model selection

Hypothesis class of functions

Supervised learning on dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$; $x_n \in \mathbb{R}^D$

Estimate a predictor parametrized by θ

$$f(\cdot, \theta) : \mathbb{R}^D \to \mathbb{R}$$

We hope to "find" "good" parameters θ^* so that we "fit" the data well

$$f(\mathbf{x}_n, \theta^*) \approx y_n \qquad \text{for all } n = 1, \ldots, N$$

Notation: let $\hat{y_n} = f(\mathbf{x}_n, \theta^*)$ represent predictor's output

Loss function for training

What does it mean to fit the data "well"?

We need to specify a **loss function**

$$\ell(\underbrace{y_n}_{\text{True label}}, \underbrace{\hat{y}_n}_{\text{Predictor's output}}) \to \underbrace{\mathbb{R}^+}_{\text{"Loss"}}$$

representing 'how big' an error we made on this particular prediction

Our goal for finding a good parameter vector θ^* is to **minimize the average loss** on the set of N training examples

Loss example: Squared Loss

$$\ell(y_n, \hat{y_n}) = (y_n - \hat{y_n})^2$$

Minimizing so-called ‘empirical risk’

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}, \theta))^2$$

Generalization

We're interested in generalization performance, not how predictor works on training data

We always split our data:

- The **training set** is used to fit the model
- The **test set** is used to evaluate generalization performance

Warning: Hazard!! Test data leakage!!

Test set not seen by the machine learning algorithm during training

Evaluation

1 Notation

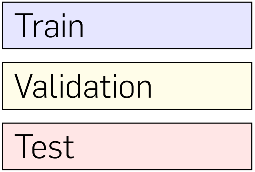
2 Supervised Machine Learning basics

3 Evaluation

Train/Dev/Test data splits

Training and Test data

Development (Validation) set used for optimizing hyper-parameters



Train

Validation

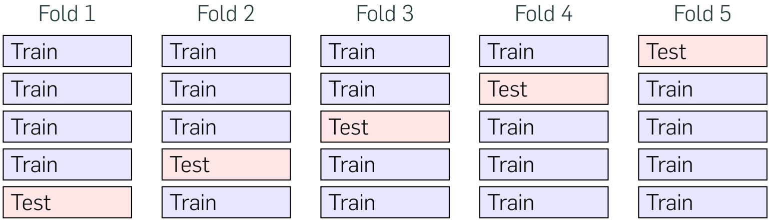
Test

Cross validation

K-fold cross-validation partitions the data into K chunks

$K - 1$ of which form the training set $\mathcal{R}$

The last chunk serves as the test set $\mathcal{V}$ (or validation)



Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Train	Train	Train	Train	Test
Train	Train	Train	Test	Train
Train	Train	Test	Train	Train
Train	Test	Train	Train	Train
Test	Train	Train	Train	Train

Figure 1: Example of 5-fold CV

Evaluation

Evaluation of text classification

Confusion matrix (binary case)

Two classes: Positive and Negative

Confusion matrix

	Pred. Negative	Pred. Positive
Act. Negative	True negative (TN)	False positive (FP)
Act. Positive	False negative (FN)	True positive (TP)

Act. Negative = Actually negative = Gold label

Ordering of columns and rows is **arbitrary**!

Accuracy

Accuracy of classifier f on test set T:

$$\mathrm{Acc}_T(f) = \frac{1}{|T|} \sum_{i=1}^{|T|} I(f(x_i) = y_i)$$

N. Japkowicz and M. Shah (2011). *Evaluating Learning Algorithms: A Classification Perspective*. (not freely available). Cambridge University Press

Example (Disease detection)

	Pred. Negative	Pred. Positive
Act. Negative	168	33
Act. Positive	48	37

$37 + 48 + 33 + 168 = 286 \rightarrow$ Test set size $|T| = 286$

$\mathrm{Acc}_T(f) = \frac{1}{286}(37 + 168) = 0.7186$

Precision, recall, F-1 score

Confusion matrix

	Pred. Negative	Pred. Positive
Act. Negative	True negative (TN)	False positive (FP)
Act. Positive	False negative (FN)	True positive (TP)

Precision (for class positive) = TP / (TP + FP)

Recall (for class positive) = TP / (TP + FN)

F-1 score (for class positive) = 2PR / (P + R)

