IMPERIAL

Text Classification

Shamsuddeen Hassan Muhammad
Advanced Research Fellow/Google DeepMind Fellow ,
Imperial College London

NLP at AIMS Cameroon

1

Previous Class: Language Modelling

- What is a language model ?
- Do we have different Language models?
- What is the simplest Language Model?
- What N-Gram Language Model? What N- means in N-Gram Language Model?
- What is an example of Probabilistic Language Model?
- What is the markup assumption in language modelling and what does it solves?
- What is the of Limitation of N-Gram Language Model and What is the solution?
- What is Perplexity?
- What is extrinsic evaluation and intrinsic evaluation? Which is better?
- What is Laplace Smoothing and why is it important?

2

Reference

These slides are based on the course material by Daniel Jurafsky :

Chapter 4: https://web.stanford.edu/~jurafsky/slp3/4.pdf

Book: Speech and Language Processing (3rd ed. draft)

3

Text Classification

- Classification lies at the heart *of both human* and *machine intelligence*.
- Deciding what letter, word, or image has been presented to our senses, recognizing faces or voices, sorting mail, assigning grades to homeworks.
- These are all examples of assigning a category to an input.
- The goal of classification is to take a single observation, extract some useful features, and thereby classify the observation into one of a set of discrete classes.
- Most cases of classification in language processing are done via supervised machine learning.

4

Example 1: Spam Classification

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

© Stanford University. All Rights Reserved.

5

Group Email

Pulling on the "heart strings"

Promise of future payoff.

Establishes response scenario.

Albert
August 27, 2017 at 12:38:33 PM EDT
To: undisclosed-recipients:;
Reply-To: <mr.truswell@mailbox.sk>
[External] Donation for Humanitarian support.

-- External email --

Dear Beloved,

Compliment of the day, May I have the pleasure of introducing myself ? I am Mr. Albert Joshua Truswell, I am 82 years old British dying man suffering from cancer disease for some years now. I got your contacts details after an extensive online search for a reliable person to entrust fund for charity works.

I have decided to donate my social fund 5.8Million British Pounds (Five million eight hundred thousand British Pounds) to you for charity works as I do not have wife, family or children that can inherit these funds when am gone. So I request you to contact me immediately for more details if you are interested in carrying out this project so that I will arrange the release of the funds to you from the bank before I join my ancestors.

Thank you and God bless you.

Mr. Albert Truswell
Cancer Patient

6

What is the subject of this medical article?

MEDLINE Article

?

MeSH Subject Category Hierarchy

Antogonists and Inhibitors
Blood Supply
Chemistry
Drug Therapy
Embryology
Epidemiology
...

Sentiment analysis?

- + ...zany characters and richly applied satire, and some great plot twists
- - It was pathetic. The worst part about it was the boxing scenes...
- + ...awesome caramel sauce and sweet toasty almonds. I love this place!
- - ...awful pizza and ridiculously overpriced...

Why Sentiment analysis?

- Movie: Is this review positive or negative?
- Products: What do people think about the new iPhone?
- Public sentiment: How is consumer confidence?
- Politics: What do people think about this candidate or issue?
- Prediction: Predict election outcomes or market trends from sentiment.

9

Examples of Text Classification Tasks

Text classification can be applied to various tasks, including:

1. **Sentiment Analysis** – Classifying text as positive, negative, or neutral (e.g., product reviews, social media comments).
2. **Spam Detection** – Identifying whether an email or message is spam or not.
3. **Hate Speech Detection** – Classifying text as hate speech, abusive language, or normal speech.
4. **Toxic Comment Classification** – Detecting toxic, offensive, or harmful language in online discussions.
5. **Topic Classification** – Categorizing articles, papers, or news into topics such as politics, sports, health, or technology.
6. **Intent Classification** – Determining user intent in chatbots (e.g., request, complaint, inquiry).
7. **Named Entity Recognition (NER) Classification** – Assigning labels to entities such as persons, locations, organizations, or dates.
8. **Language Identification** – Detecting the language in which a given text is written.
9. **Fake News Detection** – Classifying news articles as fake or real based on credibility.
10. **Emotion Detection** – Identifying emotions such as joy, sadness, anger, fear, or surprise in text.

