# Evaluating NLP systems

Statistical Methods in NLP 2
ISCL-BA-08

Çağrı Çöltekin
ccoltekin@sfs.uni-tuebingen.de

University of Tübingen
Seminar für Sprachwissenschaft

Summer Semester 2026

version: 1eee596+ @2026-07-22



## Evaluation in ML/NLP

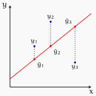
- We want to know how well a model does what it is intended to do
  - In real-world usage, it is important to know limitations of the system
  - Being able to quantify system performance helps us choose better systems, in particular during development
- Typically, model performance is evaluated based on a *metric*
- In some cases, the most reliable evaluation is human evaluation
- Evaluation should always be done on a *held-out* data set
- *Data* is (often) a hidden factor in evaluation. The scores are as indicative as the evaluation data
- Model evaluation should be tuned to the problem at hand: the success on all real-world problems can not be captured by a single metric/method





## Regression: root mean squared error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2}$$



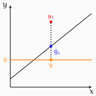

- Measures average error in the units compatible with the outcome variable





## Regression: coefficient of determination

$$R^2 = \frac{\sum_i^n (\hat{y}_i - \bar{y})^2}{\sum_i^n (y_i - \bar{y})^2} = 1 - \frac{\mathrm{MSE}}{\sigma_y^2}$$




- $r^2$ is a standardized measure in range $[0, 1]$
- Indicates the ratio of variance of $y$ explained by $x$
- With a single predictor it is the square of the correlation coefficient $r$





## Regression: mean absolute error

- (Mean) *absolute error* is another, sometimes better choice (less sensitive to outliers)

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- Normalized (percentage) version is also common

$$\mathrm{MAE\%} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$





## Classification: accuracy, Precision, recall, F-score

$$\mathrm{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\mathrm{precision} = \frac{TP}{TP + FP}$$

$$\mathrm{recall} = \frac{TP}{TP + FN}$$

$$F_1\text{-score} = \frac{2 \times \mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

| | | true value | |
|---|---|---|---|
| | | positive | negative |
| predicted | pos. | TP | FP |
| | neg. | FN | TN |





## Classification: confusion matrix

- A confusion matrix is often useful for multi-class classification tasks

| | | predicted | | |
|---|---|---|---|---|
| | | negative | neutral | positive |
| true value | negative | 10 | 2 | 0 |
| | neutral | 3 | 12 | 7 |
| | positive | 4 | 8 | 7 |

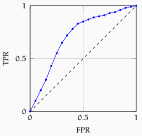
- Given the confusion matrix, we can observe classifier behavior better than single-number metrics
- All earlier metrics can be computed from the confusion matrix





## Classification: ROC and AUC

- Receiver Operating Characteristic (ROC) graph is another common metric (alternatively: precision–recall curve)
- The curve displays model's performance with varying probability thresholds
- ROC plots true-positive rate (TPR) against false-positive rate (FPR)
- Area under curve (AUC) is used as a summary measure
- AUC is threshold invariant. In most practical applications, what we care about is a single threshold








## Evaluating natural language generation

- Evaluating systems that generate natural language (e.g., *machine translation, summarization, question answering, image captioning, data-to-text, dialog agents, story generation*) is more difficult than evaluating simple regression and classification
  - We need to evaluate structured / sequence output
  - There are multiple 'correct' answers
- Most appropriate evaluation is often *human evaluation*
- However, human evaluation is expensive, and difficult to integrate into a development environment
- Automatic evaluation can be based on
  - General measures of 'high quality' text (especially when no clear gold-standard is available)
  - Measures of resemblance with a ideal output, if we have gold-standard output





## Intrinsic or reference-free metrics

- Cross entropy or perplexity is the most common method for intrinsic evaluation
- Without a reference, typical aspects of text that are measured include
  - Fluency
  - Coherence
  - Diversity
  - lack of redundancy
  - Readability
- There are some 'unsupervised' metrics (e.g., from readability research)
- Otherwise, most typical approaches include training models to assign scores in these dimensions





## BLUE

- BLUE (Bilingual Evaluation Understudy) is a simple metric that is widely used, in particular in machine translation
- BLUE measures n-gram precision: it is based on the overlap between the predicted text and a set of reference texts
- BLUE is one of the oldest metrics for machine translation
- Despite many shortcomings, BLUE is still common for evaluation text generation systems





## BLUE with example

**Prediction**

The cat sits on a carpet

**References**

The cat is on the mat
There is a kitten on the mat

- Calculating BLEU (unigrams only):
  - TP = 4
  - BLUE = 4/6
- The calculation is repeated with varying n-gram sizes (typically 1 to 4), and averaged (geometrically)
- The score is averaged over all predictions
- Scores over 0.3 considered reasonable

## Problems with BLUE

- Precision only: no sense of recall
- Easy to cheat:
  - A translation 'the a' is likely to get very high BLUE for any sentence in English
  - A 'brevity penalty' is often included in practical implementations
- N-gram overlap does not consider synonymy, similarity
- Problems with fluency
- Sensitive to tokenization
- Not good with morphologically complex languages