Confusion matrix – multi-class

true class:	prediction:	*money-fx*	*trade*	*interest*	*wheat*	*corn*	*grain*
money-fx		95	0	10	0	0	0
trade		1	1	90	0	1	0
interest		13	0	0	0	0	0
wheat		0	0	1	34	3	7
corn		1	0	2	13	26	5
grain		0	0	2	14	5	10

Confusion matrix — multi-class

M. Sokolova and G. Lapalme (2009). “A systematic analysis of performance measures for classification tasks”. In: *Information Processing and Management* 45.4, pp. 427–437

- We can unambiguously compute Precision and Recall for each class
- How to get the F-1 score for the complete test set across classes?
  - Macro-averaging (average of F-1 scores), or micro-averaging
  - These details might get tricky so always report exactly what you do!

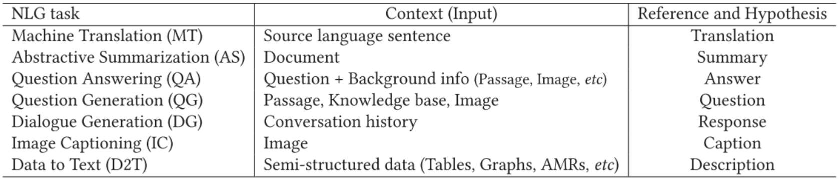
Evaluation

Evaluation of text generation

More text generation tasks

A. B. Sai, A. K. Mohankumar, and M. M. Khapra (2023). "A Survey of Evaluation Metrics Used for NLG Systems". In: *ACM Computing Surveys* 55.2, pp. 1–39

Table 2. Context and Reference/Hypothesis Forms for Each NLG Task



NLG task	Context (Input)	Reference and Hypothesis
Machine Translation (MT)	Source language sentence	Translation
Abstractive Summarization (AS)	Document	Summary
Question Answering (QA)	Question + Background info (Passage, Image, *etc*)	Answer
Question Generation (QG)	Passage, Knowledge base, Image	Question
Dialogue Generation (DG)	Conversation history	Response
Image Captioning (IC)	Image	Caption
Data to Text (D2T)	Semi-structured data (Tables, Graphs, AMRs, *etc*)	Description

Evaluating text generation is hard

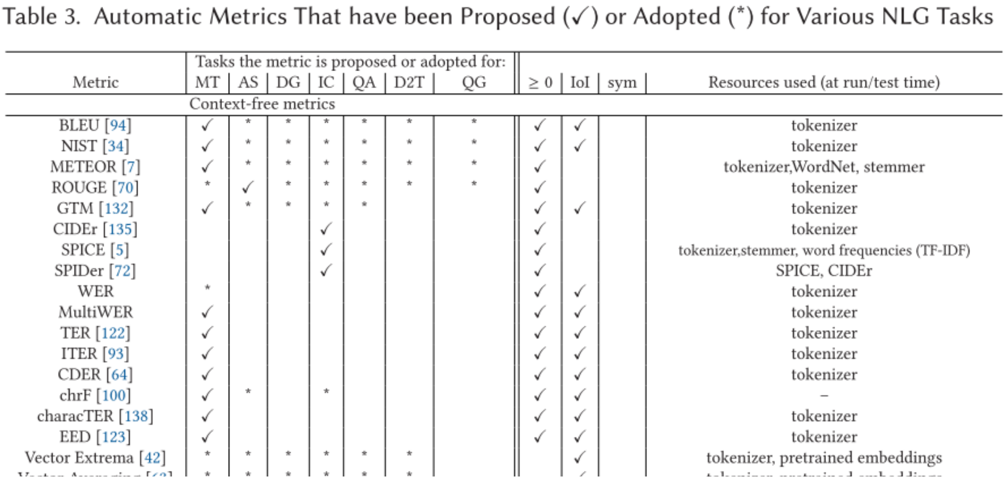


Table 3. Automatic Metrics That have been Proposed (✓) or Adopted (*) for Various NLG Tasks