10

Who Wrote This? Identifying Machine vs Human-Generated Text in Hausa

Babangida Sani[1], Aakansha Soy[1], Sukairaj Hafiz Imam[2], Ahmad Mustapha[1], Lukman Jibril Aliyu[3], Idris Abdulmumin[4], Ibrahim Said Ahmad[5], Shamsuddeen Hassan Muhammad[6]

[1]Kalinga University, [2]Bayero University, Kano, [3]Arewa Data Science Academy, [4]DSFSI, University of Pretoria, [5]Northeastern University, [6]Imperial College London

correspondence: bsani488@gmail.com

Abstract

The advancement of large language models (LLMs) has allowed them to be proficient in various tasks, including content generation. However, their unregulated usage can lead to malicious activities such as plagiarism and generating and spreading fake news, especially for low-resource languages. Most existing machine-generated text detectors are trained on high-resource languages like English, French, etc. In this study, we developed the first large-scale detector that can distinguish between human- and machine-generated content in Hausa. We scrapped seven Hausa-language media outlets for the human-generated text and the Gemini-2.0 flash model to automatically generate the corresponding Hausa-language articles based on the human-generated article headlines. We fine-tuned four pre-trained Africentric models (AfriTeVa, AfriBERTa, AfroXLMR, and AfroXLMR-76L) on the resulting dataset and assessed their performance using

The *Boko* script, existing since the 1930s, was introduced by the British colonial administration, and is used in education, government, and digital communication. The *Ajami* script, an order writing system of the Hausa language that existed in pre-colonial times, is used mostly in religious, cultural, and informal writing. For the purpose of our work, and as Hausa is widely written nowadays, we scraped and generated data based on the Latin-based script.

Large language models (LLMs) are becoming mainstream and easily accessible, ushering in an explosion of machine-generated content over various channels, such as news, social media, question-answering (QA) forums, educational, and even academic contexts (Wang et al., 2023). The human-like quality of texts generated by LLMs models for different languages including Hausa language is always advancing, allowing them to generate diverse content. LLMs, intentionally or unintentionally, have the potential to be used to create and propa-

Text Classification: Definition

Input:

- A document d
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

Output: A predicted class $c \in C$

Probabilistic classifier

- A probabilistic classifier additionally will tell us the probability of the observation being in the class.
- This full distribution over the classes can be useful information for downstream decisions; avoiding making discrete decisions early on can be useful when combining systems

13

Probabilistic classifier

Example: Medical Diagnosis System

Scenario: A hospital is using a machine learning model to classify whether a patient falls into one of the following categories:

- **Class A:** No disease
- **Class B:** Mild symptoms
- **Class C:** Severe condition

A traditional classifier would simply output a single class label, e.g., *Class B*. However, a probabilistic classifier provides a probability distribution over all classes:

$$P(A) = 0.2, \quad P(B) = 0.5, \quad P(C) = 0.3$$

Why is this Useful?

- **Better Decision-Making:** Instead of committing to one class, a doctor can see that the patient has a 30% chance of a severe condition and may decide to conduct further tests before making a final diagnosis.
- **Risk Assessment:** If the probability of a severe condition is high (e.g., $P(C) = 0.49$), even if the model predicts *Mild symptoms*, the doctor might choose to treat the patient as a high-risk case.
- **Combination with Other Systems:** The probability scores can be used to integrate the classifier's output with other models (e.g., expert systems, additional diagnostic tools) for a more reliable final decision.

14

Probabilistic classifier

By providing a full probability distribution rather than a single label, probabilistic classifiers **enhance decision-making, reduce errors, and allow for more flexible system integration**. This makes them especially useful in high-stakes domains such as healthcare, finance, and automated systems.

15

Classification Methods: Hand-coded rules

Rules based on combinations of words or other features

- Spam: *black-list-address* OR (*"dollars" AND "you have been selected"*)
- Accuracy can be high if rules carefully refined by experts
- But building and maintaining these rules is expensive

16

Classification Methods: Hand-coded rules

Rule-Based Spam Detection Example (IF-THEN Condition)

IF (email contains `"free money"` OR `"you have won"` OR `"click here to claim"`)
AND (sender is unknown)
THEN classify as spam.