## ROUGE

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is similar to BLEU, but measure recall
- Similarly, it can be calculated for n-grams (typically indicated as ROUGE-n)
- ROUGE-L is based on longest common subsequence (LCS)
- It suffers most of the problems with BLEU
- ROUGE is more common in evaluating summarization





## More metrics using symbol overlap

- METEOR, is another overlap metric used commonly
  - Harmonic mean of precision and recall, with recall with higher weight
  - Also uses factors in stemming and synonymy into the overlap
- chrF is another common metric based on character n-gram overlap
- TER (translation error rate) calculates the minimum number of edits to obtain the reference from the prediction





## BERTScore

- BERTScore makes use of modern language models to evaluate the match between prediction and reference texts
- BERTScore calculates pair-wise cosine similarity between all tokens in the prediction and the reference
- Each token is matched with the most similar token in the other sentence

$$P = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} x_i \cdot \hat{x}_j \qquad R = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} x_i \cdot \hat{x}_j$$

- BERTScore fixes some of the problems (e.g., synonymy), with some limitations (e.g., computational cost, interpretability)





## Evaluating generation: summary

- There are multiple dimensions for evaluating natural language generation systems
- Evaluation can be reference-free, or based on one or more reference texts
- Human evaluation is the most 'convincing' metric. However, it is costly and not practical
- Token overlap based metrics (e.g., BLEU, ROUGE) are common and easy to compute
- Model-based metrics (e.g., BERTScore) improve some of the limitations
- Also becoming common
  - trained models predicting human scores
  - ask an LLM
- It is important to understand the metric you use, pay attention to their fit to the task at hand, and beware of their limitations





## Model selection and hyperparameter tuning

- Our aim is to reduce the error on unseen data
- The evaluation practice should reflect that
- We can estimate the test error on a *development* set (*validation* or *held-out* data):
  - Split the data at hand as *training* and *development* set
  - Train alternative models (different hyperparameters) on the training set
  - Choose the model with best development set performance





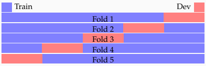
## Cross validation

- To avoid overfitting, we want to tune our models on a *development set*
- But (labeled) data is valuable
- Cross validation is a technique that uses all the data, for both training and tuning with some additional effort
- Besides tuning hyper-parameters, we may also want to get 'average' parameter estimates over multiple folds





## K-fold Cross validation




- At each fold, we hold part of the data for testing, train the model with the remaining data
- The special case where k equal to the number of data points is called *leave-one-out cross validation*





## The choice of k in k-fold CV

- Increasing k
  - reduces the bias: the estimates converge to true value of the measure (e.g., $R^2$) in the limit
  - increases the variance: smaller held-out sets produce more varied parameter estimates
  - is generally computationally expensive
- 5- or 10-fold cross validation is common practice (and found to have a good balance between bias and variance)





## Comparing with a baseline

- The performance measures are only meaningful if we have something to compare against

random: does the model do anything useful at all?
majority class: does the classifier work better than predicting the majority class all the time?
state-of-the-art: how does your model compare against known (non-trivial) models?

- In comparing different models we use another split of the data, *test set*
- Ideally test set is used only once – we want to avoid tuning the system on the test data
- Differences between models are exactly repeatable when the same test set is used (by different studies)
- Differences are reliable if your test set size is large enough
- Use statistical tests when comparing different models/methods





## Assessing (test) data quality

- The test / benchmark data is at the center of most evaluation methods
- The test data should include diverse examples from the real-world data for the task
- Data quality is equally important
- A common indication of dataset consistency (quality) is the *inter annotator agreement* (IAA)
- Common IAA metrics include *Cohen's* $\kappa$, *Fleiss'* $\kappa$, *Scot's* $\pi$, *Krippendorff's* $\alpha$
- Unlike classification evaluation metrics, IAA metrics are symmetric
- Common discounts expected agreement $A_e$ from the observed agreement ($A_o$)

$$\frac{A_o - A_e}{1 - A_o}$$

- Different metrics mainly differ with respect to the way expected agreement is calculated
- Disagreement is not always an error, for many tasks it is expected (and should be modeled)





## Checks for model comparison

- Is the held-out data (development set) representative for the actual use case?
- Are the hyperparameters of all models tuned properly?
- Is the model variation taken into account?
- Is the variation due to development set taken into account?
- Is the held-out data large enough?

## Reproducibility

- Reproducibility is an essential part of scientific method
- A scientific finding is more reliable if it can be reproduced by others
- Reproducibility is not only important for research: practical decisions / policies should be based on reproducible experiments
- Why reproduction may fail?
  - Misconduct, sloppy work
  - Measurement error: metrics are not precise enough
  - Non-representative, non-random, or small test set
  - Confirmation bias
  - Insufficient information on the original experiment





## Summary: a bag of concepts on evaluation

- Correct evaluation of the NLP systems is as important as (a prerequisite to) developing good ones
- Choose your evaluation metrics carefully
- Compare alternative models correctly
  - Always test on a held-out data set
  - Carefully tune your model hyperparameters
  - Compare models to a baseline
- Beware of issues in your test data
- Make sure your experiments are reproducible



## Additional reading, references, credits