Metric	Tasks the metric is proposed or adopted for:										
	MT	AS	DG	IC	QA	D2T	QG	≥ 0	IoI	sym	Resources used (at run/test time)
Context-free metrics											
BLEU [94]	✓	*	*	*	*	*	*	✓	✓		tokenizer
NIST [34]	✓	*	*	*	*	*	*	✓	✓		tokenizer
METEOR [7]	✓	*	*	*	*	*	*	✓			tokenizer,WordNet, stemmer
ROUGE [70]	*	✓	*	*	*	*	*	✓			tokenizer
GTM [132]	✓	*	*	*	*			✓	✓		tokenizer
CIDEr [135]				✓				✓			tokenizer
SPICE [5]				✓				✓			tokenizer,stemmer, word frequencies (TF-IDF)
SPIDer [72]				✓				✓			SPICE, CIDEr
WER	*							✓	✓		tokenizer
MultiWER	✓							✓	✓		tokenizer
TER [122]	✓							✓	✓		tokenizer
ITER [93]	✓							✓	✓		tokenizer
CDER [64]	✓							✓	✓		tokenizer
chrF [100]	✓	*		*				✓	✓		–
characTER [138]	✓							✓	✓		tokenizer
EED [123]	✓							✓	✓		tokenizer
Vector Extrema [42]	*	*	*	*	*	*			✓		tokenizer, pretrained embeddings
Vector Averaging [63]	*	*	*	*	*	*			✓		tokenizer, pretrained embeddings

BLEU (Bilingual Evaluation Understudy)

Almost first and most popular metric for MT

- Precision-based metric that computes the n-gram overlap between the reference and the hypothesis
- In particular, BLEU is the ratio of the number of overlapping n-grams to the total number of n-grams in the hypothesis.

Corpus-level metric, i.e., BLEU gives a score over the entire corpus (as opposed to scoring individual sentences)

Major drawbacks of BLEU: (i) it does not take recall into account and (ii) it only allows exact n-gram matching

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu (2002). "BLEU: a Method for Automatic Evaluation of Machine Translation". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA: Association for Computational Linguistics, pp. 311–318

ROUGE (Recall-Oriented Understudy for Gisting Evaluation)

C.-Y. Lin (2004). “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, pp. 74–81

ROUGE metric includes a set of variants: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S

- ROUGE-N is similar to BLEU-N in counting the n-gram matches between the hypothesis and reference, however, it is a recall-based measure unlike BLEU which is precision-based
- ROUGE-L measures the longest common subsequence (LCS) between a pair of sentences

Evaluation

Caveats of NLP benchmarking

The ‘gold’ data paradigm might not always fit

The assumption of a ground truth makes sense when humans highly agree on the answer

- “Does this image contain a bird?”
- “Is ‘learn’ a verb?”
- “What is the capital of Italy?”

This assumption often does not make sense, especially when language is involved

- “Is this comment toxic?”

Human label variation impacts all steps of the traditional ML pipeline, and is an opportunity, not a problem

B. Plank (2022). “The “Problem” of Human Label Variation: On Ground Truth in Data, Modeling and Evaluation”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 10671–10682

Human annotators are biased

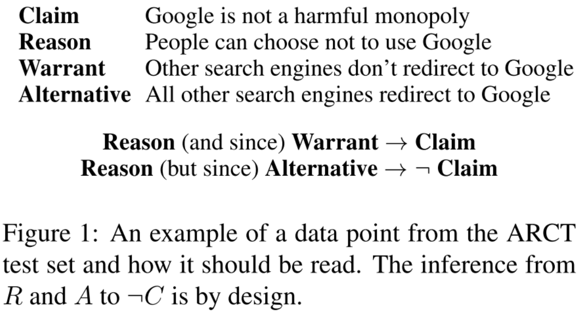
Datasets are often constructed using a small number of annotators, and humans are biased

- Concerns about data diversity, especially when workers freely generate sentences
- Models do not generalize well to examples from annotators that did not contribute to the training set

M. Geva, Y. Goldberg, and J. Berant (2019). "Are We Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1161–1166

Artifacts in datasets

Datasets have artifacts (spurious statistics) that can be exploited



Claim	Google is not a harmful monopoly
Reason	People can choose not to use Google
Warrant	Other search engines don't redirect to Google
Alternative	All other search engines redirect to Google

Reason (and since) **Warrant** → **Claim**
Reason (but since) **Alternative** → ¬ **Claim**

Figure 1: An example of a data point from the ARCT test set and how it should be read. The inference from R and A to $\neg C$ is by design.

I. **Habernal**, H. Wachsmuth, I. Gurevych, and B. Stein (2018). "The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants". In: *Proceedings of NAACL*. New Orleans, LA, pp. 1930–1940

T. Niven and H.-Y. Kao (2019). "Probing Neural Network Comprehension of Natural Language Arguments". In: *Proceedings of ACL*. Florence, Italy, pp. 4658–4664

License and credits

Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
`https://www.aclweb.org/anthology`