17

Supervised machine learning

- The most common way of doing text classification in language processing is instead via supervised machine learning.
- In supervised machine learning learning, we have a data set of input observations, each associated with some correct output (a 'supervision signal').
- The goal of the algorithm is to learn *how to map from a new observation to a correct output*

18

Classification Methods: Supervised Machine Learning

Input:

- A document d
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- A training set of m hand-labeled documents: $(d_1, c_1), (d_2, c_2), \ldots, (d_m, c_m)$

Output:

- A learned classifier $\gamma : d \rightarrow c$

19

Classification Methods: Supervised Machine Learning

Any kind of classifier can be used:

- Naïve Bayes
- Logistic regression
- Neural networks
- k-Nearest Neighbors

20

Supervised Learning Problems

- We have training examples $x^{(i)}$, $y^{(i)}$ for $i = 1, \ldots, m$. Each $x^{(i)}$ is an input, each $y^{(i)}$ is a label.
- Task is to learn a function f mapping inputs x to labels $f(x)$.
- Conditional models:
  - Learn a distribution $p(y|x)$ from training examples.
  - For any test input x, define $f(x) = \arg\max_y p(y|x)$.

Generative and Discriminative Models

Classifiers can be broadly categorized into two types: **Generative** and **Discriminative** models.

- Generative classifiers model the joint probability distribution p(x, y), where x represents the input features and y represents the class label.
- Discriminative classifiers directly model the conditional probability p(y|x) without assuming a data generation process.

Feature	Generative	Discriminative
Models	$p(x, y)$	$p(y\|x)$
Examples	Naïve Bayes, HMMs	Logistic Regression, SVM
Focus	Data generation	Decision boundary
Strengths	Works with small data	More accurate
Weaknesses	Independence assumptions	Needs more data

Generative Models

- Given training examples $x^{(i)}$, $y^{(i)}$ for $i = 1, \ldots, m$. The task is to learn a function f that maps inputs x to labels $f(x)$.
- Generative models:
  - Learn the joint distribution $p(x, y)$ from the training examples.
  - Often, we have $p(x, y) = p(y)p(x|y)$.
  - Note: We then have

$$p(y|x) = \frac{p(y)p(x|y)}{p(x)} \quad \text{where} \quad p(x) = \sum_y p(y)p(x|y).$$

Classification with Generative Models

- Given training examples $x^{(i)}$, $y^{(i)}$ for $i = 1, \ldots, m$. The task is to learn a function f that maps inputs x to labels $f(x)$.
- Generative models:
  - Learn the joint distribution $p(x, y)$ from the training examples.
  - Often, we have $p(x, y) = p(y)p(x|y)$.
- Output from the model:

$$\begin{aligned} f(x) = \arg\max_y p(y|x) &= \arg\max_y \frac{p(y)p(x|y)}{p(x)} \\ &= \arg\max_y p(y)p(x|y) \end{aligned}$$

Naive Bayes Intuition

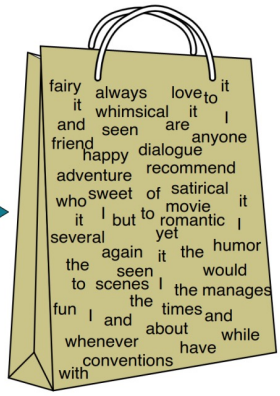
Naive Bayes is a simple (”naive”) classification method based on Bayes’ rule.

- Relies on a very simple representation of a document: *Bag of words*

25

The Bag of Words Representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

26

Bayes' Rule Applied to Documents and Classes

For a document d and a class c:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

27

Stop Words

Stop words are frequently used words like "the" and "a" that are often considered to have little or no significance in text classification. Some systems choose to ignore stop words in the classification process. Here is how it is typically done:

- Sort the vocabulary by word frequency in the training set.
- Create a stopword list by selecting the top 10 or 50 most frequent words.
- Remove all stop words from both the training and test sets, treating them as if they were never there.

However, removing stop words doesn't usually improve the performance of Naive Bayes classifiers. Therefore, in practice, most Naive Bayes algorithms use all words and do not utilize stopword lists.

28

Sentiment Classification: Dealing with Negation

I really like this movie

I really **don't** like this movie

Negation changes the meaning of "like" to negative.

Negation can also change negative to positive-ish

- **Don't** dismiss this film
- **Doesn't** let us get bored

Sentiment Classification: Dealing with Negation

A very simple baseline that is commonly used in sentiment analysis to deal with negation is the following: during text normalization, prepend the prefix **NOT** to every word after a token of logical negation (n't, not, no, never) until the next punctuation mark.

Add NOT_ to every word between negation and following punctuation:

```
didn't like this movie , but I
```

```
didn't NOT_like NOT_this NOT_movie but I
```

Sentiment Classification: Dealing with Negation

Newly formed ‘words’ like **NOT_like**, **NOT_recommend** will thus occur more often in negative document and act as cues for negative sentiment,

while words **like NOT_bored**, **NOT_dismiss** will acquire positive associations.

Slides adapted from Jure Leskovec

31

Sentiment Classification: Lexicons

- Sometimes we don't have enough labeled training data to train accurate ML model e.g., naive Bayes classifiers
- In that case, we can make use of pre-built word lists Called lexicons
- There are various publically available lexicons

32

Sentiment Lexicon

- General Inquirer (Stone et al., 1966),
- LIWC (Pennebaker et al., 2007),
- the opinion lexicon General Inquirer
- LIWC of Hu and Liu (2004) and
- the MPQA Subjectivity Lexicon (Wilson et al., 2005).

Slides adapted from Jure Leskovec

33

MPQA

- subjectivity lexicon has 6885
- Words each marked for whether it is strongly or weakly biased positive or negative. Some examples:

$+$: *admirable, beautiful, confident, dazzling, ecstatic, favor, glee, great*
$-$: *awful, bad, bias, catastrophe, cheat, deny, envious, foul, harsh, hate*

Slides adapted from Jure Leskovec

34

Evaluation of Text Classification

Task	Metric	Use Case
Classification	Accuracy	Balanced datasets
Classification	Precision/Recall	Imbalanced datasets
Classification	F1-Score	Trade-off between precision & recall
Classification	ROC-AUC	Probabilistic models
Classification	Log Loss	Measures confidence in predictions
Regression	MAE	Average absolute error, interpretable
Regression	MSE	Penalizes large errors, smooth gradients
Regression	RMSE	Similar to MSE, but interpretable
Regression	R^2 Score	Measures variance explanation

35

Accuracy

Measures the proportion of correctly classified instances:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Works well for balanced datasets but can be misleading for imbalanced data.

36

Example: Calculating Accuracy

Consider a binary classification problem (spam detection). Suppose we test a model on 10 emails, with actual and predicted labels as follows:

Email	Actual Label (y)	Predicted Label ($\hat{y}$)
1	1 (Spam)	1 (Spam)
2	0 (Not Spam)	0 (Not Spam)
3	1 (Spam)	0 (Not Spam) ×
4	0 (Not Spam)	0 (Not Spam)
5	1 (Spam)	1 (Spam)
6	1 (Spam)	1 (Spam)
7	0 (Not Spam)	1 (Spam) ×
8	0 (Not Spam)	0 (Not Spam)
9	1 (Spam)	1 (Spam)
10	0 (Not Spam)	0 (Not Spam)

37

Example: Calculating Accuracy

Step 1: Count Correct Predictions

- Correct Predictions (): 8
- Incorrect Predictions (×): 2
- Total Predictions: 10

Step 2: Compute Accuracy

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{8}{10} = 0.8 \quad (80\%)$$

However, accuracy may not be reliable for imbalanced datasets.

38

Example: Calculating Accuracy

accuracy may not be reliable for imbalanced datasets.

39

Example: Why Accuracy is Not Reliable for Imbalanced Datasets

Consider a medical diagnosis problem where we predict whether a patient has cancer (Positive = 1) or does not have cancer (Negative = 0). However, only 5% of the patients in the dataset actually have cancer.

Dataset Statistics:

- Total Patients = 1000
- Patients with Cancer (Positive class, 1) = 50
- Patients without Cancer (Negative class, 0) = 950

Now, consider a **simple classifier** that always predicts **"No Cancer" (0)** for every patient.

Predictions:

- Model predicts **"No Cancer"** for all 1000 patients.
- Out of 1000 predictions, it correctly classifies 950 patients without cancer.
- However, it misclassifies all 50 patients with cancer.

40

Example: Why Accuracy is Not Reliable for Imbalanced Datasets

Step 1: Calculate Accuracy

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{950}{1000} = 95\%$$

Step 2: Why is Accuracy Misleading?

- The model never predicts cancer (1), meaning all 50 cancer cases were missed.
- Even though accuracy is 95%, the model is completely useless for detecting cancer.

41

High accuracy does not mean good performance in imbalanced datasets. Better Metrics for Imbalanced Datasets?

- **Precision**: How many predicted cancer cases are actually cancer?
- **Recall (Sensitivity)**: How many actual cancer cases were correctly detected?
- **F1-Score**: Balances precision and recall.

42

Instead of accuracy we use precision and recall

	System Positive	System Negative
Gold Positive	True Positive (TP)	False Negative (FN)
Gold Negative	False Positive (FP)	True Negative (TN)

Recall (also known as **Sensitivity** or **True Positive Rate**):

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

43

Understanding Precision, Recall, and F1-Score

Imagine you are a doctor diagnosing cancer.

Precision (How accurate are the positive predictions?):
Precision measures how many of the predicted cancer cases are actually cancer.

$$Precision = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Positives (FP)}}$$

Example: If you diagnosed 10 people with cancer, but only 7 actually had cancer, then:

$$\text{Precision} = \frac{7}{10} = 70\%$$

44

Understanding Precision, Recall, and F1-Score

Imagine you are a doctor diagnosing cancer.

Recall (Sensitivity) (How many actual cases were detected?):
Recall measures how many of the actual cancer cases were correctly identified.

$$Recall = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)+False Negatives (FN)}}$$

Example: If there were 10 real cancer patients but you only identified 7 of them, then:

$$\text{Recall} = \frac{7}{10} = 70\%$$

45

Understanding Precision, Recall, and F1-Score

Imagine you are a doctor diagnosing cancer.

Trade-off: Precision and recall often compete—if you are too cautious, you may miss some real cancer cases (low recall). If you diagnose too many, you may misidentify healthy people (low precision). The F1-score helps balance both.

F1-Score (Balances precision and recall):
The F1-score is the harmonic mean of precision and recall, helping to balance both metrics.

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

46

Domains Where Precision is More Important

Precision is crucial in cases where **false positives** have serious consequences.

Precision is crucial in cases where **false positives** (incorrect positive predictions) have serious consequences.

- **Spam Detection**: A high-precision spam filter ensures that only real spam emails are marked as spam. A false positive (important email wrongly marked as spam) could result in missing crucial information.
- **Medical Diagnosis for Rare Diseases**: If a test falsely diagnoses a patient with cancer, it can cause unnecessary stress and costly further tests. High precision ensures that when the model predicts cancer, it is highly likely to be correct.
- **Fraud Detection**: If a bank wrongly flags a legitimate transaction as fraudulent, it can inconvenience customers. High precision helps minimize false fraud alerts.
- **Legal Document Classification**: A legal search system should retrieve only the most relevant case files, rather than overwhelming the user with irrelevant documents.

47

Domains Where Recall is More Important

Recall is crucial in cases where false negatives (missing actual positive cases) are costly or dangerous

- **Medical Screening for Contagious Diseases (e.g., COVID-19)**: Missing an actual positive case (false negative) means an infected person continues spreading the disease. High recall ensures that most infected individuals are detected.
- **Criminal Investigation & Security**: A face recognition system used for identifying criminals should not miss potential suspects. A false negative (missing a real criminal) is worse than a false positive (which can be manually verified).
- **Fire or Hazard Detection (Smoke/Intrusion Alarms)**: A smoke detector should always detect fire, even if it occasionally gives false alarms. High recall ensures that real hazards are rarely missed.
- **Search Engines & Recommendation Systems**: A search engine should retrieve all relevant results for a query. If recall is low, users might miss important information.

48

When to Balance Both (F1-Score is Important)

Some domains require a balance between precision and recall, making the F1-score a crucial metric

- **Autonomous Vehicles (Collision Detection)**: The system must avoid both false positives (unnecessary stops) and false negatives (failing to detect an obstacle), requiring a balance between precision and recall.
- **Chatbots & Virtual Assistants**: Precision ensures responses are relevant, while recall ensures that all user queries are properly handled.
- **Sentiment Analysis in Social Media Monitoring**: A good balance between precision and recall ensures accurate detection of both positive and negative sentiments.

Can we use Accuracy in multi-class problems?

Macro-Averaging and Micro-Averaging

In multi-class classification problems, evaluating the performance of a model can be approached using two methods: **micro-averaging** and **macro-averaging**. These methods differ in how they account for each class's contribution to the overall metrics.

51

Micro-Averaging (Equal Weight to Every Instance)

Micro-averaging aggregates the contributions of all classes to compute the average metric. This approach considers each individual prediction equally, regardless of its class.

1. Compute the total true positives (TP), false positives (FP), and false negatives (FN) across all classes:

$$\text{Total TP} = \sum_{i=1}^{N} \text{TP}_i, \quad \text{Total FP} = \sum_{i=1}^{N} \text{FP}_i, \quad \text{Total FN} = \sum_{i=1}^{N} \text{FN}_i$$

   where N is the number of classes.

2. Calculate precision and recall:

$$\text{Micro Precision} = \frac{\text{Total TP}}{\text{Total TP} + \text{Total FP}}, \quad \text{Micro Recall} = \frac{\text{Total TP}}{\text{Total TP} + \text{Total FN}}$$

3. Compute the F1-score:

$$\text{Micro F1-Score} = \frac{2 \times \text{Micro Precision} \times \text{Micro Recall}}{\text{Micro Precision} + \text{Micro Recall}}$$

52

Example Micro-Averaging

Consider a model classifying emails into three categories: Spam, Promotional, and Important. Suppose we have the following confusion matrix:

Class	TP	FP	FN
Spam	30	10	5
Promotional	40	5	10
Important	50	8	12

Calculations:

$$\text{Total TP} = 30 + 40 + 50 = 120$$

$$\text{Total FP} = 10 + 5 + 8 = 23$$

$$\text{Total FN} = 5 + 10 + 12 = 27$$

$$\text{Micro Precision} = \frac{120}{120 + 23} \approx 0.839$$

$$\text{Micro Recall} = \frac{120}{120 + 27} \approx 0.816$$

$$\text{Micro F1-Score} = \frac{2 \times 0.839 \times 0.816}{0.839 + 0.816} \approx 0.827$$

Use Case: Micro-averaging is particularly useful when class distributions are imbalanced, as it gives equal weight to each instance, making it more representative of the overall performance.

53

Macro-Averaging (Equal Weight to Every Class)

Macro-averaging computes the metric independently for each class and then takes the average, giving equal weight to each class regardless of its size.

1. Compute precision and recall for each class:

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \quad \text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}$$

2. Average the precision and recall across all classes:

$$\text{Macro Precision} = \frac{1}{N} \sum_{i=1}^{N} \text{Precision}_i, \quad \text{Macro Recall} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall}_i$$

3. Compute the F1-score using the averaged precision and recall:

$$\text{Macro F1-Score} = \frac{2 \times \text{Macro Precision} \times \text{Macro Recall}}{\text{Macro Precision} + \text{Macro Recall}}$$

54

Example Macro-Averaging

Using the same data:

$$\text{Precision}_{\text{Spam}} = \frac{30}{30+10} = 0.75, \quad \text{Recall}_{\text{Spam}} = \frac{30}{30+5} \approx 0.857$$

$$\text{Precision}_{\text{Promotional}} = \frac{40}{40+5} \approx 0.889, \quad \text{Recall}_{\text{Promotional}} = \frac{40}{40+10} \approx 0.800$$

$$\text{Precision}_{\text{Important}} = \frac{50}{50+8} \approx 0.862, \quad \text{Recall}_{\text{Important}} = \frac{50}{50+12} \approx 0.806$$

$$\text{Macro Precision} = \frac{0.75 + 0.889 + 0.862}{3} \approx 0.834$$

$$\text{Macro Recall} = \frac{0.857 + 0.800 + 0.806}{3} \approx 0.821$$

$$\text{Macro F1-Score} = \frac{2 \times 0.834 \times 0.821}{0.834 + 0.821} \approx 0.827$$

Use Case: Macro-averaging is beneficial when all classes are equally important, ensuring that the performance metric reflects the model's ability across all classes, regardless of their frequency.

55

Micro-Averaging vs Macro-Averaging?

- **Micro-Averaging:** Suitable for imbalanced datasets; emphasizes overall performance by giving equal weight to each instance.
- **Macro-Averaging:** Suitable when all classes are equally important; emphasizes performance across all classes by giving equal weight to each class.

56

Development Test Sets (”Devsets”)

- To avoid overfitting and provide a more conservative estimate of performance, we commonly use a three-set approach: training set, devset, and testset.

  - **Training set**: Used to train the model.
  - **Devset**: Used to tune the model and select the best hyperparameters.
  - **Testset**: Used to report the final performance of the model.
- This approach ensures that the model is not tuned specifically to the test set, avoiding overfitting.
- However, it creates a paradox: we want as much data as possible for training, but also for the devset.
- How do we split the data?

57

Cross-validation: Multiple Splits

- Cross-validation allows us to use all our data for training and testing without having a fixed training set, devset, and test set.
- We choose a number k and partition our data into k disjoint subsets called folds.
- For each iteration, one fold is selected as the test set while the remaining $k - 1$ folds are used to train the classifier.
- We compute the error rate on the test set and repeat this process k times.
- Finally, we average the error rates from these k runs to obtain an average error rate.

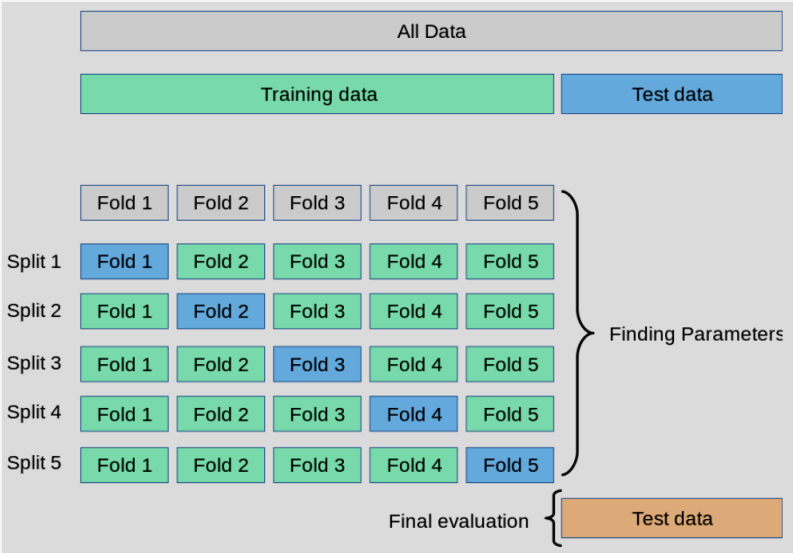
58

Cross-validation: Multiple Splits

- 10-fold cross-validation, for example, involves training 10 models on 90% of the data and testing each model separately.
- The resulting error rates are averaged to obtain the final performance estimate.
- However, cross-validation requires the entire corpus to be blind, preventing examination of the data for feature suggestion or understanding system behavior.
- To address this, a fixed training set and test set are created, and 10-fold cross-validation is performed within the training set.
- The error rate is computed conventionally in the test set.

59

Cross-validation: Multiple Splits



60

IMPERIAL

Thank you

61